

Afstudeer verslag van Bas van der Heijden
Bedrijf:



www.nikhef.nl

Project: MuonLab II

Datum: 30-3-2006

Versie: FINAL

Stage begeleider: Kees Rijsenbrij

Bedrijf begeleider: Hans Verkooijen



Inhoudsopgaven

Inhoudsopgaven	2
1. Voorwoord	3
2. Wat is Muonlab II	4
2.1 Voor wie is Muonlab II	4
2.2 Het meten van muonen	4
2.3 Het PMT signaal	5
2.4 Muon lifetime	6
2.5 Muon delta time	8
3. Muonlab II hardware	9
3.1 De FPGA	10
3.2 De Comparator	10
3.3 De DAC's	10
3.4 De USB Interface	11
3.5 De Printplaat	11
4. VHDL	12
4.1 Wat is VHDL	12
4.2 De DCM's	13
4.3 DeltaTime_calc	14
4.4 LifeTime_calc	15
4.5 DAC_Control	15
4.6 USB_Control	15
5. De software	17
5.1 De afweging voor de programmeertaal	17
5.2 Wat zijn threads	17
5.3 Threads in Muonlab	18
5.4 De Com thread	18
5.5 File thread	19
5.6 De decodeer/histogram thread	19
5.7 Communicatie tussen de threads	19
5.8 De GUI	20
6. Wat is een Muon	21
6.1 Relativiteit en de muonklok	21
6.2 Kosmische straling in de atmosfeer	21
6.3 De oorsprong van kosmische straling	22
6.4 Samenstelling van kosmische straling	23
7. Ondervonden problemen	24
7.1 Problemen met de voeding	24
7.2 Problemen met JTAG	24
7.3 Problemen met hoge datarate usb	24
7.4 Problemen met de LifeTime data	25
7.5 De Jitter in de DeltaTime meting	25
8. Conclusie	26
9. Gebruikte software	27
10. Reverenties	27

1. Voorwoord

Dit verslag gaat over mijn afstudeer stage bij het Nationaal Instituut voor Kernfysica en Hoge Energie Fysica (NIKHEF) op de afdeling Elektronica Technologie.

Het NIKHEF bestaat uit een collaboratie tussen het Fundamenteel onderzoek der materie (FOM), de Universiteit van Amsterdam, de Vrije Universiteit Amsterdam de Radboud Universiteit Nijmegen en de Universiteit Utrecht. Het NIKHEF coördineert en ondersteund alle activiteiten in experimentele subatomaire en hoge energie fysica in Nederland.

Tevens voert het NIKHEF haar wetenschappelijk programma, op het gebied van Kernfysica en Hoge-Energie-Fysica, uit in internationale samenwerkingsverbanden. Met name op CERN in Geneve en DESY in Hamburg, maar ook op diverse andere plaatsen.

Ten behoeve van dit programma ontwikkelt de Groep Elektronica Technologie hoogwaardige elektronische apparatuur en systemen, in nauwe samenwerking met fysische werkgroepen binnen het NIKHEF en met werkgroepen in wetenschappelijke internationale collaboraties waarin NIKHEF participeert.

De opdracht die ik heb gekregen was het ontwikkelen van de software voor een Muon meet instrument.

De opdracht bestaat uit twee delen:

- Het programmeren van de FPGA
- Een applicatie schrijven

Bij het lezen van dit verslag is enige kennis van programmeren en digitale systemen nodig.

2. Wat is Muonlab II

Muonlab II heeft 2 functies.

- het muon lifetime experiment
- muon delta time meting experiment.

In het volgende stuk staan beide experimenten uitgelegd. Hier zal ook duidelijk worden aan wat voor specificaties Muonlab II moet voldoen om te kunnen functioneren.

2.1 Voor wie is Muonlab II

Muonlab is ontwikkeld in opdracht van een fysicus binnen het NIKHEF. Het doel van Muonlab is het leren over astro/particle fysica voor middelbare scholieren. Leerlingen kunnen zelf experimenten doen met het doel te leren over de fysica.

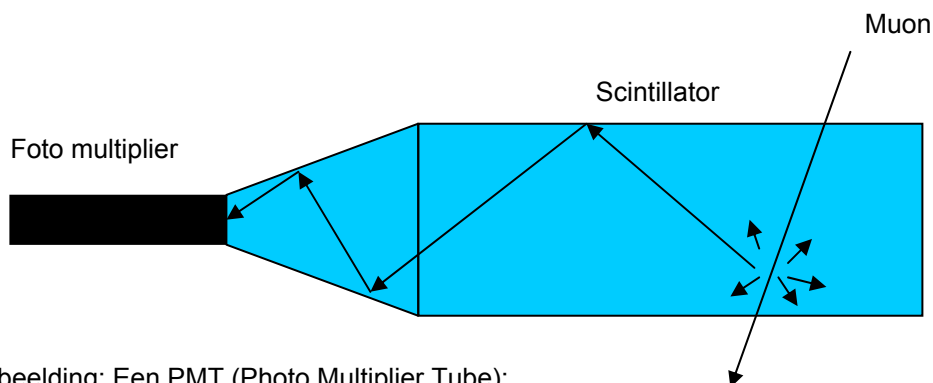
Het is de bedoeling dat de middelbare scholen en universiteiten contacten met elkaar onderhouden zodat de apparatuur geleend of geruild kan worden.

2.2 Het meten van muonen

Hoe detecteer je een muon (voor uitleg over een muon zie hoofdstuk 6) Dit gebeurt door gebruik te maken van scintillator materiaal. Dit is een speciaal soort materiaal (poly-vinyl toluene (PVT)) dat als het geraakt wordt door een geladen deeltje ioniseert. Hierdoor raken de elektronen in het PVT in een hogere baan. Omdat dit een instabiele toestand is zal het elektron terug vallen naar zijn oorspronkelijke baan, hier komt energie vrij in de vorm van (blauw) licht. De vorm van de scintillator is zo ontworpen dat de lichtpuls binnen het materiaal blijft tot het bij het eind is gekomen.

Aan het uiteinde van de scintillator is een PMT “fotomultiplier” gekoppeld. Deze zal de lichtpuls opvangen, versterken en omzetten in een elektrisch signaal.

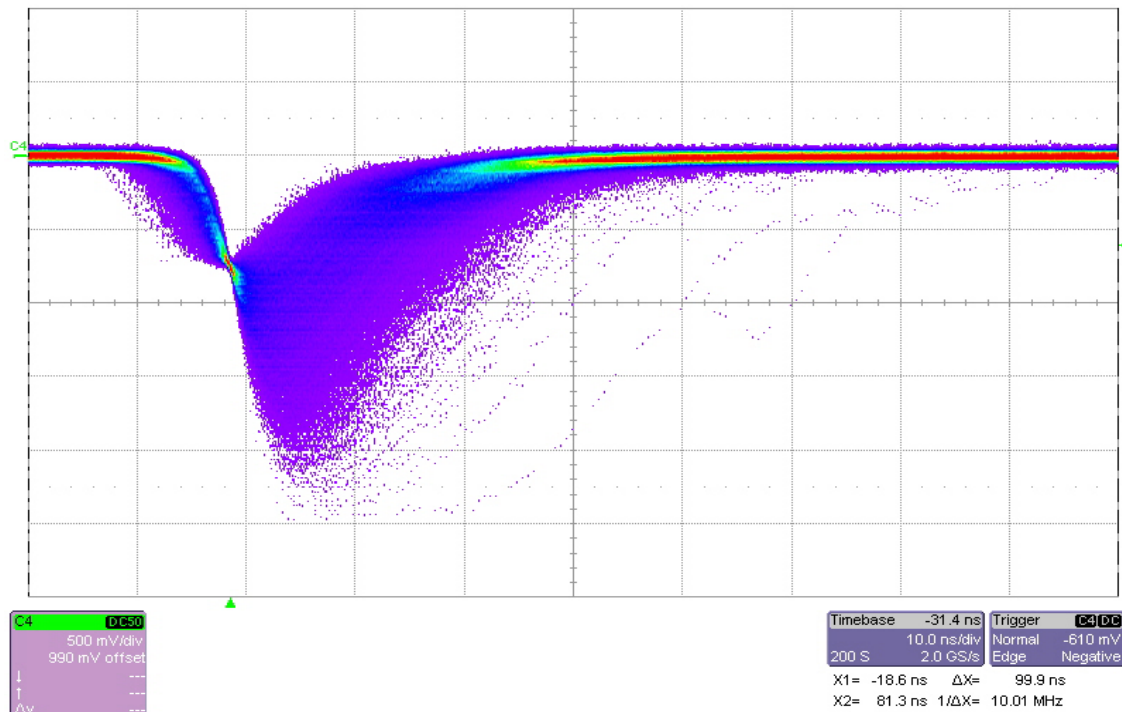
.



Afbeelding: Een PMT (Photo Multiplier Tube):

2.3 Het PMT signaal

In de volgende figuur is te zien hoe het signaal dat van de PMT afkomt er uit ziet. Deze meting is gemaakt in de “persistence mode” Dit houdt in dat ieder signaal wat gemeten wordt in de plot wordt getekend. De kleur geeft aan hoeveel signaal er op hetzelfde punt is geweest.



Afbeelding: Het signaal van de PMT gemeten over een tijdsverloop.

Aan de plot is te zien dat het signaal van de PMT niet constant is. Deze spreiding wordt veroorzaakt door het licht dat in het scintillatie materiaal weerkaatst en niet op het zelfde moment bij de PMT terecht komt. Daarnaast zit er nog ruis in het signaal wat standaard veroorzaakt wordt door de PMT.

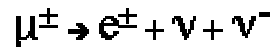
De in de volgende hoofdstukken besproken resoluties zijn zonder het “scintillatie effect” meegerekend.

Bij een meting met het scintillatie materiaal en PMT's treed een vervlechting in de meetnauwkeurigheid op.

2.4 Muon lifetime

In het muon lifetime experiment wordt gekeken naar de levensduur van een muon. De benodigdheden voor dit experiment zijn een stuk scintillatie materiaal en een photomultiplier (PMT).

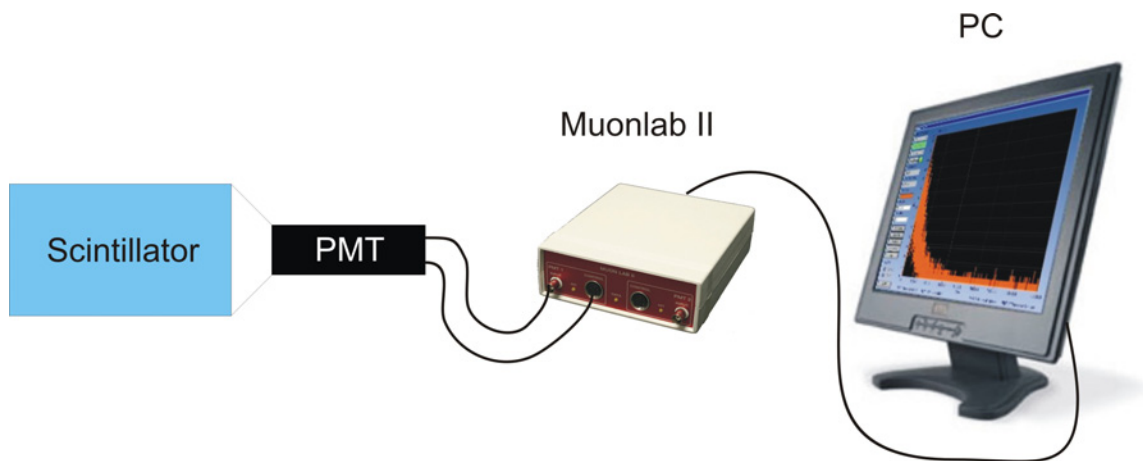
Het muon vervalt in een elektron of positron en 2 neutrino's



De gemiddelde levenstijd (τ_{μ}) van een muon is theoretisch $2.2\mu\text{s}$:

$$N(t) = N_0 \exp(-t/\tau_{\mu})$$

Deze stelling wordt bewezen in het lifetime experiment.



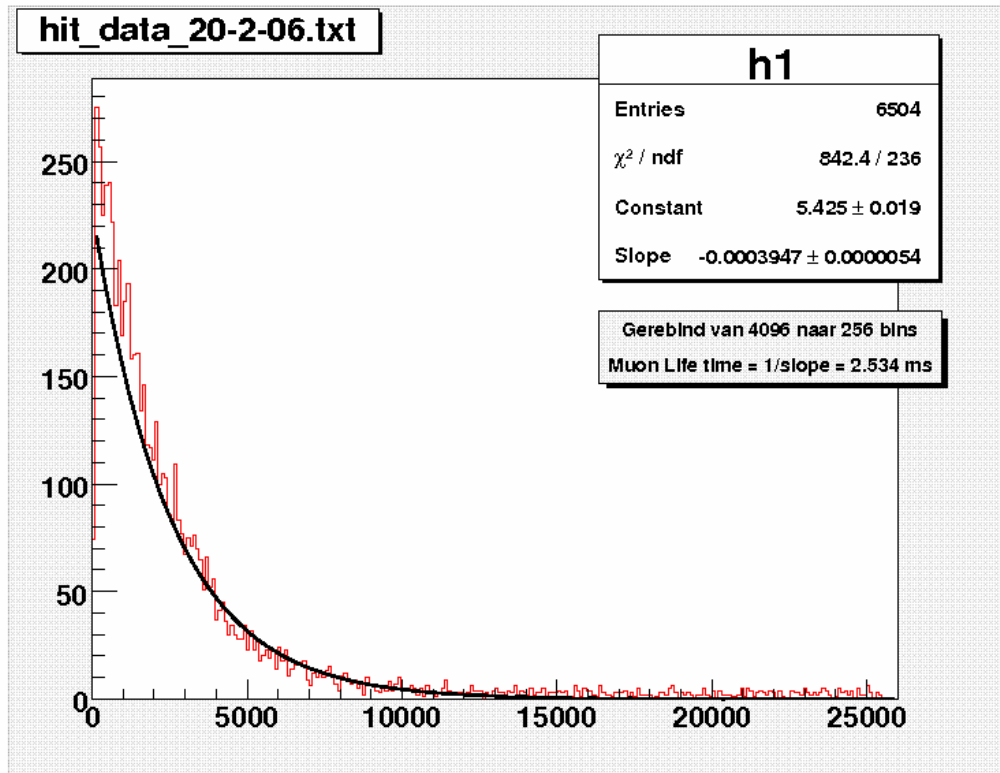
Figuur: De opbouw van de lifetime meting.

De meeste muonen passeren door het scintillatie materiaal en genereren pulsen licht. Deze lichtpulsen worden vervolgens omgezet in een elektrisch signaal door de photomultiplier.

Soms zal een muon gestopt worden door het scintillatie materiaal. Hierdoor ontstaat er 2 lichtpulsen achter elkaar. De eerste lichtpuls is van het verval van het muon en de tweede is van het scintillatie materiaal.

De tussentijd van deze twee signalen wordt over een bepaalde tijd gemeten om statistiek te verzamelen. Uiteindelijk zal er een exponentiële distributie ontstaan zoals in elk radioactief verval.

Met de curve die ontstaat uit de meting is doormiddel van een "curve fit" (de best passende functie) te bepalen wat de muon levenstijd is.



Figuur: Een curve fit van een lifetime meting gemaakt in het programma Root.

Om een nette verval curve te krijgen moet de maximale meettijd rond de $25\mu\text{s}$ liggen. De resolutie hoeft niet heel hoog te zijn omdat het anders erg lang duurt om genoeg statistiek te verzamelen.

De meetresolutie voor de lifetime meting is 6.25ns en de maximale meettijd bedraagt $25.6\mu\text{s}$

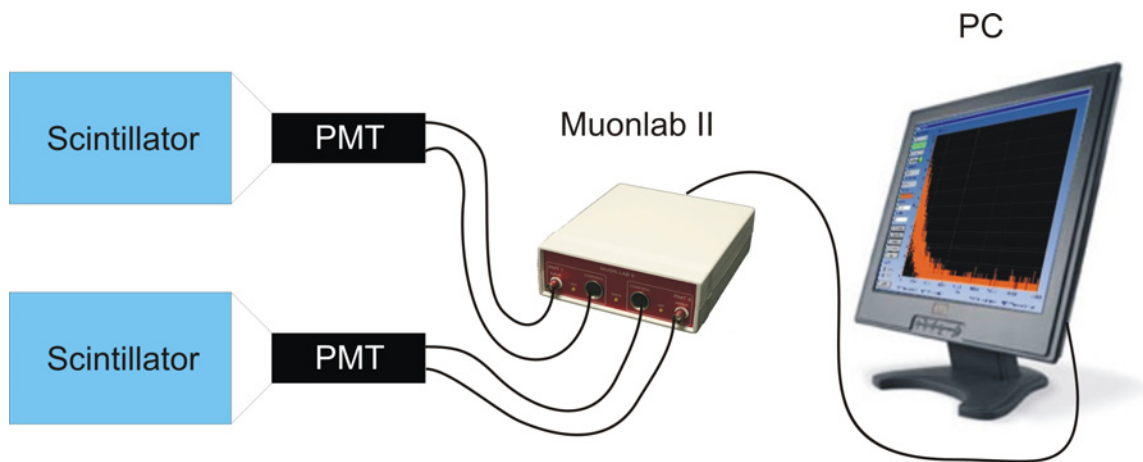
2.5 Muon delta time

Met het muon delta time experiment kan een coïncidentie worden bepaald tussen 2 meetpunten. Als er een coïncidentie optreedt wordt de verschil (delta) tijd gemeten.

Als er een puls van de eerste photomultiplier komt wordt de meting gestart. Er wordt vervolgens gewacht tot er een puls van de tweede photomultiplier afkomt. Deze tussentijd wordt gemeten en weergegeven in een histogram.

Een voorbeeld voor een meting met 2 scintillatoren:

Als twee scintillator staven boven elkaar gelegd worden zal er een muon door de eerste en vervolgens door de tweede staaf gaan. Aan de hand van de gemeten verschildtijd is te bepalen hoe snel het deeltje zich voortbeweegt.



Afbeelding: Een meetopstelling met 2 scintillatoren.

Aangezien de meeste muonen zich met bijna de lichtsnelheid voortbewegen moet de meetresolutie hoog genoeg zijn om dit waar te nemen. (licht legt een meter af in ongeveer 3.3ns)

De meetresolutie voor de delta time meting is 0.52ns en de maximale meettijd 2133ns

Om een resolutie te krijgen van 0.52ns is er een snelle klok nodig. In de code (VHDL) van Muonlab II is een voorziening gemaakt om te meten met bijna 2GHz ($1/0.52\text{ns}$) als frequentie. Dit zal uitgelegd worden in het VHDL hoofdstuk.

3. Muonlab II hardware

In het ontwerp van Muonlab was het belangrijk om de gebruiksvriendelijkheid hoog te houden, maar het moest ook goedkoop te produceren blijven.

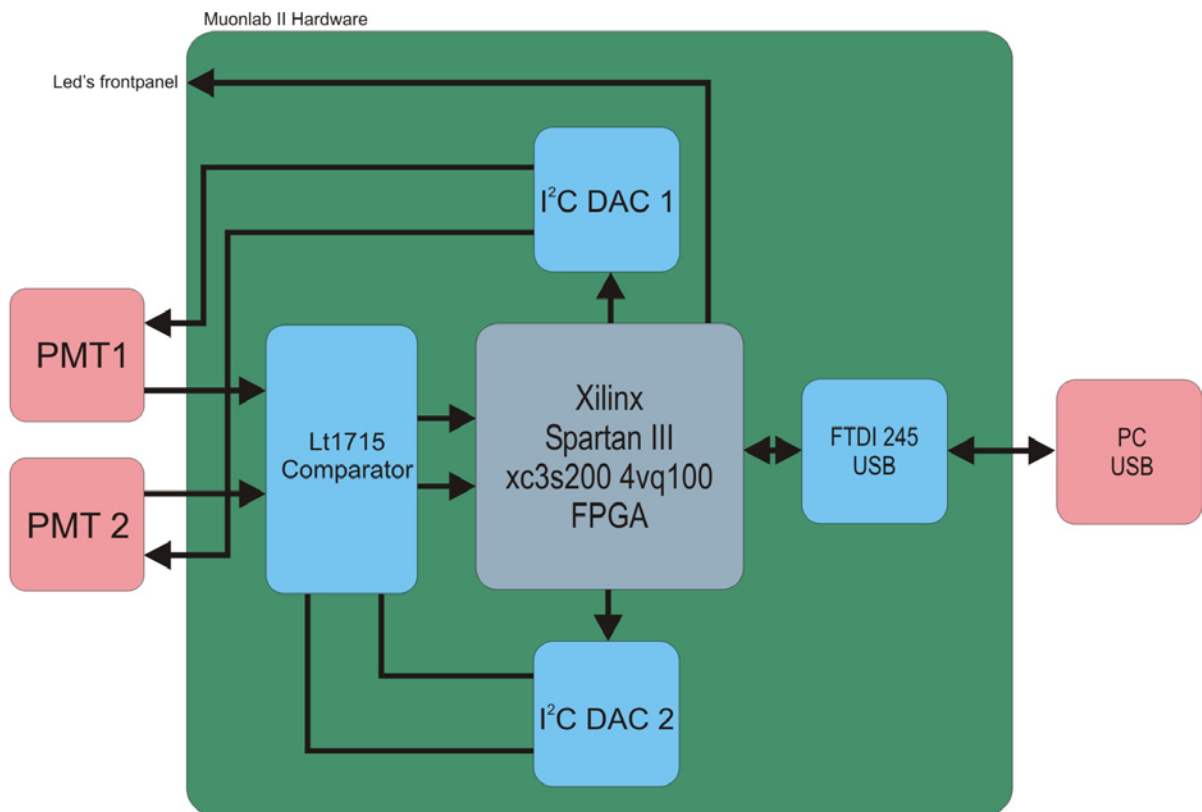
In dit hoofdstuk worden de volgende componenten besproken:

De FPGA

De comparator

De DAC's

De USB interface



Afbeelding: Het hardware schema.

3.1 De FPGA

Het centrale punt van de Muonlab hardware is de FPGA. Dit is een type chip, opgebouwd uit duizenden programmeerbare logische poorten. Een voordeel hierbij is dat de ontwikkelingstijden en ontwikkelingskosten erg laag zijn. FPGA's worden vaak gebruikt voor maatwerk of in de ontwikkelfase van een product.

De FPGA wordt geprogrammeerd door middel van de programmeertaal VHDL. Van dit programma wordt een bestand gemaakt die in de FPGA geladen kan worden door middel van JTAG. JTAG is een speciaal protocol voor deze toepassing.

Als de spanning van de FPGA af gaat zal het programma verloren gaan. Daarom heeft deze FPGA een EPROM geheugen. Als de FPGA opstart zal deze het programma ophalen van de EPROM.



Afbeelding: Een FPGA

3.2 De Comparator

De gebruikte FPGA heeft alleen digitale I/O. Om informatie van de buitenwereld naar binnen (en naar buiten) te kunnen halen is eerst andere hardware nodig.

Om de signalen van de PMT's om te zetten naar een digitaal signaal wordt er gebruik gemaakt van een comparator. Deze zorgt er voor dat een analoog signaal op een bepaalde drempel omgezet wordt in een digitale puls.

3.3 De DAC's

De drempel voor de comparator is in te stellen doormiddel van de DAC's. Muonlab bevat twee van deze "digitaal naar analoog" omzetter, waarvan elk met twee aparte kanalen.

De eerste DAC wordt gebruikt om de stuurspanning voor de PMT's te maken.

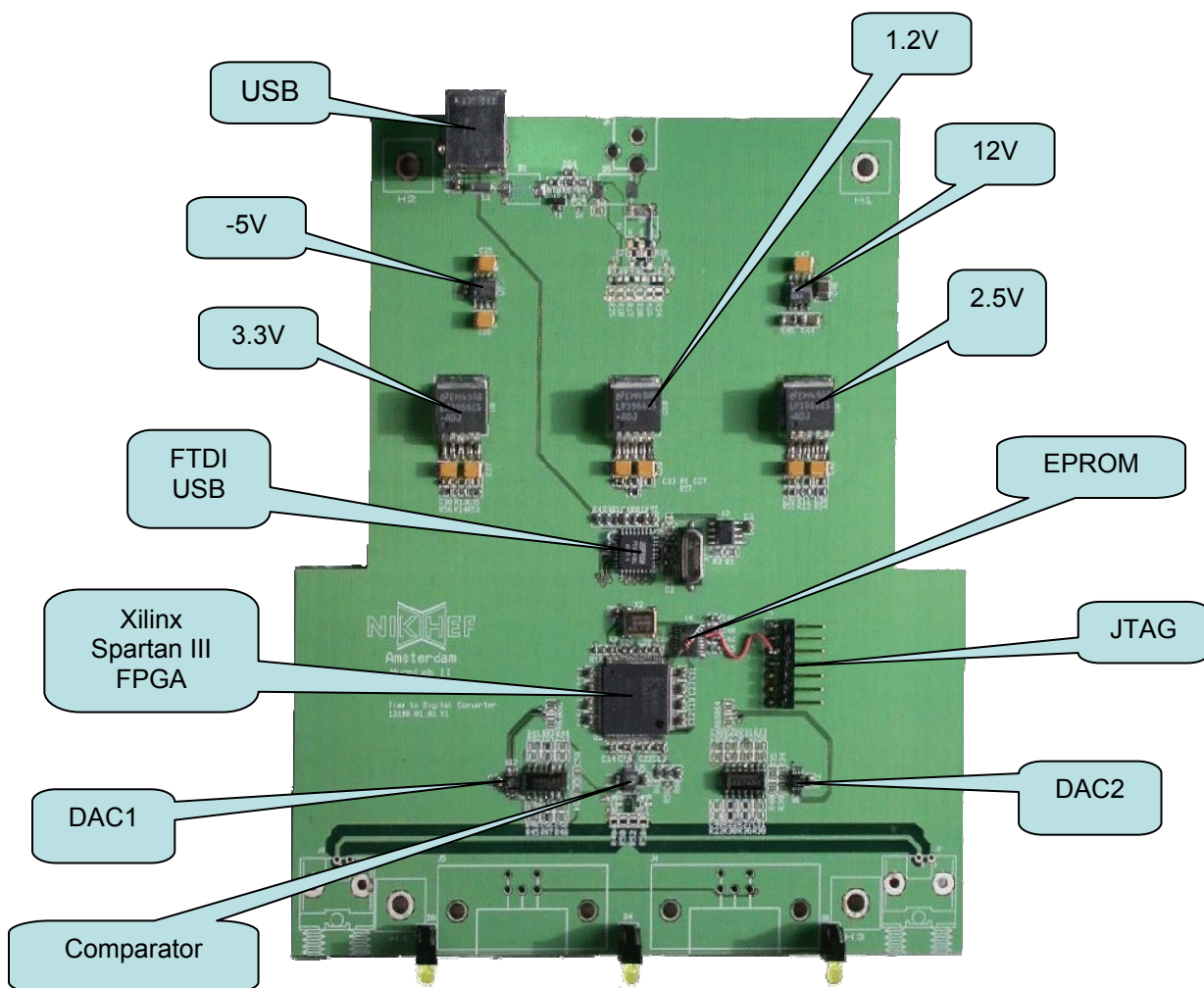
De tweede DAC wordt gebruikt om de drempelspanning op de twee analoge ingangen te regelen.

3.4 De USB Interface

Om de Muonlab aan de PC te koppelen wordt er gebruik gemaakt van een USB interface. Dit component zorgt voor de communicatie tussen de applicatie en de hardware.

Er wordt gebruik gemaakt van de FTDI FT245 USB FIFO. Hiermee is het mogelijk om een byte te versturen naar de PC. De byte wordt ontvangen op de PC door de FTDI driver die een virtuele compoort simuleert. Aangezien de compoort een stuk makkelijker uit te lezen is dan direct de USB versimpeld dit het programma aanzienlijk. Tevens is het mogelijk om een byte te versturen naar de USB interface. Alle handshaking en USB protocollen worden afgehandeld door de FTDI chip en de bijbehorende driver.

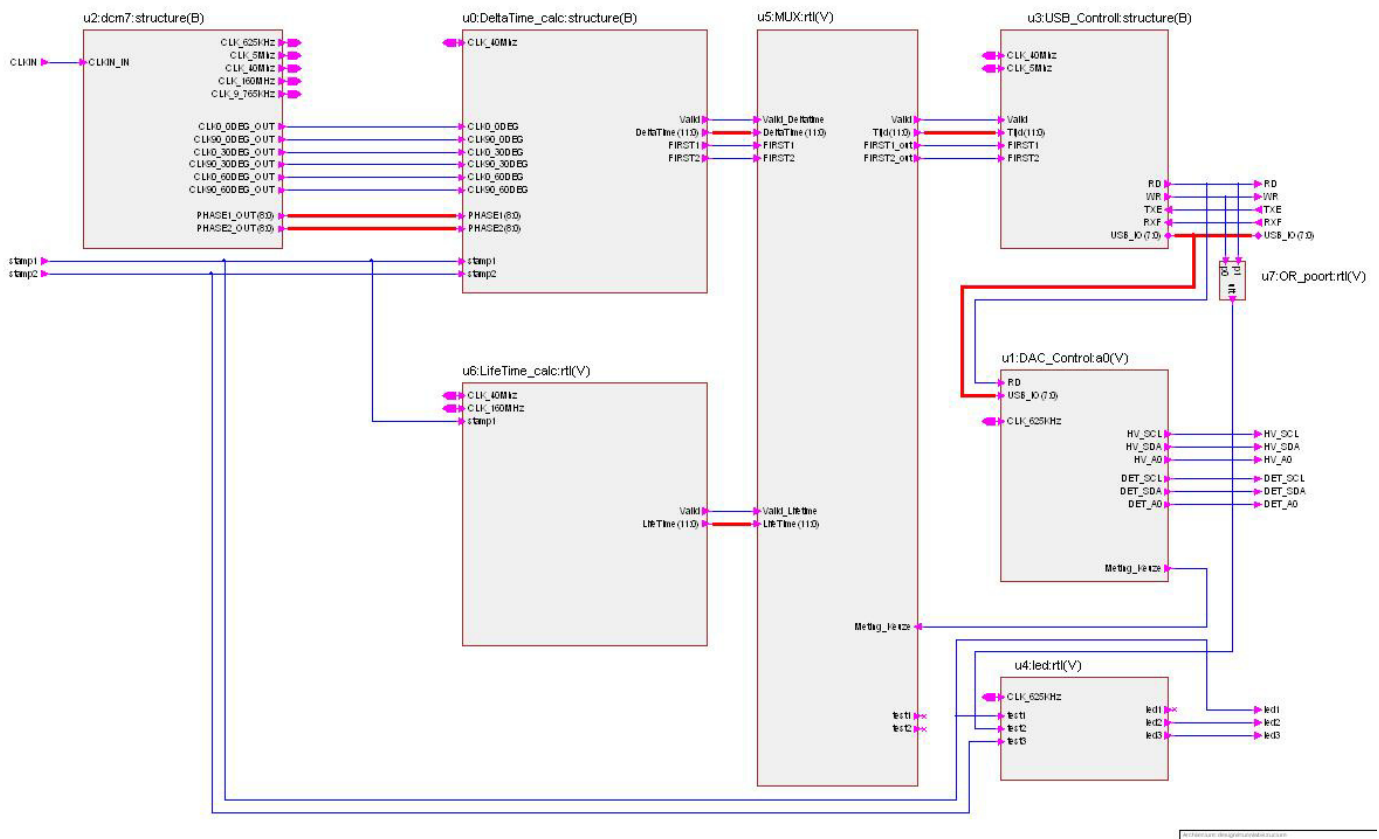
3.5 De Printplaat



4. VHDL

De VHDL code voor de FPGA heb ik geprogrammeerd in het programma Ease. Dit is een VHDL ontwerp pakket waarmee de VHDL in blokken gezet kan worden. Hierdoor wordt de code een stuk overzichtelijker en makkelijker te begrijpen en debuggen.

Voor het simuleren en synthetiseren heb ik ModelSim in combinatie met Xilinx Project Navigator gebruikt. Hiermee kon ik de VHDL in zijn geheel of in delen simuleren. Dit pakket heb ik ook gebruikt om de FPGA te programmeren.



Afbeelding 1: Het Ease blok schema van de VHDL.

In de volgende hoofdstukken staat de functie van de blokken VHDL uitgelegd.

4.1 Wat is VHDL

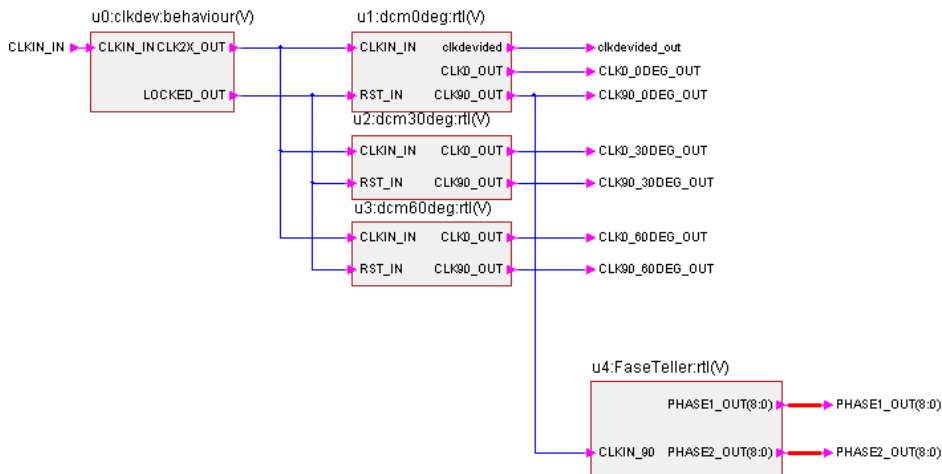
VHDL staat voor Very High Speed Integrated Circuits Hardware Description Language en is dus een taal om de Hardware mee te beschrijven. VHDL is een van de belangrijkste HDL's die wordt gebruikt door scholen en de ontwerp industrie. VHDL is als sinds 1987 een IEEE standaard en een FIPS (Federal Information Processing Standard). VHDL is de eerste echte HDL, het is afgeleid van Very High Speed Integrated Circuits (VHSIC) dat ontwikkeld is door de Amerikaanse regering.

4.2 De DCM's

De Digital Clock Managers (DCM's) bieden geavanceerde mogelijkheden om de binnenkomende klok signalen te manipuleren. De Spartan 3 beschikt over vier DCM's die allemaal gebruikt worden in dit project.

Een DCM biedt de volgende mogelijkheden:

- vermenigvuldigen of delen van een binnenkomend kloksignaal of een combinatie van deze.
- conditionering van het signaal voor een schone 50% duty cycle.
- Fase verschuiving met een vaste fractie van een klok signaal
- Uitsluiten van "Clock Skew" binnen de fpga of naar buiten toe.
- Een klok spiegelen, forwarden of rebufferen om een signaal bijvoorbeeld te converteren van LVTTTL naar LVDS.
- Een combinatie van bovenstaande.

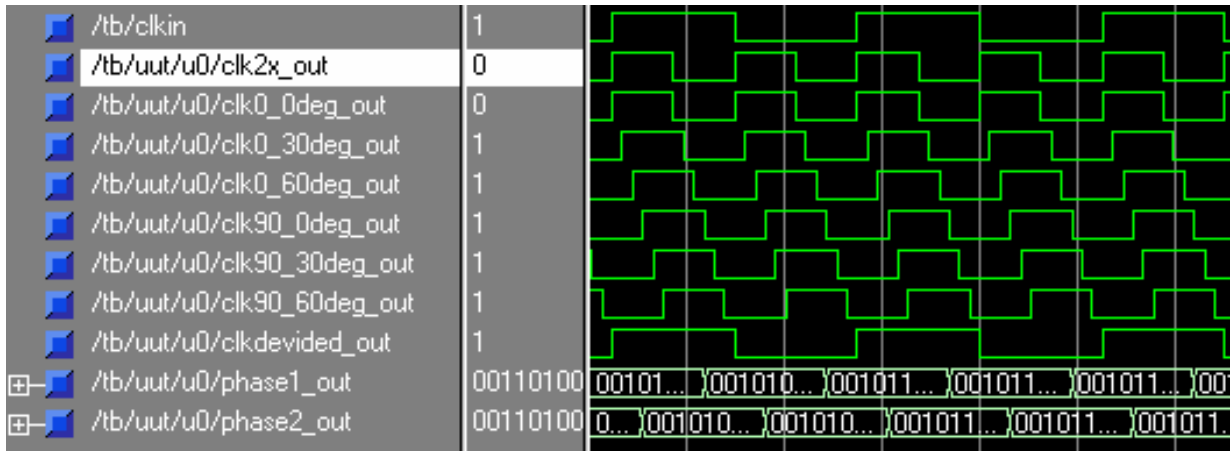


Afbeelding 2: De DCM's

Op de Muonlab print is een 80Mhz kristal aanwezig. Als deze frequentie gebruikt wordt om te het photomultipliersignaal te bemonsteren is een maximale resolutie van 12.5 ns mogelijk. Om deze resolutie te vergroten maken we als eerste gebruik van de x2 optie van een DCM. Vervolgens wordt deze klok opgedeeld in 12 stappen door middel van faseverschuiving. In de bovenstaande afbeelding is schematisch weergegeven hoe de vier DCM's gekoppeld zijn.

- u0 vermenigvuldigd de klok met 2
- u1 geeft de klok ongewijzigd en met 90 graden verschil door. Verder wordt in deze DCM de klok door 2 gedeeld voor overige logica.
- u2 verschuift de fase met 30 en 120 graden
- u3 verschuift de fase met 60 en 150 graden

Verder worden in u4 nog 2 tellers gegenereerd die nodig zijn om een langzamere klok te maken. Er zijn twee uit fase lopende klokken nodig om de snelle DCM klokken met de langzamere klok te synchroniseren.



Afbeelding 3: een simulatie van de klokken

In de bovenstaande afbeelding is de faseverschuiving van de klokken te zien. In een periode van clkin zijn nu 12 stappen te onderscheiden.

4.3 DeltaTime_calc

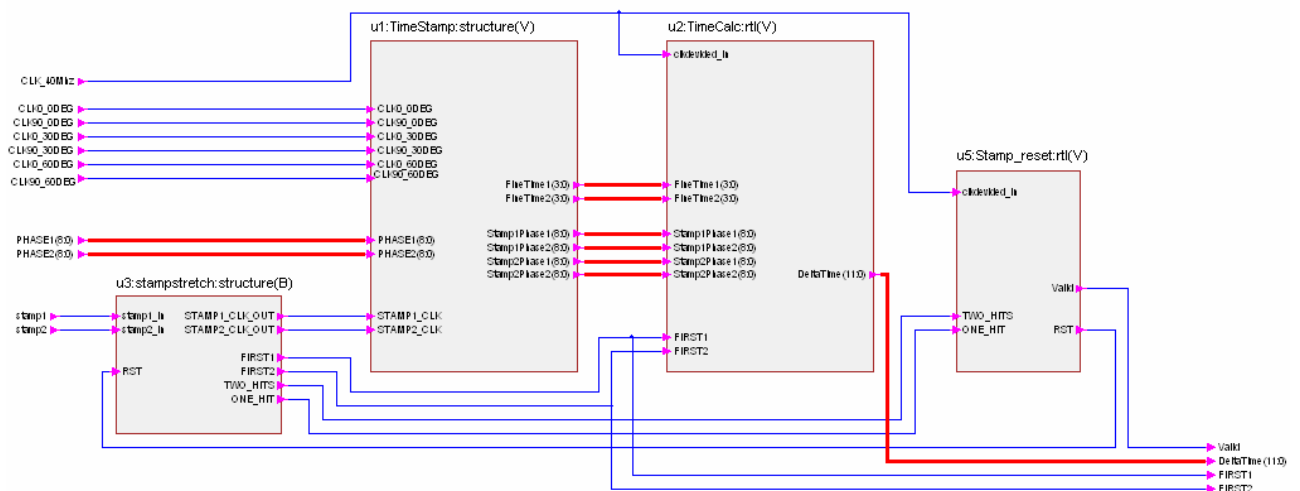
Muonlab bestaat uit 2 meetmogelijkheden. De DeltaTime meting en de LifeTime meting.

In het DeltaTime blok wordt de verschiltijd berekend tussen pulsen van de twee ingangen.

Eerst wordt er bepaald welke ingang de eerste puls heeft gekregen. Vervolgens wordt de toestand ingeklokt die op dat moment op de uitgang van het DCM blok staat. Aan de hand van deze data is te bepalen op welk punt in de langzame klok de puls is geweest.

Nu wordt gekeken naar de andere ingang voor de volgende puls en wordt de eigen ingang geblokkeerd. Als de tweede lijn pulst zal de het aantal klokpulsen van de langzame klok en de twee snelle klokken worden opgeteld tot een totale verschiltijd.

De maximale waarde die naar buiten kan worden geschreven is 12 bits. Bij een teller van 160MHz is dat 25.6µs. Omdat de periodetijd van 160 MHz nog door 12 wordt gedeeld is de resolutie 520ps.



Afbeelding 4: Het schema van het DeltaTimeCalc blok.

4.4 LifeTime_calc

Het LifeTime calc blok is om het muon verval te meten. Dit verval is een natuurconstante van ongeveer $2.2\mu\text{s}$.

Voor deze meting is slechts één ingang nodig. Op deze ingang wordt gekeken naar de verschiltijd tussen twee pulsen. Er kan gemeten worden op de 160MHz klok aangezien er geen grote nauwkeurigheid nodig is.

Deze meting werkt verder het zelfde als de DeltaTime meting. Op het moment dat de eerste puls binnenkomt wordt de teller gestart. Als de tweede puls komt wordt de teller gestopt en naar buiten gestuurd. Als er geen tweede puls binnen de maximale teller periode valt zal de meting gereset worden. De teller is net als de delta time 12 bits, maar omdat de waarde niet door 12 wordt gedeeld is de maximale meettijd $25.6\mu\text{s}$ met een resolutie van 6.25ns

4.5 DAC_Control

In het dac control blok worden de twee I2C DAC's (digitaal naar analoog omzetter) aangestuurd. De data die terug komt van de USB poort wordt opgevangen door het dac control blok. Aan de hand van deze gegevens kan de uitgangsspanning van de DAC's worden ingesteld. Beide DAC's zijn op een eigen bus aangesloten.

4.6 USB_Control

Het usb control blok bestaat uit twee onderdelen. Het eerste onderdeel is de interface die zorgt voor het protocol tussen Muonlab en de applicatie. Het tweede onderdeel bestuurt de FTDI chip.

Omdat de USB interface alleen bytes kan versturen moet het 12 bits getal opgesplitst worden. Dit gebeurt als volgt:

Er worden twee bytes gemaakt (een higher byte en een lower byte) die achterelkaar worden verstuurd.

Het eerste bit is geeft aan of het byte het higher of het lower byte is.

Het tweede bit geeft aan welke ingang de eerste puls heeft gegeven. Een 1 in het higher byte betekent dat op ingang 1 de eerste puls is geweest. Een 1 in het lower byte betekent de eerste puls op ingang 2.

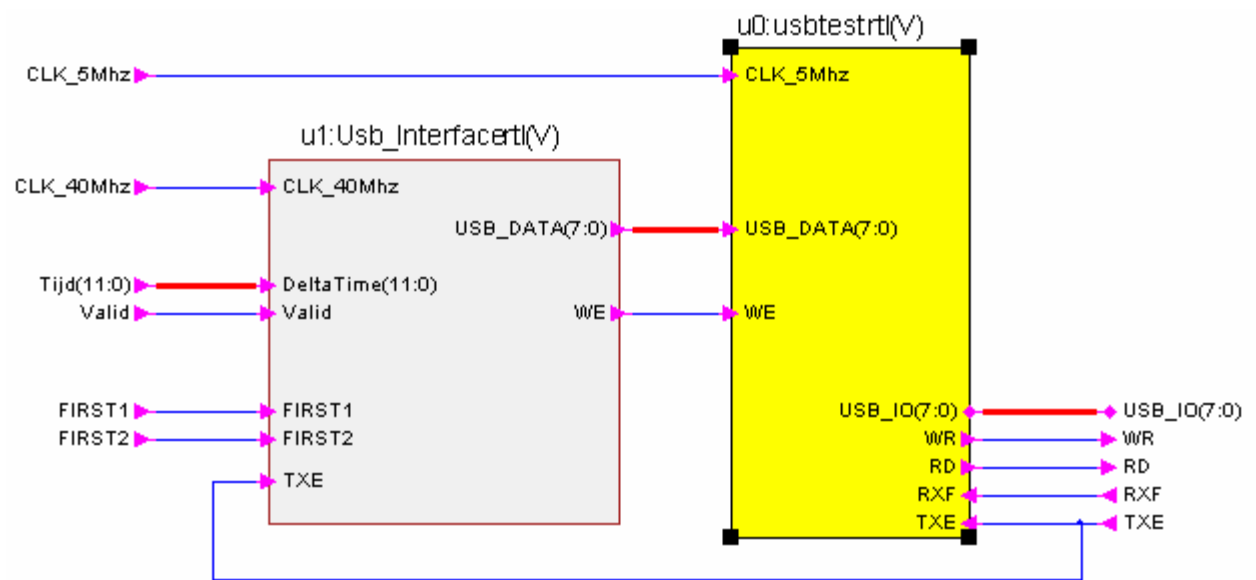
Het 12 bits antwoord van de berekening wordt opgesplitst in de laatste 6 bits van het higher en lower byte.

higher byte

high/low	First hit	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6
----------	-----------	--------	--------	--------	--------	--------	--------

Lower byte

high/low	First hit	Data 7	Data 8	Data 9	Data 10	Data 11	Data 12
----------	-----------	--------	--------	--------	---------	---------	---------



Afbeelding 5: de interface en de USB besturing.

5. De software

Naast de VHDL moest er voor Muonlab een applicatie geschreven worden. De voorwaarden voor deze applicatie waren dat het er “gelikt” uit moest zien, maar ook eenvoudig te bedienen zijn.

5.1 De afweging voor de programmeertaal

De programmeertaal om de Muonlab applicatie in te schrijven was nog niet vastgesteld bij de aanvang van de stage. Van te voren was wel uitgezocht of alle randvoorwaarden mogelijk waren. Hier bij was de programmeertaal Labwindows gebruikt.

Ik had de keuze uit Java, MS Visual C++ of Labwindows, aangezien deze programmeertalen snel voor handen waren en ik eerdere ervaring er mee had.

Omdat het programma een grafische user interface moest hebben, moest de programmeertaal een voorziening hebben om eenvoudig een gui in elkaar te zetten.

De afwegingen op een rijtje:

- Eerdere ervaring
- Simpel een gui maken
- Het liefst object georiënteerd zijn
- Thread voorziening hebben

Ik heb uiteindelijk voor Labwindows gekozen. Het grote voordeel van Labwindows is dat een gui maken erg simpel is. Ik was in het begin bang dat ik het de gui niet genoeg kon vormen naar mijn wensen, maar dit bleek uiteindelijk goed te doen te zijn.

Labwindows maakt als basis gebruik van C met eigen bibliotheken. Aangezien ik al eerder met C heb geprogrammeerd was dit goed te doen.

Echter een groot nadeel van Labwindows is het feit dat het niet object georiënteerd is. (C maar geen C++) Hierdoor kan het Labwindows programma snel onoverzichtelijk worden.

Labwindows werkt door middel van zogenaamde “Callback functies”. Dit zijn methodes die aangeroepen worden als er bijvoorbeeld op een knop wordt gedrukt. Met deze functie kan vervolgens bijvoorbeeld een thread gestart worden.

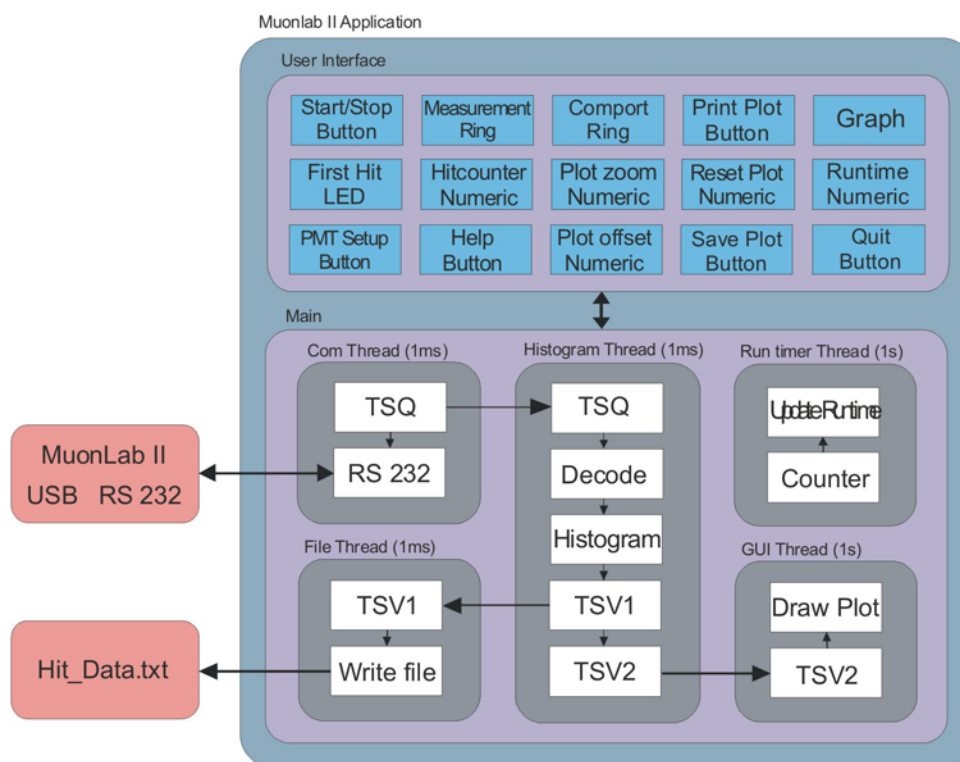
5.2 Wat zijn threads

Threads zijn als het ware een opsplitsing van de applicatie in aparte (onzichtbare)applicaties. De reden om een applicatie op te splitsen in threads is om een programma niet te laten wachten op een gebeurtenis. Er zal bijvoorbeeld continu naar de UI (User Interface) gekeken moeten worden of er op een knop wordt gedrukt. Een applicatie zonder threads zou hier op blijven hangen.

5.3 Threads in Muonlab

De Muonlab applicatie bestaat uit een aantal verschillende threads, omdat er verschillende zaken tegelijk moeten gebeuren.

- Er zal continu naar de UI gekeken moeten worden of er een variabele wordt veranderd of een knop wordt ingedrukt.
- Er moet gekeken worden of er data op de COM poort aangeboden wordt.
- Het bestand moet geschreven worden met de data.
- De data moet “gehistogrammeerd” worden.
- Tot slot loopt er nog een runtime die alleen de secondes telt, maar dit wel onafhankelijk doet van de rest van de applicatie.



Afbeelding: schematische voorstelling van de threads en functies in de Muonlab applicatie.

5.4 De Com thread

Deze thread bekijkt of er data in de buffer van de (virtuele) COM poort staat. Om te zorgen dat de PC niet continu aan het pollen is waardoor het CPU verbruik constant 100% is zit er een delay in van 1ms. Deze delay functie zorgt er voor dat de thread 1 milliseconde slaapt. De USB naar COM poort interface heeft een buffer van een aantal Kilobyte waardoor er niet snel dataverlies op zal treden. Verder zit de delay functie in een else statement zodat de thread alleen wacht als er geen data in de buffer staat.

5.5 File thread

Als de optie “save data” aan staat zal deze thread de data die van de COM poort af komt in een bestand opslaan. In deze thread zit ook een delay om de rest van het programma niet op te hangen als er in een bestand wordt geschreven.

5.6 De decodeer/histogram thread

In de histogram thread worden twee functies afgehandeld. Eerst wordt de data van de Com thread gedecodeerd. De data van de Com thread zijn de bytes die gemaakt zijn in de FPGA van de data. Deze bytes bevatten het 12 bits datawoord voor de meetwaarde, de higher of lower byte en de First hit poort. Het 12 bits woord wordt uit deze twee bytes gehaald en naar de histogram functie gestuurd.

In de Muonlab applicatie is de histogram een array van 4096 double's. Omdat de gemeten waarde maximaal 12 bits is zijn hier alle mogelijkheden aanwezig. Op het moment dat er een 12 bits datawoord binnen komt zal die plek in het array met 1 worden opgehoogd. Zo zal er na verloop van tijd een histogram ontstaan van de meetwaardes.

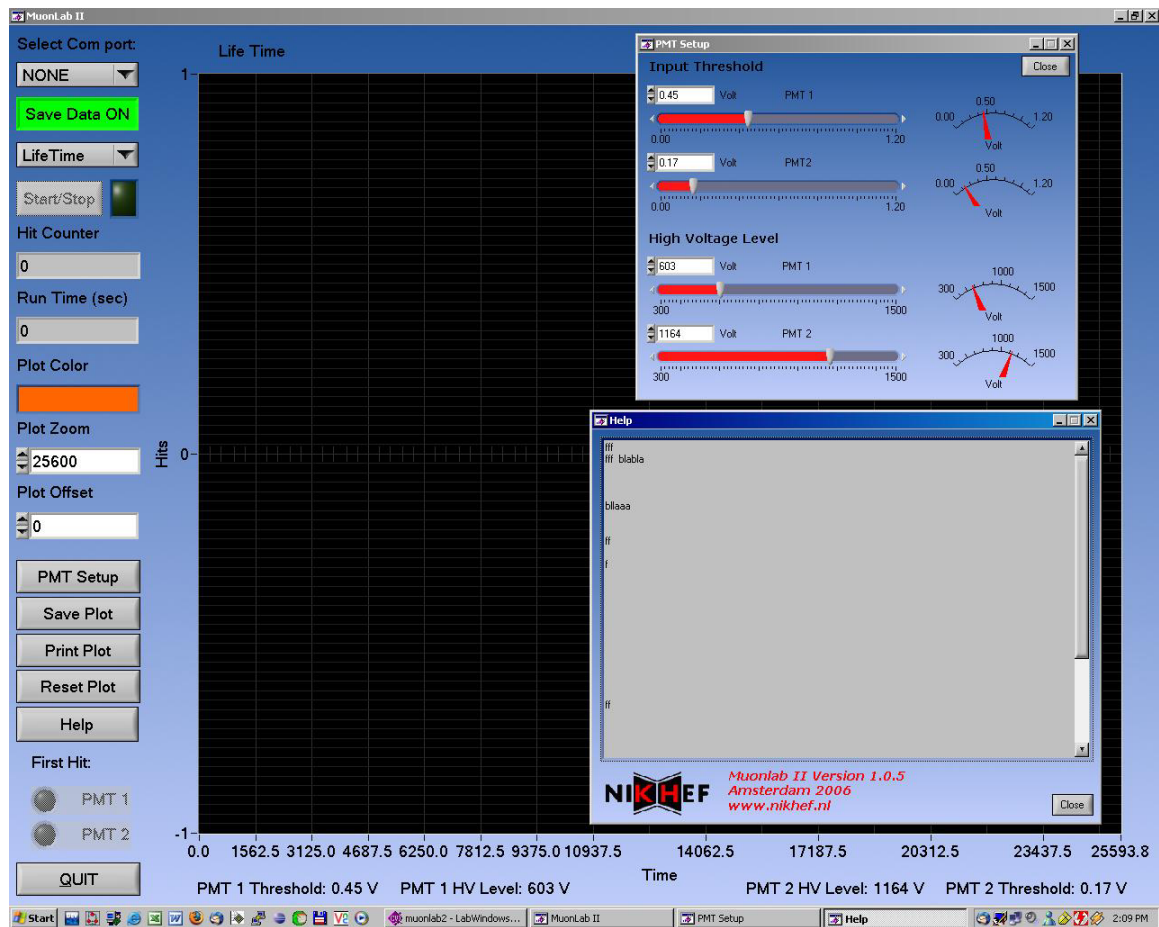
5.7 Communicatie tussen de threads

De communicatie tussen de threads verloopt via “thread save queues” en variabelen. Omdat threads los van elkaar functioneren is het lastig om data tussen de threads uit te wisselen. Als de ene tread data in een (global)variable veranderd en de andere thread moet dit vervolgens uitlezen weet je nooit zeker welke van de twee het eerste is en of de informatie overgedragen is. Om dit op te lossen wordt er gebruik gemaakt van thread save queues en variabelen. Aan deze overdrachtvormen is een mechanisme gekoppeld, die de toegang tot de informatie serializeerd zodat threads niet gelijktijdig een waarde kunnen veranderen of lezen. Doormiddel van de queues kan een thread eenvoudig reageren of deze data aangeboden krijgt of moet versturen.

5.8 De GUI

Met het maken van de GUI heb ik zoveel mogelijk proberen het scherm overzichtelijk te houden. Daarom heb ik een aantal functies in een apart scherm geplaatst. Hierdoor wordt het beeld minder snel onoverzichtelijk.

In de volgende afbeelding is een schreenshot te zien van de Muonlab applicatie. (Het help scherm moet hier nog gevuld worden)



Afbeelding: Muonlab screenshot.

Het PMT instellingen scherm en het help scherm wordt opgeroepen door knoppen op het hoofdscherm.

De PMT waardes staan onder de plot op het hoofdscherm zodat bij het printen en opslaan de informatie erbij staat.

6. Wat is een Muon

In het volgende hoofdstuk zal een stukje theorie worden uitgelegd over het Muon. Deze informatie is verder niet belangrijk voor het de opdracht.

Het muon is één van de exotische deeltjes die in de kosmische straling zijn ontdekt. Het muon is een zwaardere versie van het elektron, met tweehonderd keer zoveel massa, maar met dezelfde elektrische lading. In tegenstelling tot het elektron is het niet stabiel. Het vervalt gemiddeld na ongeveer twee miljoenste seconde (500ns), waarbij het overgaat in een elektron en nog lichtere ongeladen deeltjes: neutrino's.

Tegen de tijd dat kosmische straling vanaf buiten ons melkwegstelsel de aarde bereikt zegt het standaard model dat ze een maximale energie kunnen hebben van 5×10^{19} eV, bekend onder de naam Greisen-Zatsepin-Kuzmin (GZK) limiet. Dit is omdat de kosmische straling energie verliest door botsingen met fotonen van de kosmische microgolf achtergrond als ze door de ruimte racen met een snelheid dichtbij de lichtsnelheid.

6.1 Relativiteit en de muonklok

Een muon leeft gemiddeld slechts twee miljoenste seconde. Zelfs als het met de lichtsnelheid beweegt kan het in die tijd nog geen kilometer afleggen. De meeste muonen ontstaan rond een hoogte van tien kilometer en dat betekent dat ze het aardoppervlak helemaal niet kunnen bereiken. Toch meet zelfs de simpelste deeltjes detector op aarde elke seconde gemiddeld 180 muonen per vierkante meter.

De relativiteit theorie van Einstein geeft de verklaring: als objecten sneller bewegen gaat hun klok langzamer t.o.v. onze klok. Omdat de muonen bijna met de lichtsnelheid gaan, leven ze in onze tijdsbeleving vijf tot tien keer langer. Voor een groot aantal muonen is dat lang genoeg om de aarde te bereiken.

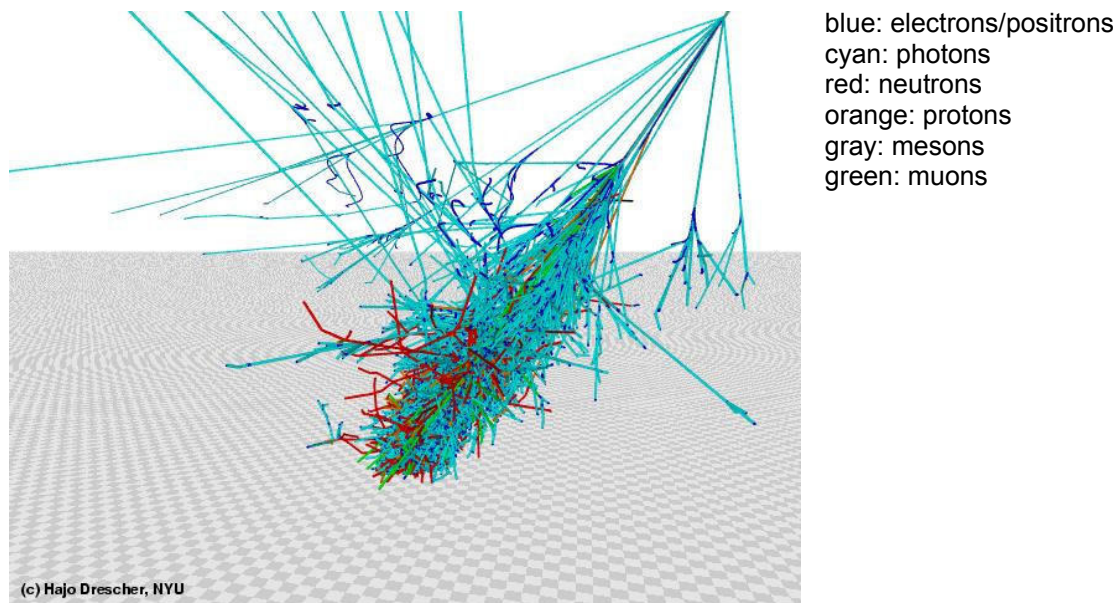
6.2 Kosmische straling in de atmosfeer

Kosmische straling wordt op aarde nooit rechtstreeks waargenomen. We meten de effecten van de wisselwerking tussen de straling en de atmosfeer.

Als een kosmisch deeltje de dampkring binnendringt, zal het botsen met stikstof of zuurstofatomen in de lucht. Dit ontketent een kettingreactie: de brokstukken van deze botsing raken andere atomen, en zo gaat het verder. Het resultaat is een regen van deeltjes in de atmosfeer, een zogenaamde airshower.

De meeste van deze deeltjes hebben relatief weinig energie. Ze vervallen, of ze worden verstrooid of opgenomen in de dampkring voor ze het aardoppervlak kunnen bereiken. Alleen deeltjes met heel veel energie, of deeltjes die 'lang genoeg leven', bereiken de grond. Een veelvoorkomend deeltje is het muon.

In de afbeelding hier onder is een simulatie te zien van een proton die de atmosfeer raakt:



Afbeelding: Een airshower.

Meer simulaties zijn hier te vinden:

<http://www.th.physik.uni-frankfurt.de/~drescher/CASSIM/>

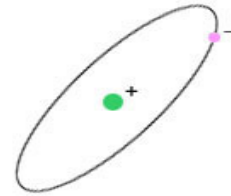
6.3 De oorsprong van kosmische straling

Het bestaan van kosmische straling is nu al meer dan 100 jaar bekend. Er is in die tijd veel onderzoek naar gedaan. Zo weten we dat de zon en andere gewone sterren enorme hoeveelheden laagenergetische deeltjes de ruimte in spuiten. Kosmische straling met veel meer energie komt van zware sterren die aan het eind van hun leven uiteenspatten, de zogenaamde supernova.

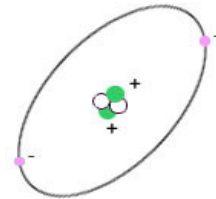
Maar zelfs een supernova is niet in staat om de extreem hoogenergetische deeltjes te maken. Natuurkundigen begrijpen gewoonweg nog niet hoe deeltjes zoveel energie kunnen krijgen. Dat kan oplopen tot een biljoen keer meer dan de energierijkste zonnedeeltjes. Ze moeten van erg ver komen, zeker van buiten ons zonnestelsel en zelfs van buiten de Melkweg. Want iets in het heelal dat dergelijke ultra energetische deeltjes aanmaakt zal met zeer heftige processen gepaard gaan, en die hebben we zelfs in de verre omgeving van de Melkweg nog niet kunnen waarnemen.

6.4 Samenstelling van kosmische straling

Elektronen, protonen en zware atoomkernen komen ook als vrije deeltjes in de natuur voor. Zo'n 75% van alle kosmische straling bestaat uit waterstof (protonen) en bijna 25% uit heliumkernen, ook bekend als alfadeeltjes. De rest bestaat uit elektronen, zware atoomkernen en gammastraling, een zeer energetische vorm van röntgenstraling.



Een waterstof atoom.



Een Helium atoom.



7. Ondervonden problemen

Gedurende mijn stage ben ik tegen een aantal problemen aangelopen. In de volgende hoofdstukken staat omschreven wat deze problemen inhielden en hoe ik ze heb opgelost.

7.1 Problemen met de voeding

Bij het ontwerp van de Muonlab print was er een voorziening voor een externe adapter en voeding via de USB. Aangezien het nog niet bekend was hoeveel stroom de hardware en FPGA zouden trekken was dit een voorzorgsmaatregel. Nadat de VHDL af was en alle hardware op de print zat bleek het stroomverbruik onder de 200mA te blijven. USB heeft de mogelijkheid om 500mA te leveren voor alle apparaten op de bus, dus Muonlab zit hier ruim onder. Op de productie versie van de print zal de voorziening voor een externe adapter niet aanwezig zijn.

Er was ook een probleem met het opstarten van de FPGA. In sommige gevallen sprong de FPGA in "latch up". Dit houdt in dat het programma niet goed in de FPGA is geladen en de FPGA intern een kortsluiting kan veroorzaken. Dit probleem bleek veroorzaakt te worden door het missen van een pullup weerstand op de enable lijn tussen de FPGA en de EPROM.

7.2 Problemen met JTAG

Bij de eerste poging om de FPGA te programmeren werkte het niet. JTAG werkt door middel van een chain (ketting) van apparaten. In dit geval de FPGA en de EPROM. Ieder apparaat moet op zijn beurt de juiste data doorgeven. In mijn geval bleek de FPGA de data niet goed door te geven. Dit heb ik uiteindelijk niet op kunnen lossen. Omdat de FPGA in principe niet direct geprogrammeerd hoeft te worden en de koppeling tussen de EPROM en de FPGA via een andere bus gaat, heb ik de FPGA uit de JTAG chain gehaald.

Als het programma in de EPROM wordt geladen haalt de FPGA bij opstarten het programma weer uit de EPROM.

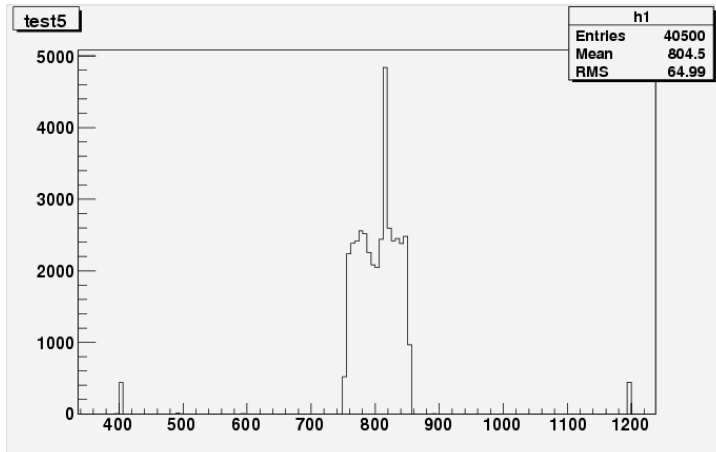
7.3 Problemen met hoge datarate usb

Tijdens het testen bleek er een probleem te ontstaan bij het oversturen van veel data over de USB verbinding. In het geval dat er veel hits snel na elkaar komen kan de fifo buffer van de USB overlopen. Dit veroorzaakte een fout in de applicatie die vervolgens bleef hangen. Na een aanpassing in het programma en in de VHDL was dit probleem opgelost.

Muonlab kan afhankelijk van hoe snel de PC is, hits verwerken tot ongeveer 100.000 keer per seconde

7.4 Problemen met de LifeTime data

Bij het testen van de LifeTime meting bleek onjuiste data van de meting te komen.



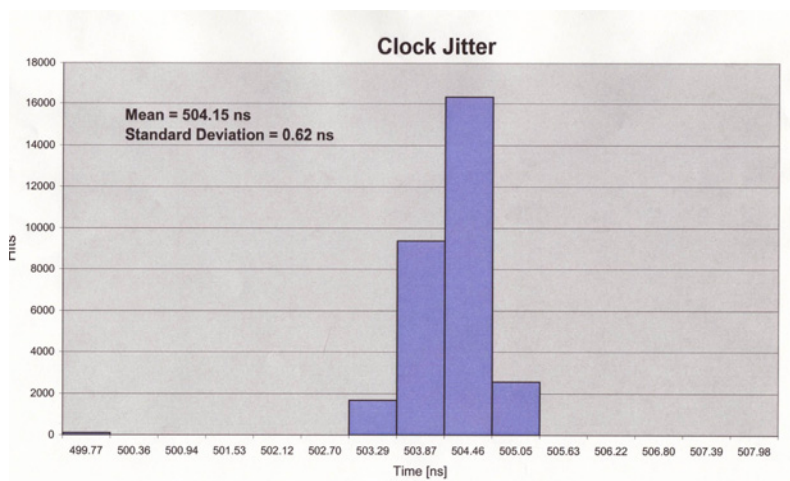
Afbeelding: een test met een gelijk aantal hits van 750 tot 850ns. Deze pieken horen niet.

Dit werd uiteindelijk veroorzaakt door dat de applicatie soms de higher en lower byte door elkaar haalde. Hierdoor ontstaan er vage pieken in de plot. Dit komt doordat bij het samenvoegen van de data, de higher en lower byte achter elkaar wordt geplaatst (zie hoofdstuk 4.6). Het lower byte is minder significant dan het higher byte. Hierdoor ontstaat op bepaalde meetwaardes onjuiste data.

7.5 De Jitter in de DeltaTime meting.

In de praktijk blijkt de interne klok toch een beetje te "jitteren" dit houdt in dat het tijdsvak niet altijd even groot is. De meeste hits zullen in het juiste tijdsvak vallen, maar een aantal zullen er naast vallen.

In de volgende afbeelding is een plot te zien van hoe de data van een vast tijdsverschil er uit ziet:



Afbeelding: de jitter in de meting

Deze jitter is niet op te lossen in De FPGA. De meetresolutie blijft echter nog onder 1ns, wat nauwkeurig genoeg is voor de toepassing. Deze meting is zonder PMT's gedaan.

8. Conclusie

Deze stage heeft me veel nieuwe aspecten van embedded systemen geleerd. Zaken als VHDL worden niet veel behandeld in de Ti opleiding. Terwijl dit een belangrijk onderdeel is van het ontwerpen en ontwikkelen van embedded systemen.

Een ander aspect van deze stage is dat ik de opdracht alleen heb moeten volbrengen. Ik had natuurlijk wel de mogelijkheid om gebruik te maken van de kennis van mijn collega's, maar de planning en keuzes waren uiteindelijk mijn verantwoordelijkheid.

Iets wat ik in het bijzonder heb geleerd is het documenteren terwijl je bezig bent. Ik heb dit in het begin van de opdracht gedaan, echter na verloop van tijd is dit toch minder geworden. Het voordeel van de documentatie tijdens de opdracht maken is dat het minder werk, en vooral makkelijker is om uiteindelijk het verslag te maken.

Het voordeel van de Muonlab opdracht is dat op het eind een volledig product afgeleverd wordt. Hierdoor blijft de motivatie hoog om een goed product af te leveren. Daarom heb ik ook veel aandacht besteed aan het testen van Muonlab. Ik heb een aantal scintillatoren kunnen bemachtigen waarmee ik kon testen zoals Muonlab ook gebruikt zal worden. (dat is het voordeel om bij een instituut als het NIKHEF te werken) Hierdoor kwamen er zelfs nog een aantal bugs uit. Bijvoorbeeld de meetfout in de LifeTime meting die je met alleen met de pulsgenerator niet had gevonden.

Ik heb bij het NIKHEF mezelf uitstekend vermaakt. Iedereen was bijzonder aardig en hulpvaardig. Aangezien het NIKHEF geen commercieel bedrijf is, is hier waarschijnlijk ook meer ruimte voor. Voor een stage in Embedded systemen kan ik dit stagebedrijf zeker aanbevelen.

9. Gebruikte software

De volgende software pakketten heb ik in dit project gebruikt:

- Labwindows/CVI v7.1
- Ease 5.2R9
- Xilinx project navigator 7.1.04i
- ModelSim SE PLUS 5.8e
- Dev C++ 4.9.9.2
- The Gimp 2.2.9
- Root 5.08

10. Reverenties

NIKHEF Homepage

<http://www.NIKHEF.nl>

Muonlab II homepage

<http://www.NIKHEF.nl/~basvdh/muonlab/>

The European Particle Physics Outreach Group:

<http://eppog.web.cern.ch/eppog/ExperimentsForSchools.html>

Informatie over airshowers

<http://www.th.physik.uni-frankfurt.de/~drescher/CASSIM/>

De producent van de FPGA

<http://www.xilinx.com>

National Instruments, de producent van LabWindows

<http://www.ni.com/>