

Számításigényes algoritmusok hatékony megvalósítása párhuzamos számítási eszközökön



Nemes Csaba

A PhD Disszertáció tézisei

Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar
Multidiszciplináris Műszaki és Természettudományi
Doktori Iskola

Témavezető:
Nagy Zoltán, PhD
Konzulens:
Szolgay Péter, DSc

Budapest, 2014

1. Bevezetés

Különböző fizikai jelenségek számításigényes szimulációja gyakran elkerülhetetlen lépése egyes mérnöki vagy tudományos indíttatású problémák megoldásának. Szimulációk segítségével tesztelni lehet olyan termékeket, amelyeket meg sem építettek, vagy előre meg lehet jósolni új fizikai vagy kémiai rendszerek egyes tulajdonságait. A számítástechnika a kezdetektől fogva foglalkozik a szimulációk gyorsításával, hogy rövidítse az új termékek elkészítésének idejét, vagy javítson a termékek minőségén a tervezési lehetőségek kiszélesítésével. Amióta a processzorok működési frekvenciájának emelése elérte a fizika korlátait, a processzortervezők arra koncentrálnak, hogyan lehet több-, illetve sokmagos processzorokkal életben tartani a Moore-törvény által előrevetített tendenciákat. A modern nagy teljesítményű számítástechnika célja, hogy választ találjon arra, hogyan lehet az új párhuzamos architektúrákban, mint például a grafikus processzor (GPU) vagy a *Field Programmable Gate Array* (FPGA), rejlő számítási kapacitást a számításigényes feladatok szolgálatába állítani.

Kutatói tevékenységem alatt két konkrét feladat gyorsításával foglalkoztam, melyek során a következő kérdések megválaszolására törekedtem. Melyik a legmegfelelőbb párhuzamos architektúra az adott feladat számára? Hogyan fejleszthető a tervezési metodika? Milyen maximális teljesítményt lehet elérni az adott architektúrán, és milyen kompromisszumok adódnak a sebesség, a fogyasztás és a felület szempontjából?

Az első feladat a *parciális differenciálegyenletek* (PDE) FPGA-n történő numerikus megoldása volt. Nagy Z. és társai demonstrálták, hogy az FPGA-ra tervezett emulált digitális CNN-UM architektúra általánosítható, és alkalmas számos megmaradási tétel hatékony szimulációjára a véges térfogat módszerből adódóan egy olyan matematikai kifejezést (ún. *numerikus sémát*) kell kiértékelni minden egyes iterációban és minden egyes cellára, mely leírható egy szinkron adatfolyam gráffal. A hatékony szimulációhoz a matematikai kifejezés kiértékelésére egy nagy teljesítményű, futószalagszerűen kialakított (angolul pipelined) *aritmetikai egységet* (AE) érdemes tervezni, amely magas működési frekvencián üzemeltethető. Ilyen aritmetikai egységet kapunk, ha a kifejezés minden operátorát, azaz a gráf csúcsait, egy dedikált lebegőpontos *műveletvégző egységgel* (ME) valósítjuk meg. A modern FPGA-kon számos műveletvégző egység megvalósítható és magas működési frekvencián üzemeltethető, ugyanakkor a globális vezérlőjelek, melyek a műveletvégző egységeket összekötik, lelassíthatják a teljes áramkört.

A kutatási céloom az volt, hogy megalkossak egy új tervezési metodikát,

amelyik képes nagy teljesítményű, lokálisan vezérelt aritmetikát generálni az adatfolyam gráf reprezentáció alapján. Kérdéseim a következők voltak. Hogyan érdemes particionálni a gráfot, hogy hatékonyan vezérelhető partició osztályokat kapjak? Hogyan érdemes vezérelni az egyes osztályokat, és hogyan érdemes őket összekötni, hogy elkerüljük a szinkronizációs problémákat? Mi az ára a sebesség növelésének a felület és a fogyasztás szemszögéből? Végezetül hogyan érdemes automatizálni az aritmetika generálását, hogy csökkentsük az újabb szimulációk fejlesztési idejét?

A második feladat az ún. sűrűségmátrix renormalizálási csoport (DMRG) algoritmus [10] gyorsítása volt. Az algoritmus egy variációs numerikus módszer, mely napjaink egyik vezető algoritmusára erősen összefüggő, láncszerű összefonódást mutató kvantumfizikai rendszerek alacsony energiaszintjeinek a vizsgálatára [11]. A módszer lehetővé teszi a felhasználó számára, hogy az effektív Hilbert tér mérete és a szimuláció pontossága között egyensúlyt teremtsen. Az algoritmus futásidejét a Hamilton operátor iteratív diagonalizálása dominálja. Mivel az algoritmus legidőigényesebb része, a diagonalizálás projekciós lépése, kifejezhető sűrű mátrix-mátrix szorzások sorozataként, az algoritmus ideális jelölt arra, hogy kihasználja a modern párhuzamos architektúrákban rejlő számítási kapacitást.

Mivel a DMRG algoritmust tudtommal még nem gyorsították modern párhuzamos architektúrán, az első kérdés az volt, hogy melyik architektúrán lehet a leghatékonyabban gyorsítani. Egy olyan nagy teljesítményű, párhuzamos és flexibilis megvalósítást tűztem ki célul, amelyik az algoritmus futási konfigurációinak széles skáláját képes kezelni.

2. Módszerek

Egy C/C++ nyelven megírt keretrendszert készítettem, amely a PDE-t megoldó numerikus séma szöveges vagy SystemC leírásából automatikusan legenerálja az aritmetikai egységet. A keretrendszerben a numerikus séma gráfként ábrázolódik, ami lehetővé tette, hogy különböző particionáló algoritmusokat tervezek és teszteljek. Később fejlesztettem egy új gráf reprezentációt, amely képes mind a particionálással, mind az elhelyezéssel kapcsolatos információk leírására. Az algoritmus és reprezentáció tervezés kapcsán az áramkör elhelyezés [12, 13], a gráf particionálás [14] és a gráf vizualizáció [15] témakörben elérhető szakirodalomra támaszkodtam.

A keretrendszer kimenete az aritmetikai egység VHDL nyelvű leírása, amely közvetlenül használható a hagyományos FPGA szintézis eszközeivel. A vezérlő logikát és az ún. mixer egységeket, amelyek az aritmetikai egység-

get adatokkal látják el, VHDL nyelven készítettem a Xilinx ISE Design Suite 13.1. segítségével. Az aritmetikai egységben felhasznált lebegőpontos műveletvégző egységeket a Xilinx IP Core könyvtár segítségével valósítottam meg. Az áramkör elhelyezésére vonatkozó kényszerfeltételeket a Xilinx User Constraint File (UCF) segítségével adtam meg. A kényszerfeltételek kézi finomhangolását a Xilinx PlanAhead szerkesztő segítségével végeztem.

A Xilinx elhelyező és huzalozó algoritmus, az alkalmazott valószínűségi heurisztikáknak köszönhetően, érzékeny a kezdeti paraméterekre. A teljesítmény kiértékelése során a legenerált áramköröket különböző kezdeti paraméter értékek esetén is implementáltam, és mindig a legnagyobb frekvenciát nyújtó eredményt választottam.

Az áramlástan (CFD) szimulációkhoz tartozó aritmetikai egységeket, amelyeken keresztül a tervezett keretrendszert bemutattam, egy Xilinx Virtex-6 SX475T FPGA-n implementáltam -1-es sebesség besorolás mellett. A lapkán 74.000 slice, 2.016 dedikált szorzó blokk (DSP48E1) és 38.304 Kb belső memória (BRAM) található. A lapka az egyik legnagyobb Virtex-6 FPGA, melyben a szorzó és memória blokkok aránya kimagasló.

A DMRG algoritmust C/C++ nyelven implementáltam, és fordítható csak CPU, illetve hibrid GPU-CPU módban is. CPU módban az alapvető lineáris algebrai szubrutinokat a BLAS interfészen keresztül az Intel MKL függvénykönyvtár segítségével gyorsítottam. Hibrid üzemmódban a műveletek egy részét a GPU-n valósítottam meg a CUDA 5.0 környezet segítségével. A Davidson iteráció projekciós műveletéhez kapcsolódó mátrix-mátrix szorzásokat az NVidia CuBLAS függvénykönyvtár segítségével implementáltam, míg az aszimmetrikus mátrix-vektor szorzáshoz saját CUDA kernelt terveztem.

Modellnév	CUDA magok száma	Órajel frekvencia	Eszköz memória	Compute Capability*
NVidia GTX 570	480	1464 MHz	1280 MB	2.0
NVidia K20	2496	706 MHz	5120 MB	3.5

* A GPU architektúra verziószáma.

1. táblázat. A felhasznált grafikus kártyák főbb paraméterei.

A hibrid üzemmód teljesítményét összevettem a CPU teljesítményével egy középkategóriás (Intel Core-i7 2600 3.4 GHz CPU + NVidia GTX 570 GPU) és egy felsőkategóriás (Intel Xeon E5-2640 2.5 GHz CPU + NVidia K20 GPU) konfiguráció esetén. A konfigurációban szereplő grafikus kártyák főbb paraméterei az 1. táblázatban szerepelnek. Az összehasonlítást két különböző

modell (Heisenberg és Hubbard) esetén is elvégeztem, és hasonló mértékű gyorsulásokat kaptam.

3. Új tudományos eredmények összefoglalása

A disszertáció két téziscsoport köré szerveződik. Az első téziscsoport lokálisan vezérelt *aritmetikai egységek* adatfolyam gráf leírásból történő generálásával foglalkozik. A második téziscsoport tárgya pedig a DMRG algoritmus első modern párhuzamos számítógép-architektúrákon történő megvalósítása.

I. Tézis **Magas működési frekvenciájú lokális vezérlést terveztem szinkron adatfolyam gráfok FPGA-n történő implementációjához, és módszert adtam a lokálisan vezérelt egységek számának és struktúrájának meghatározására a sebesség és a felület figyelembevételével.**

I.1. **Megterveztem és megvalósítottam egy lokálisan elosztott vezérlési módot, mely segítségével elkerülhetőek a globális vezérlőjelek és a vezérlés működési frekvenciája a felület mérsékelt növekedése árán növelhető. [3]**

Kísérletileg megmutattam, hogy a szinkron adatfolyam gráfokból generált aritmetikai egységek esetén (pl. parciális differenciálegyenletek numerikus megoldása) a globális vezérlő jelek visszafogják az aritmetikai egység működési frekvenciáját, ha a ki- és bemenetek száma meghalad egy határértéket. Megterveztem és megvalósítottam egy lokálisan elosztott vezérlési módot, amely lehetővé teszi, hogy a lassú globális vezérlést elkerüljük a felület mérsékelt növekedése mellett.

Megmértem a tervezett vezérlés működési frekvenciáját a műveletvégző egységek nélkül, különböző számú ki- és bemenet esetén, hogy a vezérlés sebessége és a ki- és bemeneteinek száma között fennálló kapcsolatot meghatározzam. A nagy teljesítményű számításokhoz tervezett Virtex-6 FPGA esetén a maximálisan 10 ki- és bemenetet kezelő vezérlés 510 MHz frekvenciát ért el, amely körülbelül 20%-os gyorsulás a 20 ki-és bemenetet kezelő referencia esethez képest. A méréssel igazoltam, hogy a ki- és bemenetek számának

korlátozóságával a vezérlés sebessége a szükséges mértékig növelhető, és lényegesen meghaladja a 450 MHz működési frekvenciát, melyet az áramkör többi része esetén (pl. lebegőpontos műveletvégzők) feltételezhetünk. A vezérlés esetén érdemes magasabb frekvenciát kitűzni célként, mivel gyakorlatban, amikor a műveletvégző egységek is megvalósításra kerülnek, jóval alacsonyabb működési frekvencia érhető el.

Optimalizálási feladatot terveztem az áramkör lokálisan vezérelt komponenseinek meghatározására, amely particionálja az adatfolyam gráfot, ha a ki- és bemenetek száma meghaladja a gyors működéshez szükséges határértéket. A létrejövő partició osztályok egymástól függetlenül vezérelhetőek, az osztályok közötti szinkronizációt pedig FIFO (First-In-First-Out) bufferek biztosítják, melyek növelik a felhasznált konfigurálható logikai egységek számát (az áramkör felületigényét). A tervezett optimalizálási feladat célja, hogy a lokális vezérlés miatt kialakuló felületnövekedést minimalizálja a gyors működési frekvenciát biztosító particionálási feltételek mellett.

I.2. Kifejlesztettem egy mohó particionáló algoritmust, amelyik felülmúlta az egyik leggyakrabban használt korszerű particionáló eljárást a megfogalmazott optimalizálási feladat esetén. [4]

Megmutattam egy komplex áramlástan feladat aritmetikai egységének a megvalósításán keresztül, hogy a particionálás során megfogalmazható célkitűzések önmagukban nem elégségesek a magas órajel eléréséhez, és a generált áramkör elhelyezhetőségét is figyelembe kell venni. Terveztem egy mohó particionáló algoritmust, amely figyelembe veszi az elhelyezhetőség szempontjait, és segítséget nyújt a műveletvégző egység magas szintű manuális elhelyezéséhez. A mohó algoritmust és a Xilinx fejlesztőkörnyezet nyújtotta manuális elhelyezési lehetőségeket felhasználva 370 MHz működési frekvenciát értem el egyszeres pontosság mellett, felülmúlva a korszerű hMetis [16] algoritmust körülbelül 13%-kal. A generált áramkör manuális elhelyezés nélkül 328 MHz-et ért el egyszeres pontosság, és 296 MHz-et dupla pontosság esetén.

I.3. Kifejlesztettem egy új particionáló algoritmust, amely mind a particionálásbeli, mind az elhelyezésbeli szempontokat figyelembe veszi, és magas működési frekvenciát eredményez manuális elhelyezés nélkül is. [1, 5–7]

Egy új, a magas szintű szintézis során alkalmazható megközelítést javasoltam, mely a hagyományos, egymásra épülő lépésekből álló megközelítéssel szemben már a particionálás során figyelembe veszi az elhelyezési

szempontokat. Terveztem egy szimulált lehűtésre épülő algoritmust a javasolt megközelítés bemutatására. Az algoritmus működését két komplex áramlástanai példán keresztül is megvizsgáltam, megmérve a maximális működési frekvenciát a ki- és bemenetek számának függvényében. A nem partitionált áramkörhöz képest mért legnagyobb gyorsulást (15-25%) akkor kaptam, amikor a partíció osztályok ki- és bemeneteinek a száma kisebb volt mint 9, illetve 10. Mindkét áramlástanai példa körülbelül 320-325 MHz működési frekvenciát ért el dupla pontosság esetén.

II. Tézis Az első hibrid CPU-GPU architektúrán megvalósított DMRG (Density Matrix Renormalization Group) implementáció hatékonyságának javítására ütemezőt terveztem a legidőigényesebb lépés mátrix-mátrix szorzásaihoz, és egy aszimmetrikus mátrix-vektor szorzó algoritmust dolgoztam ki GPU-hoz.

Elemeztem az algoritmus futásidőjét, és az iteratív diagonalizáló eljárás (Davidson) projekciós lépését találtam a legidőigényesebbnek, amely kiszámolható egy sor sűrű mátrix-mátrix szorzás segítségével. Megvizsgálva a GPU és az FPGA architektúrák teljesítményét a sűrű mátrix-mátrix szorzásokra nézve azt találtam, hogy kellően nagy méretű mátrixok esetén a művelet mindkét architektúrán jó kihasználtság mellett elvégezhető, ugyanakkor teljes kihasználtságot feltételezve a GPU architektúra körülbelül 5-ször nagyobb teljesítményre képes, mint az FPGA. A CUDA környezet segítségével elkészítettem az algoritmus hibrid GPU-CPU implementációját, amely az algoritmus első modern párhuzamos architektúrán történő gyorsítása. A hibrid megoldás körülbelül 3.5-szörös sebességnövekedést ért el a csak CPU-t használó verzióhoz képest.

II.1. Új ütemező algoritmust terveztem a DMRG legidőigényesebb lépéséhez, a projekciós műveletet leíró mátrix-mátrix szorzásokhoz, amely alkalmazkodva a GPU architektúra limitációihoz biztosítja a GPU magas kihasználtságát változó méretű mátrixok esetén. [8, 2]

Megvizsgáltam a projekciós operáció mátrix-mátrix szorzásaiban résztvevő mátrixok méretét a Heisenberg és a Hubbard modell esetén, melyek különböző számú szimmetriát tartalmaztak. Az eredményeket kiértékelve azt találtam, hogy a mátrix méretek széles skálán változnak mind az algoritmus iterációi között, mind magukon az iterációkon belül, és a mátrixok átlagos

mértét nagyban befolyásolja a választott fizikai modell és a modellben szereplő szimmetriák száma.

Mérésekkel igazoltam a választott architektúrák teljesítménye és a mátrixok mérete közötti összefüggést a CPU esetén az MKL, a GPU esetén pedig a CuBLAS függvénykönyvtár segítségével. A méréseim alapján a GPU kihasználtsága hatékonyan növelhető több művelet párhuzamos végrehajtásával, amelyet mind a CUDA keretrendszer, mind a DMRG algoritmuson belüli alkalmazás lehetővé tesz.

A projekciós művelet gyorsítására egy hibrid megoldást javasoltam, melyben a szorzás műveletek szétosztásra kerülnek az elérhető számítógépek között. A GPU kihasználtságának javítására a GPU-n végrehajtandó műveletekhez egy olyan új ütemező algoritmust terveztem, amely két stratégia szerint képes a műveleteket ütemezni. Az egyszálú futtatáshoz szánt ütemezési stratégiát a nagyobb mátrixokhoz terveztem, amelyeknél a GPU kihasználtsága egy művelet végrehajtása során is megfelelő. Ebben a stratégiában a műveletek sorrendjét csak az határozza meg, hogy a kommunikáció és a munkavégzés átlapolható legyen. A többszálú stratégiát a kisebb mátrixokhoz terveztem, amelyek esetén relative több GPU memória áll rendelkezésre, de a párhuzamos műveletvégzés indokolt. A többszálú stratégiához tartozó ütemezés megvalósítására egy olyan algoritmust terveztem, amely nem csak a kommunikáció és a munkavégzés átlapolását hanem a CUDA kernelek ütemezésének a korlátait is figyelembe veszi.

A felsőkategóriás K20 GPU esetén a többszálú stratégia a kisebb méretű (400-600) mátrixok szorzását jelentős mértékben (44%) gyorsította, ugyanakkor az algoritmus teljes futási idejét tekintve csak mérsékelt (5%) gyorsulást eredményezett, mivel az alkalmazott modellekben viszonylag nagy mátrixok szerepeltek. A gyakorlatban összetettebb modellek is előfordulnak, melyekben több szimmetria szerepel, ami kisebb átlagos mátrix méretet és nagyobb sebességnövekedést vetít előre.

II.2. Új algoritmust fejlesztettem GPU-ra, mely jelentősen megnövelte a GPU teljesítményét a DMRG-ben szereplő speciális, aszimmetrikus mátrix-vektor szorzások számítása során. [2]

Egy hibrid gyorsítást terveztem a DMRG algoritmus második legidőigényesebb részét képező szélsőségesen aszimmetrikus mátrix-vektor szorzásokhoz, amelyben az elvégzendő számítás az elérhető szabad GPU memória és az architektúrák teljesítménybeli különbsége alapján kerül szétosztásra. A GPU-ra jutó rész hatékonyságának növelésére egy új algoritmust terveztem a transzponált mátrix-vektor szorzás megvalósítására CUDA környezetben, amelyik a DMRG-s alkalmazás során, ahol a mátrix sorainak száma 1

és 24 között változott, 4-5-ször gyorsabbnak bizonyult, mint az NVidia CuBLAS könyvtár függvényei. A bemutatott gyorsítást a Heisenberg és a Hubbard modell esetén mérésekkel ellenőriztem, és a CPU-GPU kommunikációt is figyelembe véve körülbelül 2.4-szer gyorsabb aszimmetrikus mátrix-vektor szorzásokat eredményezett az NVidia K20 esetén.

4. Alkalmazási területek

A disszertációban két különböző típusú számításigényes probléma párhuzamos számítógép-architektúrákon történő gyorsíthatóságát vizsgáltam, különös tekintettel arra, hogyan lehet a gyorsítás során alkalmazott tervezési metodikán javítani. A problémákat más-más architektúrán valósítottam meg, és a kutatás eredményeit egy-egy téziscsoportban foglaltam össze.

Az I. Tézisben javasolt tervezési metodika minden olyan AE tervezése során alkalmazható, ahol jelentős számú ki- és bemenet mellett az áramkör sebessége elsőbbséget élvez a felületigénnyel szemben. Esetemben az AE tervezését komplex PDE-ek numerikus szimulációjának a gyorsítása motiválta, ugyanakkor más feladatokban is szükség lehet hasonló, számos ki- és bemenettel rendelkező aritmetikára (például Monte Carlo szimulációk).

A komplex PDE-ek FPGA-n történő numerikus szimulációja számos alkalmazás esetén felmerül, például áramlástan [1], elektromágneses [17] vagy szeizmikus [18] hullámok szimulációja. A szimulációk gyorsításából számos iparág és tudományterület profitálhat, például autó-, szélerőmű- és repülőipar, áramkörtervezés és szeizmológia.

A javasolt tervezési metodika legfontosabb eleme, hogy a magas szintű elhelyezés során nyert információ visszacsatolható a tervezési fázishoz, tovább általánosítható. Az ismertetett feladatban a műveletvégző egységek particionálása az a lépés, melyet szabadon állíthatunk be, hogy a legkedvezőbb elhelyezést elérjük. Ugyanakkor a módszer a tervezési fázisban felmerülő más szabad paraméter beállítására is használható. A javasolt metodika beépíthető más magas szintű szintézist támogató programokba, illetve a szintézis bármely pontján, ahol egy szabad paramétert a sebesség szempontjából optimalizálni kell.

A II. Tézis eredményeit elsődlegesen a DMRG algoritmus grafikus processzoron történő gyorsítása során használtam fel, de más alkalmazásokban is használhatóak, ahol hasonló feladatok merülnek fel. A mátrix-mátrix szorzásokra javasolt ütemezések felhasználhatók az ún. Tensor Network (TN) módszerek [19] megvalósítása során, melyek egy bővebb osztályt képviselnek, aminek a DMRG is a része. Az aszimmetrikus mátrix-vektor szorzásokra javasolt kernel felhasználható a Davidson iterációt alkalmazó kvantumkémiaili problémákban (pl. [20]).

Mivel a DMRG algoritmus az egyik vezető módszer az erősen korrelált, láncszerű összefonódottságot mutató kvantum rendszerek alacsony energiaszintjeinek a tanulmányozására, az elkészített alkalmazás jól használható anizotrop agyagok (pl. polimerek [21]) szimulációjára, vagy molekulák d pályájának az elektronikus szerkezetének [22] a leírására. Továbbá a DMRG alkalmas a lézerrel csapdázott ultrahideg atomokból felépített láncok tanulmányozására [23], melyeket az építendő kvantumszámítógépek egyik lehetséges építőelemének tartanak.

Köszönetnyilvánítás

Mindenekelőtt szeretnék köszönetet mondani Nagy Zoltán témavezetőmnek és konzulensemnek, Szolgay Péter professzor úrnak, hogy az évek során támogatta és egyengette a kutatói munkámat.

Nagy köszönettel tartozom Legeza Örsnek és Barcza Gergelynek, hogy kvantumfizikát tanulhattam tőlük. Barcza Gergelynek külön köszönettel tartozom, hogy barátként és gyermekem keresztapjaként is rendelkezésemre állt.

Kiemelt köszönet illeti Roska Tamás és Csurgay Árpád professzor urakat, valamint Nyékyné Gaizler Judit professzor asszonyt, hogy bátorítottak és lehetővé tették, hogy a kutatásaimat az egyetemen végezhessem.

Nagyra értékeltem Varga Balázssal és Balogh Ádámmal elköltött közös ebédeket és beszélgetéseket. Volt kollégám és barátom, Fülöp Tamás józan gondolatai mindig segítségemre voltak, hogy pontosan meghatározzam a helyzetem akár GPS nélkül is. Köszönet jár legközelebbi kollégáimnak, Kiss Andrásnak, Ruszinkó Miklósnak, Csík Árpádnak, Füredi Lászlónak, Hiba Antalnak és László Endrének az eredményes közös munkáért.

A legnagyobb köszönet a családomat illeti. Köszönöm feleségemnek, Fruzsinnak és fiamnak, Zentének, hogy szeretetükkel támogattak, és elviseltek, ha nem álltam a rendelkezésükre. Végezetül szerencsés vagyok, hogy szüleim, nagyszüleim és a dédnagymamám támogatták a véget nem érő tanulmányaimat.

A szerző publikációi

A szerző folyóirat publikációi

- [1] Z. Nagy, C. Nemes, A. Hiba, Á. Csík, A. Kiss, M. Ruszinkó, and P. Szolgay, "Accelerating unstructured finite volume computations on field-

programmable gate arrays”, **Concurrency and Computation: Practice and Experience**, vol. 26, no. 3, pp. 615–643, 2014.

- [2] C. Nemes, G. Barcza, Z. Nagy, Ö. Legeza, and P. Szolgay, “The density matrix renormalization group algorithm on kilo-processor architectures: implementation and trade-offs”, **Computer Physics Communications**, 2014. DOI: 10.1016/j.cpc.2014.02.021.

A szerző konferencia publikációi

- [3] C. Nemes, Z. Nagy, M. Ruzsinkó, A. Kiss, and P. Szolgay, “Mapping of high performance data-flow graphs into programmable logic devices”, in **Proceedings of the 2010 International Symposium on Nonlinear Theory and its Applications**, 2010, pp. 99–102.
- [4] C. Nemes, Z. Nagy, and P. Szolgay, “Efficient mapping of mathematical expressions to fpgas: exploring different design methodologies”, in **Circuit Theory and Design (ECCTD), 2011 20th European Conference on**, 2011, pp. 717–720.
- [5] C. Nemes, Z. Nagy, and P. Szolgay, “Automatic generation of locally controlled arithmetic unit via floorplan based partitioning”, in **Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on**, 2012, pp. 1–5.
- [6] Z. Nagy, C. Nemes, A. Hiba, A. Kiss, A. Csik, and P. Szolgay, “Fpga based acceleration of computational fluid flow simulation on unstructured mesh geometry”, in **Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on**, 2012, pp. 128–135.
- [7] Z. Nagy, C. Nemes, A. Hiba, A. Kiss, Á. Csík, and P. Szolgay, “Accelerating unstructured finite volume solution of 2-d euler equations on fpgas”, in **Conference on Modelling Fluid Flow (CMFF’12)**, 2012.
- [8] C. Nemes, G. Barcza, Z. Nagy, Ö. Legeza, and P. Szolgay, “Implementation trade-offs of the density matrix renormalization group algorithm on kilo-processor architectures”, in **Circuit Theory and Design (ECCTD), 2013 21th European Conference on**, 2013, pp. 100–104.

A témához kapcsolódó irodalom

- [9] Z. Nagy, Z. Vörösházi, and P. Szolgay, “Emulated digital cnn-um solution of partial differential equations”, **International Journal of Circuit Theory and Applications**, vol. 34, no. 4, pp. 445–470, 2006.
- [10] S. R. White, “Density matrix formulation for quantum renormalization groups”, **Phys. Rev. Lett.**, vol. 69, pp. 2863–2866, 19 1992.
- [11] Ö Legeza, R. Noack, J. Sólyom, and L. Tincani, “Applications of quantum information in the density-matrix renormalization group”, in **Computational Many-Particle Physics**, ser. Lecture Notes in Physics, vol. 739, Berlin Heidelberg: Springer-Verlag, 2008.
- [12] G.-J. Nam and J. Cong, **Modern Circuit Placement: Best Practices and Results**, 1st. Springer Publishing Company, Incorporated, 2007.
- [13] C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, **Handbook of Algorithms for Physical Design Automation**, 1st. Boston, MA, USA: Auerbach Publications, 2008.
- [14] A. Kahng, J. Lienig, I. Markov, and J. Hu, **VLSI Physical Design: From Graph Partitioning to Timing Closure**. Springer, 2011, ISBN: 0133016153.
- [15] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, **Graph Drawing: Algorithms for the Visualization of Graphs**, 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
- [16] G. Karypis and V. Kumar, “HMETIS 1.5: A Hypergraph Partitioning Package”, Department of Computer Science, Tech. Rep., 1998, <http://www-users.cs.umn.edu/~karypis/metis>.
- [17] J. Durbano and F. Ortiz, “Fpga-based acceleration of the 3d finite-difference time-domain method”, in **Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on**, 2004, pp. 156–163.
- [18] H. Fu, W. Osborne, R. G. Clapp, O. Mencer, and W. Luk, “Accelerating seismic computations using customized number representations on fpgas”, **EURASIP J. Embedded Syst.**, vol. 2009, 3:1–3:13, Jan. 2009.
- [19] J. I. Cirac and F. Verstraete, “Renormalization and tensor product states in spin chains and lattices”, **Journal of Physics A: Mathematical and Theoretical**, vol. 42, no. 50, p. 504 004, 2009.

- [20] C. Vömel, S. Z. Tomov, O. A. Marques, A. Canning, L.-W. Wang, and J. J. Dongarra, “State-of-the-art eigensolvers for electronic structure calculations of large scale nano-systems”, **J. Comput. Phys.**, vol. 227, no. 15, pp. 7113–7124, Jul. 2008.
- [21] W. Barford, **Electronic and Optical Properties of Conjugated Polymers**. Oxford University Press, 2005.
- [22] G. Barcza, Ö. Legeza, K. H. Marti, and M. Reiher, “Quantum-information analysis of electronic states of different molecular structures”, **Phys. Rev. A**, vol. 83, p. 012 508, 1 2011.
- [23] M. Lewenstein, A. Sanpera, V. Ahufinger, B. Damski, A. Sen(De), and U. Sen, “Ultracold atomic gases in optical lattices: mimicking condensed matter physics and beyond”, **Advances in Physics**, vol. 56, no. 2, pp. 243–379, 2007.