Parciális Differenciál Egyenletek Numerikus Szimulációjának Optimális Leképezése Emulált-digitális CNN-UM Architektúrákra

Ph.D. disszertáció tézisei

Kiss András

Tudományos vezető
Dr. Szolgay Péter

Konzulens:
Dr. Nagy Zoltán

Pázmány Péter Katolikus Egyetem
Információs Technológiai Kar

Budapest, 2011

1. Bevezetés, a probléma felvetése

A számítástechnika gyors ütemű fejlődésének köszönhetően előtérbe kerültek azok a problémák, ahol sok processzáló elem van egy geometriailag rendezett struktúrában (tömbprocesszorok). A processzáló elemek nagy számának és nagy sebességüknek köszönhetően immár fontos tulajdonság lett a processzor térbeli elhelyezkedése is. Ezek a processzorok képesek több feladatot egyszerre párhuzamosan futtatni. Tehát annak érdekében, hogy hatékonyan végrehajtódó algoritmust tudjunk létrehozni, figyelembe kell venni a processzáló elemek egymáshoz viszonyított távolságát, a lokalitás precedenciáját is. Ez a diszciplína megköveteli az alapvető műveleteink hardverre történő implementálásának újragondolását.

Disszertációm megírása során olyan megoldásokat kerestem, ahol a felületet, a disszipált teljesítményt minimalizálni és az implementált processzorszámot, a sebességet és a memória hozzáféréseket maximalizálni tudom. Amikor optimális megoldást keresünk parciális differenciál egyenlet megoldó implementációjára, akkor ezen paramétertérben keresünk és a megoldást a paramétertér néhány változójára optimalizáljuk (pl.: sebesség, felület, sávszélesség). A keresést minden esetben szűkíteni fogja az adott hardware sajátosságai.

Számos nehéz feladat ismert, amiket nem tud valós időben számolni a korábbi számítástechnika, vagy csak lassan. Kutatásom célja ezen nehéz problémák vizsgálata, konkrétabban folyadékok és gázok áramlásának szimulációjának vizsgálata, és hozzá adott hardware-re architektúra kidolgozása volt. A dolgozaban elemzésre kerülnek azok a módszerek, amik lehetőséget

teremtenek a nehéz feladatok megoldásának könnyítésére.

2. A kutatás módszerei

Disszertációm célja a parciális differenciál egyenletekre, kiemelten a gázok és folyadékok áramlására, egy metodika kidolgozása volt, mely segítségével optimálisan lehet leképezni ezen problémákat kötött és nem kötött architektúrákra. Ennek elérése érdekében két kísérleti platform vizsgálatával foglalkoztam, az IBM Cell Broadband Engine architektúrával és a Xilinx Field Programmable Gate Array újrakonfigurálható számítógépeivel.

Az IBM Cell processzor egy kötött architektúrát képvisel, amely heterogén processzorokból épül fel. Bár maga a Cell processzor marketing szempontból megbukott, mégis jelentős újításai (pl.: heterogén processzorok, gyűrűs busz struktúra) továbbra is megfigyelhetők a mai modern processzorokban (pl.: IBM Power 7, Intel Sandy Bridge). A processzor tulajdonságainak megfelelően vektorizált adatokkal dolgoztam, melyek lebegőpontos számok voltak. A szoftverek fejlesztéséhez pedig az IBM ingyenes SDK-ját használtam C nyelven programozva.

A Xilinx FPGA-k régóta az újrakonfigurálható számítógépek élvonalába tartoznak. Gyors Konfigurálható Logikai Blokkjainak (CLB) és nagyszámú összeköttetéseinek köszönhetően tetszőleges áramkört valósíthatunk meg. Annak érdekében, hogy bizonyos műveleteket gyorsabban lehessen végrehajtani, dedikált egységeket (pl.: DSP blokkok) is megvalósítottak rajtuk. Az FPGA CLB-je és DSP-je felfogható különböző típusú processzornak és ezek különböző feladatokat oldanak meg hatékonyan. Az

FPGA konfigurálhatóságának köszönhetően tetszőleges számábrázolással számolhatunk. Kutatásom során megvizsgáltam mind a fixpontos és a lebegőpontos számításokat különböző mantisszahosszal annak érdekében, hogy kiderüljön, hogy kvalitatíve elfogadható megoldáshoz hány bites pontosság szükséges. Az architektúrák létrehozásához a Xilinx Foundation ISE szoftvereit használtam és VHDL nyelven írtam le az architektúrákat. Az architektúrák szoftveres szimulációjához a MentorGraphics Modelsim SE nevű programot használtam.

3. Új tudományos eredmények

1. Tézis: *Parciális differenciál egyenletek numerikus szimulációjának optimális leképe-zése inhomogén és újrakonfigurálható architektúrára: Összehasonlítottam egy komplex tér-időbeli dinamika szimulációjának optimális (felület, idő, disszipált teljesítmény) leképezését Xilinx Virtex FPGA-án és IBM Cell architektúrán, és erre egy keretrendszert alkottam. A keretrendszert sikeresen teszteltem egy CFD szimuláció gyorsításával. Célom mindvégig a lehető leggyorsabb feldolgozás volt. Az architektúra ennek megfelelően lett kialakítva figyelembe véve a hardware sajátosságait.*

1.1. **Létrehoztam egy új felület, idő, disszipált teljesítmény, sávszé-lesség szempontjából hatékony architektúrát parciális differenciál egyenletek strukturált rácson történő megoldására. Újraterveztem a Falcon processzor aritmetikai egységeit a diszkretizált parciális differenciál egyenleteknek megfelelően az FPGA dedikált erőforrásaira (BlockRAM, szorzó) optimali-zálva.**

Eljárást adtam a processzáló elemek és a memória között a sávszélesség optimális kezelésének problémájára Xilinx Virtex és IBM Cell architektúrákon, amely lehetővé teszi a processzáló elemek folyamatos ellátását adatokkal.

Mindkét esetben kísérletileg sikerült igazolnom, hogy a működési sebességre jótékonyan hat egy processzor

közeli tárterület kialakítása, mely a feladat dimenziójától függetlenül legalább egy nagyságrendnyi sebességnövekedést biztosít.

1.2. **Kísérletileg igazoltam, hogy egy kötött architektúra, mint az IBM Cell és egy a Xilinx Virtex FPGA-ra tervezett, optimalizált architektúra között egy nagyságrendi gyorsulást lehetséges elérni azonos felület, disszipált teljesítmény és pontosság esetén.** A Xilinx Virtex 5 SX240T 410 MHz-en 8-szor gyorsabb volt, mint a 8 db szinergikus processzor elemet tartalmazó IBM Cell architektúra 3.2 GHz-en CFD-t szimulálva görbült hálón. A disszipált teljesítményük és felületük azonos nagyság-rendbe tartozik, rendre 85 Watt, 253 mm^2 és 30 Watt, 400 mm^2. **Az IBM Cell processzor egy wattra jutó számítási teljesítményét egységnyinek tekintve a Xilinx Virtex 5 SX240T FPGA 8-szoros sebesség-növekedés mellett a számítás hatékonysága 22 szeres. Az egy nagyságrendnyi sebességkülönbség köszönhető az FPGA teljesen párhuzamosan működő műveletvégző egységeinek, illetve a megvalósítható aritmetikai egységek számának.** A ma használatos általános célú mikroprocesszorokhoz (pl.: Intel x86 processzorok) képest az IBM Cell processzor CFD-t szimulálva 2, az FPGA alapú gyorsító 3 nagyság-rendnyi gyorsulást ért el.

2. Tézis: *Parciális differenciál egyenleteket megoldó architektúrák pontosságának vizsgálata FPGA-n: Tézisemben megmutattam, hogy az állapot pontosság-ának csökkentésével jelentős*

sebességnövekedés érhető el. Ez a számítási pontosság csökkentés elfogadható lehet, hiszen nem minden mérnöki alkalmazás követel meg 14-15 helyiértéknyi pontosságot. A pontosság csökkentésével néhány különösen bonyolult probléma is leképezhető az FPGA-ra. A megoldás előírt pontossága és a rácstávolság ismeretének tükrében kidolgoztam egy eljárást, amivel a szükséges minimális számábrázolás pontosság megadható, ezáltal az FPGA-val adható legnagyobb számí-tási teljesítmény érhető el. Természetesen a szükséges pontosság meghatározása csak egzakt megoldás esetén adató meg pontosan és ez kevés esetben áll rendelkezésünkre.

2.1. **Eljárást adtam arra vonatkozólag, hogy hogyan határozható meg az aritmetika minimális szükséges pontossága ismert lépés-köz, térbeli felbontás és a megoldás elvárt pontossága esetén. Analitikus megoldással rendelkező probléma vizsgálata esetén tesztelt eljárást adtam a problémát megoldó architektúra aritmetikai egységeinek pontosságára.** Azon problémák esetén, ahol nincs analitikus megoldása a prob-lémának, ezen csökkentett pontosságú eredményeket a referenciának tekinthető 64 bites lebegőpontos pontossághoz lehet viszonyítani. A módszer továbbá alkalmas arra is, hogy a kerekítési és a levágási hibák ismeretével meghatározható adott pontosság mellett a rács legfinomabb felbontása.

2.2. **Megmutattam, hogy az advekciót leíró parciális differenciál-egyenlet (1) megoldása során hogyan lehet a pontosság ro-vására kevesebb erőforrás felhasz-**

nálásával nagyobb teljes-ítményt elérni.

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}=0 \qquad (1)$$

ahol a t az időt, u egy fentartósági tulajdonságot, c az advekció sebességét jelenti. **A pontosság vizsgá-latára használt advekciós egyenletet megoldó architektúra esetén, az aritmetikai egységek pontossá-gának 40 bitről 29 bitre csökkentésével a felhasznált felületigénye 20-25%-al csökkent az alkalmazott diszkretizációs eljárástól függetlenül.** Sebességnövekedés jelentős része az FPGA-ra implementálható művelet-végző egységeknek nagyobb számának kö-szönhető, az órajel pedig számottevően nem növekszik a pontosság csökkentésével.

2.3. **Kísérlettel igazoltam, hogy megfelelő normalizálás esetén a fixpontos aritmetika adott pontosság mellett további felületnyereséggel jár. A pontosság vizsgálatára használt advekciós egyenletet megoldó architektúra esetén a 33 bit pontos fix és 40 bit pontos lebegőpontos (29 bit mantissza) aritmetikai egység megoldás hibája ugyanabba a nagyságrendbe esik, ellenben az aritmetikai egység felülete fixpontos esetben 15-ödére csökken.** Fixpontos aritmetikát használva a sebességnövekedés jelentős része az FPGA-ra implementálható műveletvégző egységek nagyobb számának köszönhető.

3. Tézis: *Globális Analogikai Vezérlő Egység implementációja emulált-digitális CNN processzorhoz FPGA architektúrán: A*

Falcon processzor különböző számábrázolási pontossággal, különböző méretű template-kkel, több rétegben tudja a CNN dinamikát kiszámolni. Annak érdekében, hogy komplexebb analogikai algoritmusokat is időben hatékonyan végre lehessen hajtani, ki kellett egészíteni egy Globális Analogikai Vezérlő Egységgel (GAPU), továbbá az aritmetikai és logikai műveletek elvégzésére egy Vektor Processzort kellett készíteni. A GAPU-nak nemcsak programszervezési és I/O periféria kezelési feladata van, hanem lokális logikai és aritmetikai műveleteket, vala-mint analóg utasításokat is tudnia kell kezelni. A GAPU feladata továbbá a Falcon processzor megfelelő időzítő- és vezérlő-jeleinek beállítása is.

Az általam javasolt módosítások implementálásra is kerültek és egy pédán keresztül tesztelve is lettek. Az eszközölt módosításoknak és a GAPU, illetve a Vektor Processzorral való kiegészítésnek köszön-hetően létrehozható egy önálló képfeldolgozó rendszer, egy Celluláris Hullámszámítógép.

3.1. **Emulált digitális CNN-UM kialakításához javaslatokat tettem a GAPU felépítésére (pontosságára) vonatkozólag.** A Falcon processzor kibővítése a GAPU-val, annak érdekében, hogy komplexebb algoritmusokat is végre tudjon időben hatékonyan hajtani, kézenfekvő volt az eredeti CNN-UM-nek megfelelően. **A GAPU-t úgy kellett implementálni, hogy ne foglaljon el sok helyet az FPGA-n, és ne lassítsa számot-tevően a Falcon processzor működési sebességét, hiszen a cél a lehető legnagyobb számítási teljesítmény elérése. A GAPU szerepének betöltés-ére egy jól konfigurált**

MicroBlaze-t, dedikált PPC-t, vagy ARM-ot javasoltam használni. Továbbá megfontolásokat tettem a vezérlő és állapot regiszterek fajtájára és a template és állapot memória konfigurálhatóságára is annak érdekében, hogy a rendszer adaptálható legyen a hozzá csatlakoztatott Falcon Processzáló Egység fajtájához. Pl.: ha csak fekete-fehér képen dolgozunk, vagy szürkeárnyalatos képeken is akarunk műveleteket végezni, különböző Falcon egységet célszerű használni.

3.2. **Olyan új architektúrát dolgoztam ki ami lehetővé teszi, hogy a beágyazott mikroprocesszor, a vezérlő áramkörök, a memória és a Falcon processzáló egység különböző órajeleken működhessen. A belső struktúra átalakítása mellet a külső elérést is biztosítottam egy dedikált FIFO-n keresztül. Az új architektúra lehetővé teszi a MicroBlaze, a vezérlő egység és a Falcon processzor számára a külső memó-ria konkurrens hozzáférését.**

Az új generációs FPGA-k dedikált műveletvégző egységei gyorsulnak, de ezt nem követi a beágyazott processzor-architektúra és az alkalmazott busz sebessége. Az új FPGA-k esetében a Falcon processzor nagyobb működési sebességre is képes, mint a mellette be-ágyazott mikroprocesszor- és busz-rendszer.

4. Eredmények alkalmazási területei

4.1. Áramlások szimulációjának alkalmazása

Az összenyomható és összenyomhatatlan folyadékok/gázok szimulációja a parciális differenciál egyenletek megoldásának egyik legérdekesebb területe, mert ezek az egyenletek sok fontos alkalmazásnál megjelennek, pl.: aerodinamika, meteorlológia és óceánográfia. Óceán áramlások modellezése fontos szerepet játszik mind a középtávú időjárás előrejelzésben mind a globális felmelegedés szimulációjában. Általánosságban elmondható, hogy az óceán modellek az óceán változó sűrűségét, légköri nyomását, hőterjedésének a reakcióját írják le. A legegyszerűbb barotropikus (a nyomás a sűrűség függvénye és fordítva) óceán modelben egy terület vízoszlopa vertikálisan egységesített, hogy a vertikálisan különböző horizontális áramlások egy változó értékéhez hozzájussunk. Ennél is pontosabb modellek néhány horizontális réteget is figyelembe vesznek, hogy az óceán mélyebb területeinek mozgását is modellezni tudják. A Princeton Óceán Modell (POM) is egy ilyen modell, ahol szigma koordináta modellt alkalmazunk, amely során a vertikális koordináták a vízoszlop mélységével skálázódnak.

A numerikus áramlástan (Computational Fluid Dynamics - CFD) a tudományos modellezése a gáz- és a folyadék- áramlások időbeli lefolyásának, ami napjaink szuperszámítógépeinek hatalmas számítási teljesítményének határait feszegetik. Összetett alakzatok körüli folyadékok áramlásának szimulálásához napjaink szuperszámítógépeinek is néhány hétre van szükségük. Az általam létrehozott FPGA-ra implementált CFD szimulációs ar-

chitektúra nagyságrendekkel gyorsabb a hagyományos mikroprocesszorokhoz képest.

4.2. Eredmények pontosságának vizsgálata

Mérnöki alkalmazások során dupla pontosságú lebegőpontos számokat alkalmaznak, hogy elkerüljék a kerekítés által okozott hibákat. Érdemes viszont megvizsgálni a szükséges pontosságot, ha a számításra fordítható erőforrás, disszipált teljesítmény, vagy a felület korlátos, vagy a számításokat valós időben kell végrehajtani.

FPGA-ra implementált parciális differenciál egyenleteket megoldó architektúra sebessége nagyban növelhető, ha a megoldó aritmetikai egység pontosságát csökkentjük, ezáltal több processzáló egységet lehet implementálni egységnyi felületen. Ez a tézis ott hasznosul, hogy meg lehet nézni, hogy meddig lehet valós idejű számításokat végezni és ennek milyen korlátai vannak. Egy egyszerűsített advekciós egyenltet megoldó architektúrát vizsgáltam, mely esetében az analitikus megoldás ismert. Egy ilyen analitikus megoldással rendelkező probléma minimális módosításával valószínűsíteni lehet, hogy a megtalált pontosság a felvetett probléma során is alkalmazható marad.

4.3. Globális Analogikai Vezérlő Egység fontossága

Annak érdekében, hogy nagy rugalmasságot biztosíhassunk a fent leírt CNN számításokban, érdekes kérdés, hogy hogy tudunk relatív lassú párhuzamosan végrehajtó egységek szomszédos

összeköttetéseivel, amik reguláris elrendezésűek, nagy számítási teljesítményt elérni. Az architektúra konfigurálható paramétereinek nagyfokú változtathatóságának köszönhetően (pl.: állapot-, template-pontosság, template-k mérete, a feldolgozó egységek sorainak és oszlopainak száma, rétegek száma, képek mérete, stb...) létrehoztam egy implementációt, ami a legalkalmasabb a kiválasztott alkalmazásnak (pl.: áramlások modellezésének szimulációja, kép-, videó-feldolgozás). Eddig, a GAPU kiegészítés nélkül, amikor különböző típusú PDE-ket kellett megoldani, CNN template-k egy csoportját kellett létrehozni a kapcsolódó PC-n: a képet le kellett tölteni az FPGA kártyára (egy viszonylag lassú kommunikációs csatornán keresztül), kiszámítani a tranzienst, és végül az eredményt visszatölteni a kapcsolódó számítógépre, ahol a logikai, aritmetikai és programszervezési lépések hajtódtak végre.

5. Köszönetnyilvánítás

Nem nehéz Doktori fokozatot szerezni, ha tehetséges, motivált, optimista, bölcs emberekkel van körülvéve az ember, akik egy pillanatig sem hezitálnak, hogy utat mutassanak, mikor elakadsz és tudást adnak a kezedbe, hogy legyűrd a nehézségeket. Két ember bíztatott engem az egyetemi éveim után, hogy folytassam tanulmányaimat és folyamatosan haladásra késztettek, hogy elérjem szerény céljaimat. Ismerték az utam, mert Ők már jártak rajta. Ez a munka nem jöhetett volna létre a témavezetőm és mentorom Szolgay Péter Professzor Úr és konzulensem és barátom Dr. Nagy Zoltán nélkül.

Hálás vagyok a közeli munkatársaimnak is, hogy kisegítettek a nehéz helyzetekben, Dr. Vörösházi Zsoltnak, Kocsárdi Sándornak, Kincses Zoltánnak, Sonkoly Péternek, Füredi Lászlónak és Nemes Csabának.

Továbbá szeretnék köszönetet mondani a tehetséges kollégáimnak is, akik folyamatosan szenvedtek az őrült ötleteimtől és akik ezért nem kergettek el vasvillával, különösen Bankó Évának, Hermann Petrának, Soós Gergelynek, Hegyi Barnának, Weiss Bélának, Szolgay Dánielnek, Bérci Norbertnek, Benedek Csabának, Tibold Róbertnek, Pilissy Tamásnak, Treplán Gergelynek, Fekete Ádámnak, Veres Józsefnek, Tar Ákosnak, Tisza Dávidnak, Cserey Györgynek, Oláh Andrásnak, Feldhoffer Gergelynek, Giovanni Pazienzának, Kósa Endrének, Balogh Ádámnak, Kárász Zoltánnak, Kovács Andreának, Kozák Lászlónak, Szabó Vilmosnak, Varga Balázsnak, Fülöp Tamásnak, Tornai Gábornak, Zsedrovits Tamásnak, Horváth Andrásnak, Koller Miklósnak, Gergelyi Domonkosnak, Kovács Dánielnek, Laki Lászlónak, Radványi Mihálynak, Rák Ádámnak, Stubendek Attilának és még a többi fel nem sorolt kollégámnak.

Hálás vagyok a Pázmány Péter Katolikus Egyetemnek, ahol a doktori éveimet töltöttem és a Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutatóintézetnek (MTA-SZTAKI).

Köszönettel tartozom Keserű Katalinnak a MTA-SZTAKIból és a Pázmány Péter Katolikus Egyetem különböző irodai dolgozóinak a gyakorlatias és hivatalos segítségnyújtásukért.

Nagyon hálás vagyok Édesanyámnak és Édesapámnak és az egész családomnak, akik mindig tolerálták a ritka találkozásainkat és minden lehetséges úton támogattak.

6. Publikációs lista

6.1. A szerző folyóiratbeli publikációi

[1] Z. Nagy, L. Kék, Z. Kincses, A. Kiss, and P. Szolgay, „Toward Exploitation of Cell Multi-processor Array in Time-consuming Applications by Using CNN Model," *International Journal of Circuit Theory and Applications*, vol. 36, no. 5-6, pp. 605–622, 2008.

[2] Z. Vörösházi, A. Kiss, Z. Nagy, and P. Szolgay, „Implementation of Embedded Emulated-Digital CNN-UM Global Analogic Programming Unit on FPGA and its Application," *International Journal of Circuit Theory and Applications*, vol. 36, no. 5-6, pp. 589–603, 2008.

6.2. A szerző nemzetközi konferencia publikációi

[3] Z. Vörösházi, Z. Nagy, A. Kiss, and P. Szolgay, „An Embedded CNN-UM Global Analogic Programming Unit Implementation on FPGA," in *Proceedings of the 10th IEEE International Workshop on Cellular Neural Networks and their Applications*, (Istanbul, Turkey), CNNA2006, August 2006.

[4] Z. Vörösházi, A. Kiss, Z. Nagy, and P. Szolgay, „FPGA Based Emulated-Digital CNN-UM Implementation with GAPU," in *Proc. of CNNA'2008*, (Santiago de Compostella), pp. 175–180, 2008.

[5] Z. Nagy, L. Kék, Z. Kincses, A. Kiss, and P. Szolgay, „Toward Exploitation of Cell Multi-Processor Array in Time-Consuming Applications by Using CNN Model," in *Proc. of CNNA'2008*, (Santiago de Compostella), pp. 157–162, 2008.

[6] Z. Vörösházi, A. Kiss, Z. Nagy, and P. Szolgay, „A Standalone FPGA Based Emulated-Digital CNN-UM System," in *Proc. of CNNA'2008*, (Santiago de Compostella), 2008.

[7] Z. Nagy, A. Kiss, S. Kocsárdi, and Á. Csík, „Supersonic Flow Simulation on IBM Cell Processor Based Emulated Digital Cellular Neural Networks," in *Proc. of ISCAS'2009*, (Taipei, Taiwan), pp. 1225–1228, 2009.

[8] Z. Nagy, A. Kiss, S. Kocsárdi, and Á. Csík, „Computational Fluid Flow Simulation on Body Fitted Mesh Geometry with IBM Cell Broadband Engine Architecture," in *Proc. of ECCTD'2009*, (Antalya, Turkey), pp. 827–830, 2009.

[9] Z. Nagy, A. Kiss, S. Kocsárdi, M. Retek, Á. Csík, and P. Szolgay, „A Supersonic Flow Simulation on IBM Cell Processor Based Emulated Digital Cellular Neural Networks," in *Proc. of CMFF'2009*, (Budapest, Hungary), pp. 502–509, 2009.

[10] A. Kiss and Z. Nagy, „Computational Fluid Flow Simulation on Body Fitted Mesh Geometry with FPGA Based Emulated Digital Cellular Neural Networks," in *Proceedings of 12th International Workshop on Cellular Nanoscale Networks and their Applications*, (Berkeley, CA, USA), CNNA2010, 2010.

[11] L. Füredi, Z. Nagy, A. Kiss, and P. Szolgay, „An Improved Emulated Digital CNN Architecture for High Performance FPGAs," in *Proceedings of the 2010 International Symposium on Nonlinear Theory and its Applications*, (Krakow, Poland), pp. 103–106, NOLTA2010, 2010.

[12] C. Nemes, Z. Nagy, M. Ruszinkó, 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*, (Krakow, Poland), pp. 99–102, NOLTA2010, 2010.