

2014•2015
FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN
master in de industriële wetenschappen: elektronica-ICT

Masterproef

Computational Video Mass Storage Array System

Promotor :
De heer Bart STUKKEN

Promotor :
prof. dr. ir. LUC CLAESEN

Senne Vanoppen
Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven

2014•2015
Faculteit Industriële
ingenieurswetenschappen
master in de industriële wetenschappen: elektronica-ICT

Masterproef

Computational Video Mass Storage Array System

Promotor :
De heer Bart STUKKEN

Promotor :
prof. dr. ir. LUC CLAESEN

Senne Vanoppen
Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Woord vooraf

Bij de aanvang van mijn studies Master Industriële Ingenieurswetenschappen afstudeerrichting Elektronica-ICT maakte ik de keuze voor de masterproef “Computational Video Mass Storage Array System”. Dit was een onderwerp waar ik weinig kennis van had, maar mij toch enorm aansprak. Van deze keuze heb ik dan ook geen spijt gehad. Gedurende het academiejaar kwam ik namelijk door mijn masterproef in contact met veel verschillende technologieën. Dit heeft mij dan ook de kans gegeven erg veel bij te leren.

Graag wil ik Prof. Dr. Ir. Luc Claesen bedanken voor het mogelijk maken en begeleiden van mijn masterproef. De tweede persoon die ik wil bedanken is Dhr. Bart Stukken – mijn interne promotor – voor zijn vele goede raad. Als laatste wil ik de aanwezigheid van het lokaal B205 – het lokaal waar mijn opstelling stond – aan de PXL Diepenbeek bedanken voor de leuke werksfeer.

Inhoudsopgave

1	Introductie.....	15
2	Literatuurstudie.....	19
2.1	Dwords.....	19
2.2	CRC.....	19
2.3	8b/10b-encoding.....	20
2.4	LVDS.....	21
2.5	SATA.....	22
2.5.1	Primitives.....	23
2.6	SATA-layers.....	24
2.6.1	Transport layer.....	25
2.6.2	Link layer.....	26
2.6.3	Physical layer.....	27
2.6.4	Hardware.....	29
2.7	PCI Express.....	31
2.8	PCI Express layers.....	32
2.8.1	Transaction layer.....	33
2.8.2	Data link layer.....	34
2.8.3	Physical layer.....	35
3	Materiaal en methoden.....	37
3.1	Virtex-6 ML605.....	37
3.1.1	FMC XM104.....	37
3.1.2	Seagate ST3750528AS.....	38
3.1.3	Opstelling.....	38
3.2	Open source SATA 3.0 Gbit/s-core.....	39
3.2.1	Application layer.....	40
3.2.2	Command layer.....	42
3.2.3	Transport layer.....	43

3.2.4	Link layer.....	45
3.2.5	Physical layer	46
3.3	Proefprocedure.....	47
4	Conclusie	49
5	Referenties	51
6	Bijlagen.....	53
6.1	SATA primitives	53
6.2	SATA Voedingsconnector.....	54

Lijst van tabellen

Tabel 1 SATA-revisies.....	22
Tabel 2 Vaakvoorkomende primitives.....	23
Tabel 3 Basiscommando's	27
Tabel 4 gegevensconnector [4]	29
Tabel 5 PCI Express revisies	31
Tabel 6 Statusregister [4]	40
Tabel 7 ATA-commands [4]	42
Tabel 8 DMA Write EXT FIS [4]	43
Tabel 9 DMA Read EXT FIS [4]	43
Tabel 10 Primitives [3]	53
Tabel 11 Slimlineconnector [6].....	54
Tabel 12 Microconnector [6]	54

Lijst van Figuren

Figuur 1 Kleinschalig multicamerasysteem (21 camera's) rond voetbalveld.....	15
Figuur 2 bit, byte, word, dword.....	19
Figuur 3 8b/10b-encoding	20
Figuur 4 LVDS met het oogdiagram.....	21
Figuur 5 Opbouw SATA-interface	24
Figuur 6 SATA-frame.....	25
Figuur 7 Bursts van commando's met $T_1=106.7$ ns, $T_2=320$ ns.....	27
Figuur 8 Initialisatie van connectie.....	28
Figuur 9 SATA-connector [4].....	29
Figuur 10 PCI Express layers.....	32
Figuur 11 TLP na transaction Layer	33
Figuur 12 TLP na data link layer.....	34
Figuur 13 TLP na Physical Layer en voor encoding en serialisatie	35
Figuur 14 Afbeelding opstelling.....	38
Figuur 15 SATA 3.0Gbit/s core.....	39
Figuur 16 Flowchart SATA initialisatie	40
Figuur 17 Flowchart SATA Write.....	41
Figuur 18 Flowchart SATA Read.....	41
Figuur 19 SATA core.....	42
Figuur 20 Write DMA Ext.....	44
Figuur 21 Read DMA Ext.....	44
Figuur 22 Transport layer schema	45
Figuur 23 Link initialisatie	46
Figuur 24 Resultaten testopstellingen.....	48

Verklarende woordenlijst

ACK	Acknowledge
ATA	Advanced Technology Attachment
ATAPI	Advanced Technology Attachment Packet Interface
DMA	Direct Memory Access
Dword	Double word
EMI	Elektromagnetische Interferentie
FMC	FPGA Mezzanine Connector
FPGA	Field-programmable gate array
FSM	Finite State Machine
HDD	Hard Disk Drive
HPC	High Pin Count
IDE	Integrated Drive Electronics
JTAG	Joint Test Action Group
LBA	Logical Block Addressing
LPC	Low Pin Count
LPSR	Linear Phase Shift Register
LSB	Least Significant Bit
LVDS	Low Voltage Differential Signaling
MPMC	Multi-Port Memory Controller
MSB	Most Significant Bit
NAK	Negative Acknowledge
ODD	Optical Disk Drive
PLB	Processor Local Bus

Abstract

Multicamerasystemen staan synoniem voor rauwe gegevensstromen en hoge bandbreedte. Voor het wegschrijven van dit soort gegevens gaat een besturingssysteem – te vinden op traditionele opslagsystemen – zorgen voor een grote overhead en fragmentatie. Hierdoor is er nood aan onderzoek voor een efficiëntere oplossing.

Voor de video mass storage array zijn er drie eisen vooropgesteld. Ten eerste is de architectuur geschreven in HDL, waardoor er grote vrijheidsgraden zijn in zaken zoals schaalbaarheid. Ten tweede is de architectuur bidirectioneel, zodat zowel schrijf- als leesoperaties mogelijk zijn. Ten derde, om fragmentatie te vermijden, zijn deze schrijfoperaties sequentieel.

Er is een uitgebreide literatuurstudie uitgevoerd van verschillende interfaces – namelijk SATA en PCI-Express – waarna de geschikte hardware is gekozen. Er is gekozen voor een Xilinx ML605 FPGA verbonden via een XM104 Mezzanine-kaart met een mechanische harde schijf. De FPGA is geïmplementeerd met een open source beta SATA core ontwikkeld door de University of North Carolina. De application layer is geschreven in C en verricht sequentiële schrijf- en leesoperaties via DMA. Testen tonen een snelheidswinst van 10-20% aan ten opzichte van een desktopopstelling met identieke opslagmedia.

Abstract in English

Multi-camera systems are synonymous to raw data streams and high bandwidth. Writing these kinds of data, an operating system – found on traditional storage systems – will cause a large overhead and disk fragmentation. As a result, calling the research for a more efficient solution.

There are three set requirements for the video mass storage array. The first requirement calls for architecture to be coded in HDL, allowing large degrees of freedom in qualities such as scalability. The second requirement calls for an architecture that is bidirectional, enabling read as well as write operations. The third requirement calls for these read and write operations to be sequential, hence avoiding fragmentation.

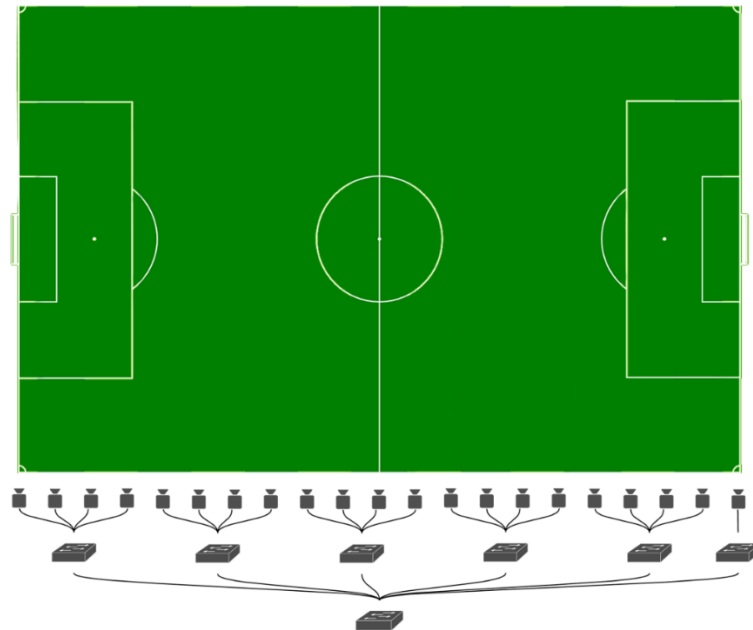
The first step is the literature review of different interfaces – namely SATA and PCI Express – after which the appropriate hardware is selected. The setup is a Xilinx ML605 FPGA connected with a mechanical hard drive through a XM104 Mezzanine card. The FPGA is implemented with an open source beta SATA core developed by the University of North Carolina. The application layer is coded in C and utilizes sequential write and read operations through DMA. Tests show a speed advantage of 10-20% compared to a desktop system with identical storage media.

1 Introductie

Situering

Eén van de speerpunten van de Faculteit Industriële Ingenieurswetenschappen (*FIW*) aan de Universiteit Hasselt (*UHasselt*) is het onderzoek naar multicamerasystemen. Dit is een opstelling van een backend en meerdere camera's met het doel een enkelvoudig beeld te construeren.

Het voordeel van multicamerasystemen is dat er door verschillende beelden te combineren, er de mogelijkheid is hogere resoluties en grotere kijkhoeken – zogenaamde panoramabeelden – te bereiken. De gegevens bevatten meer informatie en zijn dus interessanter voor naverwerkingstoepassingen.



Figuur 1 Kleinschalig multicamerasysteem (21 camera's) rond voetbalveld

Een voorbeeld van een praktische toepassing is:

- Een voetbalveld omcirkelen met een multicamerasysteem:
 - o Bij overtredingen kan de scheidsrechter een herhaling oproepen en correctere beslissingen nemen op basis van meerdere invalsperspectieven.
 - o Kijkers kunnen individueel hun favoriete spelers volgen.
 - o De algemene beeldkwaliteit verbeteren.

Probleemstelling

De bandbreedte van een kleinschalig multicamerasysteem met 21 camera's (Figuur 1) in 1080p (1920x1080 resolutie, 60 beelden per seconde) en 1 byte per kleurenpixel bedraagt:

$$21 \text{ camera's} \times 1920 \text{ hres} \times 1080 \text{ vres} \times 24 \text{ bits} \times 60 \text{ fps} = 62.71 \text{ Gbit/s}$$

Een voetbalwedstrijd van 90 minuten komt dit neer op een benodigde opslagcapaciteit van:

$$\frac{62.71 \text{ Gbit/s}}{8 \text{ bits}} \times 90 \text{ min} \times 60 \text{ sec} = 42.33 \text{ TB}$$

Voor het schrijven van rauwe gegevensstromen met deze bandbreedtes zijn traditionele opslagsystemen zoals servers een inefficiënte oplossing. Het draaien van een volwaardig besturingssysteem is voor deze toepassing overbodig en gaat door zijn overhead en fragmentatie de opstelling zelfs vertragen.

Doelstelling

Het doel van deze proef is het uitbouwen van een snelle, bidirectionele en schaalbare architectuur voor een video mass storage array. Tegen het einde van de masterproef is er een uitgedachte architectuur met de volgende vooropgestelde eisen:

- HDL: de architectuur is geschreven in HDL, dit laat grote vrijheidsgraden toe in zaken zoals schaalbaarheid.
- Bidirectioneel: naast het wegschrijven van gegevens moeten deze ook worden uitgelezen.
- Sequentieel wegschrijven van gegevens: het voorkomen van fragmentatie heeft voordelen voor zowel HDD's als SSD's. Bij HDD's heeft dit het voordeel dat de zoektijden (het verplaatsen van de leeskop) verminderen.

Methode

De proef wordt gedaan in twee fasen. In de eerste fase wordt er onderzoek uitgevoerd naar verschillende interfaces, namelijk SATA en PCI Express. Vervolgens worden verschillende bestaande opstellingen geanalyseerd. Tegen het einde van deze fase is er een geschreven literatuurstudie en is er met de reeds opgedane kennis de geschikte hardware gekozen.

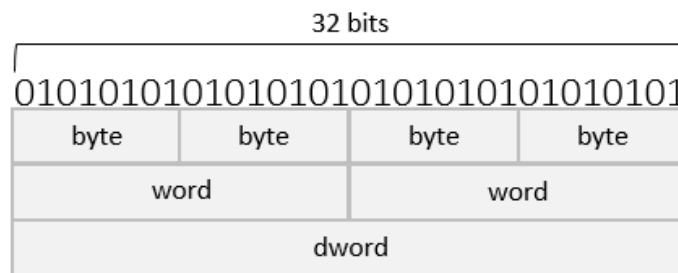
In de tweede fase – de ontwikkelingsfase – wordt de FPGA-opstelling opgebouwd en de lagen geïmplementeerd in HDL. Vervolgens wordt er een application layer geschreven en worden de lagen aangepast aan het doel van de proef. De applicatie layer is geschreven in C – op basis van de Linux block drivers – en zorgt voor de interface met de gebruiker.

2 Literatuurstudie

In de literatuurstudie worden eerst enkele veelvoorkomende begrippen en technologieën bekeken. Hierna worden twee interfaces (namelijk SATA en PCI Express) besproken. Dit begint met een inleiding en vervolgens een analyse van de verschillende lagen. De literatuurstudie is de eerste stap van deze thesis, waarop de basis is gelegd voor de keuze van de hardware.

2.1 Dwords

De kleinste stukjes gegevens in zowel de PCI Express- als de SATA-specificatie zijn dwords (kort voor double word). Een dword bestaat uit twee words, vier bytes oftewel 32 bits. Een dword kan zowel een primitive (uitgelegd in hoofdstuk 2.5.1) als gegevens voorstellen. Aan een dword kan er worden afgeleid dat er in PCI Express en SATA geen oneven aantal bytes voorkomen.



Figuur 2 bit, byte, word, dword

2.2 CRC

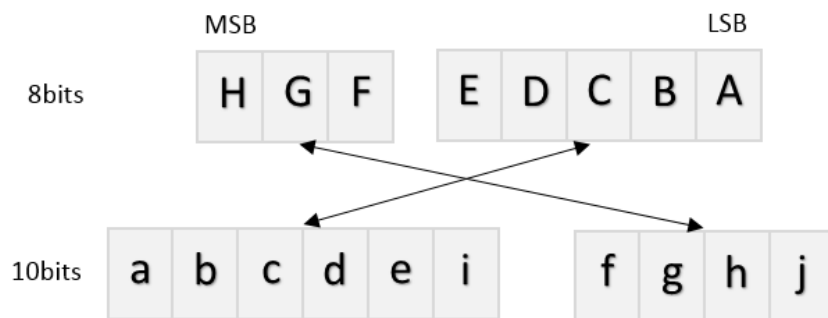
CRC of cyclic redundancy check is een methode voor het controleren van de gegevensintegriteit. Gegevens waarop een CRC zijn uitgevoerd zijn voorzien van een CRC-waarde. Deze CRC-waarde is berekend door de zender aan de hand van een polynoom. Bij het ontvangen worden de gegevens met deze CRC-waarde terug omgerekend. Wanneer het resultaat van de controlewaarde overeenkomt met de gegevens, heeft er geen corruptie opgetreden tijdens de verzending. SATA en PCI Express maken voornamelijk gebruik van 32-bit CRC. Dit betekent dat de CRC-waarde is berekend aan de hand van een 32-bit polynoom. PCI Express maakt langs 32-bit CRC ook gebruik van 16-bit CRC. [1]

2.3 8b/10b-encoding

De 8b/10b-encoding is een veelvoorkomende encodingstechniek en is te vinden in vele seriële communicatieprotocollen, zoals SATA, PCI Express, HDMI en USB 3.0. [3] Het doel van de 8b/10b-encoding is het behalen van een DC-balans. Dit wil zeggen dat er op lange termijn evenveel bits zijn met de waarde 0 als 1. Er wordt ook gezorgd voor meer bitovergangen, waardoor het kloksignaal nauwkeuriger uit het signaal kan worden afgeleid.

De 8b/10b-encoder bestaat uit twee delen: een 5b/6b-encoder en een 3b/4b-encoder. Van de 8 bits (HGFEDCBA) zet de 5b/6b-encoder de eerste vijf bits (EDCBA) en mapt deze naar 6-bits (abcdei). De laatste drie bits (HFG) gaan naar de 3b/4b-encoder en worden gemapt naar 4 bits (fghj). Voor het resultaat worden beide stukjes samengeplakt tot 10 bits (abcdeifghj).

De DC-balans wordt behaald door middel van een techniek genaamd running disparity. Elke 8-bit heeft twee corresponderende 10-bits, één met een positieve dispariteit (RD+, meer 1 dan 0) en één met negatieve dispariteit (RD-, meer 0 dan 1). Wanneer er bij de verzending de DC-balans verstoord geraakt gaat de encoder dit compenseren door van dispariteit (bijvoorbeeld RD+ in plaats van RD-) te veranderen.



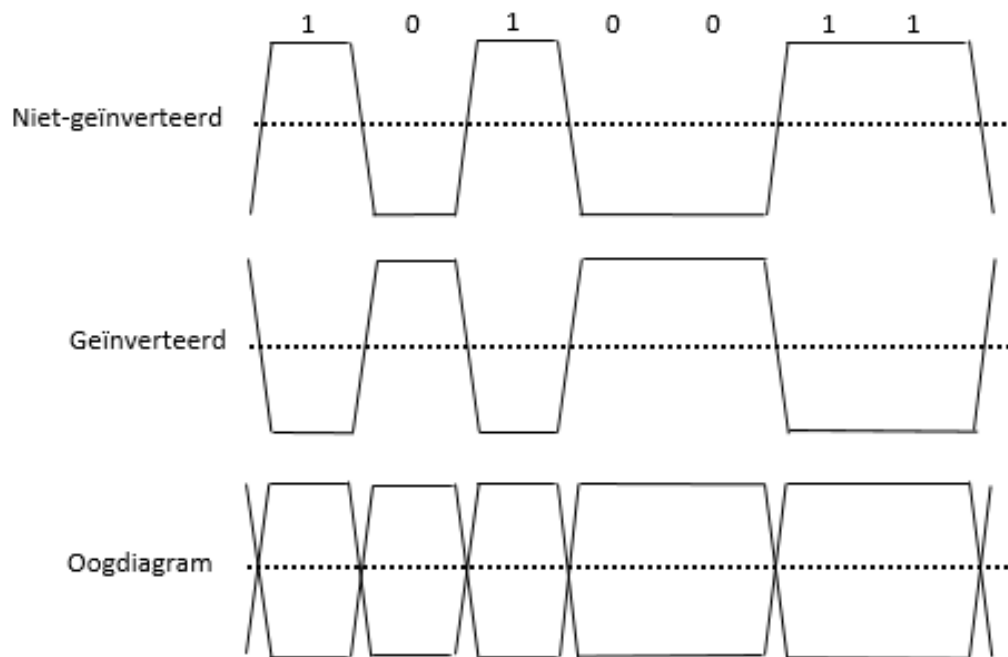
Figuur 3 8b/10b-encoding

Eén van de nadelen van 8b/10b-encoding is dat er in totaal 8 bits naar 10 bits worden gemapt en hierdoor de bandbreedte 25% groter is. Dit is ook hoofdzakelijk de reden waarom bij bijvoorbeeld SATA 6.0Gb/s slechts een praktische bandbreedte heeft van 4.8Gb/s. Deze overhead wordt kleiner in vergelijkbare technieken zoals 130b/132b-encoding, te vinden in PCI Express (vanaf 3.0). [2]

2.4 LVDS

LVDS of low voltage differential signaling is een techniek gebruikt in zowel SATA, PCI Express als andere seriële communicaties van hoge bandbreedte. Hierbij worden de gegevens differentieel doorgestuurd; dit wil zeggen dat er naast de lijn van de gegevens (de niet-geïnverteerde lijn) er ook nog een geïnverteerde lijn is. Wanneer de niet-geïnverteerde lijn een positief signaal draagt, draagt de geïnverteerde lijn een negatief signaal, en omgekeerd. Aangezien ruis beide lijnen hetzelfde aantasten kan deze eenvoudig worden weg gefilterd. Dit gebeurt door beide lijnen aan de ontvangerszijde te vergelijken.

De optelling van beide signalen toont de derde grafiek, ook wel een oogdiagram genoemd. Uit het oogdiagram kan eenvoudig de kwaliteit van een signaal worden afgeleid. [3]



Figuur 4 LVDS met het oogdiagram

2.5 SATA

SATA (Serial Advanced Technology Attachment) is een computerbus ontworpen voor het transport van gegevens tussen host en massaopslag (zoals HDD, SSD, ODD) en is de opvolger van IDE (Integrated Drive Electronics), later PATA (Parallel Advanced Technology Attachment) genoemd. SATA heeft hoofdzakelijk de volgende voordelen over PATA:

- **Snellere gegevensoverdracht:** door hogere kloksnelheden.
- **Kleinere kabelgrootte:** gebruik van 7 geleiders in plaats van 40 of 80 (zogenaamde flat cables). Hierdoor zijn de kabels goedkope, kleiner (minder ruimte, betere verluchting) en is er minder last van cross talk en EMI (elektromagnetische interferentie).
- **Point-to-point:** in tegenstelling tot de master-slave van PATA.

SATA 1.x/2.x/2.x

De volgende tabel weergeeft de eerste revisies van de SATA-specificatie [4]:

Revisie	Commerciële naam	Doorvoersnelheid	Jaar uitgebracht	Compabiliteit
SATA 1.x	SATA 1.5Gb/s	1.2Gb/s of 150MB/s praktisch*	2001	n.v.t.
SATA 2.x	SATA 3Gb/s	2.4Gb/s of 300MB/s praktisch	2004	SATA 1.0 op 1.5Gb/s
SATA 3.x	SATA 6Gb/s	4.8Gb/s of 600MB/s praktisch*	2009	SATA 1.0 op 1.5Gb/s SATA 2.0 op 3.0Gb/s

Tabel 1 SATA-revisies

*Het verschil tussen theoretische en praktische snelheid komt door 8b/10b-encoding

SATA 3.2

SATA 3.2 is in 2013 uitgebracht. De technologie maakt gebruik van de van de SATA Express-specificatie. Deze naast de gewone SATA-specificatie ook nog eens gebruik van de bestaande PCI-Express 3.0-specificatie. Om deze reden wordt deze revisie niet in dit hoofdstuk besproken. [5]

2.5.1 Primitives

Het opzetten en regelen van de seriële communicatie gebeurt bij het SATA-interface door middel van primitives. Dit zijn dwords die beginnen met een controlekarakter K – in tegenstelling tot D voor gewone gegevens – gevolgd door drie karakters. Primitives zijn in hoofdstuk 2.4 gekenmerkt door een lage index p (zoals ALIGN_p). ALIGN_p wordt alleen gebruikt in de physical layer. De rest van de primitives worden gebruikt door de link layer. De meest voorkomende worden getoond in tabel 2. Bijlage 6.1 toont een volledige lijst. [3]

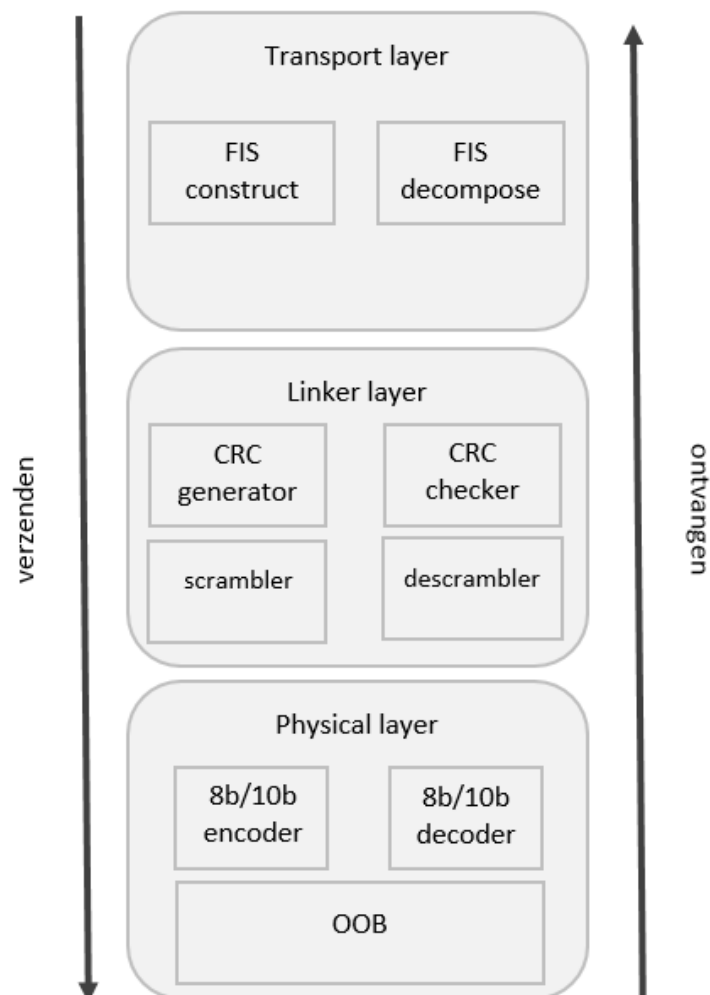
Primitive	Naam	Functie
ALIGN_p	Physical Layer Control	Regelen van physical layer
EOF_p	End of frame	Markeert einde van FIS
SOF_p	Start of frame	Markeert begin van FIS
SYNC_p	Synchronization	Synchroniseren van host en device

Tabel 2 Vaakvoorkomende primitives

2.6 SATA-layers

In de volgende hoofdstukken wordt de architectuur van het SATA-interface besproken in functie van de drie lagen. De lagen worden besproken in de volgorde van het versturen van gegevens van een host naar een device.

Voor het communiceren van informatie tussen de host en de device maakt het SATA-interface gebruik van een frame. Deze frame wordt gemaakt in de transport layer waarna – in de volgende lagen – de CRC wordt toegevoegd en ten slotte de FIS wordt geëncodeerd voor de verzending. De ontvangerszijde breekt deze informatie dan weer af tot bruikbare informatie voor de application layer. Figuur 5 toont de opbouw van het SATA-interface.

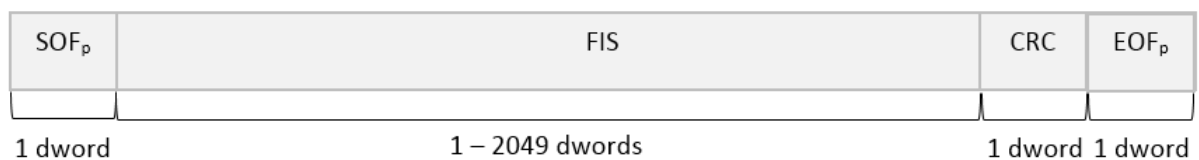


Figuur 5 Opbouw SATA-interface

2.6.1 Transport layer

De transport layer is verantwoordelijk voor het construeren en ontbinden van de FIS (Frame Information Structure). Figuur 6 weergeeft de opbouw van het volledige frame. De FIS is het belangrijkste deel van het frame. Deze bevat namelijk de te verzenden informatie en heeft een lengte van minimaal één dword tot een maximum van 2049 dwords.

Naast de FIS is er de CRC (Cyclic Redundancy Check) met een vaste lengte van één dword. Het begin en het einde van het frame worden gemarkeerd met een SOF_p (Start Of Frame primitive) en een EOF_p (End Of Frame-primitive). Deze worden besproken in de volgende lagen. [3]



Figuur 6 SATA-frame

Error handling

De transport layer is naast het construeren van de FIS ook verantwoordelijk voor het behandelen van fouten, dit houdt in [3]:

- 8b/10b-dispariteits- en CRC-fouten
- Verbreken van het protocol
- Frame errors

Energieprofielen

Voor toepassingen waar energiebesparing belangrijk is – zoals onder andere mobiele toepassingen – zijn er verschillende profielen. De profielen zijn leveranciersafhankelijk maar hebben enkele vooropgestelde richtlijnen. Zo is het zogenaamde ‘slumber profile’ het meest energiezuinige, maar heeft een ontwaaktijd van 10 ms. Hiernaast is er ook nog het zogenaamde ‘partial profile’ met een ontwaaktijd van 10 μ s. [3]

2.6.2 Link layer

De link layer regelt het verkeer van het SATA-interface door middel van primitives. Om de integriteit van de gegevens te controleren aan de ontvangerszijde, voegt de link layer een CRC-waarde toe. In de volgende stap worden de gegevens door de scrambler gehaald. Voordat het frame de link layer verlaat wordt er voor- en achteraan een SOF_p (start of frame) en een EOF_p (end of frame) geplaatst. Deze primitives markeren het begin en einde van het frame.

Cyclic Redundancy Check

De CRC (cyclic redundancy check) van een frame is afhankelijk van het aantal dwords in de FIS. De CRC-waarde is een dword – het is dus een 32-bit CRC – en staat standaard op de waarde 0x52325032. Aan de ontvangerszijde wordt de CRC-dword omgerekend om de integriteit van de data te controleren.

De CRC wordt berekend met behulp van de volgende 32-bit polynoom [4]:

$$G(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

Scrambler

Nadat de CRC wordt uitgevoerd worden alle gegevens vermengd. Als er tijdens de verzending van de gegevens er storingen optreden, worden deze verspreid over het hele frame. Dit verkleint de kans dat de gegevens onleesbaar gaan worden voor de ontvanger. De vermenging wordt gedaan door middel van een LFSR (linear phase shift register). De waarde van de scrambler staat standaard op 0xFFFF. Aan de ontvangerszijde is er een descrambler die de gemengde gegevens weer omzet naar bruikbare gegevens.

De LFSR implementeert het volgende polynoom [4]:

$$G(X) = X^{16} + X^{15} + X^{13} + X^4 + 1$$

2.6.3 Physical layer

8b/10b-encoding

SATA maakt gebruik van 8b/10b-encoding, deze techniek is reeds uitgelegd in hoofdstuk 2.3.

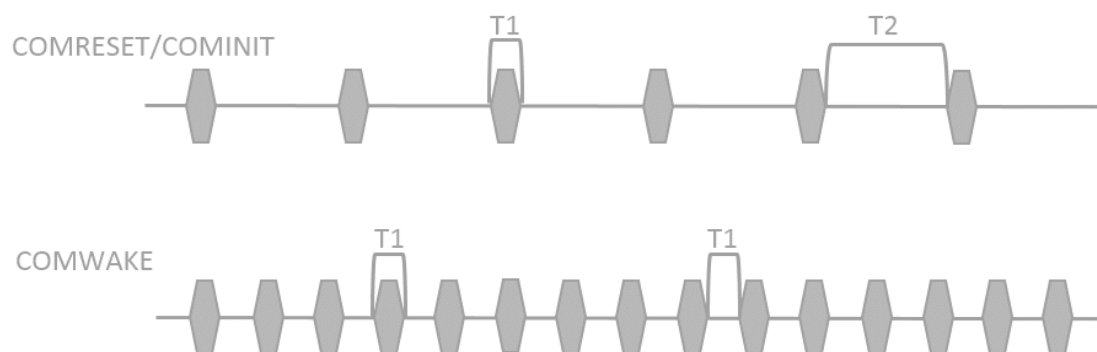
OOB

Door middel van OOB-signalisatie (Out-of-band) zet de physical layer de connectie tussen host en device tot stand. Dit gebeurt aan de hand van de volgende basiscommando's:

Naam	Functie
COMINIT	Device vraagt initialiseren van connectie
COMRESET	Host forceert reset van device
COMWAKE	Host/Device vraagt wake up van Physical layer

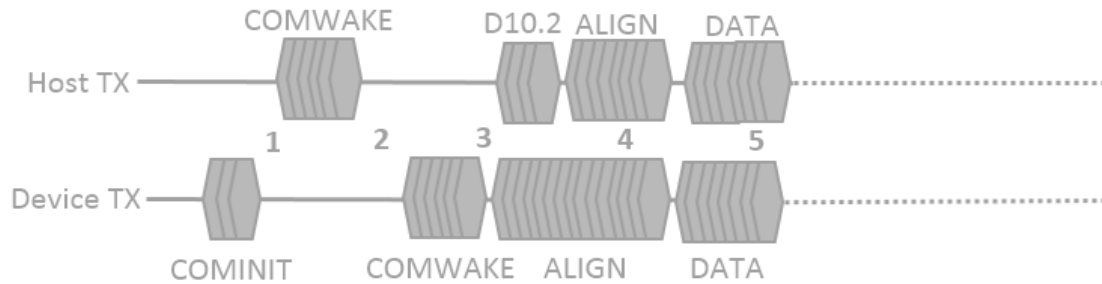
Tabel 3 Basiscommando's

Een commando wordt gegenereerd aan de hand van zes bursts (figuur 7). Een burst is opeenvolging van meerdere $ALIGN_p$ -primitives. De duur van de tijdsintervallen tussen de verschillende bursts en de duur van de burst zelf bepaalt het commando. De ontvanger moet maar vier van de zes bursts ontvangen voor het desbetreffende commando te kunnen identificeren. Bij een burstduur van 106.7 ns en een rustduur van 320 ns gaat het over een COMRESET of COMINIT. Wanneer beide een duur hebben van 106.7 ns hebben wordt er een COMWAKE gelezen. [4]



Figuur 7 Bursts van commando's met T1=106.7 ns, T2=320 ns

Initialisatie van connectie



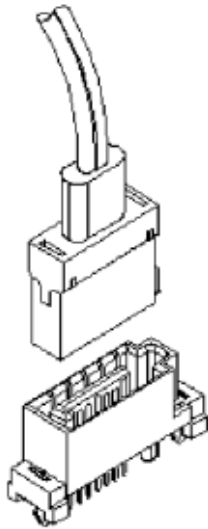
Figuur 8 Initialisatie van connectie

Figuur 7 toont de initialisatie van een SATA-verbinding. Deze initialisatie gebeurt in de volgende vijf stappen (ook weergegeven in figuur 8):

1. De device vraagt om een COMINIT
2. De host kalibreert (optioneel) en reageert met een COMWAKE
3. Vervolgens gaat de device ook een COMWAKE versturen. Hierna gaat deze een sequentie van ALIGN-bursts versturen op zijn hoogste snelheid. Wanneer er na 54.6 μ s geen reactie is van de host, gaat de device veronderstellen dat de host op deze snelheid niet kan communiceren. De device gaat opnieuw bursts versturen maar dan op een lagere snelheid voor een duur van 54.6 μ s. Wanneer er nog steeds geen respons is op de laagste ondersteunde snelheid wordt er overgegaan op een foutmelding.
Wanneer de host de sequentie heeft ontvangen gaat deze een D10.2 versturen op zijn laagste snelheid om crosstalk tegen te gaan.
4. Wanneer de host de sequentie van de device heeft ontvangen aan een ondersteunde snelheid, gaat hij dezelfde sequentie terugsturen op dezelfde snelheid.
5. De SATA-device verbindt zich met de host en zendt SYNC_n-primitives om aan te tonen dat hij klaar is voor gebruik. Wanneer de host deze primitives heeft ontvangen is de verbinding tot stand gebracht. [4]

2.6.4 Hardware

Gegevensconnector



Figuur 9 SATA-connector [4]

Pin #	Naam	Functie
1	GND	Massa
2	Tx+	Transmit+
3	Tx-	Transmit-
4	GND	Massa
5	Rx-	Receive-
6	Rx+	Receive+
7	GND	Massa
-		

Tabel 4 gegevensconnector [4]

De gegevenskabel van SATA maakt gebruik van zeven geleiders - drie massalijnen en vier gegevenslijnen. De vier gegevenslijnen zijn verdeeld in twee paren – een verzend- (Tx) en een ontvangpaar (Rx). Deze paren zijn differentieelparen waarvan de principes zijn beschreven in hoofdstuk 2.4.

Hoewel de gegevensconnector zowel een verzend als een ontvangpaar heeft, is SATA geen full-duplex architectuur. Het SATA-interface ondersteunt maar één gegevensstroom tegelijkertijd. In dit geval wordt de term half-duplex gehanteerd.

Om EMI (elektromagnetische interferentie) verder tegen te gaan zijn beide paren aan de buitenkant en van elkaar gescheiden door een massalijn. De signalen hebben een maximum van 0.5 V (± 250 mV) – in tegenstelling tot 5 V bij PATA. Dit maakt SATA energiezuiniger en is het effect crosstalk kleiner. [4] De maximumlengte van een SATA-kabel is 1 meter. (Essential Guide SATA)

Voedingsconnector

SATA heeft een slimlineconnector met een 5V-voeding en een microconnector met zowel een 5V- als 3.3V-voeding. De pinuitgangen van de voedingsconnectoren zijn te vinden in de bijlage (hoofdstuk 6.2).

2.7 PCI Express

PCI Express (Peripheral Component Interconnect Express) is een high-speed seriële computer expansion bus ter vervanging van de oudere PCI (Peripheral Component Interconnect), PCI-X (Peripheral Component Interconnect eXtended) en AGP (Accelerated Graphics Port). [6]

Enkele kenmerken van PCI Express zijn:

- **Point-to-point topologie:** meerdere seriële connecties tussen device en host.
- **1x tot 32x lanes,** laat toe PCI Express te gebruiken in zowel intensieve (10 Gbit-internet, GPU, Fibre Channel) als kosteffectieve applicaties.

Technologieën zoals SATA Express en Thunderbolt maken gebruik van het PCI Express-interface.

PCI Express 1.0/2.0/3.0/4.0

Revisie	Jaar	Transfer rate	Bandbreedte	
			1x	4x
1.0	2003	2.5GT/s	2 Gbit/s	8 Gbit/s
2.0	2007	5GT/s	4 Gbit/s	16 Gbit/s
3.0	2010	8GT/s	7.877 Gbit/s	31.508 Gbit/s
4.0	2011	16GT/s	15.754 Gbit/s	63.016 Gbit/s

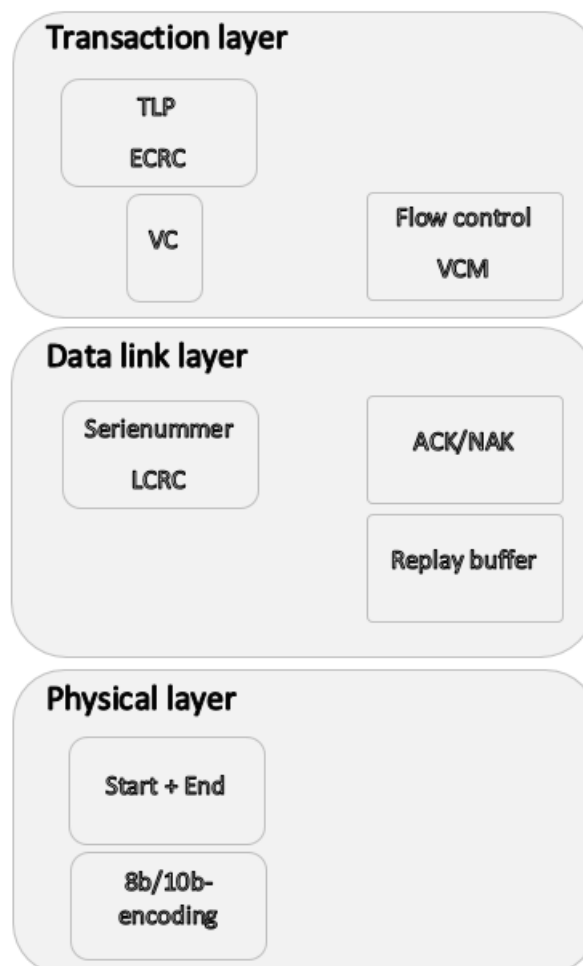
Tabel 5 PCI Express revisies

2.8 PCI Express layers

In dit hoofdstuk wordt de architectuur van PCI Express besproken in functie van de drie lagen. De lagen worden besproken in de volgorde van het versturen van gegevens.

Voor het communiceren van informatie maakt PCI Express gebruik van pakketten – namelijk TLP's of transaction layer packets. Deze pakketten worden aangemaakt in de transaction layer en worden – laag per laag – uitgebreid met extra blokken. Deze blokken hebben het doel de verzending zo vlot mogelijk te laten verlopen. De ontvangerszijde disintegreert deze pakketten weer tot leesbare informatie voor de software layer.

Figuur 10 toont een blokschema van de verschillende lagen. Naast TLP's zijn er ook nog DLLP's of data link layer packets, waarmee het verkeer over de connectie wordt geregeld. [6]



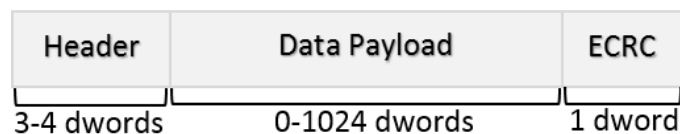
Figuur 10 PCI Express layers

2.8.1 Transaction layer

De bovenste layer van de architectuur is de transaction layer. De transaction layer vertaalt de verzoeken van de software layer om naar TLP's. Bij het ontvangen van een TLP vormt de transaction layer deze om naar bruikbare gegevens voor de software layer.

Een TLP heeft de vorm van Figuur 11. De header bestaat uit 3-4 dwords en bevat informatie zoals onder andere de grootte van het pakket, het adres en het type operatie en de traffic control (TC). De payload bevat de gegevens en is 0 tot 1024 dwords groot.

De transaction layer biedt de optie voor 32-bit ECRC (end-to-end cyclic redundancy check) achter de payload. Deze maakt foutendetectie op het niveau van de transaction layer mogelijk. [6]



Figuur 11 TLP na transaction Layer

Flow Control

De flow control (FC) stuurt de zender de grootte van de vrije bufferruimte. Deze grootte wordt geschat door de flow control aan de hand van een complex kredietensysteem. De FC stuurt deze kredieten door naar de ACK/NAK van de data link layer.

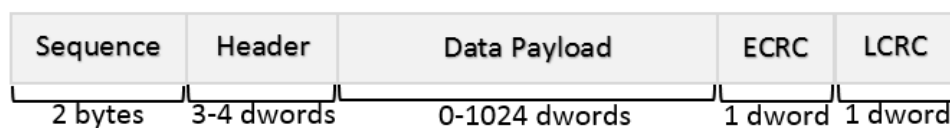
Virtual channels

Virtual channels (VC) worden gebruikt voor het scheiden van verschillende soorten verkeer. Het aantal VC's zijn gedefinieerd in de software layer. De transaction layer leidt dit aantal af aan de hand van de waarde van traffic control (TC) in de header van de TLP. Een TC-waarde van 0 hoort altijd geïmplementeerd te zijn. In dit geval is er één VC beschikbaar – namelijk VC0. De waarde van TC varieert van 0 tot 7, voor een maximum van acht VC's. [PCI Express System Architecture Ravi Budruk, Don Anderson, Tom Shanley]

2.8.2 Data link layer

De middelste layer is de data link layer. De taken van de data link layer is het beheer van de connectie, gegevensintegriteit, foutendetectie en -correctie.

Bij het verzenden van gegevens accepteert worden de TLP's van de transaction layer geaccepteerd en vooraan een serienummer geplaatst. Aan het einde van de TLP wordt er een 32-bit LCRC-waarde (link layer cyclic redundancy check) berekend en geplaatst achter de payload. In tegenstelling tot ECRC is deze niet optioneel. Het controleren van de gegevensintegriteit in de data link layer is dus een vereiste. [6]



Figuur 12 TLP na data link layer

Serienummer

De data link layer voegt vooraan de TLP een uniek 16-bit serienummer toe, waarvan laatste 12 bits het eigenlijke serienummer bevatten. Het serienummer wordt na elke transmissie met 1 verhoogd tot een maximumwaarde van 4095. Hierna wordt de teller terug op 0 wordt gezet.

ACK/NAK

ACK/NAK (acknowledge of negative acknowledge) zorgt voor het beheer van de connectie in de PCI Express-specificatie. Dit gebeurt aan de hand van DLLP's (data link layer packet), niet te verwarren met TLP's.

Wanneer de zenderzijde na het versturen van een TLP een ACK ontvangt, wordt er aangenomen dat de verzending correct is verlopen en wordt de replay buffer leeggemaakt. Bij het ontvangen van een NAK gaat de replay buffer de TLP opnieuw versturen.

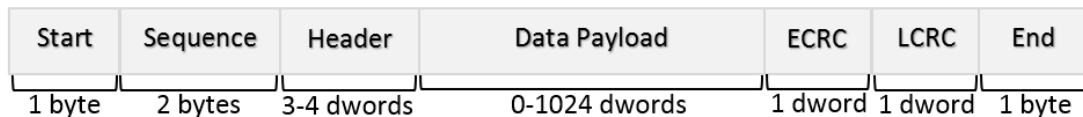
Een ACK/NAK bestaat uit één dword en een 16-bit CRC. De eerste acht bits van het dword wordt identificeert de DLLP al seen ACK of een NAK. De laatste twaalf bits bevat het serienummer van van de betreffende TLP. [6]

Replay buffer

Bij de detectie van fouten (zie ACK/NAK) gaat de data link layer van de ontvangerszijde vragen voor een hertransmissie. Dit gebeurt maximaal vier keer – dus een total van drie hertransmissies – waarna de data link layer een foutmelding geeft.

2.8.3 Physical layer

De physical layer voorziet de TLP's voorzien van een start- en een stopbyte. Deze worden voor- en achteraan de TLP geplaatst. Bij ontvangen van een TLP gaat de physical layer deze twee bytes verwijderen. Om een DC-balans te bekomen maakt PCI Express gebruik van 8b/10b-encoding of 128b/130b-encoding bij recentere revisies. [6]



Figuur 13 TLP na Physical Layer en voor encoding en serialisatie

8b/10b-encoding

De PCI Express-specificatie 1.0 en 2.0 maakt gebruik van 8b/10b-encoding. Uitgelegd in hoofdstuk 2.1.

128b/130b-encoding

Nieuwere specificaties van PCI-Express zoals 3.0 en 4.0 maken gebruik van 128b/130b-encoding. Het principe hiervan is dezelfde als 8b/10b-encoding. Het enige verschil is dat er bij 128b/130b-encoding 128 bits worden gemapt over 130 bits. Dit heeft het voordeel dat er slechts een overhead is van 1.56% – dit in tegenstelling tot 25% bij 8b/10b-encoding – en zo er een grotere bandbreedte beschikbaar is.

3 Materiaal en methoden

In het eerste hoofdstuk van materiaal en methoden wordt de gekozen hardware en de opbouw van de opstelling besproken. Hoofdstuk 3.2 bespreekt de opbouw en de implementatie van de SATA 3.0Gbit/s-core.

In het laatste hoofdstuk van materiaal en methoden wordt de schrijfsnelheid van de opstelling getest. De prestaties worden vergeleken met twee desktopopstelling met een identieke schijf en eenvoudig programma geschreven in C.

3.1 Virtex-6 ML605

De Virtex-6 ML605 maakt deel uit van de Xilinx Virtex-familie en staat door zijn uitbreidbaarheid en flexibiliteit bekend als een sterk platform voor het ontwikkelen van cores.

De FPGA-chip op de ML605 is een XC6VLX240T-1FFG1156. Deze chip is gemaakt op het 40nm-procedé en beschikt over 241,152 logische cellen. Het BRAM heeft in totaal 14,976 Kb geheugen.

Het bord beschikt over 20 GTX-transceivers, deze hebben een maximale bandbreedte 6.6Gbit/s. De ML605 is voorzien van twee FMC-connectoren: eenn HPC-connector (high pin count-connector) met 8 GTX-transceivers en één LPC-connector (low pin count-connector) met 1 GTX-transceiver. [7]

Dit project maakt gebruik van de tweede revisie van het ML605-bord. Deze revisie beschikt over 1 GB DDR3 in tegenstelling tot 512 MB in de eerste revisie. Gebruikers van de nieuwe revisie kunnen hierbij compatibiliteitsproblemen krijgen bij het importeren van een project ontwikkeld op een ouder bord. Dit probleem is op te lossen door met behulp van VHD het RAM te reduceren tot 512 MB. [8]

3.1.1 FMC XM104

Het ML605-bord beschikt zelf niet over een fysieke SATA-connector. Hierdoor is het nodig de ML605 uit te breiden met een XM104-module. Geconnecteerd met een HPC-connector is er de mogelijkheid twee SATA-poorten te benutten. Naast de twee SATA-poorten bevat de XM104 tevens ook acht SMA-connectoren en een CX4-connector, deze worden echter niet gebruikt in dit project. [9]

3.1.2 Seagate ST3750528AS

De gebruikte HDD is een Seagate Barracuda 7200.12 ST3750528AS. Deze heeft een capaciteit van 750GB met een sectorgrootte van 512 bytes, dit komt neer op een totaal van 1.465.149.168 sectoren. De HDD heeft een SATA 3.0Gbit/s-interface en is hierdoor compatibel met de SATA 3.0Gbit/s-core. De theoretische maximumsnelheid van de HDD is 125MB/s. [10]

3.1.3 Opstelling

De ML605 is beveiligd voor EMI door een antistatische mat en –armband. Deze zijn verbonden met zowel de massa van de FPGA als de aarding. De FPGA is verbonden met de HDD via de XM104-kaart. De HDD wordt gevoed door de voeding van de PC, aangezien er drie massalijnen door de SATA kabel loopt hebben de PC en de FPGA dezelfde massa. De FPGA is verbonden via JTAG en UART met de PC, respectievelijk voor de FPGA te programmeren en om ermee te communiceren.

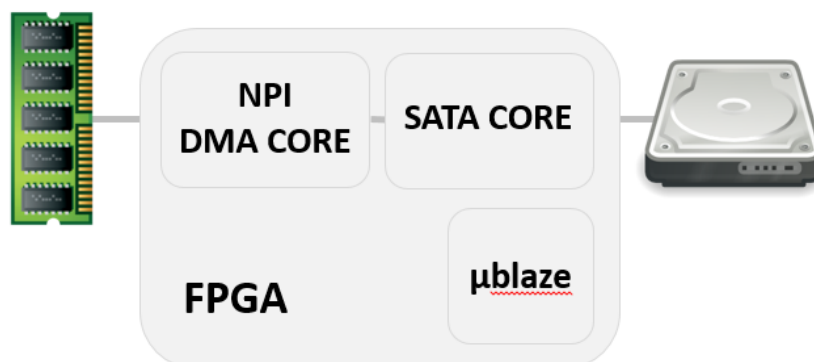


Figuur 14 Afbeelding opstelling

3.2 Open source SATA 3.0 Gbit/s-core

Er is gebruik gemaakt van de open source beta SATA 3.0 Gbit/s-core geschreven door Ashwin Mendon, Bin Huang en Ron Sass van de University of North Carolina. [11] De core implementeert de command, transport en link layer in VHDL, terwijl de physical layer ontworpen is in Verilog. Het ontwerp is gebouwd voor de GTX-transceivers te vinden op de Xilinx Virtex-6. De core biedt een interface voor alle SATA 3.0 Gbit/s-toestellen – zowel HDD's als SSD's.

Door de beperkte bandbreedte van de PLB (processor local bus) wordt er gebruik gemaakt van NPI (native port interface). Deze is ontworpen door Xilinx als een interface voor de MPMC (multiport memory controller) van de FPGA. Omdat NPI werkt op dezelfde hoge kloksnelheid werkt van de MPMC – en zonder inbreng van de processor – zorgt deze voor een interface van hoge bandbreedte en lage latency tussen het geheugen en de core. Hoewel de gegevens wordt verstuurd over de NPI, worden de command- en statusregisters nog steeds over de PLB worden verstuurd. [12]



Figuur 15 SATA 3.0Gbit/s core

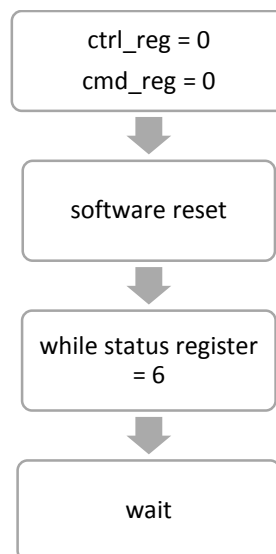
3.2.1 Application layer

De application layer is de bovenste layer en is geschreven in C. Dit is de laag waar de gebruiker mee communiceert met een terminal via UART. De code is gebaseerd op de Linux block drivers. Gegevens zoals adres, het aantal sectoren en de soort operatie van de application layer doorgestuurd naar de command layer.

Bit	7	6	5	4	3	2	1	0
Name	BSY	RDY			DRQ			ERR

Tabel 6 Statusregister [4]

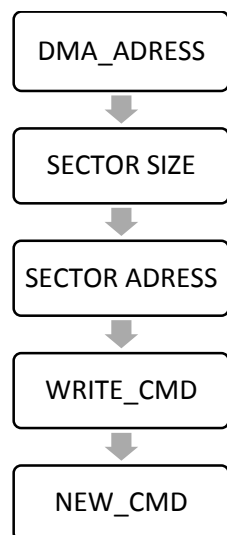
Bij het opstarten van de opstelling gaat de schijf initialiseren. Zowel de command- als het control register worden leeggemaakt. Hierna wordt er een software reset aangevraagd via het control register. De wachperiode hierna varieert sterk en kan soms duren tot enkele seconde. Wanneer er een statusregister van de device komt met de waarde 6 (RDY), is de SATA-link opgezet.



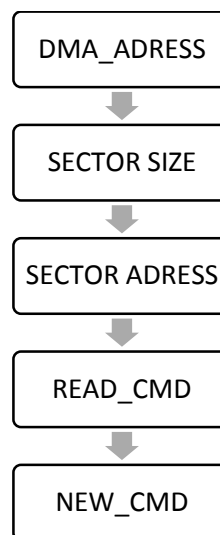
Figuur 16 Flowchart SATA initialisatie

Na de initialisatie moet de gebruiker de gegevensgrootte van het frame afkomstig van het multicamerasysteem definiëren. De grootte van de frames worden vervolgens gelijkgesteld aan het aantal sectoren per DMA-aanvraag – met andere woorden wordt er per DMA-aanvraag één frame gestuurd. Het aantal sectoren per DMA is maximaal 16-bit ofwel 65536 sectoren (zie hoofdstuk 3.2.3). Voor een schijf van 512 bytes per sector komt dit neer op een maximale framegrootte van 32MB. Deze keuze is gemaakt omdat het zoeken van individuele frames bij een leesoperaties zo eenvoudiger wordt.

In de volgende stap kiest de gebruiker of het gaat om een lees- of schrijfoperatie. Het adres van het RAM zowel als van de HDD – het LBA (logic block adresssing) – worden meegegeven. Na de keuze voert de opstelling sequentiële lees- of schrijfoperaties uit. Het LBA-adres loopt per operatie op afhankelijk van de gekozen grootte van de frames; wanneer het statusregister opnieuw een 6 (RDY) teruggeeft is de operatie voltooid en worden het LBA-adres verhoogt voor de volgende operatie.



Figuur 17 Flowchart SATA Write



Figuur 18 Flowchart SATA Read

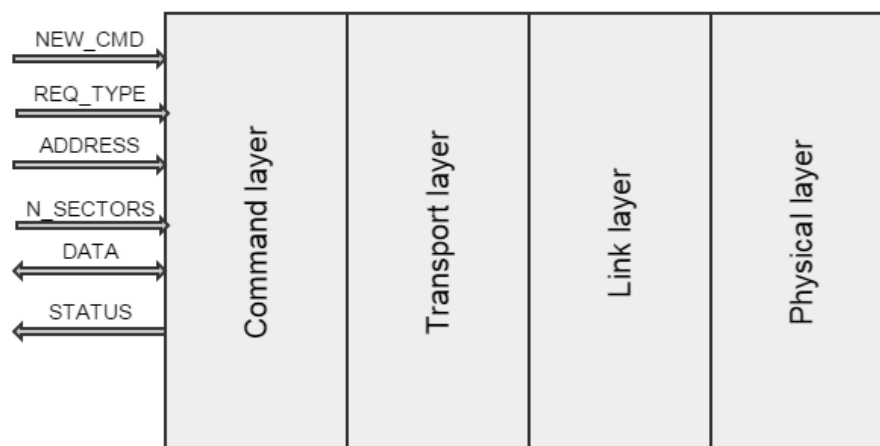
3.2.2 Command layer

De instructies komen binnen van de application layer en worden aan de hand van een FSM omgezet naar het correcte commando. Deze zijn gedefinieerd in de ATA (Advanced Technology Attachment) of ATAPI (ATA Packet Interface). ATA/ATAPI is een standaard dat compatibiliteit biedt tussen PATA- en SATA-interfaces. De gebruikte commando's worden getoond in tabel 7.

Commando	Code
WRITE DMA EXT	35h
READ DMA EXT	25h
REG FIS VALUE	27h
DATA FIS VALUE	46h

Tabel 7 ATA-commands [4]

De command layer is aangepast voor het mogelijk maken van 48-bit LBA-adressering. Hierbij is het mogelijk capaciteiten tot 128 PiB te adresseren in tegenstelling tot enkel 8.5GB voor 24-bit LBA-adressering. Het activeren van 48-bit LBA wordt gedaan door gebruik van de EXT-commando's (namelijk WRITE DMA EXT en READ DMA EXT) die gedefinieerd zijn in de ATA/ATAPI standaard. Deze worden herkend door de opslagmedia en maken de transport layer duidelijk dat er gebruik wordt gemaakt van de uitgebreide (extended) registers in de FIS. [13]



Figuur 19 SATA core

3.2.3 Transport layer

Het hoofddoel van de transport layer is het construeren en deconstrueren van de FIS. De parameters zoals het soort operatie, het LBA-adres, het aantal sectoren en de gegevens worden van de command layer ontvangen en in het FIS-formaat gegoten. Bij het ontvangen wordt de FIS ontbonden tot het statusregister en de gegevens.

Tabel 8 en tabel 9 tonen respectievelijk H2D-registers voor het wegschrijven en lezen van gegevens door middel van 48-bit LBA-adressen:

- Het type van de FIS is gedefinieerd in byte 0 van dword 0. De waarde 27h toont aan dat het om een H2D-register gaat. Bij een D2H-register heeft de byte de waarde 34h, voor DMA Activate-register 35h. Wanneer de FIS enkel gegevens bevat is deze 46h.
- De type operatie – schrijf- of leesoperatie – wordt geplaatst in byte 2 van woord 0. In dit geval gaat het om een “DMA Write Ext” en een “DMA Read Ext”. Dit betekent dat er gebruik wordt gemaakt van de uitgebreide registers (ext). Hierbij wordt het maximaal aantal sectoren per aanvraag verhoogd van 8-bits naar 16-bit. Het maximale LBA-adres wordt verhoogd van een waarde van 24 bits naar 48 bits.
- Door middel van de features-bytes kan er gebruik worden gemaakt van onder andere instellingen voor energiebesparing (sleep, slumber, etc).

dword	Byte 3	Byte 2	Byte 1	Byte 0
0	Features	Command (35h)	Interrupt	FIS Type (27h)
1	Device	LBA High	LBA Mid	LBA Low
2	Features (ext)	LBA High (ext)	LBA Mid (ext)	LBA Low (ext)
3	Control	Reserved	Sector count (ext)	Sector count
4	Reserved	Reserved	Reserved	Reserved

Tabel 8 DMA Write EXT FIS [4]

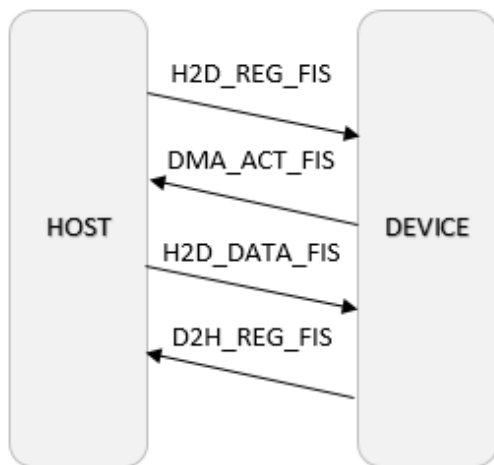
dword	Byte 3	Byte 2	Byte 1	Byte 0
0	Features	Command (25h)	Interrupt	FIS Type (27h)
1	Device	LBA High	LBA Mid	LBA Low
2	Features (ext)	LBA High (ext)	LBA Mid (ext)	LBA Low (ext)
3	Control	Reserved	Sector count (ext)	Sector count
4	Reserved	Reserved	Reserved	Reserved

Tabel 9 DMA Read EXT FIS [4]

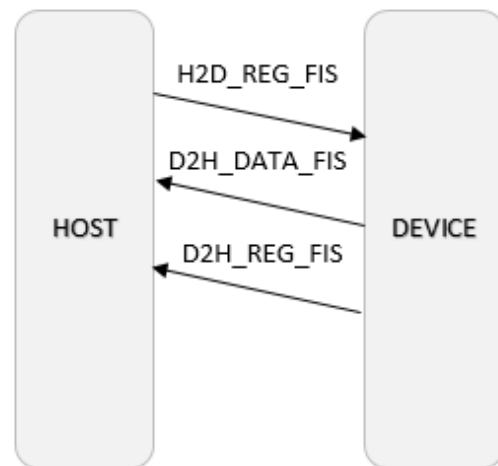
Een schrijf- of leesoperatie is een opeenvolging van meerdere frames. Figuur 20 en figuur 21 tonen de transmissies van schrijf- en lees operatie tussen host en device. Beiden beginnen allebei met een H2D_REG_FIS – waarvan de opbouw eerder is gezien in tabel 8 en 9.

Bij een schrijfoperatie stuurt de device een DMA_ACT_FIS waarmee hij aantoont dat de host de toestemming heeft de gegevens (H2D_DATA_FIS) door te sturen. Het opvragen van leesoperatie gaat de device de gegevens direct doorsturen naar de host.

Wanneer de transmissie voltooid is stuurt de device een D2H_REG_FIS naar de host. Deze wordt gecontroleerd op fouten. De H2D_FIS wordt achtergehouden wanneer eventuele fouten optreden zodat deze opnieuw kan worden verzonden. [4]



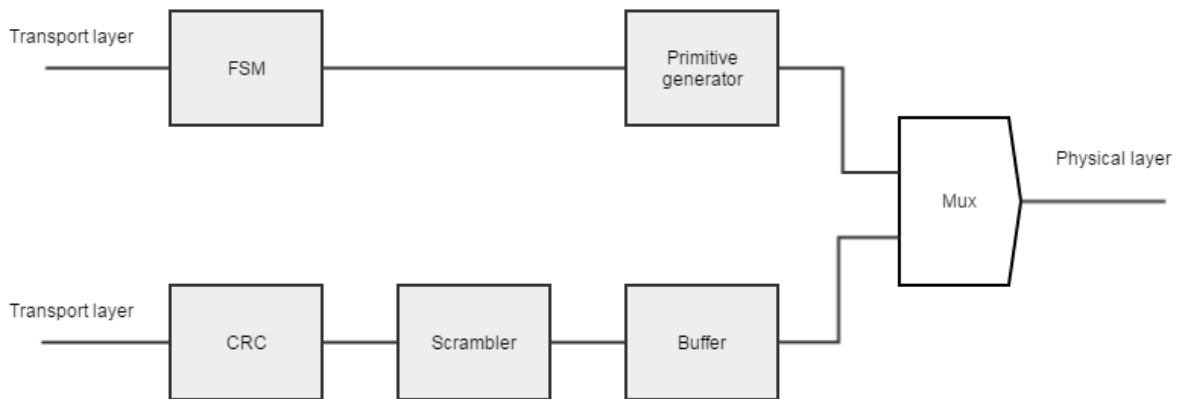
Figuur 20 Write DMA Ext



Figuur 21 Read DMA Ext

3.2.4 Link layer

De link layer is verantwoordelijk voor het ontvangen en afleveren van de frames. Hiervoor maakt deze laag gebruik van primitives gedefinieerd in de SATA-protocol (gezien in de literatuurstudie 2.4.1, volledige lijst in bijlage). De Link layer is onaangepast gebleven. Figuur 22 toont de opbouw van de transport layer.



Figuur 22 Transport layer schema

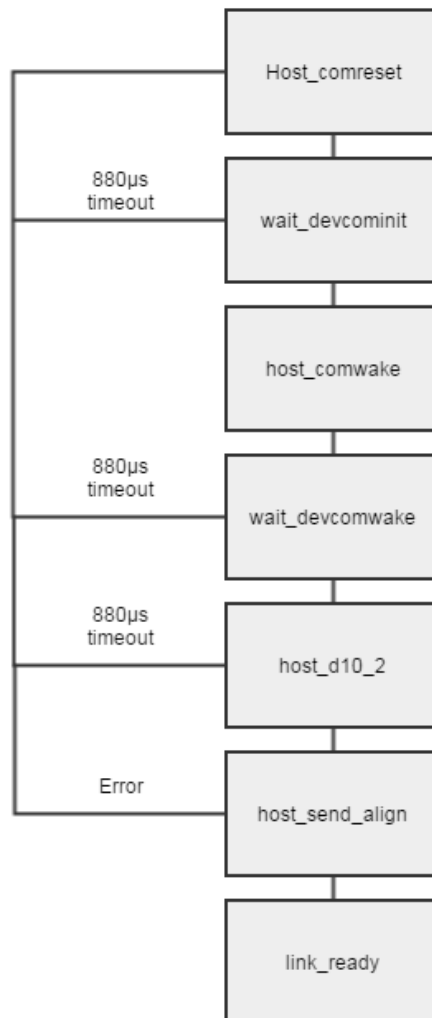
Voor het verzenden van gegevens maakt de link layer wordt de 32-bit CRC van de frame berekend en de CRC-waarde vervolgens erachter geplakt. Om de impact van EMI te verminderen is er een scrambler. De scrambler haalt de gegevens door een LFSR. Hierdoor wordt de impact van ruis op de kabels verspreidt over de hele frame. Aangezien physical layer gebruik maakt van de primitives blijven deze onaangetast. De architectuur maakt gebruik van een multiplex die wisselt tussen de gegevens die komen van de scrambler en de primitives.

Het ontvangen van gegevens gebeurt door de RX-FSM van de link layer. De RX-FSM ontvangt een FIS van de physical layer. Deze gaat hij eerst door de descrambler, waarna de gegevens op integriteit wordt getest door de CRC te calculeren. Bij een fout wordt de zender gevraagd voor de FIS opnieuw te sturen. Als de CRC correct is wordt de FIS doorgegeven naar de transport layer.

3.2.5 Physical layer

De GTX-transceivers zijn verantwoordelijk voor het serialiseren en deserialiseren van de gegevens. De physical layer is ook verantwoordelijk voor 8b/10b-encoding. Deze functionaliteit zit al in de GTX-transceivers ingebouwd. Een groot deel van de physical layer is daardoor gegenereerd door Xilinx Coregen – zogenaamde *wrappers*. [11] De GTX-transceivers zijn een dword breed en zijn voorzien van een klok van 75MHz. Dit komt neer op een snelheid van 300MB/s. Deze waarde komt overeenkomt met de praktische bandbreedte van SATA 3.0Gbit/s.

Naast de GTX-transceivers moet de physical layer voorzien zijn van een initialisatie, synchronisatie en reset van de link. Dit gebeurt aan de hand van OOB-signalen. Deze signalen zijn geïmplementeerd door middel van een FSM. Figuur 23 toont een flowchart van de link initialisatie van de SATA-core. [14]



Figuur 23 Link initialisatie

3.3 Proefprocedure

De snelheid wordt gemeten over verschillende opstellingen. De FPGA-opstelling wordt dus getest en vervolgens vergeleken met een desktopopstelling, waarbij de metingen zijn uitgevoerd met een identieke HDD door middel van een eenvoudig programma geschreven in C.

De desktopopstelling is getest met een installatie van zowel Windows 7 32-bit als Puppy Linux 32-bit. De processor is een Core 2 Duo op 1.86GHz met 3GB RAM. Tijdens het lopen van de test bleef de processor laag bezet, wat duidt op een bottleneck van de HDD. Op deze manier kan de maximumsnelheid van de HDD kan worden getest. De gebruikte HDD voor elke opstelling is een Seagate ST3750528AS. Deze heeft een capaciteit van 750GB met 512 bytes per sector. Dit komt neer op 1.465.149.168 sectoren. De datasheet van de HDD toont een maximale bandbreedte is 125MB/s. [10]

In de proefprocedure wordt er een bestand van 512 GB weggeschreven, dit is $2/3^{\text{de}}$ van de capaciteit van de schijf. Deze bestandsgrootte is gekozen om een beeld te scheppen analoog aan dat van een multicameraopstelling. Bij bestanden van deze grootte gaat de invloed van een aantal optimalisaties van een besturingssystemen minimaal zijn.

FPGA

De VHDL-code is voorzien van een parameter waarmee de application layer een parameter ontvangt klokcycli voor de uitgevoerde operatie. Aangezien de transceivers werken op een kloksnelheid van 75MHz kan er eenvoudig de bandbreedte worden berekend.

De opstelling behaalt schrijfsnelheden van 100.6MB/s. Deze snelheden liggen relatief dicht bij de theoretische maximumsnelheid van de schijf (125MB/s).

Windows

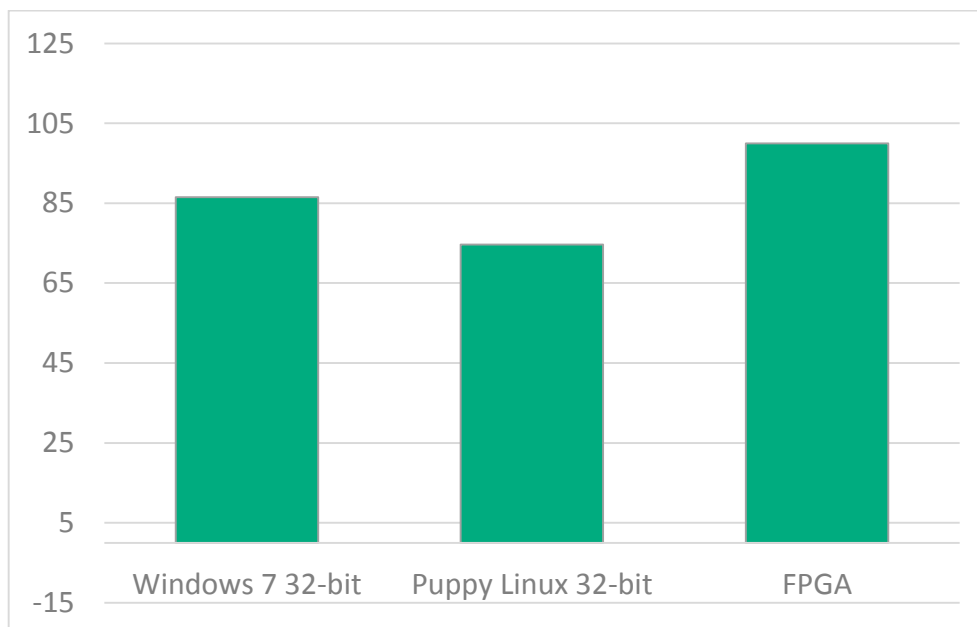
De opstelling met de Windows 7 32-bit-installatie is vervolgens getest met dezelfde harde schijf aangesloten via het SATA-interface en geformatteerd in NTFS.

De opstelling met Windows behaalt een snelheid van 86,5 MB/s. Deze snelheden liggen significant lager dan de FPGA-opstelling. De reden hiervoor is dat het besturingssysteem voor een overhead gaat zorgen en er na een tijd schijffragmentatie optreedt.

Linux

De laatste opstelling is deze met Puppy Linux 32-bit. Bij deze opstelling is de schijf wederom aangesloten via het SATA-interface maar nu geformatteerd in ext3.

De opstelling met Linux behaalt een snelheid van 74.7MB/s. Deze snelheid ligt lager dan de opstelling met Windows. De reden hiervoor is vermoedelijk dat het hier om een ouder bestuursysteem gaat en deze minder optimalisaties bevat dan een nieuwer besturingssysteem zoals Windows 7 – denk bijvoorbeeld aan NCQ (Native Command Queuing).



Figuur 24 Resultaten testopstellingen

4 Conclusie

De uitkomst is deze masterproef is een werkende FPGA-opstelling met een functionerende SATA 3.0Gbit/s-interface. Na het instellen van enkele parameters, zoals onder andere de framegrootte en de keuze van de operatie, kan de FPGA-opstelling individueel functioneren.

De architectuur is bidirectioneel; er kunnen dus zowel lees- als schrijfoperaties worden uitgevoerd. De schrijfoperaties worden sequentieel uitgevoerd zodat er geen last is van schijffragmentatie. In de testen worden er schrijfsnelheden gehaald die significant hoger liggen dan traditionele opslagsystemen.

Discussies

Hoewel de SATA-core functioneel is zijn er nog enkele functies die kunnen worden geïmplementeerd.

De SATA-core is niet voorzien van een onderhandelingsproces waarbij de FPGA ook compatibel is met media met een ouder SATA 1.5Gbit/s-interface. Het implementeren van deze functie vereist enkele extra stappen in het OOB-proces in de physical layer.

Het SATA-protocol ondersteunt de mogelijkheid tussen verschillende energieprofielen te wisselen. Echter is deze functionaliteit niet in de core geïmplementeerd. Voor deze functionaliteit toe te voegen moeten zowel de transport layer als de link layer worden aangepast.

De ML605 biedt de mogelijkheid meerdere SATA-schijven aan te sluiten. Dit biedt de mogelijkheid een softwarematig of zelfs hardwarematig een RAID-systeem te implementeren. De opstelling zou dan in een RAID0-configuratie te draaien voor een snelheidswinst, ofwel in een RAID1-configuratie voor een hogere betrouwbaarheid.

5 Referenties

- [1] Thomas Schmidt, "CRC Generating and Checking". Microchip Technology Inc, 2000.
- [2] N.Kiran Babu, P.S.Srinivas Babu. "Design of Physical Coding Sublayer using 8B/10B Algorithm.International Journal of Recent Technology and Engineering", vol 2, nr. 2, Mei 2014.
- [3] David A. Deming, "The Essential Guide To Serial ATA and SATA Express", CRC Press, 2015.
- [4] SATA-IO. "Serial ATA International Organization: Serial ATA Revision 3.1, 2011.
- [5] SATA-IO. "SATA-IO Unveils Revision 3.2 Specification", 2013.
- [6] Ravi Budruk, Don Anderson, Tom Shanley "PCI Express System Architecture", MindShare Inc, 2004.
- [7] Xilinx. "Virtex-6 Family Overview", 2012
- [8] Xilinx. "ML605 - SO DIMM shipped with newer version" 24 april 2012 [Online], Beschikbaar: <http://www.xilinx.com/support/answers/44814.html>, [Geraadpleegd op 1 maart 2015]
- [9] Xilinx. "FMC XM104 Connectibbity Card User Guide", 2010.
- [10] Seagate. "Barracuda 7200.12 Serial ATA", 2009.
- [11] Ashwin A. Mendon, Bin Huang, Ron Sass, "A High Performance, Open source SATA2 Core", University of North Carolina, 2012.
- [12] Ron Sass, Andrew G. Schmidt, "Embedded Systems Design with Platform FPGAs Principles and Practices", Elsevier, 2010.
- [13] T13, "AT Attachment 8 - ATA/ATAPI Command Set (ATA8-ACS)", 2006.
- [14] Matt DiPaolo, Simon Tam, "Serial ATA Physical Link Initialization with the GTP Transceiver of Virtex-5 LXT FPGAs", Xilinx, 2008.

6 Bijlagen

6.1 SATA primitives

Tabel 10 toont alle primitives gebruikt door de link layer en de physical layer van het SATA-interface.

Primitive	Naam	Functie
ALIGN_p	Physical Layer Control	Gebruikt voor regelen van physical layer
CONT_p	Continue repeating previous primitive	Toont aan dat vorige non-ALIGN _p onderdrukt is geweest
DMAT_p	DMA terminate	Eindigen DMA-transfer
EOF_p	End of frame	Indiceert dat een FIS en CRC zijn verzonden
HOLD_p	Hold data transmission	Door transmitter verzonden dat gegevens niet klaar zijn
HOLDA_p	Hold acknowledge	Respons op HOLD _p
PMACK_p	Power management acknowledge	Respons op PMREQ_P _p of PMREQ_S _p wanneer zender klaar is voor power mode te veranderen
PMNAK_p	Power management denial	Respons naar PMREQ_P _p of PMREQ_S _p wanneer zender niet klaar is of het niet wordt ondersteund
PMREQ_P_p	Power management request to partial	Transceiver gaat in partial power wanneer PMACK _p is ontvangen
PMREQ_S_p	Power management request to slumber	Transceiver gaat in slumber power wanneer PMACK _p is ontvangen
R_ERR_p	Reception error	Error in ontvangen payload
R_IP_p	Reception in progress	Zegt dat payload wordt ontvangen
R_OK_p	Reception with no error	Zegt dat payload ontvangen is zonder error
R_RDY_p	Receiver ready	Zegt dat ontvanger klaar is
SOF_p	Start of frame	Zegt dat volgende primitive Dword het eerst Dword van een payload is
SYNC_p	Synchronization	Synchronizeren van host en device
WTRM_p	Wait for frame termination	Wordt verzonden na EOF _p tot ook R_OK _p of R_ERR _p is ontvangen
X_RDY_p	Transmission data ready	Payload is klaar voor transfer

Tabel 10 Primitives [3]

6.2 SATA Voedingsconnector

Pin #	Volgorde	Functie
-		
1	3 ^{de}	Detectie aangesloten
2	2 ^{de}	5V
3	2 ^{de}	
4	2 ^{de}	Testpin fabriek
5	1 ^{ste}	Massa
6	1 ^{ste}	

Tabel 11 Slimlineconnector [6]

Pin #	Volgorde	Functie
1	3 ^{de}	3.3 V
2	2 ^{de}	
3	1 ^{ste}	Massa
4	1 ^{ste}	
5	2 ^{de}	5 V
6	3 ^{de}	
7	3 ^{de}	Gereserveerd
-		
8	3 ^{de}	Leverancierafhankelijk
9	2 ^{de}	

Tabel 12 Microconnector [6]

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Computational Video Mass Storage Array System

Richting: **master in de industriële wetenschappen: elektronica-ICT**

Jaar: **2015**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Vanoppen, Senne

Datum: **1/06/2015**