Opinnäytetyö (AMK)

Elektroniikka

Tietoliikennejärjestelmät

2015

Mikael Rekola

NÄYTÖNOHJAIMEN SUORITUSKYKYTESTIT

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Elektroniikan koulutusohjelma | Tietoliikenne

2015 | 36 + 1 liite

Ohjaaja: Yliopettaja Juha Nikkanen

Mikael Rekola

NÄYTÖNOHJAIMEN SUORITUSKYKYTESTIT

Opinnäytetyössä on perehdytty Nvidia oy suunnittelemiin näytönohjain arkkitehtuureihin ja asioihin, jotka vaikuttavat näytönohjainten suorituskykyyn. Työssä käydään arkkitehtuureista läpi vain Fermi-, Kepler- ja Maxwell-arkkitehtuurit. Pascal-arkkitehtuurista selvitetään vain pääasiat, koska Pascal-arkkitehtuuria ei ole vielä julkaistu.

Opinnäytetyössä tutkittiin, miten ajurit, ohjelmointirajapinnat ja laitteiston pullonkaulat vaikuttavat näytönohjaimen suorituskykyyn. Aluksi ajurien suorituskykyä testattiin yli 20 eri ajurilla 3DMark suorituskykytestissä. Peleissä testattiin 5 eri ajuria. Testeissä huomattiin, että ajureilla on mahdollista parantaa suorituskykyä peleissä 30 %:lla, mutta vain 5 % suorituskykytestissä. Joskus Nvidian julkaisemat GameReady-ajurit eivät parantaneet suorituskykyä laisinkaan.

Ohjelmointirajapinnoista huomattiin, että DirectX on suuressa suosiossa peleissä Windows-käyttöjärjestelmissä. Yleensä DirectX suoriutuu paremmin kuin OpenGL, mikä voi johtua huonommista ajureista ja optimoinneista OpenGL:lle. DirectX suorituskyky oli yleensä noin 20 % parempi kuin OpenGLn. Dota 2 -peliä testattaessa OpenGL suoriutui paremmin kuin DirectX 11, koska OpenGL oli optimoitu paremmin, mutta DirectX 9 oli keskiarvollisesti hiukan nopeampi.

Laitteiston rajoituksissa huomattiin, että tehonrajoittajan rajan ja grafiikkasuorittimen taajuuden kasvattaminen paransi suorituskykyä 12 %, mutta samalla se nostaa näytönohjaimen lämpötilaa. Kun näytönohjaimen lämpötila nousee liian korkealle, grafiikkasuoritin alkaa pienentämään omaa kellotaajuuttaan. Näytönohjaimen yksi pullonkauloista on videomuisti, jota tarvitaan datalle. Modernit pelit rupeavat tarvitsemaan suurempia määriä ja kaistanleveydeltään nopeampia videomuisteja. Seuraavan sukupolven näytönohjaimet yrittävät parantaa tämän ongelman 3D muistilla, joka parantaa huomattavasti kaistanleveyttä ja mahdollistaa suuremmat määrät muistia.

ASIASANAT:

näytönohjain, grafiikkasuoritin, Nvidia, OpenGL, DirectX

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics | Telecommunications systems

2015 | 36 + 1 attachment

Instructor: Juha Nikkanen, Lic. Tech. Principal Lecturer

Mikael Rekola

GRAPHICS CARD PERFORMANCE TESTS

This thesis focuses on graphics card architectures designed by Nvidia and explores factors that affect the performance of graphics cards. This thesis examines only Fermi, Kepler and Maxwell architectures. In case of Pascal architecture, the thesis only discusses differences because the Pascal architecture has not been published.

The objective of this thesis was to study how drivers, application programming interfaces, and hardware bottlenecks affect graphics card performance. Firstly, driver performance was tested with over 20 different drivers in GPU benchmark utility and 5 drivers for games. It was found that driver performance can be improved by 30% in games but only 5% in GPU benchmark utility. However sometimes GameReady drivers did not display improved performance at all.

In application programming interface performance tests, it was observed that DirectX is a very popular application programming interface in games for Windows. Usually, DirectX performs better than OpenGL because DirectX has been used as a lead API and received better optimization. The performance of DirectX was usually 20% better than that of OpenGL. In Dota 2 OpenGL was better than DirectX 11 because of better optimization but DirectX 9 still performs slightly better.

On the basis of graphics card hardware testing, it was discovered that increasing the power limit and GPU frequency improve performance by 12 % but, at the same time this raises the temperature of the graphics card. When the graphics cards temperature rises too high, the GPU begins to reduce its own clock frequency to remain cool. One of the bottlenecks is video memory, which is needed for data. Modern games uses more and more video memory and after the memory is full of data, the game can crash or slowdown. In the next generation graphics cards this problem will be mostly solved by 3D memory which increases memory bandwidth and amount of memory.

KEYWORDS:

Graphics card, graphics processing unit, Nvidia, OpenGL, DirectX

SISÄLTÖ

LIITTEET

KUVAT

TAULUKOT

KÄYTETYT LYHENTEET

ALU	Arithmetic logic unit, aritmeettinen logiikkayksikkö
CPU	Central processing unit, suoritin
CWD	CUDA work distributor, CUDA työn jakelija
DRAM	Dynamic random access memory, dynaaminen RAM-muisti
ECC	Error correcting code, virheen korjauskoodi
FMA	Fused multiply-add, sulautettu kerto- ja lisäys-yksikkö
FPA	Floating point arithmetic, liukulukuaritmetiikka
FPS	Frames per second, kehysnopeus
FPU	Floating point unit, liukulukuyksikkö
GMU	Grid managment unit, verkon hallintayksikkö
GPU	Graphics processing unit, grafiikkasuoritin
HPC	High computing performance, korkea laskentateho
MPI	Message passing interface, viestin kuljettajan rajapinta
PTX	Paraller thread execution, rinnakkaisten säikeiden suoritus
RDMA	Remote direct memory access, linkki kahden tietokoneen välillä
ROP	Raster operations pipeline, rasterointioperaatio yksikkö
SFU	Special function units, erikoisfunktioyksikkö
SM	Streaming multiprocessor, Fermi grafiikkapiirin osa
SMM	Streaming multiprocessor Maxwell, Maxwell grafiikkapiirin osa
SMX	Next generation streaming multiprocessor, Kepler grafiikkapiirin osa
VXGI	Voxel global illumination, vokseli valaistustekniikka

1 JOHDANTO

Näytönohjainten tekniikka ja käyttökohteet ovat vuosien saatossa muuttuneet. Tässä työssä perehdytään Nvidian suunnittelemiin arkkitehtuureihin ja tutkitaan, mitkä asiat vaikuttavat näytönohjainten suorituskykyyn. Vaikuttaviin asioihin kuuluvat ajurit, ohjelmointirajapinnat ja laitteiston pullonkaulat. Työssä käydään myös hieman läpi uusia käyttökohteita GPGPU-laskennalle.

Tämä työ keskittyy pelkästään Nvidian suunnittelemiin arkkitehtuureihin, koska suorituskykymittaukset tehtiin Nvidian suunnittelemalla GeForce GTX 670 -näytönohjaimella ja AMD:n lisääminen olisi laajentanut työtä liikaa. Työssä käydään aluksi läpi, mitä näytönohjain sisältää ja miten se toimii. Esimerkkeinä käytettiin PNY XLR8 GeForce GTX 670 -näytönohjainta ja Nvidian referenssimallia.

Luvussa 3 käsitellään Nvidia tuotteita ja Nvidian uusimmat arkkitehtuurit. Työssä esitetyt arkkitehtuurit ovat Fermi, Kepler, Maxwell ja Pascal. Fermi- ja Kepler-arkkitehtuurista käsitellään vain uusin ja kokonainen arkkitehtuuri, koska valmistajat voivat muokata kokoonpanoa mieleisekseen. Maxwell-arkkitehtuurista käsitellään vain GM204-versio. Pascal-arkkitehtuuri on Nvidian tuleva arkkitehtuuri, josta käydään vain kolme pääasiaa läpi, jotka ovat mixed-precision computing, 3D muisti ja NVLink. Nvidia ei ole julkaissut enempää tietoa tulevasta Pascal-arkkitehtuurista

Lopuksi käsitellään ajureiden, ohjelmointirajapintojen ja laitteiston pullonkaulojen vaikutusta peleihin ja suorituskyky testeihin. Testit tehtiin 3DMark FireStrike-suorituskykytestillä, Unigine Heaven/Valley Benchamark -testeillä ja Half Life 2, Dota 2- sekä Bioshock Infinite -peleillä. Osa OpenGL -testeistä tehtiin myös Ubuntun käyttöjärjestelmässä.

Ville Suvanto on tehnyt opinnäytetyön prosessorin korvaamisesta näytönohjaimella yleishyödyllisissä ohjelmissa [1]. Työssä testataan sovellusten toimintaa näytönohjaimella prosessorin sijaan ja käsitellään Nvidian- ja AMD- näytönohjainarkkitehtuureja (Fermi ja TeraScale 2) ja ohjelmointikirjastoja.

2 NÄYTÖNOHJAIN

Näytönohjain on oleellinen osa tietokonetta, koska se vastaa grafiikan piirtämisestä näytölle. Näytönohjain on tyypillisesti erillinen komponentti tietokoneessa, mutta useimmissa prosessoreissa eli CPU:ssa on integroitu näytönohjain. Aiemmin integroitu näytönohjain oli emolevyssä. Huonomman suorituskykynsä takia integroitu näytönohjain soveltuu vain vanhoihin ja kevyisiin peleihin [2].

Kuvassa 1. on esiteltynä PNY:n valmistama näytönohjain, joka perustuu Nvidian suunnittelemaan GeForce GTX 670 näytönohjaimen referenssipiiriin. PNY XLR8 versiossa taajuudet ovat samat kuin referenssipiirillä ja jäähdyttäjänä käytetään samankaltaista tuuletinta. [3]

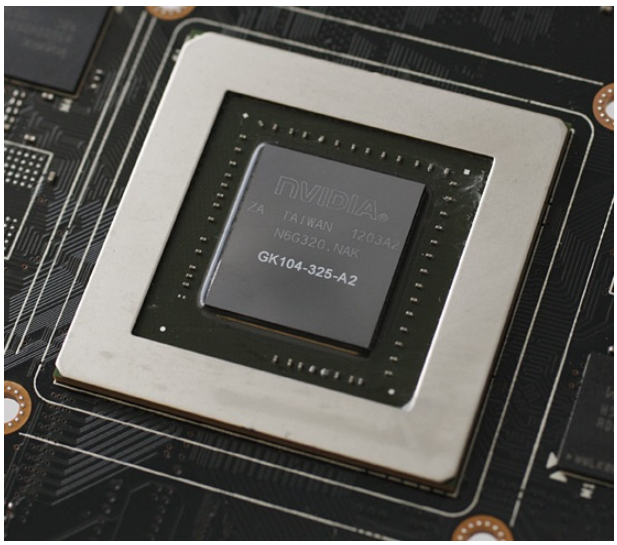
Kuva 1. Nvidia Geforce GTX 670 näytönohjain.

2.1 Näytönohjain sisältä

Kuvassa 2. on kuvattuna näytönohjaimen sisältö. Tärkeimmät näytönohjaimen osat ovat grafiikkasuoritin eli GPU, näyttömuisti, jäähdytyslaitteisto ja ulostuloportit. Näytönohjain kiinnitetään emolevyssä olevaan PCIe 3.0 -tiedonsiirtoväylään, jotta saadaan käyttöön mahdollisimman suuri siirtonopeus [4].

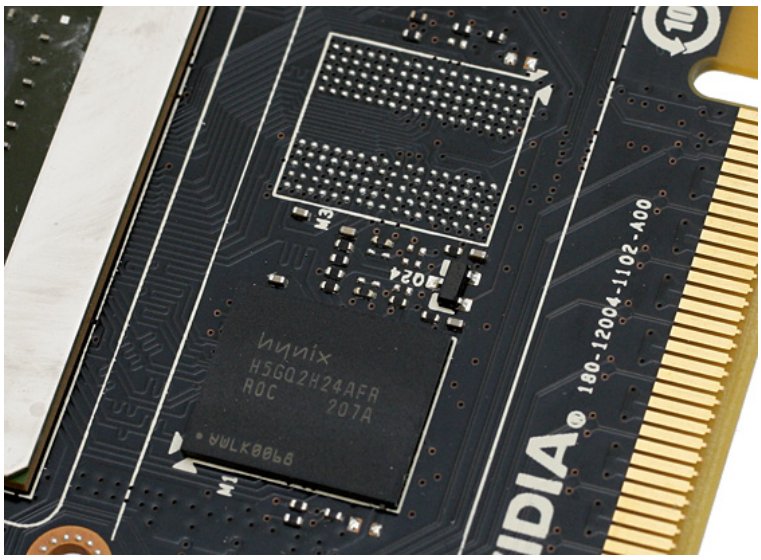
Kuva 2. GTX 670 näytönohjaimen sisältö. [2]

Näytönohjaimen tärkein osa on grafiikkasuoritin, joka hallitsee kuvan piirtoa. Nvidian Geforce GTX 670 -referenssimallissa kuvassa 3. grafiikkasuoritin sisältää 1 344 CUDA -ydintä eli varjostinta, jotka on ryhmitelty 7 SMX:ään eli Keplerin streaming multiprocessoriin. Kellotaajuutena toimii 915 MHz ja GPU Boost -ominaisuudella taajuus kasvaa 980 MHz:iin, mutta käytännössä kellotaajuus voi nousta yli 1 000 MHz:iin. [5]



Kuva 3. Nvidia GeForce GTX 670 grafiikka suoritin. [6]

GeForce GTX 670:ssä käytetään GDDR5-muistipiirejä, jotka toimivat 1 502 MHz:n kellotaajuudella eli yhteensä 6 008 MHz [6]. Kuvassa 4. on kuvattuna yksi muistipiiri ja vapaa paikka toiselle muistipiirille.



Kuva 4. GTX 670 muistipiiri. [6]

Piirejä on neljä kappaletta ja jokainen on 256 Mt:n kokoinen eli yhteensä 2 Gt ja ovat sijoitettu grafiikka suorittimen ympärille. Referenssilevy tukee myös 4 gigabitin muistia. Muistiväylä on 256-bittinen ja väylänsiirtonopeus on 192.2 Gt/s. [6]

Kuvasta 5. nähdään näytönohjaimen liittimet, jotka sijaitsevat laitteen takaosassa. Kuvassa näkyy myös tuuletusritilä, josta lämminilma puhalletaan ulos.

Kuva 5. GTX 670 liittimet. [6]

Näyttöliittiminä on kaksi kaksilinkkistä DVI-liitintä, joihin voidaan liittää näyttö, joka tukee 2 560 x 1 600 resoluutiota. Näiden vieressä on HDMI, joka on 1.4a-standardin mukainen, ja DisplayPort-liitin, joka on 1.2-standardin mukainen. HDMI tukee 3 GHz:n taajuutta ja Stereo 3D:tä. HDMI:n maksimiresoluutio on 3 840 x 2 160. [6]

Virtansa GeForce GTX 670 saa virtalähteestä kahden kuusipinnisen virtaliittimen avulla. On suositeltavaa, että virtalähteenä olisi vähintään 500 W, jos haluaa ylikellottaa grafiikkasuorittimen ja jos on monia virtaa kuluttavia laitteita koneessa kiinni. NVIDIA ilmoittaa GeForce GTX 670 -grafiikkasuorittimen TDP-arvoksi 170 W [5]. TDP ei tarkoita tehonkulutusta vaan suurinta lämpötehon poistoa, minkä jäähdytysjärjestelmä pystyy poistamaan [7].

2.2 Grafiikan piirtäminen

Grafiikkasuoritin sisältää yleensä yli tuhat varjostinprosessoria, joita Nvidia kutsuu CUDA -prosessoreiksi. Varjostinprosessorit piirtävät grafiikan näytölle, kun taas grafiikkasuoritin jakaa työn eri varjostinprosessoreille. [8]

Ennen varjostinprosessoreita näytönohjaimet piirsivät monikulmioita ja näytönohjaimen nopeus laskettiin siitä, kuinka monta polygonia pystyttiin piirtämään sekunnissa. Varjostinprosessoreilla pystytään tekemään muutakin kuin monikulmion värittämistä. Varjostinohjelmalla pystytään esimerkiksi mallintamaan veden pinnan käyttäytymistä. Prosessori vain ilmoittaa, mihin kuuluisi vettä, jolloin näytönohjain laskee veden pinnan käyttäytymistä itsenäisesti. Directx 10 myötä varjostinprosessorit korvasivat verteksi- ja pikselivarjostinyksiköt. Ne ovat yhdistettynä varjostinprosessorien alle helpottaen ohjelmointia. [8]

Muistin suuruus ja nopeus vaikuttaa kuinka paljon yksityiskohtia kuvassa voilla. Kapeassa muistiväylässä tietoa ei pystytä siirtämään tarpeeksi nopeasti grafiikka suorittimelle. Tyypillinen muistiväylän koko on 256-bittinen. [8]

3 NVIDIA OY

Nvidia perustettiin vuonna 1993 ja se kehitti ensimmäisen GPU:n vuonna 1999. Ajan kuluessa GPU -laskenta on tullut peleistä elokuvatuotantoon, tuotesuunnitteluun, kuvakäsittelyyn ja lisättyyn todellisuuteen. [9]

Vuonna 1999 Nvidia julkaisi GeForce 256 -näytönohjaimen, jossa oli ensimmäinen GPU ja kannettaviin tuli GeForce2 vuonna 2000. Vuonna 2001 julkaistiin GeForce 3, joka mahdollisti GPU:n ohjelmoinnin. Vuonna 2004 julkaistiin SLI -teknologia, joka mahdollistaa monen GPU:n liittämisen yhteen. Vuonna 2006 Nvidia julkisti CUDA -ytimet. Fermi -arkkitehtuuri julkaistiin vuonna 2009 ja vuonna 2012 markkinoille tuli Kepler -arkkitehtuuri. Maxwell -arkkitehtuuri julkaistiin vuonna 2014. [10]

Nvidian tuotteet keskittyvät pelaamiseen, ammattitason visualisointiin ja suunnitteluun ja suorituskykyä vaativaan laskentaan joihin tarjotaan prosessorit, ohjelmat, työkalut, markkinointi, ammattitaito ja palvelut. [9]

Pelaajille Nvidia tarjoaa näytönohjainten lisäksi GeForce Experience™ -ohjelman, jolla on 25 miljoonaa käyttäjää. Ohjelmalla pystytään optimoimaan pelit automaattisesti omalle kokoonpanolle ja ohjelma huolehtii automaattisesti ajurien päivityksistä. Ohjelmalla pystytään myös tallentamaan videoita peleistä ja suoratoistaa kuvaa Twitch:iin tai Nvidia SHIELD -laitteille. [9]

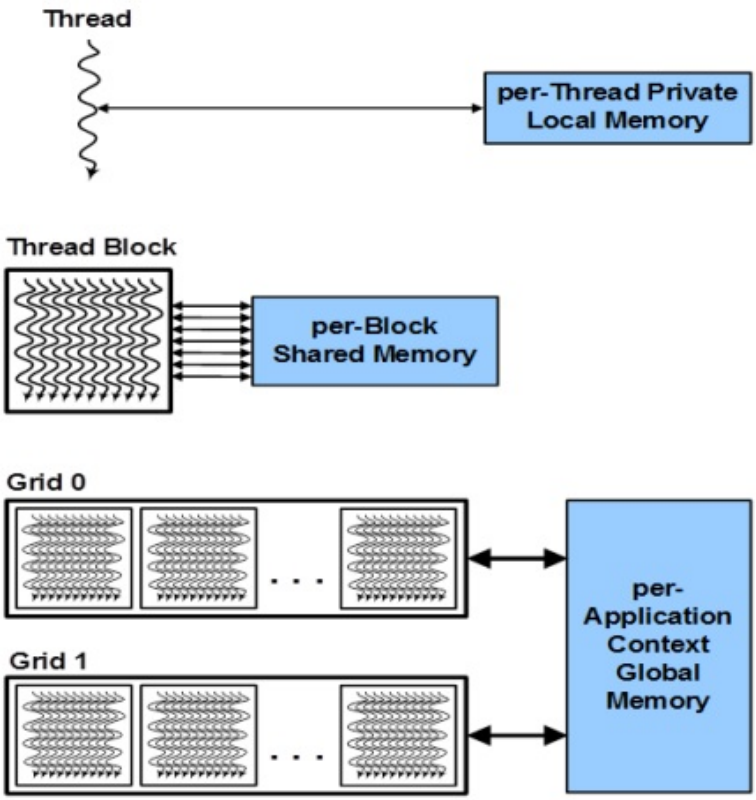
Ammattitason visualisoinnissa käytetään Quadro -näytönohjaimia, joita käytetään suunnittelussa, lääketieteellisissä kuvauksissa ja digitaalisen sisällön luonnissa ja filmaamisessa. Quadroa käytetään 80 % maailman työasemista ja sen työkalut ja algoritmit ovat sulautettu melkein kaikkiin suuriin suunnittelutyökaluihin. [9]

Tieteellisissä tutkimuksissa käytetään Tesla -näytönohjaimia, joilla mallinnetaan aivoja ja lääkkeen löytämistä sairauksia vastaan. Nvidia tarjoaa työkaluja, kirjastoja ja ammattilaisia avuksi. Nvidian näytönohjaimia nähdään monessa maailman nopeimmissa supertietokoneissa. Nvidia on tuonut grafiikkasuorittimet palvelinkeskuksiin. Ciscon, Dellin, Fujitsun, Hitatchin, HP:n, IBM;n ja muiden palvelimet käyttävät Nvidian GRID -teknologiaa virtualisoimaan tietokoneita liikkuvalle työvoimalle ja nopeuttavat ohjelmistoja, kuten suuren datan analysointia. [9]

Nvidia on tuonut mobiilimarkkinoille Tegra -malliston, jota käytetään puhelimista auton viihdejärjestelmiin ja avusteisiin, kuten ohjaamon digitalisointi ja avustaminen itsestään ajavissa autoissa. [9]

3.1 CUDA-yhdistelmäalusta

CUDA on NVIDIAn kehittämä laitteiston ja ohjelmiston yhdistelmäalusta, joka mahdollistaa NVIDIA grafiikkasuorittimen suorittaa ohjelmia, jotka ovat kirjoitettu C, C++, Fortran ja muilla ohjelmistokielillä. CUDA-ohjelma herättää rinnakkaisia funktioita eli kerneleitä, jotka suoritetaan useassa rinnakkaisessa säikeessä. Säie koostuu suoritettavan prosessin osista. Ohjelmoija tai kääntäjä järjestelee nämä säikeet säielohkoihin ja säielohkoverkkoihin, kuten kuvasta 6. nähdään. [11]



Kuva 6. CUDA:n säiehierarkia. [8]

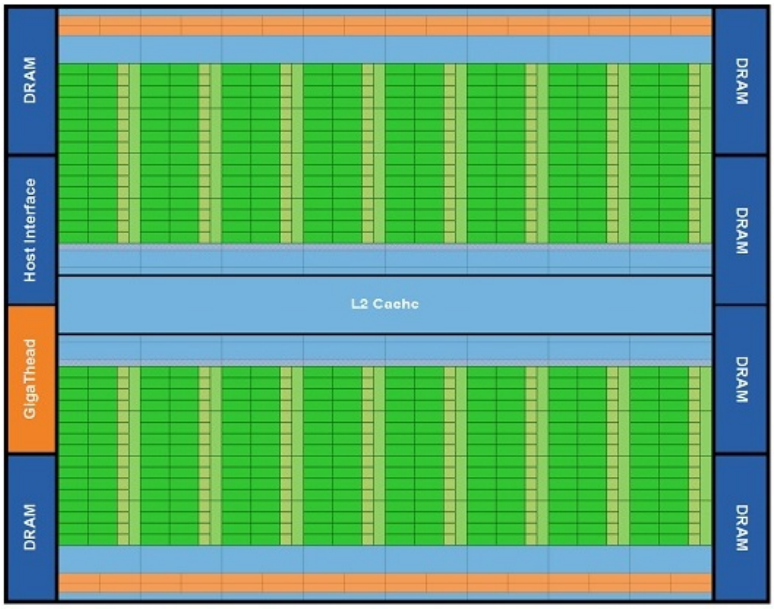
Grafiikkasuoritin suorittaa yhden tai useamman kerneliverkon ja SM-yksikkö eli streaming multiprocessor suorittaa yhden tai useamman säielohkon. SM-yksikössä olevat CUDA ytimet ja muut suorittimet suorittavat säiekäskyt. SM suorittaa säikeet 32 kappaleen säieryhmissä joita kutsutaan warpeiksi. Ohjelmoijat voivat olla huomioimatta

warp suorituksen toiminnallisen virheettömyyden vuoksi ja keskittyä ohjelmoimaan itsenäisiä skalaarisäikeitä. Ne voivat parantavat suorituskykyä, koska säikeet warpissa suorittavat saman koodipolun ja pääsevät käsiksi muistiin. [11]

3.2 Fermi-arkkitehtuuri

Nvidian ensimmäinen Fermi-arkkitehtuuriin perustuva GPU sisälsi 3.0 miljardia transistoria ja 512 CUDA-ydintä. CUDA-ytimet muodostavat 16 SM:ä eli streaming multiprocessoria, joista jokainen sisältää 32 ydintä. GPU:lla on kuusi kappaletta 64-bitin muistiosioita 384-bitin muistiväylällä, jotka tukevat maksimissaan 6 Gt GDDR5 DRAM -muistia. [12]

Kuvassa 7. on esiteltynä Fermin-arkkitehtuuri, jossa SM:t ovat sijoitettu L2 välimuistin ympärille. Jokainen SM sisältää vuorottajan ja lähettäjän, joita kuvataan oranssilla. Vihreä kuvastaa suoritusyksiköitä ja vaaleansinen kuvastaa rekisteritiedostoja ja L1 välimuistia. [12]



Kuva 7. Fermin arkkitehtuuri. [12]

3.2.1 Ohjelmointimalli

Fermin-arkkitehtuuria hallinnoi monitasoinen ohjelmointimalli, joka mahdollistaa ohjelmointikehittäjille keskittymisen algoritmien suunnitteluun, kuin algoritmin kartoituksen laitteistolle. [12]

Säikeet ovat ryhmitelty säielohkoihin, jotka sisältävät suurimmillaan 1 536 säiettä. Jokainen säie lohkossaan ajetaan yhdellä SM:llä, jolloin säikeet pystyvät tekemään yhteistyötä ja jakamaan muistia. Säielohkot pystyvät koordinoimaan globaalin jaetun muistin käyttöä keskenään ja ne voidaan suorittaa rinnakkain tai järjestyksessä. [13]

Säielohkot ovat jaettu 32 säikeen ryhmiin eli warpeihin. Tämä on olennainen yksikkö SM:n lähettäjänä. Fermissä kaksi warppia voidaan laskea ja suorittaa rinnakkain, vaikka olisivat eri säielohkoista. Tämä parantaa laitteiston hyödyntämistä ja energiatehokkuutta. [13]

Jokaisella säikeellä ja säielohkolla on omat tunnisteensa, jotka määrittävät niiden suhteen kerneliin eli ydinfunktioille. Näitä tunnisteita käytetään indeksinä syöte ja tulos datalle ja jaetun muistin sijainnille. [13]

Fermi tukee monen kernelin samanaikaista suorittamista samalta ohjelmalta. Kernelit jaetaan SM:ien kesken. Tällä saavutetaan, ettei kerneli käytä vain pientä osaa laitteesta. [13]

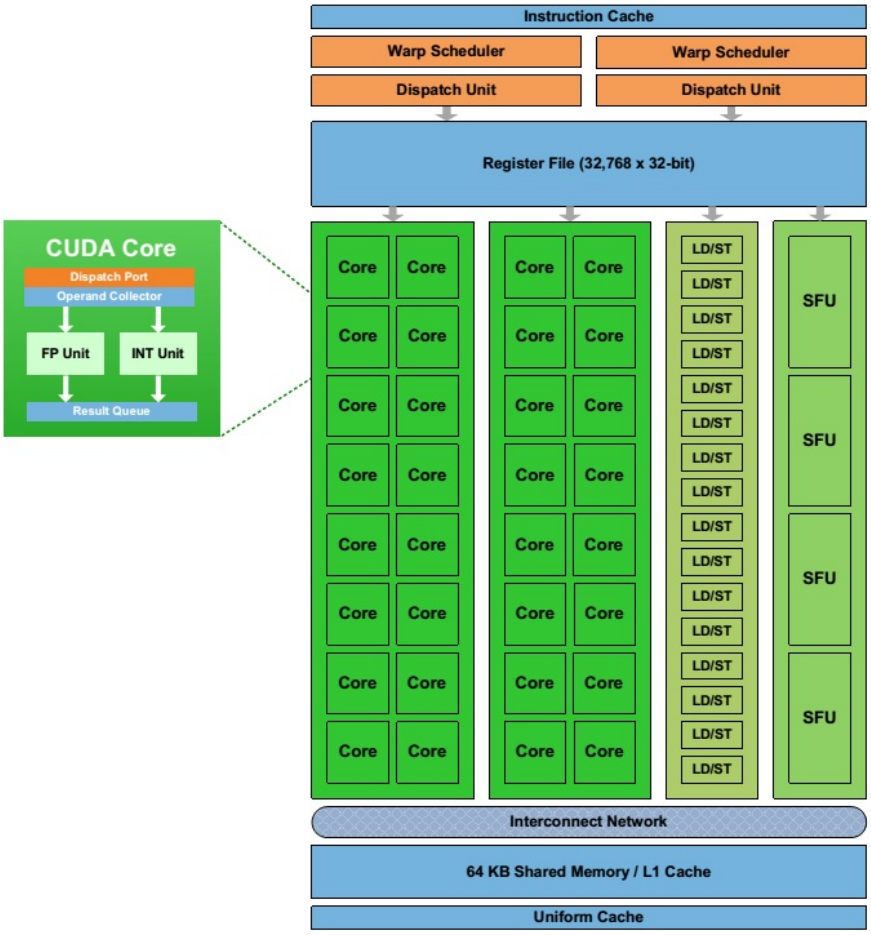
Vanhempaan sukupolveen nähden Fermi on nopeampi vaihtamaan sovelluksesta toiseen. Tämä vaihtoaika on tarpeeksi lyhyt, jolloin Fermi pystyy ylläpitämään korkeaa hyötykäyttöä, vaikka suorittaisi monta laskenta- ja grafiikka -koodia samanaikaisesti. Vaihtamista hoitaa sirutason GigaThread-säiejärjestelijä, joka hoitaa samanaikaisesti 1536 aktiivista säiettä jokaiselle SM:lle. [13]

3.2.2 Streaming Multiprocessor

SM sisältää 32 CUDA ydintä, joita on neljä kertaa enemmän kuin aiemmissa malleissa. Jokaisessa CUDA-ytimessä on ALU ja FPU. ALU eli arithmetic logic unit on aritmeettinen logiikkayksikkö, joka hoitaa aritmeettisia ja bittitason loogisia operaatioita. FPU eli floating-point unit on liukulukuyksikkö, joka suoritta liukulukuoperaatioita. Aiemmin FPU:na oli IEEE 754.1985 FPA. Fermissä käytetään IEEE 754-2008 liukulukustandardia tarjoten FMA:n käskyt yhdellä tai kahdella liukulukutarkkuudella eli 32- tai 64 -bittisenä. FMA eli fused multiply-add suorittaa kerto-ja lisäyslaskut liukulukuoperaatioina. [12]

Kuvassa 8. on kuvattuna SM:n rakenne johon kuuluu 32 ydintä, 16 lataus-tallennusyksikköä, neljä SFU:ta, jotka suorittavat erityisiä operaatioita, kuten sini, kosini, potenssi ja

käänteisarvoja, 32 kt muunneltavaa RAMia eli käyttömuistia ja säiekontrollilogiikka. Jokainen ydin sisältää liukuluvun ja kokonaisluvun suorittavat yksiköt. [12]



Kuva 8. SM:n rakenne [13]

Jokainen ydin voi suorittaa yhdentarkkuuden FMA-operaation per kellojakso ja kaksoistarkkuuden kahdella kellojaksolla. Sirutasolla Fermi suorittaa 8 kertaa enemmän kaksoistarkkuuden operaatioita per kellojakso kuin aiempi sukupolvi. FMA tuen hyötynä on parantunut tarkkuus matemaattisissa operaatioissa. [13]

Fermissä on uudenlainen kokonaisluku ALU, joka tukee 32-bittistä standardin mukaista tarkkuutta jokaiselle ohjelmointikielen käskylle. ALU tukee myös 64-bittistä ja laajennettuja tarkkuusoperaatioita. [12]

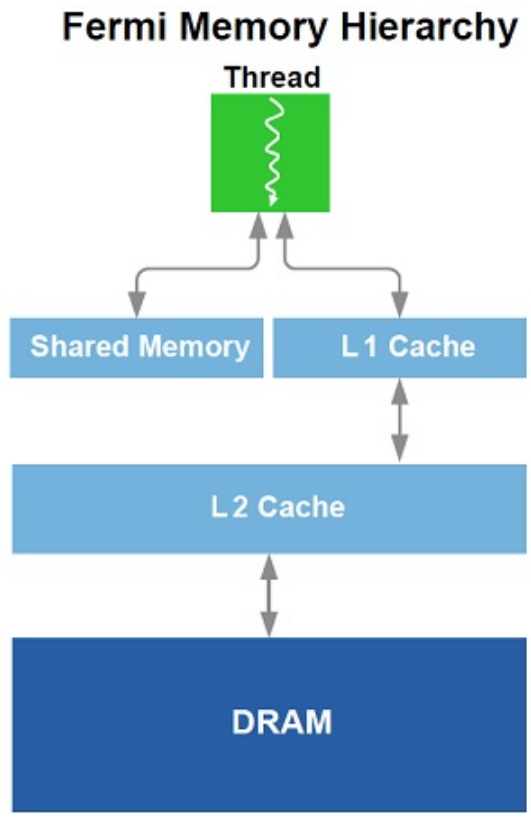
Muistioperaatioita käsittelevät lataus-tallennusyksiköt joita on 16 kappaletta jokaisessa SM:ssä jolloin lataus-tallennuskäskyt voivat viitata muistiin kaksiulotteisessa taulukossa. Dataa voidaan kääntää formaatista toiseen kuten liukuluvusta kokonaisluvuksi ja toisinpäin, joka parantaa GPU:n optimointia. [13]

3.2.3 Välimuisti ja hierarkia

Fermi osaa käyttää osan paikallisesta muistista L1 välimuistina globaalille muistiviitteelle. Paikallinen muisti on 64 kt kokoinen ja se voidaan jakaa 16 kt:un välimuistia ja 48 kt jaettua muistia ja toisinpäin. Tavallisesti jaettua muistia käytetään SM:n paikallisena muistina tarjoten matalan viiveen datan hakemiselle. [13]

Fermi GPU sisältää L2 välimuistin, jonka koko on 768 kt. L2 välimuisti mahdollistaa keskeytymättömät luku-muokkaus-kirjoitusoperaatioita, koska ne toimivat ydinoperaatioina, jolloin ne ovat hyviä hoitamaan dataan pääsyä, joka täytyy jakaa säielohkojen ja kerneleiden kesken. [13]

Kuvassa 9. havainnollistetaan muistin hierarkia, jossa säietasolla tapahtuvat ydinoperaatiot, josta noustaan ylöspäin hierarkiassa L1 välimuistiin jne.



Kuva 9. Fermin muisti hierarkia. [12]

Fermin ydinoperaatiot on toteutettu kokonaisluku ALU:illa, jotka voivat estää pääsyn muistiosoitteeseen, kun samaan aikaan luku-muokkaus-kirjoitusjakso suoritetaan. Tämä muistiosoite voi olla järjestelmän muisti, GPU:n DRAM tai paikka muistiväylässä. Lyhyen

eston aikana loput muistista jatkaa toimimistaan normaalisti, mutta samalla esto kunnioittaa GPU:n ydinoperaatioita. [13]

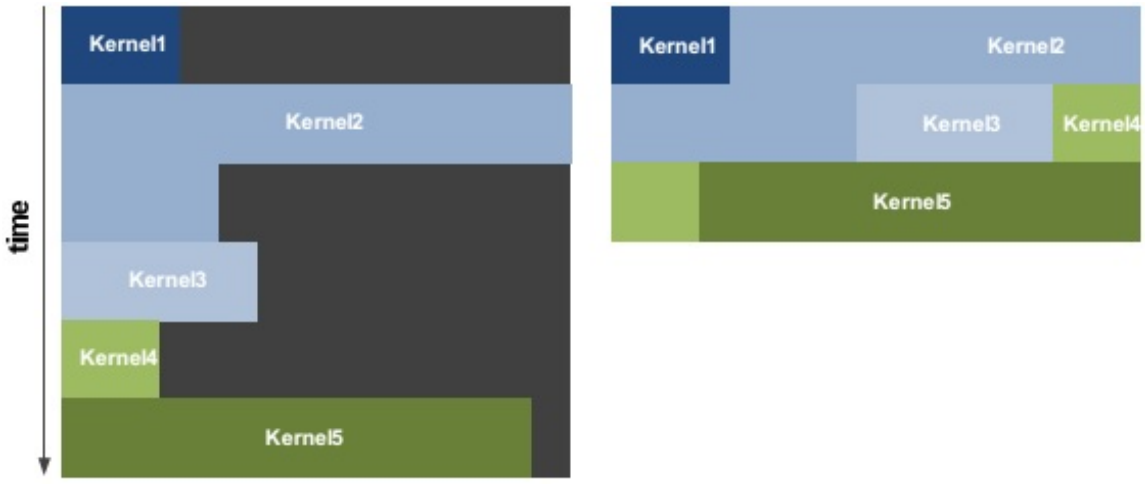
Viimeinen taso hierarkiassa on GPU:hun kiinnitetty DRAM. Fermi käyttää väylänä 64-bittistä DRAM kanavaa, joka tukee SDDR3 ja GDDR5 DRAMia, jota voi olla yhteensä 6 Gt. [13]

3.2.4 PTX 2.0 ISA -käskyjoukko

Fermi oli ensimmäinen, joka tuki PTX 2.0 -käskyjoukkoa. PTX on matalan tason virtuaalikone ja tukee rinnakkaissäieytimen operaatioita, jonka avulla saavutetaan paremman GPU-ohjelmoinnin, tarkkuuden ja suorituskyvyn. PTX 2.0 sisältää IEEE 32-bittisen liukulukustandardin tarkkuuden, yhtenäisen osoiteavaruuden jokaiselle muuttujalle ja osoittimelle, 64-bittisen osoittamisen ja uudet käskyt OpenCL:lle ja DirectComputelle. PTX 2.0 tukee C++ kieltä kokonaisuudessaan. [12]

3.2.5 Kernelien suorittaminen rinnakkain

Fermi tukee kernelien suoritusta samanaikaisesti, jolloin eri kernelit samassa sovellussessa pystytään suorittamaan samanaikaisesti GPU:lla kuvan 10. mukaisesti. [12]



Kuva 10. Kernelien suorittaminen sarjassa ja rinnan. [12]

Rinnakkaissuoritus mahdollistaa ohjelmien käyttämään koko GPU:ta, jolloin suorituskyky paranee. [12]

3.3 Kepler-arkkitehtuuri

Kepler-arkkitehtuurin aloitti GK104, joka paransi suorituskykyä ja vähensi tehon tarvetta Fermi-arkkitehtuurin nähden. Keplerin tarkoituksena oli parantaa DirectX 11-rajapinnan tesseloinnin suorituskykyä ja mahdollistaa kehittäjien käyttämään muita DirectX 11:n tuomia ominaisuuksia. Kepler GPU -arkkitehtuuri on valmistettu 28nm viivanleveydellä, joka parantaa energiatehokkuutta ja suorituskykyä. [14]

GK104 lippulaivamalli GeForce GTX 680 sisältää kahdeksan kappaletta SMX-yksikköä, joissa on yhteensä 1536 CUDA-ydintä. Keplerissä muistin kellotaajuutta nostettiin ja se toimii 6 008MHz:n taajuudella. Kepler esitteli GPU Boostin pelaajille, joka säätää GPU:n kellotaajuutta tehonkulutuksen mukaan. [14]

Kepler GK110 kuvassa 11. on paranneltu versio GK104:stä ja sisältää enemmän ominaisuuksia, joihin kuuluvat dynaaminen rinnakkain-laskenta, Hyper-Q, joka mahdollistaa CUDA-jonojen hoitamisen omalla laitteistotyöjonolla, jolloin operaatiot yhdessä jonossa eivät enää estä tosia jonoja, Verkon ohjausyksikkö (GMU) ja NVIDIA GPUDirect, jonka avulla kolmannen osapuolen laitteilla on pääsy käsiksi GPU:n muistiin. [11]

Kuva 11. GK110 sirun kaavio. [11]

GK110-arkkitehtuuri esiintyy vain huipputason näytönohjaimissa, kuten GTX 780 ja siitä paremmat. Muissa käytetään GK104 versiota. GK 210 versiota käytetään Tesla-mallistossa, joka on suunniteltu HPC-laskentaan eli suurta laskukykyä vaativille. [11]

GK110 siru kokonaisuudessaan sisältää 15 kappaletta SMX-yksikköä, jotka on sijoitettu L2 välimuistin ympärille ja sisältävät samalla itse L1 välimuistin. Reunoille on sijoitettu kuusi kappaletta muistiohjainta, jotka ovat 64-bittisiä. Kokoonpano vaihtelee tuotteiden välillä. [11]

3.3.1 SMX

SMX on yksikkö, joka sisältää 192 CUDA-ydintä ja jokaisella ytimellä on oma kokonaisluku ALU. Kepler tukee samaa IEEE 754-2008 liukulukustandardia kuten Fermikin. Kuvassa 12. on kuvattuna SMX-yksikön rakenne. [11]

Kuva 12. SMX yksikön sisältö. [11]

CUDA-yksiköiden (vaaleanvihreät) lisäksi se sisältää 64 kappaletta kaksoistarkkuudenyksikköä (DP unit) kuvassa keltaisen väriset, 32 kappaletta SFU-yksikköä, jotka ovat

vihreällä ja 32 kappaletta lataus- ja tallennus -yksikköä (LD/ST), jotka ovat kuvattuna tummanvihreällä. [11]

SMX suunnittelun lähtökohtana oli parantaa kaksoistarkkuuden suorituskykyä, koska HPC-ohjelmat hyötyvät paremmasta kaksoistarkkuuden aritmetiikasta, joten Keplerissä on kahdeksan kertaa enemmän SFU yksiköitä kuin Fermissä. [11]

GK110 kuten GK104 käyttää ensisijaista GPU-kelloa kuin aikaisemmissa arkkitehtuureissa olevaa kaksinkertaista kelloa varjostimilla. Keplerissä varjostimien eli CUDA-ytimien kellotaajuus pienennettiin ja suoritinyksiköiden määrää kasvatettiin jolloin tehon tarve laski ja suorituskyky pysyi samana. [11]

3.3.2 Quad Warp järjestelijä

SMX järjestelee säikeet 32 kappaleen säieryhmiin, joita kutsutaan warpiksi. Jokainen SMX sisältää neljä warp järjestelijää ja kahdeksan kappaletta käskyn lähetysyksikköä. Jolloin neljä warppia voidaan jakaa ja suorittaa samanaikaisesti. Quad warp -järjestelijä valitsee neljä warpia, jossa kaksi itsenäistä käskyä voidaan lähettää jokaisessa warpissa per kellojakso. [11]

3.3.3 ISA-koodaus

GK110-rekisterien määrä per säie on kasvanut nelin kertaiseksi Fermiin verrattuna, tarjoten jokaiselle säikeelle 255 rekisteriä. Tästä on hyötyjä koodeissa, jotka tarvitsevat paljon rekistereitä. GK210-arkkitehtuurissa rekisterien määrää on suurennettu kaksin kertaiseksi per SMX kuin GK110:ssä. Tällöin ohjelmat voivat käyttää enemmän rekistereitä per säie ilman, että tarvitsisi uhrata säikeiden määrää, jotka mahtuvat samanaikaisesti SMX:ään. [11]

3.3.4 Shuffle-käsky

Kepleri tukee Shuffle-käskyä, joka sallii säikeiden jakamaan tietoa warpin sisällä. Aiemmin data jaettiin säikeiden kesken erillisillä tallennus- ja lataus -operaatioilla, jotta pystytään siirtämään dataa jaetun muistin läpi, joka on hidasta. Shuffle-käskyllä pystytään

parantamaan suorituskykyä ja pienentämään jaetun muistin tarvetta per säielohko, koska data vaihtuu suoraan warp-tasolla. [11]

3.3.5 Ydinoperaatiot

Rinnakkaisohjelmoinnissa ydinmuistioperaatiot ovat tärkeitä, koska ne sallivat rinnakkaisten säikeiden luku-muokkaus-kirjoitusoperaatioiden toimia oikein jaetussa datatietuessa. Ydinmuistioperaatiot ovat laajasti käytetty rinnakkaislajitteluun, vähennys operaatioihin ja datatietueiden rakentamiseen rinnakkaisesti ilman keskeytyksiä. [11]

Globaalin muistin ydinoperaatiot ovat parantuneet Keplerissä Fermiin verrattuna. Ydinoperaatiot voidaan suorittaa usein samalla nopeudella kuin globaalit latausoperaatiot. Tämän ansiosta ydinoperaatioita voidaan käyttää kernelin sisäisissä silmukoissa. GK110 myös laajentaa tuen 64-bittisille ydinoperaatioille globaalissa muistissa. [11]

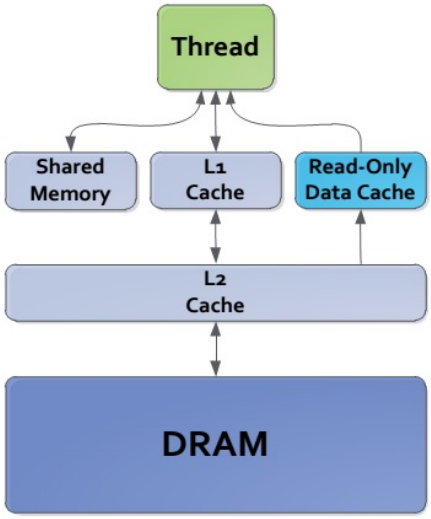
3.3.6 Tekstuuriyksiköt

GPU:n laitteisto tekstuuriyksiköt ovat tarpeellisia laskentaohjelmille, joiden tarvitsee näytteistää ja suodattaa kuvadataa. Tekstuuriyksiköiden suorituskyky on kasvanut Fermiin verrattuna. Jokaisella SMX yksiköllä on 16 tekstuurisuodatusyksikköä eli neljä kertaa enemmän kuin Fermissä. Lisäksi Kepler on muuttanut miten tekstuurien tilaa hallitaan. Fermi-arkkitehtuurissa tarvitaan sidontataulukko, jossa on paikka tekstuurille, jolloin GPU voi noutaa tekstuurin sieltä. Sidontataulukossa on vain rajallinen määrä paikkoja, jolloin ohjelma voi käyttää 128 tekstuuria samanaikaisesti. Keplerissä tekstuurit tallennetaan objektina muistiin ja laitteisto hakee tekstuurit tarvittaessa, jolloin ei tarvita sidontataulukkoa. Tämän ansiosta uniikkien tekstuurien määrällä ei ole rajoja joihin ohjelma voisi viitata. [11]

3.3.7 Muistialijärjestelmä

Kepler-arkkitehtuuri tukee yhtenäistä muistipyyntöpolkua lataukselle ja tallennukselle. Jokaisessa SMX-yksikössä 64 kt muistia, joka voidaan muuntaa 48 kt:un jaettua muistia ja 16 kt:un L1 välimuistia tai toisinpäin. Uudistuksena on, että muistin voi muuntaa tasan 32 kt jaettua muistia ja 32 kt L1 välimuistia. [11]

Muistihierarkian (kuva 13.) uudistuksiin kuuluu vain luku -data välimuisti, jonka määrä on 48 kt. Fermi-arkkitehtuurissa tähän muistiin pääsi käsiksi vain tekstuuriyksiköt. Keplerissä välimuistin kokoa suurennettiin ja parannettiin tekstuurien suorituskykyä. Välimuisti pääsee käsiksi SM:ään, jotta se voisi suorittaa tavallisia latausoperaatioita. Vain luku -polun käyttö on hyödyllistä, koska se vähentää jaetun muistin ja L1 välimuisti -polun käyttöä. [11]



Kuva 13. Keplerin muisti hierarkia. [11]

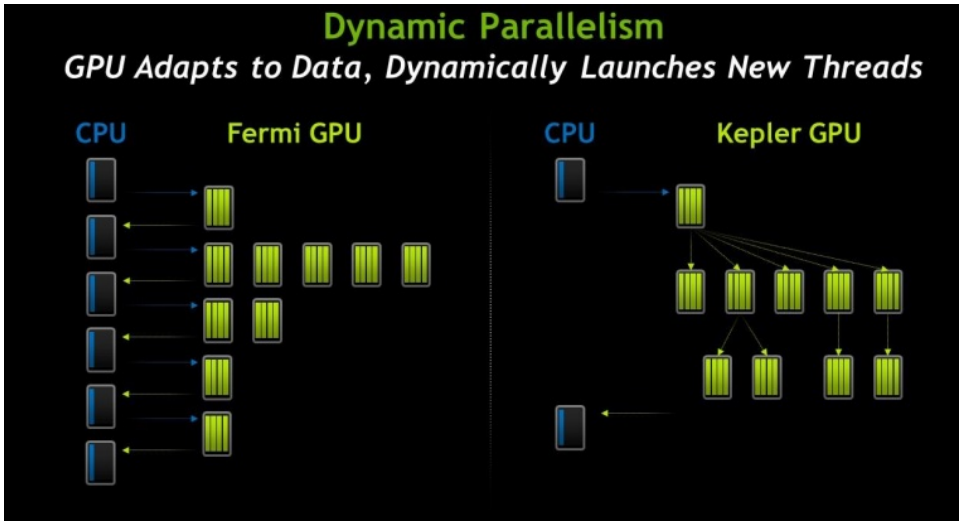
L2 välimuistin koko suurennettiin 1536 kt:un eli kaksi kertaa enemmän kuin Fermissä. L2 välimuisti on ensisijainen datan yhdistämisen paikka SMX yksiköiden välillä. Palvellen kaikkia lataus-, tallennus- ja tekstuuri -pyyntöjä ja tarjoaa nopean datan jaon ympäri GPU:ta. L2 välimuistin kaistanleveyttä suurennettiin kaksin kertaiseksi Fermiin nähden. Tästä hyötyvät ohjelmat, joissa datan osoitetta ei tiedetä ennestään tai tarvitaan monta SMX-yksikköä lukemaan samaa dataa, kuten fysiikkalaskenta ja säteenseuranta. [11]

3.3.8 Dynaaminen rinnakkaislaskenta ja GMU

Dynaaminen rinnakkaislaskenta sallii GPU:n luoda uuden työn itselleen, synkronoida tuloksiin ja kontrolloida työn aikataulutusta järjestelmän polkuja pitkin ilman CPU:n puuttumista. Fermi pystyi käsittelemään suuria rinnakkaisia datarakenteita, kunhan ongelman koko ja parametrit olivat tiedossa kernelin aloituksen aikana. Työt, jotka lähtivät CPU:lta

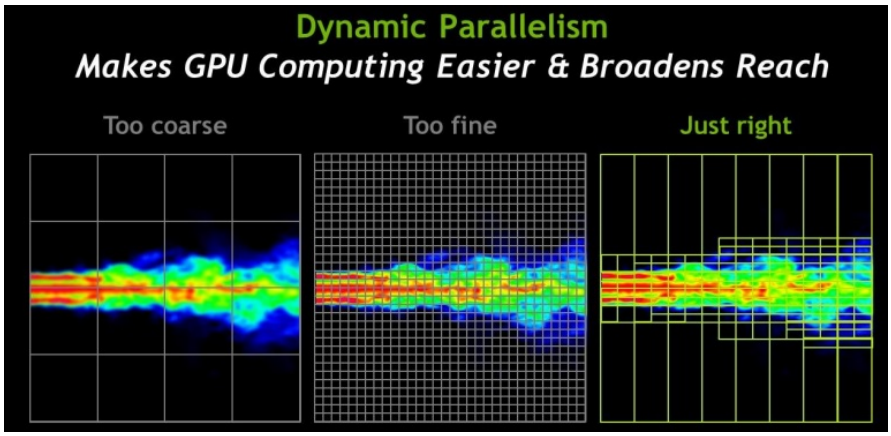
suoritettaan ja palautetaan takaisin CPU:lle. Tulokset käytettäisiin osana ratkaisua tai analysoitaisiin CPU:lla, joka lähettäisi kutsuja takaisin GPU:lle käsiteltäväksi. [11]

Keplerissä kerneli voi aloittaa toisen kernelin ja luoda tarvittavat jonot, tapahtumat ja hoitaa riippuvuuksia, joita tarvitaan käsittelemään ylimääräistä työtä ilman, että tarvitaan CPU:n vuorovaikutusta kuvan 14. mukaisesti. Tämän ansiosta ohjelmat voidaan kehittää toimimaan GPU:lla, jolloin vapautuu CPU:lta resursseja muihin tehtäviin. [11]



Kuva 14. Dynaaminen rinnakkaislaskenta vähentää CPU käyttöä. [8]

Dynaamisen rinnakkaislaskennan avulla verkon resoluutio voidaan määrittää dynaamisesti suorituksen aikana kuvan 15. mukaisesti. Simulaatio pystyy tarkentamaan verkkoa kohdissa, joissa tapahtuu muutosta ja pitää verkon karkeana muualla, missä ei tapahdu muutoksia. Tällöin saavutetaan hyvä tarkkuus alueilla, jotka sitä tarvitsevat ja säästetään resursseja. [11]

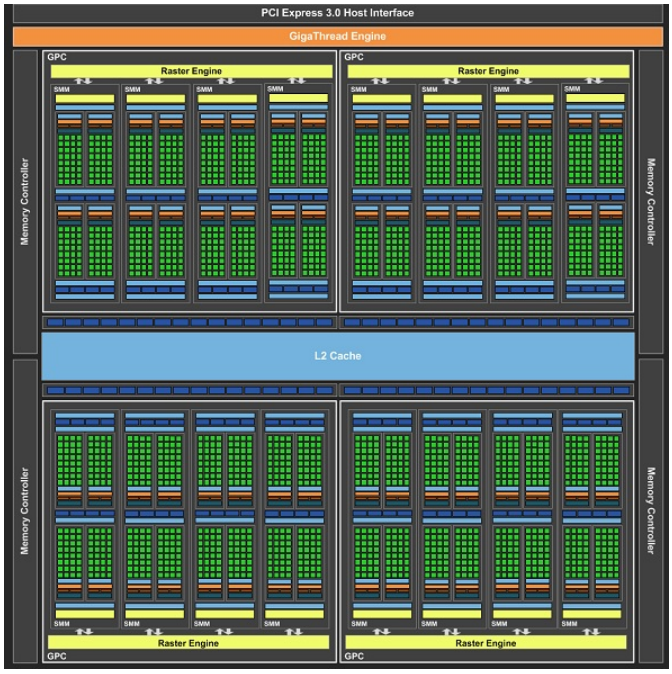


Kuva 15. Dynaamisesti muuttuva verkko. [11]

GMU (Grid Managment Unit) luotiin hallitsemaan CUDAn luomia ja CPU:n lähettämiä verkkoja. Tämä yksikkö hoitaa ja asettaa verkot tärkeysjärjestykseen, jotka siirretään CWD:lle (CUDA work distributor), jonka kautta SMX-yksikölle suoritettavaksi. CWD pitää verkot, jotka ovat valmiina lähetettäväksi ja pystyy siirtämään 32 aktiivista verkkoa samanaikaisesti. CWD viestii GMU:n kanssa kaksisuuntaisen linkin avulla, joka sallii GMU:n tauottaa uusien verkkojen lähetyksen ja viivyttämään odottavia ja keskeytettyjä verkkoja kunnes niitä tarvitaan. GMU:lla on myös suorayhteys SMX-yksikköön, jossa verkot voivat aloittaa uusia töitä GPU:lle, josta uudet työt lähetetään GMU:lle lähetykseen ja tärkeysjärjestyksen laittoon. [11]

3.4 Maxwell-arkkitehtuuri

Nvidia julkaisi Maxwell-arkkitehtuurin vuoden 2015 alussa. Uuden arkkitehtuurin tarkoituksena oli parantaa suorituskykyä, energiatehokkuutta ja valaistuksen parantamista VXGI:n avulla. Kuvan 16. mukaisesti Maxwell-arkkitehtuuri koostuu neljästä GPC (Graphics Processing clusters) ryhmästä, 16 kappaleesta SMM (Streamin Multiprocessor Maxwell)-yksiköstä ja neljästä muistiohjaimesta. Näytönohjaimista GeForce GTX 980 käyttää arkkitehtuuria kokonaisuudessaan ja muut karsitumpia versiota. Arkkitehtuuri on vieläkin valmistettu samalla 28nm valmistustavalla kuten Keplerkin. [15]

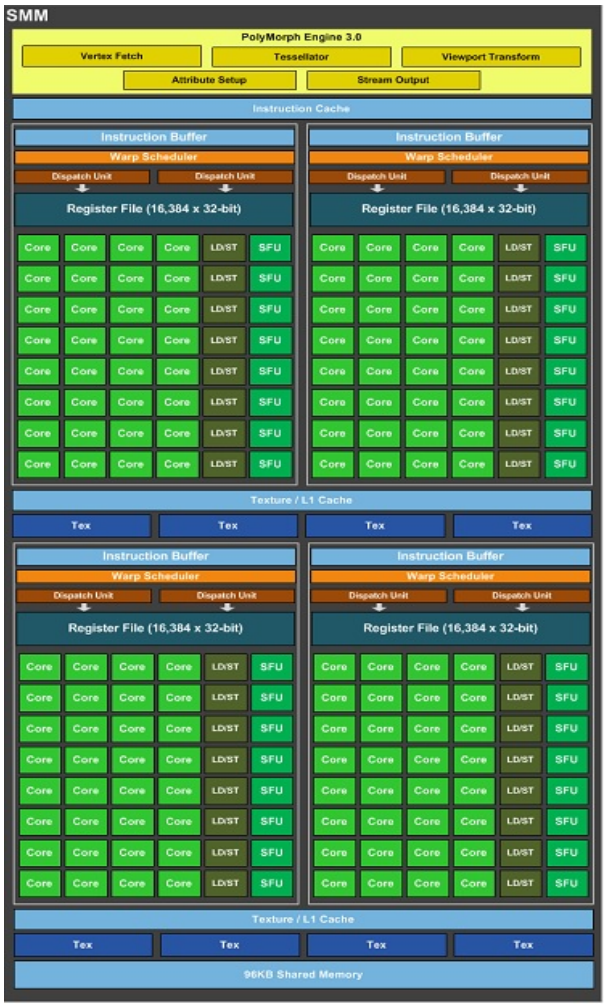


Kuva 16. Maxwell arkkitehtuuri. CUDA-ytimet (vihreä), välimuistit (sininen), PolyMorph engine (keltainen). [15]

Jokaisella GPC-ryhmällä on oma rasterointimoottori ja neljä SMM-yksikköä. Jokaisessa SMM-yksikössä on 128 kappaletta CUDA-ydintä, PolyMorph-moottori ja kahdeksan kappaletta tekstuuriyksiköitä. Yhteensä CUDA-ytimiä on 2 048 ja tekstuuriyksiköitä 128 kappaletta. Muistiohjaimia on neljä kappaletta, joista jokainen on 64-bittinen eli yhteensä 256-bittiä. Jokaisella muistiohjaimella on 16 ROP (Raster operations pipeline)-yksikköä ja 512 kt L2 välimuistia. Yhteensä 64 ROP:sia ja 2048 kt L2 välimuistia. [15]

3.4.1 Maxwell streaming multiprocessor

Maxwell-arkkitehtuurissa on uusiksi suunniteltu SM-yksiköt, joita kutsutaan nimellä SMM. Yksikön suorituskyky per watti on parantunut Keplerin yksiköihin verrattuna. Kuvan 17. mukaisesti SMM on jaettu neljään suoritinlohkoon, jotka sisältävät 32 kappaletta CUDA-ydintä. Suoritinlohkoilla on omat järjestelijä- ja suoritin -puskurit. [15]



Kuva 17. SMM yksikkö. [15]

Jokainen SMM sisältää neljä warp-järjestelijää ja jokainen warp-järjestelijä pystyy lähettämään kaksi käskyä, joka kellojaksolla. Järjestelijän logiikkaa on parannettu vähentämällä turhaa laskentaa järjestelijän päätöksissä. [15]

SMM:n muistihierarkiaa on myös muutettu. SMM-yksiköillä on 96 kt omaa jaettua muistia, kun taas L1 välimuistiin laitto toiminto on siirretty tekstuurien välimuistitoimintoihin. [15]

Näiden muutoksien ansiosta CUDA-ydin pystyy 1.4 kertaiseen suorituskykyyn Keplerin CUDA-ytimeen verrattuna ja kaksin kertaiseen suorituskykyyn per watti. SMM sisältämä PolyMorp engine 3.0 parantaa suorituskykyä lisää tesseloinnissa ja kovassa rasituksessa. [15]

3.4.2 Muistialijärjestelmä

Maxwell GM204 jokainen ROP osio sisältää 16 ROP-yksikköä ja aiemmin Keplerissä oli vain 8 ROP-yksikköä. Jokainen ROP pystyy käsittelemään yhden värin näytteen, jolloin 64 ROP-yksikön ansiosta suorituskyky kasvaa. [15]

GM204 on 256-bittinen muistirajapinta GDDR5-muistin kanssa, jonka nopeus on 7Gbps. L2 välimuistinkokona on 2048 kt, joka jaetaan pitkin GPU.ta. Arkkitehtuuriin on myös parannettu muistipakkausmenetelmiä. [15]

DRAM:n kaistanleveyttä pystytään pienentämään pakkaamalla se häviöttömästi. Kun dataa kirjoitetaan muistiin, jokainen lohko tutkitaan, jos 4 x 2 pikselin alueen lohkossa ei ole muutoksia se pakataan suhteella 8:1. Jos tämä epäonnistuu, mutta 2 x 2 pikselin alue on muuttumaton, silloin data pakataan 4:1. Jos lohkoa ei pystytä pakkaamaan millään menetelmällä, GPU kirjoittaa datan pakkaamattomana. Näiden parannuksien ansiosta Maxwell käyttää 25 % vähemmän bittejä per kehys, kuin Keplerissä. [15]

3.5 Pascal-arkkitehtuuri

Nvidian seuraavaa arkkitehtuuria kutsutaan nimellä Pascal, jonka ensiesiintyminen on luvattu vuodeksi 2016. Pascal-arkkitehtuuri lupaa kymmenkertaista suorituskykyä deep learning eli syvien neuroverkkojen sovelluksiin verrattuna Maxwell-arkkitehtuuriin. Kolme Pascalin suurinta uudistusta on mixed-precision computing, 3D muisti ja NVLink. [16]

Mixed-percision computing ominaisuuden avulla 16-bittinen liukulukutarkkuus pystytään laskemaan kaksi kertaa nopeammin kuin 32-bittinen liukulukutarkkuus. Tämä parantaa luokittelun ja konvoluution suorituskykyä, joita tarvitaan syvien neuroverkkojen sovelluksissa. [16]

3D muisti mahdollistaa kolminkertaisen kaistanleveyden ja melkein kolminkertaisen näyttömuisti määrän Maxwelliin verrattuna. Pascalin muistipiirit ladotaan päällekkäin GPU:n läheisyyteen. Tämä vähentää matkaa, jonka data joutuu kulkemaan muistista GPU:lle ja takaisin, jolloin tiedonvälitys nopeutuu ja tehon kulutus laskee. [16]

NVLink mahdollistaa datan kulkemaan 12 kertaa nopeammin GPU:n ja CPU:n välillä nykyiseen PCI-Express -standardiin verrattuna. Tämän hyödyttää ohjelmia, jotka tarvitsevat paljon GPU:n ja CPU:n välistä kommunikaatiota. Lisäksi NVLink mahdollistaa kaksinkertaisen määrän grafiikkasuorittimia järjestelmässä eli yhteensä kahdeksan kappaletta. [16]

4 VAIKUTUKSET SUORITUSKYKYYN

Monet asiat vaikuttavat näytönohjaimen suorituskykyyn, kuten laitteiston ja arkkitehtuurin rajoitukset, ajurien optimoinnit ja ohjelmointirajapinnat. Suurimman vaikutuksen suorituskykyyn tekee arkkitehtuureihin tehdyt muutokset, jolloin suorituskyky voi kasvaa peleissä 20 - 30 % uudella arkkitehtuurilla [17]. Monesti arkkitehtuurit suunnitellaan tukemaan ohjelmointirajapintojen uusia ominaisuuksia, kuten Kepler parantaa DirectX 11 -rajapinnan tesseloinnin suorituskykyä Fermiin verrattuna [14]. Tesseloinnin avulla voidaan luoda yksityiskohtaisempia pintoja, hahmoja ja animaatioita, vaikka käytössä olisi vain vähän polygoneja eli monikulmioita [18].

Näytönohjainta ei pelkästään rajoita näytönohjaimen grafiikkasuorittimen suorituskyky vaan uudeksi pullonkaulaksi on tullut videomuisti. Nykyään osa peleistä haluavat käyttöönsä enemmän ja nopeampaa videomuistia, jolloin suurempi ja nopeampi muisti parantavat suorituskykyä. Suurempi muisti voi lisätä suorituskykyä noin 30 % riippuen pelistä. [19]

Nvidia julkaisee monille uusille peleille optimoidut ajurit, jotka lisäävät pelien vakautta, suorituskykyä ja SLI -ominaisuuksia. SLI-ominaisuudella on mahdollista kytkeä kahdesta neljään samanlaista näytönohjainta rinnakkain piirtämään grafiikkaa. [20]

Testaukset tehtiin samalla koneella, mutta käyttöjärjestelmä ja tallennusmedia vaihtelevat testien välillä. Taulukossa 1. on testattavan tietokoneen kokoonpano.

Taulukko 1. Tietokoneen kokoonpano.

Tietokoneen osat	Malli
Emolevy	Asus P8Z77-V LX
Prosessori	Intel I7 3770K 4.0GHz
Näytönohjain	PNY XLR8 Nvidia GeForce GTX 670
Käyttömuisti	Kingston HyperX 2x4 Gt (KHX1600C9D3K2/8G)
Virtalähde	Cooler Master 650W GX 80Plus Bronze
Kovalevy 1.	Western Digital 1TB Caviar Black (WD1003FZEX)
Kovalevy 2.	Western Digital 1TB Caviar Green (WD10EZRX)
SSD	Kingston 120GB SSDNow V300 (SV300S37A/120G)

4.1 Ajurien vaikutus

Nvidia julkaisee uuden WHQL (windows hardware quality labs) ajurin tyypillisesti noin kuukauden välein, mutta Beta-ajurit julkaistaan nopeampaa tahtia. WHQL-ajuri on Microsoftin testaama, jotta ajuri toimisi oikein Windows-käyttöjärjestelmässä [21].

Ajuri toimii rajapintana tai rajapinnanohjaajana laitteiston ja käyttöjärjestelmän välillä. Ajurit mahdollistavat uusien ominaisuuksien lisäämisen, kuten tuen lisäämineni uusille ohjelmointirajapinnoille ja teknologioille [25]. Nvidian kehittämiä teknologioita ovat esimerkiksi 3D vision/surround, adaptive vsync ja GPU Boost [22]. Uusissa ajureissa myös parannetaan tukea vanhemmille peleille ja tuodaan tuki uusille peleille.

Ajurin vaikutus pelin suorituskykyyn suoritettiin Bioshock Infiniten omalla suorituskykytestillä ja tulokset mitattiin Fraps-ohjelman suorituskykymittausominaisuuden avulla. Grafiikka-asetukset olivat ultra-tasolla ja resoluutio 1 920 x 1 080. Taulukossa 2. ovat Bioshock Infiniten suorituskykytestin tulokset. Taulukossa käytetään ruudunpäivitysnopeudesta lyhennettä fps eli frames per second.

Taulukko 2. Bioshock Infiniten tulokset.

Ajuri	Min fps	keskiarvo fps	maksimi fps
314.14 Beta	41	55.4	80
314.14 Beta	41	55.0	79
314.22 WHQL	47	67.2	100
314.22 WHQL	47	67.1	100
327.23 WHQL	47	67.2	99
327.23 WHQL	46	66.8	99
337.88 WHQL	49	71.3	107
337.88 WHQL	48	71.0	107
358.50 WHQL	50	71.9	108
358.50 WHQL	50	71.6	107

Testi aloitettiin 314.14 Beta -ajurilla, jossa ei ollut optimointia pelille ja ajuri 314.22 WHQL sisälsi optimointeja pelille. Muut ajurit valittiin satunnaisesti ajuriin 358.50 WHQL saakka.

Tuloksissa on pieni virhe, koska suorituskyvyn mittaaminen piti aloittaa ja lopettaa käsin, jolloin jokaisessa testissä on pieni aikaero, pisimmillään kestäneessä testissä oli noin 6.0 % enemmän kehyksiä piirrettynä, kuin lyhimmillään kestäneessä testissä, mikä saattaa

muuttaa keskiarvoa toisessa desimaalissa. Tuloksiin vaikuttavat eniten tietokoneen taustalla toimivan käyttöjärjestelmän-, taustasovelluksien- ja ajurien -prosessit. Pelikään ei piirrä kaikkia efektejä samanlaisesti. Näiden takia tuloksissa on eroa, vaikka ne olisivat suoritettu samalla ajurilla. Suurin eroavaisuus oli 0.73 %, joka oli 314.14 Beta -ajurilla. Muilla ajureilla ero oli 0.4 % luokkaa.

Testien mukaan suorituskyky parani noin 21.1 %, kun siirryttiin pelille optimoituun ajuriin. Optimoidun ajurin jälkeinen ajuri ei parantanut suorituskykyä, mutta sen jälkeen ilmestyneet ajurit lisäsivät suorituskykyä. Uusimmalla ajurilla saadaan 29.8 % parempi suorituskyky kuin ajurilla jossa ei ole optimointia pelille.

Toisessa testissä testattiin 358.50 WHQL -ajuria, joka on Gaming Ready -ajuri Star Wars Battlefront -pelille. Testi toistettiin muutaman kerran vanhalla ja uudella ajurilla. Apuna käytettiin MSI Afterburner -ohjelmaa, jolla näki ruudunpäivityksen nopeuden ja informaatiota näytönohjaimesta ja prosessorista. Pelikuvaa tallennettiin Nvidian Shadowplayn avulla, minkä jälkeen videokuvasta analysoitiin kohtia, joissa on eroa ajurien kesken. Videokuva tallennettiin betan yksinpelin selviytymistehtävästä Tatooinella. Tehtävän alussa oli reaaliajassa piirretty johdatusvideo, jota käytettiin hyväksi, koska se on aina sama.

Taulukossa 3. on esiteltynä tuloksia, jotka analysoitiin videoilta. Tuloksien eroavaisuuksiin vaikuttavat taustalla toimivat ohjelmat, kuten Shadowplay-videokaappausohjelma, joka heikentää suorituskykyä rasittaen näytönohjainta ja kovalevyä.

Taulukko 3. Star Wars Battlefront Betan tulokset.

Kohtaus	355.98 WHQL	355.98 WHQL testi 2.	358.50 WHQL	358.50 WHQL testi 2.
Nainen lähellä	65.7 fps	65.1 fps	65.1 fps	68.0 fps
Nainen kuvassa ja räjähdys	63.8 fps	64.3 fps	64.1 fps	63.1 fps
Aluksia laskeutuu	63.1 - 68.0 fps	63.0 – 68.0 fps	63.0 – 67.7 fps	54.4 – 67.8 fps
Pelaaja ennen liikkeelle lähtöä	59.6 fps	57.8 fps	57.8 fps	57.7 fps
Pelaaja liikkuu suoraan eteenpäin (Max fps)	66.5 fps	67.0 fps	66.2 fps	66.1 fps

Tuloksista nähdään, että ajuri paransi suorituskykyä joissakin tilanteissa, mutta monessa tilanteessa suorituskyky ei parantunut ja jopa huononi. Monet eroavaisuudet johtuvat videokaappauksesta ja taustaohjelmista, mutta myös videon analysoinnissa tulee virheitä,

koska on vaikeaa saada verratuksi täysin samaa kuvaa videolta. Pääosin nämä virheet ovat alle prosentin luokkaa, koska kohtaukset valittiin sen perusteella, että niiden ruudunpäivitys pysyisi samana pidemmän aikaa.

Lopuksi testattiin ajureiden vaikutusta 3DMark FireStrike-suorituskykytestiin. Suorituskykytestiä varten ei ole julkaistu ajurioptimointeja. FirStrike-suorituskykytesti testaa näytönohjaimen graafista kyvykkyyttä, prosessorin suorituskykyä ja niiden yhteistoimintaa. FireStrike-testi ilmoittaa lopuksi grafiikkapisteet ja keskimääräisen ruudunpäivitysnopeuden.

Taulukossa 4. on esiteltynä tulokset kaikilta käyttöjärjestelmiltä ja prosessorin kellotaajuuksilta. Tarkemmat tiedot taajuuksista, käyttöjärjestelmistä ja tuloksista löytyvät liitteestä 1.

Taulukko 4. FireStrike-testin tulokset.

Ajuri	Grafiikka pisteet	GPU test 1	GPU test 2	3DMark pisteet
314.07 WHQL	6237	29.28 fps	25.25 fps	5719
314.22 WHQL	6265	29.39 fps	25.39 fps	5737
320.18 WHQL	6373	30.43 fps	25.44 fps	5794
320.18 WHQL	6361	30.37 fps	25.39 fps	5809
320.49 BETA	6367	30.39 fps	25.42 fps	5816
326.41 BETA	6382	30.43 fps	25.51 fps	5828
327.23 WHQL	6371	30.46 fps	25.40 fps	5794
331.65 WHQL	6456	30.78 fps	25.80 fps	5896
337.50 BETA	6493	30.57 fps	26.23 fps	5902
337.88 WHQL	6567	31.3 fps	26.25 fps	5950
340.43 BETA	6541	31.1 fps	26.20 fps	5920
340.43 BETA	6579	31.27 fps	26.35 fps	5967
340.52 WHQL	6605	31.5 fps	26.39 fps	5984
344.11 WHQL	6600	31.5 fps	26.35 fps	5987
344.48 WHQL	6629	31.52 fps	26.55 fps	6009
344.75 WHQL	6621	31.47 fps	26.53 fps	6002
347.25 WHQL	6714	32.03 fps	26.82 fps	6082
347.52 WHQL	6676	31.78 fps	26.71 fps	6051
350.12 WHQL	6698	32.02 fps	26.71 fps	6050
352.86 WHQL	6710	31.92 fps	26.86 fps	6073
353.62 WHQL	6712	32.0 fps	26.83 fps	6078
355.60 WHQL	6722	32.13 fps	26.81 fps	6079
355.98 WHQL	6666	31.89 fps	26.56 fps	6037
358.50 WHQL	6593	31.45 fps	26.34 fps	5999

Testissä hyödynnettiin samalla kokoonpanolla aiemmin tehtyjä tuloksia, mutta prosessorin kellotaajuus ja käyttöjärjestelmä vaihtuvat. Prosessorin kellotaajuudet vaihtelevat 3,72 GHz- 4,0 GHz, mutta puolet testeistä on tehty 4,0 GHz taajuudella. Käyttöjärjestelminä on käytetty Windows 7, 8.1 ja Windows 10. Käyttöjärjestelmän ja prosessorin kellotaajuuden muuttuminen muuttivat tuloksia marginaalisesti. Prosessorin kellotaajuuden muuttaminen näkyi fysiikkalaskentaa vaativissa tuloksissa, joten ne jätettiin pois.

Grafiikkapisteistä laskettuna ajurin 314.07 WHQL ja 358.50 WHQL ero on 5,7 %. 3DMarkpistemäärien ero on 4,9 % vaikka prosessorin kellotaajuus on alun jälkeen kasvanut ja käyttöjärjestelmä vaihtunut uuteen. 3DMark pisteisiin sisällytetään grafiikkatestien sekä prosessorin suorituskyvyn testin tulokset, jolloin se sisältää eniten virheitä prosessorin kellotaajuuden ja käyttöjärjestelmän muutoksien takia. Näiden muutosten takia tuloksissa on enintään prosentin verran virhettä ja ne häviävät ajurien vaikutuksien alle. Tuloksista nähdään, että ajurien vaikutukset vaihtelevat ja joskus tulokset huononevat, mutta pääsääntöisesti suorituskyky paranee sitä mukaa, mitä uudempi ajuri on.

4.2 Ohjelmointirajapintojen vaikutus

Ohjelmointirajapinnan avulla ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään. Ohjelmointirajapinta sisältää rutiineja, protokollia ja työkaluja, joilla voidaan rakentaa sovellusohjelmia. Ohjelmointirajapinnat pääsevät käsiksi tietokantoihin tai tietokoneen laitteistoihin, kuten kovalevyihin ja näytönohjaimiin. [23]

Ohjelmointirajapinnat vaikuttavat pelien suorituskykyyn paljon, jos niitä ei optimoida kunnolla. Tyypilliset ohjelmointirajapinnat ovat DirectX ja OpenGL. DirectX sisältää monia ohjelmointirajapintoja, joihin kuuluvat Direct3D ja Direct2D, joita käytetään grafiikan piirrossa. OpenGL on Silicon Graphicsen ja Khronoksen luoma ohjelmointirajapinta, joka oli suosittu 1990-luvulla. 2000-luvulla DirectX alkoi muuttumaan suositummaksi Xbox konsolin takia, jossa käytettiin rajapintana DirectX 8.1 ja Xbox 360 käytettiin DirectX 10. OpenGL on pysynyt ammattikäyttäjien suosiossa ominaisuuksien määrän ja suorituskyvyn ansiosta. [24]

Ohjelmointirajapintojen testeissä käytettiin Unigine Heaven benchmark 4.0- ja Unigine Valley benchmark 1.0 -suorituskykytestejä. Unigine testit tehtiin Windows 10 -käyttöjärjestelmässä ja ajurina käytettiin 358.50 WHQL. Pelejä testattaessa käytettiin Half Life 2 ja Dota 2. Nämä valittiin, koska niillä on tuki DirectX- ja OpenGL- ohjelmointirajapinnoille.

Taulukossa 5. ovat esiteltynä Unigine Heaven benchmark 4.0 -testin tulokset, jotka saatiin sovelluksen omalla suorituskykytestin avulla. Suorituskykytestin asetuksina olivat ultra-asetukset ja extreme tesselointi. Resoluutiona käytettiin 1 920 x 1 080 ja reunanpehmennys oli x2.

Taulukko 5. Heaven Benchmark 4.0 tulokset.

Unigine Heaven benchmark 4.0	Min	Max	Keskiarvo
Directx 9	8.0 fps	105.9 fps	46.7 fps
Directx 11	19.8 fps	99.4 fps	42.4 fps
DirectX 11 ilman tesselointia	27.8 fps	111.9 fps	56.1 fps
OpenGL tesseloinnilla	10.4 fps	90.4 fps	36.5 fps
OpenGL ilman tesselointia	10.1 fps	95.4 fps	45.5 fps

Tuloksista nähdään, että DirectX 11 suoriutui parhaiten, kun ei ole käytössä tesselointia, joka tarvitsee suorituskykyä. Tesselointi päällä DirectX 11 voittaa OpenGL:n 16.16 %:lla. OpenGL:n huonompi tulos voi johtua Nvidian ajureista, joita ei ole välttämättä kunnolla optimoitu OpenGL:ää varten. Unigine Heaven käyttää myös vanhempaa OpenGL 4.0 versiota[25].

Seuraavassa testissä käytettiin Unigine Valley Benchmark 1.0 -suorituskykytestiä. Asetuksina käytettiin ultra-asetuksia ja resoluutiona oli 1 920 x 1 080 ja reunanpehmennyksenä x2.

Taulukossa 6. on esiteltynä tulokset ja tulokset vaikuttavat melkein samalta kuin Unigine Heaven Benchmark 4.0 -testissä.

Taulukko 6. Unigine Valley benchmark 1.0 -testien tulokset.

Unigine Valley Benchmark 1.0	Min	Max	Keskiarvo
DirectX 9	21.4 fps	98.3 fps	46.3 fps
DirectX 9	23.1 fps	99.6 fps	46.5 fps
DirectX 11	22.9 fps	106.1 fps	58.3 fps
DirectX 11	22.2 fps	106.5 fps	58.8 fps
OpenGL	27.2 fps	94.6 fps	52.2 fps
OpenGL	25.6 fps	94.3 fps	51.8 fps

DirectX 11 suoriutuu parhaiten ja OpenGL seuraavana, mutta OpenGL:n ruudunpäivitys ei putoa yhtä alas kuin DirectX 11 -ohjelmointirajapinnalla. Keskiarvo ruudunpäivitysnopeudella on eroa 12,6 % DirectX 11:n ja OpenGL:n kesken. OpenGL:n minimi ruudunpäivitys on 18.8 % parempi, kuin DirectX 11. Vanha DirectX 9 suoriutuu 5.3 % paremmin maksiminopeudesta, kuin OpenGL. OpenGL ja DirectX 11 välinen ero on pienentynyt edellisestä testistä, koska OpenGL ajurit ja optimoinnit ovat parantuneet.

Viimeiseksi testattiin OpenGL:n ja DirectX:n ero pelissä. Pelinä käytettiin Half Life 2:ta, joka tukee DirectX 8-, 9- ja OpenGL -rajapintoja. OpenGL:n pystyi testaamaan vain Linux käyttöjärjestelmässä, joten OpenGL testessä käytettiin Ubuntu 14.04.3 -käyttöjärjestelmää ja Nvidian 346.96 -ajureita. DirectX 9 testattiin Windows 10 -käyttöjärjestelmässä ja Nvidian 358.50 WHQL -ajureilla. Ajuri eron takia OpenGL -tulokset voisivat olla parempia, mutta Ubuntu ei ole julkaissut kirjoitushetkellä uudempia testattuja ajureita.

Half Life 2:n grafiikka-asetukset olivat korkeimmilla laatuasetuksilla paitsi liikesumennus, joka oli poistettu käytöstä. Reunanpehmennyksenä käytettiin MSAAx8 ja resoluutiona oli 1 920 x 1 080. Peli rajoitti ruudunpäivitysnopeuden maksimissaan 301.0 kuvaan sekunnissa.

Taulukossa 7. on esitettynä Half Life 2 -testin tulokset. Testit toistettiin neljään kertaan, koska testeissä käytettiin pelattavaa aluetta, jolloin kohtaukset eivät olleet identtisiä.

Taulukko 7. Half Life 2 tulokset.

Half Life 2	Min	Max	Keskiarvo
OpenGL	124.0 fps	300.0 fps	261.0 fps
OpenGL	133.0 fps	301.0 fps	260.0 fps
OpenGL	130.0 fps	301.0 fps	260.0 fps
OpenGL	133.0 fps	301.0 fps	261.0 fps
DirectX 9	225.0 fps	301.0 fps	289.0 fps
DirectX 9	228.0 fps	301.0 fps	290.0 fps
DirectX 9	138.0 fps	301.0 fps	288.0 fps
DirectX 9	232.0 fps	301.0 fps	289.0 fps

Tuloksista nähdään, että DirectX 9 suoriutuu paremmin kuin OpenGL. Keskiarvosta laskettuna DirectX 9 on 11.1 % nopeampi, kuin OpenGL ja pitää minimiruudunnopeuden

korkeampana. DirectX 9 minimiruudunnopeus laski yhdessä testissä OpenGL tasolle, mutta se johtui uuden alueen tekstuurien latauksesta. Tuloksiin voi vaikuttaa käyttöjärjestelmien ja ajurien eroavaisuus.

Seuraavaksi testattiin Dota 2 suorituskykyä Windowsissa ja Ubuntussa. Grafiikka-asetuksina käytettiin parhainta asetusta, johon lukeutuu reunanpehmennys, jonka laatua ei ilmoitettu. Pelin resoluutiona käytettiin 1 920 x 1 080 ja pystytahdistus oli pois päältä. Peli rajoitti ruudunpäivitysnopeuden 120 kuvaan sekunnissa, rajan pystyi määrittämään laittamalla pelin cfg kansioon autoexec.cfg tiedosto, jossa luki fps_max ”300”. Tällä ruudunpäivitys rajoitettiin 300 kuvaan sekunnissa, mikä riitti testeissä käytettävälle näytönohjaimelle. Tällöin pelin rajoitin ei rajoittanut enää ruudunpäivitysnopeutta, jolloin saatiin oikeat maksimi nopeudet.

Pelissä ei ollut erillistä suorituskykytestiä, joten Windows puolella käytettiin Fraps-ohjelmaa analysoimaan ruudunpäivitysnopeuksia ja Ubuntussa käytettiin GLXOSD-ohjelmaa.

Taulukossa 8. on koottu Dota 2 pelin suorituskykytulokset DirectX- ja OpenGL -ohjelmointirajapinnoilla. Dota 2 on tehty DirectX 9 rajapinnalle, joka näkyy tuloksissa parhaana keskiarvolla, mutta Dotan tuki OpenGL:le ei jää kauheasti jälkeen.

Taulukko 8. Dota 2 suorituskyky tulokset.

Dota 2	Min	Max	Keskiarvo
Directx 9	63.0 fps	192.0 fps	154.6 fps
Directx 11	70.0 fps	155.0 fps	112.5 fps
OpenGL windows	100.0 fps	193.0 fps	146.0 fps
OpenGL ubuntu	87.0 fps	183.0 fps	129.0 fps
OpenGL ubuntu	86.0 fps	173.0 fps	135.0 fps

OpenGL suoriutuu parhaiten minimi että maksimi ruudunpäivityksestä, mutta keskiarvoltaan jää hiukan DirectX 9 jälkeen. Tämä johtuu siitä, että Dota 2 on tuonut tuen OpenGL rajapinnalle vasta lähiaikoina. Ubuntulla OpenGL suoriutuu hiukan huonommin, mikä voi johtua Ubuntun vanhemmista ajureista.

4.3 AMD:n- ja Nvidian -ajureiden vertailu

Vuonna 2014 Nvidia julkaisi 10 kappaletta WHQL GameReady -ajuria, mutta AMD julkaisi vain 4 kappaletta testattuja ajureita, paitsi Beta ajurit, jotka julkaistaan noin kuukauden välein. Aiemmin AMD:llä oli vaikeuksia ajureiden toimivuuden kannalta, mutta se ei ole ongelma nykyään. Nvidian ajurin nopeasta julkaisu tahdista johtuen ajureihin jää virheitä, mutta ne yleensä korjataan pian. Suorituskykyä ei tule lisää pelille suunnatussa optimoidussa ajurissa vaan hyöty voi tulla myöhemmissä ajureissa. [26]

AMD ajureilla on vaivana ylläpitää tasaista ruudunpäivitysnopeutta, vaikka näytönohjain olisi suorituskyvyltään hyvä. AMD:n hitaan ajurien julkaisun takia suorituskykyyn liittyvien ongelmien korjaus kestää ja peleille optimoituja ajureita saadaan odottaa. [26]

AMD:llä on myös vaikeuksia optimoida ajureita Linux käyttöjärjestelmille. Michael Larabelin tekemien testien mukaan AMD suoriutuu huomattavasti heikommin Linux käyttöjärjestelmissä, kuin Nvidian vastaavat näytönohjaimet. Nvidian keskitason näytönohjaimet suoriutuivat yhtä hyvin, kuin AMD:n huipputason näytönohjaimet. [27]

4.4 Ylikellotus ja laitteiston rajoitukset

Ylikellotuksella pystytään helposti parantamaan näytönohjaimen suorituskykyä, mutta sen seurauksena virrankulutus ja lämmöntuotto kasvavat. Taajuuksia ei voida kasvattaa loputtomasti, muuten laitteistosta tulee epätasapainoinen ja lämmöntuotto kasvaa niin suureksi, ettei jäähdytin jaksa jäähdyttää tarpeeksi. Suuren lämpötilan takia näytönohjain laskee grafiikkasuorittimen taajuutta, jolloin lämmöntuotto pienenee ja suorituskyky laskee.

Ylikellotustestissä käytettiin 3DMarkin kehittämää FireStrike-testiohjelmaa ja ylikellotusohjelmana käytettiin MSI Afterburner -ohjelmaa. Nämä ohjelmat valittiin niiden yksinkertaisuuden ja tehokkuuden vuoksi. FireStrike -testissä on kaksi grafiikka testiä, jotka rasittavat grafiikka suoritinta erilaisesti. Käyttöjärjestelmänä oli Windows 10 ja ajureina Nvidian 358.50 WHQL.

Testit aloitettiin perustaajuuksilla, jonka jälkeen testattiin pelkän VRAM:n ylikellotuksen vaikutusta. VRAM:n taajuus yhteensä oli 6 128 MHz. Enemmänkin olisi voinut lisätä,

mutta VRAM on herkkä suurille lämpötiloille. Taajuuden nostaminen lisäsi suorituskykyä noin 2,0 %.

Seuraavaksi testattiin Power limit -asetuksen vaikutusta ja se nostettiin 120 % kohdalle. Power limit rajoittaa tehonkulutusta, jolloin sen nostaminen antaa näytönohjaimelle mahdollisuuden kuluttaa enemmän tehoa ja kasvattaa suorituskykyä. Taulukossa 9. on FireStrike-testin tulokset. Näytönohjaimen lämpötila saatiin MSI Afterburner -ohjelman avulla.

Taulukko 9. FireStrike-ylikellotustestin tulokset.

Ylikellotus Firestrike testi	Pisteet	Testi 1.	Testi 2.	Lämpötila
Perustaajuuksilla	6593	31.45 fps	26.34 fps	70°C
VRAM lisätään 120MHz	6725	32.11 fps	26.85 fps	70°C
+Power limit 120%	6976	33.58 fps	27.66 fps	75°C
+GPU taajuus lisää 100 MHz	7368	35.39 fps	29.26 fps	76°C
Ylikellotukset yhteensä	7412	35.61 fps	11.80 fps	77°C

Tehonkulutusrajan noston jälkeen lisäksi lisättiin vielä grafiikka suorittimen taajuutta 100MHz jolloin grafiikkasuoritin toimi 1 202MHz taajuudella. Lämpötilan noustessa noin 73 °C:n yli suorittimen taajuus laski noin 1 189 MHz:in. Suorituskyky oli perustaajuuksiin verrattuna kasvanut noin 11.8 %

Lopuksi testattiin suorituskyky ylikellotuskokonaisuudessaan. Suorituskyky oli 12,4 % parempi, kuin perustaajuuksilla ja grafiikka testi 1. suorituskyky oli 13,2 % parempi. Grafiikkasuorittimen taajuus käyttäytyi samalla tavalla, kuin aiemmassa testissä eli taajuus laski 73 °C:n jälkeen 1 189 MHz:in. Näytönohjaimen lämpötila kasvoi entisestään ja lämpeni nopeammin. Testattavasta tietokoneesta oli kyljet auki, jolloin ilma pääsi helposti kiertämään. Kylkien kiinni ollessa lämpötila olisi noussut vielä enemmän, jonka seurauksena grafiikkasuoritin olisi laskenut taajuuksia enemmän.

Ylikellotuksella saadaan helposti lisättyä suorituskykyä yli 10,0 %, mutta sen seurauksena ovat lämpötilojen nousu ja riittävän korkeilla taajuuksilla järjestelmä muuttuu epävakaaksi.

4.5 Fyysiset rajoitukset

Laitteiston suurimmat rajoitukset ovat SM-yksiköiden määrä, jolloin fyysinen koko, tehonkulutus ja lämpötila kasvavat. Toisena pullonkaulana on VRAM-muistin määrä, nopeus ja väylän siirtonopeus. Videomuistissa pidetään tietoa tekstuureista ja muuta dataa joiden määrä kasvaa, kun kuvan resoluutiota kasvatetaan.

Testissä käytetyn näytönohjaimen videomuistin määrä oli 2 Gt ja siirtonopeus 192.0 Gt/s. Videomuistin kapasiteetti ei vaikuttanut ajuri-, rajapinta- ja ylikellotus -testeihin, koska testiohjelmat eivät käyttäneet muistikapasiteettia kokonaan. Kaistanleveyden suuruus vaikuttaa suorituskykyyn, koska ohjelmat voivat vaatia suurempaa kaistanleveyttä grafiikan piirtämistä varten. Ei ole olemassa ohjelmaa, jolla voisi seurata ohjelman käyttämää kaistanleveyttä, mutta seuraamalla muistiohjainten käyttöä saadaan suuntaa-antavia tuloksia kaistanleveydestä. Kaistanleveyden suuruuteen vaikuttavat muistiväylän suuruus ja videomuistin nopeus. Videomuistin nopeutta pystyttiin kasvattamaan MSI Afterburner -ohjelmalla. Ylikellotustestissä videomuistin taajuutta lisättiin 120 MHz:llä, jolloin siirtonopeus oli 200.19 Gt/s. Tämä lisäsi suorituskykyä noin 2.0 %:lla 3DMark FireStrike – testissä.

Kun videomuisti on käytetty loppuun, alkaa järjestelmä siirtämään dataa keskusmuistiin. Datan siirto keskusmuistiin on hidasta verrattuna videomuistiin, jolloin ruudunpäivitys alkaa hidastella ja ohjelma voi jopa kaatua. Monet pelit osaavat siirtää dataa keskusmuistille, jota ei heti tarvita ja pitää tärkeän datan videomuistissa. Tämä parantaa suorituskykyä ja on mahdollista ettei suorituskyky laske ollenkaan.

Testejä tehtiin peleillä, kuten Battlefield 4, Witcher 3 ja Warframe. Apuna käytettiin MSI Afterburner -ohjelmaa, jolla pystyttiin mittaamaan muistiohjainten ja PCI-e väylän toimintaa. MSI Afterburner -ohjelmasta saadut tulokset ovat suuntaa antavia. Testeissä PCI-e -väylästä käytettiin noin 5 - 10 % peleissä, mutta resoluution kasvaessa ja videomuistin täyttyessä PCI-e väylää ruvettiin käyttämään noin 30 %, mutta koskaan ei mennyt lähelle 100 %. PCI-e -väylän käyttö saatiin hetkellisesti 100 %:in, kun peli jätettiin taustalle ja aukaistiin uusia ohjelmia. Battlefield 4 ja Witcher 3 -peleissä saatiin helposti videomuisti täyteen 4K-resoluutiolla ja ultra-grafiikka-asetuksilla. Pelistä ei nähnyt täysinäisen videomuistin aiheuttamia hidasteluja, koska näytönohjaimen suorituskyky loppui kesken ja kehysnopeus oli keskimäärin 5 -15 fps. Warframe -pelissä nähtiin vain muutama hetkellinen jäätyminen, kun peli haki dataa keskusmuistista tai peli rasitti suoritinta.

5 YHTEENVETO

Opinnäytetyössä perehdyttiin Nvidian suunnittelemiin arkkitehtuureihin ja tutkittiin ajureiden, ohjelmointirajapintojen ja laitteiston pullonkaulojen vaikutusta suorituskykyyn. Työssä käytiin läpi Nvidian suunnittelemista arkkitehtuureista vain Fermi, Kepler ja Maxwell. Pascal-arkkitehtuurista käsiteltiin vain kolme tärkeintä asiaa, jotka ovat mixed-precision computing, 3D muisti ja NVLink.

Testeissä huomattiin, että ajureilla voi olla suurikin vaikutus suorituskyvylle, jos ajuri on optimoitu sovellusta varten. 3DMark FireStrike-suorituskykytestiohjelmassa huomattiin, että suorituskyky nousee aina jollain ajurilla ja ensimmäisen ajurin ja viimeisimmän ajurin suorituskykyerona oli yli 5 % kasvua. Bioshock Infinitessä optimoidulla ajurilla saavutettiin noin 20 %:n kasvu ja viimeisimmällä ajurilla suorituskyky oli kasvanut optimoimattomasta ajurista noin 30 %. Huomattiin myös, etteivät GameReady-ajurit aina parantaneet suorituskykyä, mikä huomattiin Star Wars Battlefront Beta -testeissä.

Ohjelmointirajapintatesteissä huomattiin, että peleissä käytetään paljon Microsoftin DirectX -rajapintaa. DirectX suoriutui paremmin, kuin OpenGL, mutta niiden ero johtuu luultavasti heikommin optimoiduista ajureista OpenGL rajapinnalle ja Unigine käyttää vanhempaa OpenGL 4.0 versiota. Dota 2 -pelin tapauksessa peliä oli optimoitu paremmin OpenGL:lle ja se suoriutui paremmin kuin DirectX 11, mutta jäi hiukan jälkeen DirectX 9 -rajapinnasta, jolle peli alun perin suunniteltiin.

Laitteiston rajoituksia testattaessa huomattiin, että tehorajan ja grafiikkasuorittimen kellotaajuuden kasvattaminen paransi suorituskykyä, mutta se aiheuttaa lisää lämpöä. Korkea lämpötila saa grafiikkasuorittimen laskemaan kellotaajuuttaan, jottei se ylikuumenisi. Ylikellottamalla tarpeeksi järjestelmä muuttuu epävakaaksi ja voi aiheuttaa ohjelmien kaatumisia. Testeissä huomattiin, että nykypelit tarvitsevat paljon ja nopeaa videomuistia. Videomuistin saa helposti täyteen, jos on vanhempi näytönohjain käytössä ja haluaa pelata suuremmilla näyttöresoluutioilla. Videomuistin täyttyessä järjestelmä siirtää dataa käyttömuistiin, mutta se on hitaampaa ja voi aiheuttaa nykimistä peleihin. Seuraavassa näytönohjain sukupolvessa käytetään 3D-muisteja, jotka mahdollistavat suuremman tiedonsiirtonopeuden.

LÄHTEET

[1] Suvanto, V. GPGPU- prosessorin korvaaminen näytönohjaimella yleishyödyllisissä ohjelmissa. Saatavilla: https://www.theseus.fi/bitstream/handle/10024/27244/opinnaytetyo_suvanto_0701884.pdf?sequence=1

[2] Futuremark, Intel HD Graphics 5500 Review, [www-sivu]. Saatavilla: http://www.futuremark.com/hardware/gpu/Intel+HD+Graphics+5500/review (Luettu:12.10.2015)

[3] Wu, V. PNY GeForce GTX 670 [www-sivu]. Saatavilla: http://www.bjorn3d.com/2012/07/pny-geforce-gtx-670/ (Luettu: 23.11.2015)

[4] Wikipedia, PCI-Express, [www-sivu]. Saatavilla: https://fi.wikipedia.org/wiki/PCI_Express (Luettu 12.10.2015)

[5] GeForce, specifications [www-sivu]. Saatavilla: http://www.geforce.co.uk/hardware/desktop-gpus/geforce-gtx-670/specifications (Luettu: 18.9.2015)

[6] Kurri, S. Muropaketti [www-sivu]. Saatavilla: http://muropaketti.com/artikkelit/naytonohjaimet/nvidia-geforce-gtx-670 (Luettu: 21.9.2015)

[7] Wikipedia, Thermal desing power [www-sivu]. Saatavilla: https://en.wikipedia.org/wiki/Thermal_design_power (Luettu: 12.11.2015)

[8] Rousu, P. Näytönohjaimet järjestyksessä, 25. vuosikerta, 1, s. 59-62, Tammikuu 2008.

[9] Nvidia, Nvidia overview [www-sivu]. Saatavilla: http://www.nvidia.co.uk/object/visual-computing-uk.html (Luettu: 18.9.2015)

[10] Nvidia, Nvidia history [www-sivu]. Saatavilla: http://www.nvidia.co.uk/object/corporate-timeline-uk.html (Luettu: 23.9.2015)

[11] Nvidia, Nvidia's Next Generation CUDA compute architecture: Kepler GK110/210 [www-sivu]. Saatavilla: http://international.download.nvidia.com/pdf/kepler/NVIDIA-Kepler-GK110-GK210-Architecture-Whitepaper.pdf (Luettu: 23.9.2015)

[12] NVIDIA, NVIDIA´s next generation CUDA compute architecture: Fermi [www-sivu]. Saatavilla: http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf (Luettu: 18.9.2015)

[13] Glaskowsky, P.N. NVIDIA´s Fermi: The first complete GPU computing architecture [www-sivu]. Saatavilla: http://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky_NVIDIA's_Fermi-The_First_Complete_GPU_Architecture.pdf (Luettu 18.9.2015)

[14] Nvidia, Whitepaper Nvidia GeForce GTX 680 [www-sivu]. Saatavilla: http://international.download.nvidia.com/webassets/en_US/pdf/GeForce-GTX-680-Whitepaper-FINAL.pdf (Luettu: 23.9.2015)

[15] Nvidia, Whitepaper NVIDIA GeForce GTX 980 [www-sivu]. Saatavilla: http://international.download.nvidia.com/geforce-com/international/pdfs/GeForce_GTX_980_Whitepaper_FINAL.PDF (Luettu: 4.9.2015)

[16] Buck, I. NVIDIA's Next-Gen Pascal GPU Architecture to provide 10x speedup for Deep Learning apps [www-sivu]. Saatavilla: http://blogs.nvidia.com/blog/2015/03/17/pascal/ (Luettu: 7.10.2015)

[17] Leadbetter, R. Nvidia GeForce GTX 980 review [www-sivu]. Saatavilla: http://www.eurogamer.net/articles/digitalfoundry-2014-nvidia-geforce-gtx-980-review (Luettu:19.11.2015)

[18] Muropaketin toimitus, AMD:n ja NVIDIAn kuvanlaatua parantavat tekniikat: Tesselointi [www-sivu]. Saatavilla: http://muropaketti.com/artikkelit/tekniikkakatsaukset/amdn-ja-nvidian-kuvanlaatua-parantavat-tekniikat/11/ (Luettu: 19.11.2015)

[19] Leadbetter, R. Nvidia GeForce GTX 960 2GB vs 4GB review [www-sivu]. Saatavilla: http://www.eurogamer.net/articles/digitalfoundry-2015-nvidia-geforce-gtx-960-2gb-vs-4gb-review (Luettu: 19.11.2015)

[20] Nvidia GeForce, Assassin's vreed syndicate& overwarch game ready driver released [www-sivu]. Saatavilla: http://www.geforce.com/whats-new/articles/geforce-359-00-whql-driver-releasedreleased (Luettu: 19.11.2015)

[21] Microsoft, Windows hardware certification, [www-sivu]. Saatavilla: https://msdn.microsoft.com/en-us/library/windows/hardware/gg463010.aspx (Luettu:27.10.2015)

[22] Nvidia, Nvidia technologies [www-sivu]. Saatavilla: http://www.nvidia.co.uk/object/nvidia-technologies-uk.html (Luettu: 12.11.2015)

[23] Wikipedia, Application programming interface [www-sivu]. Saatavilla: https://en.wikipedia.org/wiki/Application_programming_interface (Luettu: 9.11.2015)

[24] Wikipedia, Comparison of OpenGL and Direct3D [www-sivu]. Saatavilla: https://en.wikipedia.org/wiki/Comparison_of_OpenGL_and_Direct3D (Luettu:29.10.2015)

[25] Unigine Corp, Heaven Benchmark [www-sivu]. Saatavilla: https://unigine.com/en/products/benchmarks/heaven (Luettu 23.11.2015)

[26] Fenlon, W. Hardware report: Nvidia vs AMD [www-sivu]. Saatavilla: http://www.pcgamer.com/hardware-report-card-nvidia-vs-amd/#page-2 (Luettu 2.11.2015)

[27] Larabel, M. AMD vs. NVIDIA Linux gaming performance for DiRT Showdown [www-sivu]. Saatavilla: http://www.phoronix.com/scan.php?page=article&item=dirt-showdown-linux&num=3 (Luettu: 2.11.2015)

3DMark FireStrike tulokset

Ajureiden tulokset FireStrike suorituskyky testistä. Ruudunpiirtonopeutena käytetään fps (frames per second).

Firestrike tulokset	Käyttöjärjestelmä	CPU taajuus GHz	GPU core taajuus MHz	Muistin taajuus MHz	Ajuri	Grafiikka pisteet	GPU test 1	GPU test 2	3DMark pisteet
Testi 1.	64-bit W7	3,92	915	6008	314.07 WHQL	6237	29.28 fps	25.25 fps	5719
2.	64-bit W7	3,92	915	6008	314.22 WHQL	6265	29.39 fps	25.39 fps	5737
3.	64-bit W7	3,72	915	6008	320.18 WHQL	6373	30.43 fps	25.44 fps	5794
4.	64-bit W7	3,92	915	6008	320.18 WHQL	6361	30.37 fps	25.39 fps	5809
5.	64-bit W7	3,92	915	6008	320.49 BETA	6367	30.39 fps	25.42 fps	5816
6.	64-bit W7	3,92	915	6008	326.41 BETA	6382	30.43 fps	25.51 fps	5828
7.	64-bit W7	3,71	915	6008	327.23 WHQL	6371	30.46 fps	25.4 fps	5794
8.	64-bit W7	3,9	915	6008	331.65 WHQL	6456	30.78 fps	25.8 fps	5896
9.	64-bit W7	3,9	915	6008	337.50 BETA	6493	30.57 fps	26.23 fps	5902
10.	64-bit W7	3,9	915	6008	337.88 WHQL	6567	31.3 fps	26.25 fps	5950
11.	64-bit W7	3,9	915	6008	340.43 BETA	6541	31.1 fps	26.2 fps	5920
12.	64-bit W7	4.0	915	6008	340.43 BETA	6579	31.27 fps	26.35 fps	5967
13.	64-bit W7	4.0	915	6008	340.52 WHQL	6605	31.5 fps	26.39 fps	5984
14.	64-bit W7	4.0	915	6008	344.11 WHQL	6600	31.5 fps	26.35 fps	5987
15.	64-bit W7	4.0	915	6008	344.48 WHQL	6629	31.52 fps	26.55 fps	6009
16.	64-bit W7	4.0	915	6008	344.75 WHQL	6621	31.47 fps	26.53 fps	6002
17.	64-bit W8.1	4.0	915	6008	347.25 WHQL	6714	32.03 fps	26.82 fps	6082
18.	64-bit W8.1	4.0	915	6008	347.52 WHQL	6676	31.78 fps	26.71 fps	6051
19.	64-bit W8.1	4.0	915	6008	350.12 WHQL	6698	32.02 fps	26.71 fps	6050
20.	64-bit W8.1	4.0	915	6008	352.86 WHQL	6710	31.92 fps	26.86 fps	6073
21.	64-bit W10	4.0	915	6008	353.62 WHQL	6712	32.0 fps	26.83 fps	6078
22.	64-bit W10	4.0	915	6008	355.60 WHQL	6722	32.13 fps	26.81 fps	6079
23.	64-bit W10	4.0	915	6008	355.98 WHQL	6666	31.89 fps	26.56 fps	6037
24.	64-bit W10	4.0	915	6008	358.50 WHQL	6593	31.45 fps	26.34 fps	5999