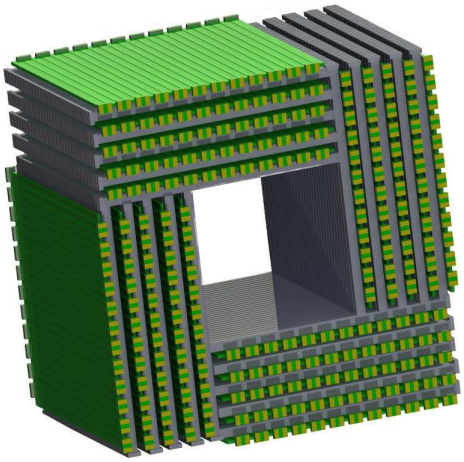
Centralized Coincidence Trigger Processing for COMPET Using Both a Synchronous and an Asynchronous Approach

Kim-Eigard Hines
kimei@fys.uio.no
Department of Physics, University of Oslo



Thesis presented for the Master of Science degree
in Instrumentation

January 2012

Abstract

A coincidence processing unit in PET detectors separates single events (noise) from coincident events (positron events) to enhance the SNR of the final image and to lower the data rate in the transfer channel. This thesis describes the work carried out to design and implement a centralized coincidence processing unit for COMPET, a pre-clinical PET scanner with a novel geometry and read-out chain. In order to minimize the coincidence time resolution, a clock distribution scheme with phase correction has been implemented. For coincidence processing both a synchronous and asynchronous approach have been designed, implemented and tested, using point-to-point cabling and UDP/IP over 1 Gbps Ethernet respectively. Both implementations were done using a FPGA situated on a Xilinx development board. Results show that the synchronous solution handles coincidence rates of 1/4 of system clock frequency (100 MHz), with a coincidence time resolution of twice the system clock time period. The asynchronous solution handles coincidence rates in the order of 5 Mcps with theoretically no upper limit on coincidence time resolution.

Acknowledgements

I though this was going to be the easy part, but where do I start? Well, thanks to my family, friends and Idunn for lending out my mental presence and enduring the few weeks of frustration. I owe my deepest gratitude to the COMPET project with Erlend Bolles steady guidance, and for letting me go to the Valencia IEEE conference to present our mind bending results. Another special thanks goes to Michael Rissi with his knowledge, patience and ability to share an office with someone who instantly wants to know why beta decays happens. To David Volgyes for his endless knowledge and experience in the fields of PET and informatics. Thanks to Torfinn Lindem for turning my work into a thesis. I'd also like to thank EPF as a whole for supplying us with what we need to be productive. Thanks to Ole Røhne for taking his time for some thorough knowledge transfer. Thanks to ELAB for helping out when needed. And Moccamaster, Nespresso and Kaldi have my gratitude. Without you, the deadlines would have never been met. Lastly, I would like to once again thank my supervisors for being that and doing it well, Torfinn Lindem, Erlend Bolle and Michael Rissi.

Contents

Chapter 1

Introduction

This thesis describes the work carried out to build and implement a centralized coincidence trigger system for a pre-clinical PET (Positron Emission Tomography) scanner using two different approaches, namely an asynchronous and a synchronous approach. A coincidence trigger is a unit which triggers data acquisition units such that events arising from positron-electron annihilation are captured while suppressing other types of events. Conventional PET detectors usually have synchronous coincidence trigger processing. An asynchronous approach based on standard network components can prove to be more flexible for a detector which is based on modules (as COMPET), such that adding more read-out cards for more channels can be done by simply connecting them to a standard network switch. To better understand the limiting factor for a PET detector the introduction starts with a general description of the basic building blocks of a typical PET scanner. The PET chapter explains how different processes are captured in the scanner and how they affect the detector performance. The COMPET chapter describes the specific system in which the centralized coincidence trigger is to be implemented in.

1.1 Gamma-ray Detection

1.1.1 Scintillators

A scintillator is a material which converts ionizing radiation to electromagnetic radiation around the visible spectrum. It absorbs the energy of the incoming particle and re-emits it as light in the visible to UV spectrum. For the scintillating material to be efficient it must have some desirable properties.

- Efficient conversion from exciting radiation to fluorescent radiation
- Transparency for its own fluorescent radiation
- Emission in a spectral range suited for photo detection
- Fast decay constant

The first point means that as much as possible of the incoming energy is converted to fluorescent radiation (light). Gamma-rays in the 100 keV range interact with matter in mainly three different processes: pair-production, Compton scattering and the photo electric effect. For PET detectors pair production is not an issue since the energy threshold needed for the process is naturally twice as large as the gammas energy after positron-electron annihilation. Compton scattering occurs when a gamma has an inelastic collision with an electron and changes direction. Depending on the energy transfered, the electron will be free (which means that the atom is ionized) or loosely bound. The cross section (which is equivalent to how likely the process happens) is proportional to proton number Z, which means that it is linearly dependent on the density of the scintillating material. Photo electric effect is similar to Compton scattering, but instead of being scattered, all the energy of the incident photon is absorbed. The cross section for photo electric effect is proportional to Z^5. The photoelectric effect is the most desired since all of the gammas energy is deposited in a single crystal.[16]

The second item states that if the scintillating material is to be efficient it must be transparent for its own radiation. This simply means that if the scintillating crystal emits light in the blue area of the visible spectra, blue light must be able to pass through the crystal so it can be detected on the outside.[16]

The third item states that the wavelength emitted from the scintillator should be in the range of the efficiency peak for the photo detector.[16]

The decay constant consist of a fast process and in some scintillators there is an additional slow process. The fast process comes from the fluorescence in the crystal, and is the process that gives a light-output from the incoming ionizing radiation. This occurs around 10^{-10} to 10^{-8} s after absorption, which is the time taken for the atomic processes to take place. The slow process comes in when the excited state is meta-stable, and this is usually referred to as afterglow. This can be from some micro seconds to hours. The resulting light output has therefore a fast rise time (in order of tens of nano seconds) with an exponential decay, see figure 1.1.[16]

Scintillating crystals are usually divided into two different categories, organic and inorganic. The organic scintillators are made by putting scintillating molecules into a plastic mold (or liquids), while inorganic scintillators are crystal structures.

Organic scintillators scintillation process is a molecular property from the fluorescent material used (fluors). The fluorescent material absorbs the gamma-rays energy by molecular excitation, and re-emits light at longer wavelengths when the de-excitation occurs. The main drawback of using organic plastic scintillators is the materials low Z. [28]

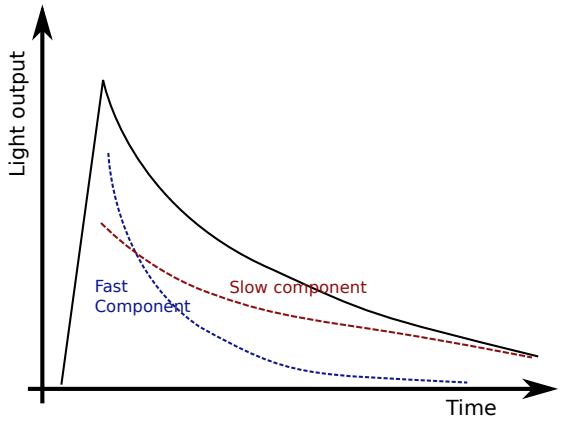


Figure 1.1: Exponential decay from a slow and fast component in a scintillating crystal with two time constants

Inorganic Crystals scintillates when an incoming particle excites an electron from the valence band to the conduction band or the exciton band[1]. Excitation to the exciton band creates a loosely tied electron-hole pair which can wander freely through the crystal lattice. The migrating hole can then ionize impurity atoms in the crystal. If then an electron arrives, it can make a transition in the impurity atom from an excited state to a ground state resulting in radiation. The impurity atoms are chosen such that the radiation is in wavelengths near UV, so that photomultipliers and other photodetectors can be used efficiently. Inorganic crystals are usually slower than organic, but because of their high stopping power (high Z) and high light output they are a popular choice in PET systems.

Table 1.1: Comparison of different crystals used for PET. sources: [19] and manufacturer web page www.omegapiezo.com for LYSO

	NaI(Ti)	BGO	LSO	LYSO	$LaBr_3$(Ce)
Density (g/cm^3)	3.67	7.13	7.40	7.40	5.1
Z-effective	51	75	65	66	47
Primary decay constant(ns)	230	300	40	42	16
secondary decay constant(ns)	10.000	-	-	-	-
emission wavelength(nm)	410	480	420	375	358
Index of refraction	1.85	2.15	1.82	1.82	1.88
(10^3)Photoelectrons/MeV	41	9	30	30	60

From table 1.1 the trade-offs between stopping power, speed and light output is apparent. Crystals containing Lutetium has an intrinsic background from the beta decay of ^{176}Lu. This will make the output noisier, but the intrinsic spectra can be useful for energy calibration of a system. The scintillator used in a PET detector sets a limit to energy resolution, timing performance and count rate performance. The scintillators energy resolution comes from

[1] located just below the conduction band and separated from the valence band by an energy gap

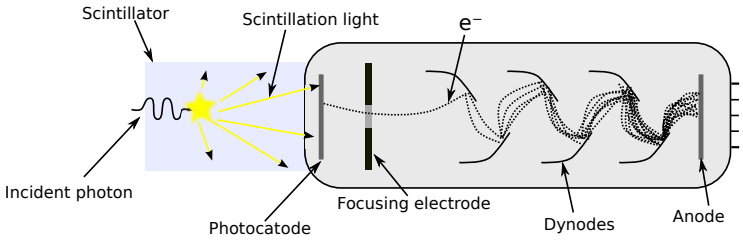


Figure 1.2: Schematic view of a photomultiplier tube

Poisson statistics from the number of photoelectrons emitted. A high light output will therefore generally give a better energy resolution. The timing performance is limited by the Poisson statistics which gives the rise time for the scintillation signal. The count rate performance could be limited by pulse pile-up in the scintillator, meaning that the previous event has not decayed before the next gamma arrives.

1.1.2 Photo Detection

The choice of photo detection method will affect the system timing resolution in the 0.1 ns range.

Photomultiplier tubes

Photomultiplier tubes (PMT) consists of a vacuum tube with a photocatode, a focusing electrode, several dynodes and an anode. The photocatode excites electrons from the photo-electric effect from an incident photon, where the incident photons usually have wavelengths from the visible spectrum to near UV. The focusing electrode directs the electrons excited from the photocatode to the first dynode which is on a higher potential than the photo electrode. Because of secondary emission from the first dynode, more electrons drifts to the second dynode which is on an higher potential than the first dynode. This repeats itself down to the anode where the electrons are captured and a charge pulse is output, see figure 1.2. The dynodes are set-up with potentials like a ladder, so the electrons will always drift to the next dynode. PMTs are known to be very (order of GHz) fast with a high gain. The current gain is mainly dependent on supply voltage, number of dynode stages, the quantum efficiency of the photo electrode and can be as high as $A_i = 10^6$. The spread of electron transit time can be under 500 ps. Others have reported a FWHM jitter of 300 ps and a rise time of under 1 ns. Drawbacks of using PMTs is that they require a lot of space, they use supply voltages of 500-2000 volts, they are sensitive to strong magnetic fields (more on that in PET/MRI section) and are noisy due to the multiplication process at the first dynode.[10][15]

Silicone Photomultipliers

A silicone photomultiplier consists of a high number $(100 - 10^5)$ of avalanche photo diodes in parallel. A photo diode is a device that use a drifting electron-hole pairs to induce a current from incident photons. When photon with enough

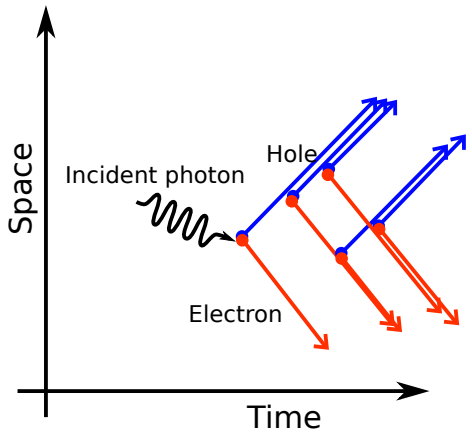


Figure 1.3: Avalanche principle from an incident photon

energy hits the active area of a photo diode it can create an electron-hole pair, since the p-n junction is reverse biased a drift field is formed where the electron and hole will drift to the anode and cathode respectively. The drift field is in the order of MV/m, which is enough to accelerate the electron and hole such that new pairs are formed. The repetition of electron-hole creation is called an avalanche. [8] SiPM pixels are photo diodes operated in geiger-mode (self-quenching). Geiger mode means that the photo diode is operated above its break down voltage such that every incoming particle with enough energy creates a self sustained avalanche. The rising edge of the current induced by this avalanche marks the arrival time of the incident particle. But operating a photo diode in geiger mode means that energy information about the incident particle is lost, therefore a SiPM is a matrix of photo diodes read out as a sum. Energy information is then found by reading out the pixels as a sum such that summed pulse height correlates to the number of geiger mode photo diodes who fired, which is correlated to the number of incoming photons which, when coupled to a scintillator, correlates to the incoming gammas energy deposition in the scintillator. Modern SiPMs report gains from $A_i = 10^5$ to 10^6 which is comparable to PMTs, with a photon detection efficiency peak that can be engineered (usually around blue to near-UV). When comparing SiPMs to PMTs the SiPMs are smaller but with lower fill factor. They use lower bias voltage (around 30-75 V) and gain parameters are not influenced by strong magnetic fields. Time resolution down to 35 ps FWHM and rise time of around 1 ns has been reported. The PMTs will have a lower temperature dependency with respect to gain and usually a lower dark count [2].[22] [9][15]

[2]Dark-count is the noise measured when there are no incident photons

1.2 Medical Imaging

1.2.1 Modalities

The following section is a brief overview of the most common modalities including X-ray Computed Tomography, MRI, SPECT and PET. PET will also be treated in depth in its own chapter. The introduction to the different modalities is here to point out the void in medical instrumentation that PET fills.

Radiography

Wilhelm Rontgen won the Nobel Prize in physics in 1901 for the detection and production of electromagnetic radiation with wavelengths around 1 nm, called X-rays. The discovery of X-rays led to the first radiograph conducted surgery in 1896 by pioneer John Hall-Edwards. A X-ray tube is a vacuum tube which accelerates electrons released from a hot cathode by the use of an high electric field. The electron then hits a (dense) target where x-ray photons are emitted by two different processes. Bremsstrahlung, which has a continuous specter and X-ray fluorescence with a discreet spectrum. The x-ray beam is then hardened by a thin metal foil to remove the lower energies which will not penetrate the subject. On the far side of the subject, a radiographic film is placed under a collimater to absorb the X-rays. Different tissue have a different attenuation, which is why the image formed on the radiographic film is a map of tissue density.[13]

X-ray Computed Tomography

X-ray Computed Tomography (CT) uses X-rays to take multiple planar images to make an image of a volume. The word tomography comes from the Latin words tomos which translates to part and graphein which means to write. Modern CT machines use a rotary gantry which spirals over a volume. High-end CT machines (for example Siemens Somatom) uses two gantries at the same time with different energies for faster acquisition time and lower radiation doses. X-ray CT is used to see different electron densities in different tissues, which makes it a very good tool to see the anatomy of the subject. From 1971 to 2006, the scan speed has gone down from 270 seconds to 0.3 seconds, the z-resolution has gone from 10 mm to 0.5 mm and what is known as the breath hold coverage (which is how long you can scan in z-direction in 30 seconds) has gone from 1 cm to 470 cm.[3]

[3] the numbers are taken from Jiang Hsieh short-course Principles and Advancements in X-ray Computed Tomography from the NSS-MIC conferanse in Orlando 2009

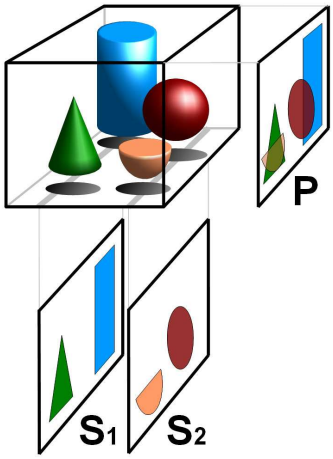


Figure 1.4: Illustration showing the basic principle of tomography

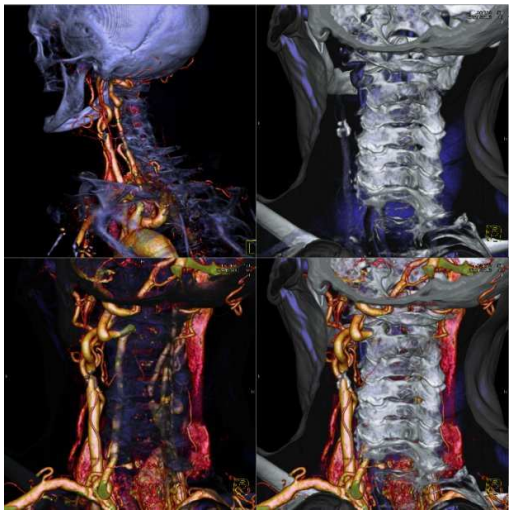


Figure 1.5: Example of image taken by the Siemens Somatom

MRI

Magnetic Resonance Imaging is done by having a strong static magnetic field (1-10 Tesla) through the subject. With this strong field it is possible to use a property called Nuclear Magnetic Resonance (NMR) where a nuclei with a magnetic axis (angular momentum) can absorb and re-emit electromagnetic radiation. ^1H which is 99.98% of the natural abundance of water in our body has a magnetic axis as such. This axis is orientated randomly unless there is a strong B field that aligns them. The frequency of which the ^1H atoms can absorb and emit electromagnetic radiation is called the Larmor frequency and is defined as $\omega = \gamma B_0$ where γ is the gyro magnet ratio and is $42.58MHz/Tesla$

for ^{+}H atoms and B_0 is the magnetic field strength. The position of the hydrogen atoms can be found by having gradient magnetic fields which alters the Larmor frequency in a region and applying a RF field with a frequency corresponding to the Larmor frequency in that region. The RF pulse will tip the magnetic axis of the atoms, and while the atoms restore their axis an RF pulse of the same frequency is emitted. Because of the physics behind MRI, the modality is best suited for soft tissue imaging. It can achieve in plane resolutions down to 0.5 mm with isotropic voxel sizes of 0.7 mm. [3] [23]

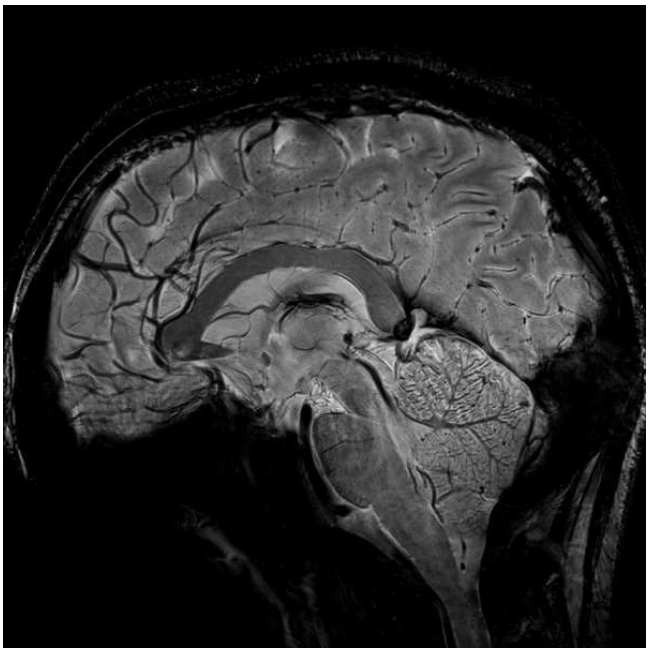


Figure 1.6: Example of 7 tesla MRI scan, courtesy of Gachon University, Seoul, Korea

SPECT

Single Photon Emission Computed Tomography is an evolvement of the gamma-camera. A gamma camera in its simplest form is a block of scintillating material with a collimater attached to an array of PMTs or SiPMs. The working principle is that a gamma emitting radionuclide is injected in the body. The radionuclide is tagged to a special radioligand which is engineered to be absorbed by the tissue of interest. A gamma camera is then used to detect the emitted photons, and by rotating the gamma camera to different angles it is possible to reconstruct a volume. Because of the colliminators the photon sensitivity is naturally lower than what you will find in a PET-scanner, but cleverly positioned pinhole collimators can give an amplification (or zoom) effect which can increase resolution with cost of detection efficiency.[7] [23]

PET

In Positron Emission Tomography a positron emitter is tagged to a radioligand and injected in the body (as in SPECT). The tracer concentration is then indirectly measured by detecting the resulting gamma photons (511 keV) from the positron annihilating when it interacts with an electron. If the electron and positron is at rest, the two resulting photons at 511 keV will be emitted 180 degrees in respect to each other.

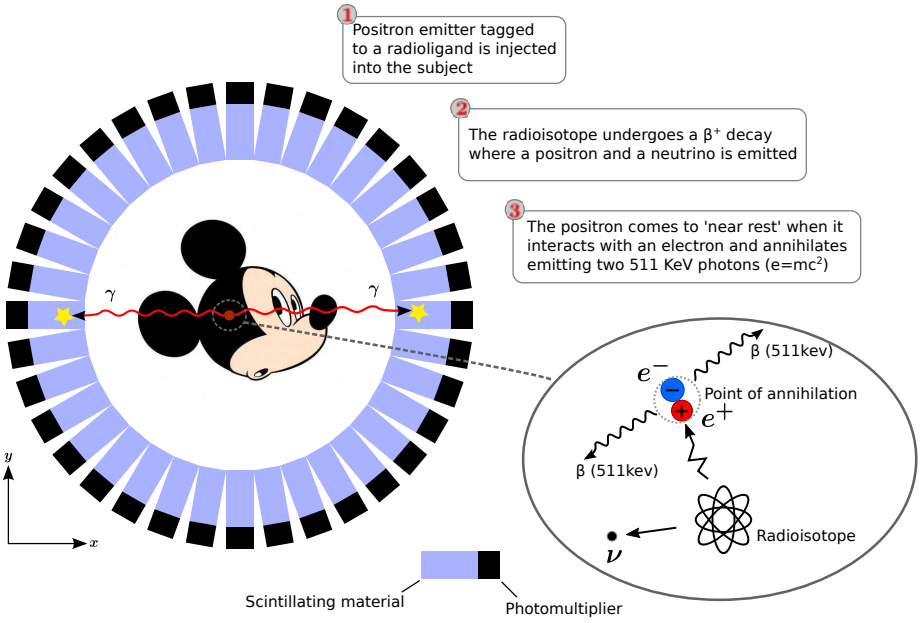


Figure 1.7: Simplified PET-ring showing a positron event, the figure shows one slice of crystals. A PET detector usually consists of several slices stacked together to form a barrel

When an event happens within a time period called 'the coincidence time window' in two different crystals, it is called a coincidence. This means that the events probably came from a positron being emitted by the radionuclide. A line of response (LOR) can then be drawn between the two co-incidental crystals. When enough LORs are captured an image can be reconstructed. PET and SPECT differ from other modalities because they do not image the structure of the body directly. Instead they image a concentration of a radio tracer. The radio tracer is engineered to probe the processes of interest. Drawbacks of using PET is short lived radionucleis, with half-life of around 120 minutes for ^{18}F down to 2 minutes for ^{15}O where the production usually requires cyclotrons. The subject must be kept still during the incubation of the radioligand, which requires some waiting time. [27]

1.2.2 Mixed modalities with PET

This is a brief summary of the advantages and issues related to PET/MRI and PET/CT.

PET/CT

Merging a CT image with a PET image gives the viewer the possibility to see the structural treats together with the image of physical processes from the PET. And today PET/CT dual modality scanners are used more than dedicated PET and CT scanners clinically. An advantage is that an attenuation map can be created from the CT scan and used for attenuation correction in the image reconstruction for PET[4]. Attenuation maps from a CT scan can decrease PET acquisition time by 30-40%[5]. Repository motion can cause artifacts in the fused image. This is due to CT scans being acquired in such a short time that the subject can hold the breath, while the subject must breath normally during a PET scan. [30]

PET/MRI

Fused PET/MRI images will offer the same main advantage of PET/CT, namely seeing anatomical structure together with physiological processes. Because MRI have a better soft tissue resolution there will be areas where PET/MRI is superior to PET/CT. Disadvantages of PET/MRI is that the MRI does not produce data that is useful in the construction of an attenuation map since the MRI intensities correlate with proton density instead of mass density. However, the creation of attenuation maps based on MRI images is an active field in research. One solution would be to use emission from a collimated point source, but lack of space inside the scanner makes this difficult. Another issue is to create a PET system which is insensitive to the strong magnetic fields from the MRI scanner.[5]

[4] Attenuation correction is necessary to remove artifacts in the image due to the fact that different tissue has different attenuation for the detected 511 keV photons

[5] Compared to using a collimated point source rotated around the subject to get the attenuation coefficients

Chapter 2

PET

2.1 PET Tracers

A pre-requisite for a radionuclide used in PET is that is a positron emitter. It is to the patients benefit that the half-life is relatively short such that acquisition time can be held short while keeping the absorbed radioactive dose low. During the scan the positrons emitted from radionuclide travel a finite length before annihilation occurs, depending on the energy of the emitted positron. The initial energy can take values up to a maximum defined by nuclear transmutation in the isotope. When the positron energy is less than the maximum energy, the excess energy is given to a neutrino. In table 2.1 some different isotopes used in PET are listed with their respective half-life and mean positron range in water. [27]

A radio tracer is a substance which is engineered to probe a process of interest. By tagging the radio tracer to a radionuclide it is possible to image different processes. Fludeoxyglucose (FDG) is an analogue of glucose which is tagged to Fluorine-18. Since FGD is analogue to glucose, it will be taken up by the cells with a high uptake of glucose and other areas with high metabolism. This makes FDG suitable to analyze glucose uptake. Oxygen-15 can be with Hydrogen to form water. The H_2O^{15} can be used for cerebral blood flow analysis

Table 2.1: List of some isotopes used in PET

Nuclei	Half-life	Range in water [mm] (mean)
Fluorine-18	$\approx$ 110 min	0.6
Carbon-11	$\approx$ 20 min	1.1
Nitrogen-13	$\approx$ 10 min	1.5
Oxygen-15	$\approx$ 2 min	2.5
Gallium-68	$\approx$ 68 min	2.9
Rubidium-82	$\approx$ 1.25 min	5.9
Technetium-94	$\approx$ 52 min	-
Iodine-124	$\approx$ 6000 min	-

2.2 Detected Event Types

Events are the output of the photo-detectors and can be identified and categorized from how they hit the detector. In PET systems there are usually four different scenarios which leads to different event types.

True coincidences are an event which arose from a positron-electron annihilation where two un-scattered 511 keV gammas are detected. The correct Line Of Response (LOR) can be drawn between the two incident channels. A true event which has not yet reached the detector material is called a "prompt event". [27]

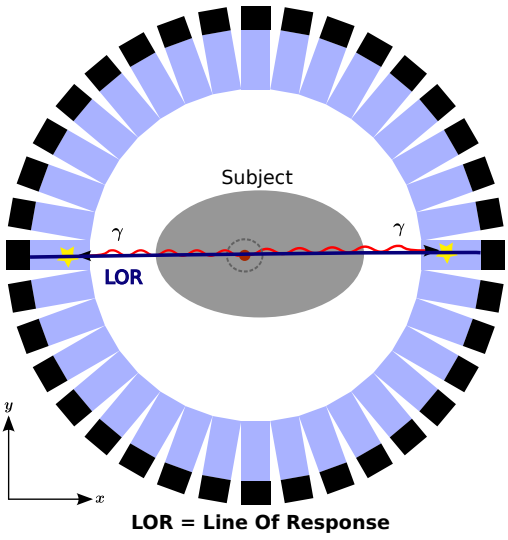


Figure 2.1: Correct Line Of Response drawn from a true coincidence

Random coincidences occurs when there are two positron-electron annihilations within the same time window which will create two different gamma pairs. If two of the gammas do not interact in the detector, the result could be an incorrect LOR between two single events from different annihilations.

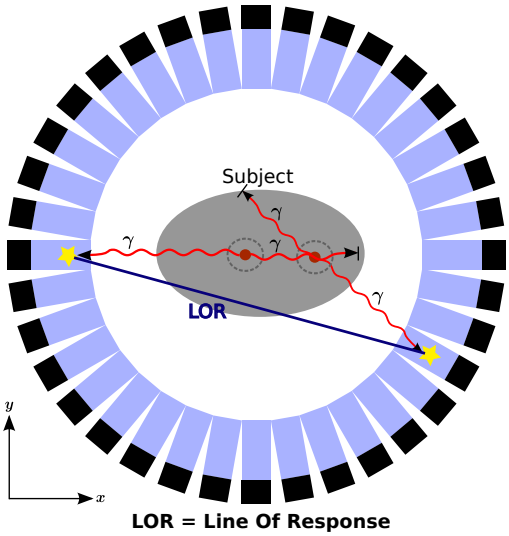


Figure 2.2: Incorrect Line Of Response drawn due to a random coincidence

Scattered coincidences arise from a single positron-electron annihilation, but one (or both) gammas undergo a Compton interaction in the subject (or between the subject and the detector). This will give an incorrect LOR because the gamma pair is no longer back-to-back. Another form of scattered events are inter-crystal scatters, which occurs when a photon has a Compton event in one crystal and a photoelectric event in one of the neighboring crystals. Scattered events can be suppressed by using an energy window close to 511 keV

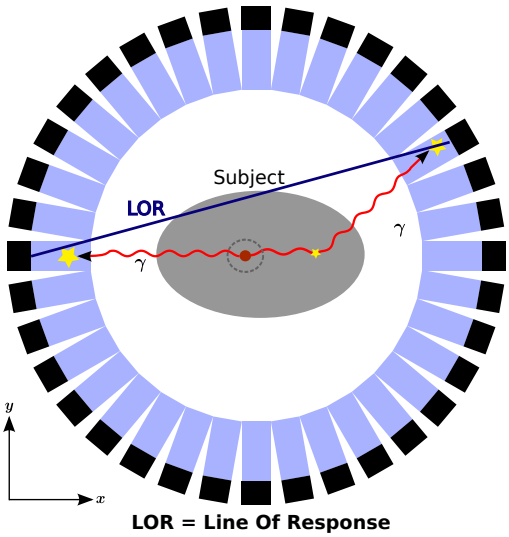


Figure 2.3: Incorrect Line Of Response drawn due to a Compton interaction

Multiple events occurs when there are multiple positron-electron annihilation where all, or more than two resulting gamma photons get detected. This will not create a line of response because it is impossible to separate the different coincidences.

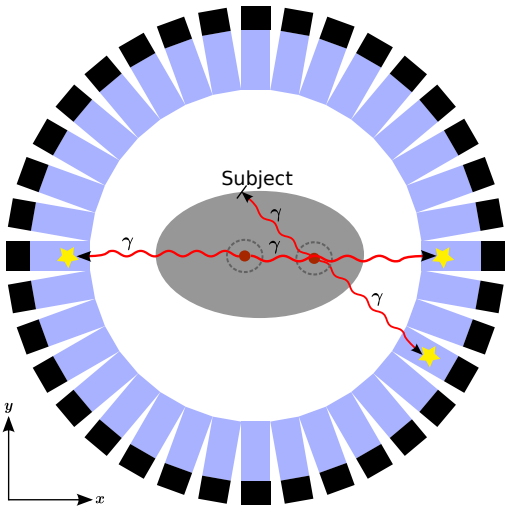


Figure 2.4: Multiple events give no LOR because they can not be separated

2.3 PET Performance Indicators

This is a summary of what constitutes performance in a PET system, for all of these performance indicators there is a defined standard for how to measure them given by National Electrical Manufacturers Association (NEMA)[1].

2.3.1 Resolution

Energy

Energy resolution is how accurate a system can measure the incident gammas energy. The energy resolution is usually given as a Full Width Half Maximum (FWHM) of the peak centered around 511 keV. For easier comparison of different systems, this value is again represented as a percentage of the center energy.

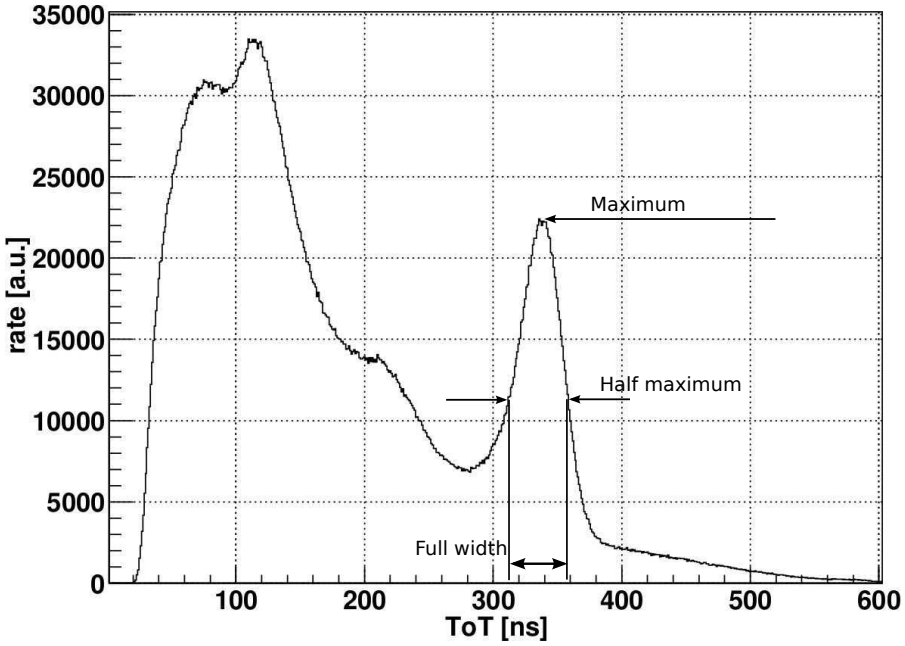


Figure 2.5: Energy spectra from a data set captured by COMPET

High energy resolution is important for two different reasons. The first is to be able to make the energy cut as close as possible to the 511 keV for scattered events suppression. The second is to be able to reconstruct inter crystal scattered events where the energy deposition is divided between crystals. A prompt event can then be found by summing up energy in neighboring crystals and see if it is near 511 keV.

Timing

Timing resolution dictates how precise a system can say when an event happened. For a Time-Over-Threshold system, where deposited energy in the crystal is proportional to the length of a pulse, the time resolution is also a factor in energy resolution. For systems with pulse height read out from an analog to digital converter, time resolution is most important for coincidence triggering (e.g. separating two single events from a positron event). The system time resolution put a limit on how narrow the coincidence window can be, where the coincidence window is the allowed arrival time difference between two events to be called a coincidence. The narrower the coincidence window, the better the random suppression. Sources for time uncertainty include crystal rise time, time-walk and clock jitter. The time-walk is an error induced from signals with a finite rise-time read out at a threshold. If the rise time is the same for two signals with different amplitude they will reach the threshold at different times even if the start time was the same, see figure 2.6. [28]

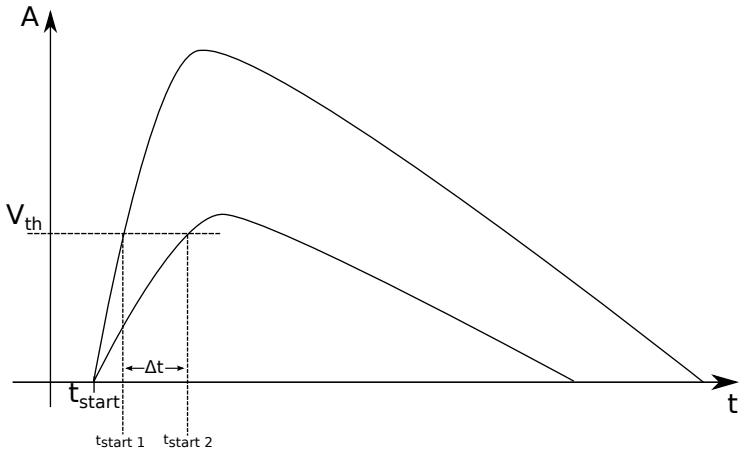


Figure 2.6: Time-walk from two signals with same start position and different amplitude reach the threshold at different times

Spatial

Spatial resolution defines how close two small points can be side-by-side and still be distinguished from each other. One method to find the spatial resolution is to have a small object (a point source) which is at least half the size as the expected resolution. The blurred image of the point source will give the point spread function. The resolution is given as a FWHM and FWTM (Full Width Tenth of Maximum) of the profile. Resolution in PET rings is not the same in the radial and trans axial field of view and must be measured separately. The NEMA standard for spatial resolution measurements states which reconstruction method to be used, such that the measurement is solely dependent on hardware and not clever reconstruction methods. Separate measures must be done for the center of the detector and off center, since the resolution is a function of position.[28]

Figure 2.7: Two point sources side by side with added blur until the objects can no longer be separated

In general the spatial resolution is given by a convolution of the three blurring factors:

1. Positron range
2. Annihilating photon non-collinearity
3. Intrinsic detector resolution

Where the positron range is dependent on the radionuclide used (see table 2.1) and the tissue it traverses. The blurring caused by annihilating photon non-collinearity is dependent on the detector diameter while intrinsic detector resolution is mostly dependent on crystal size and geometry.[17]

2.3.2 Sensitivity

Photon detection sensitivity gives the percentage of detected photons versus emitted photons $\frac{\text{Counts/s}}{\text{Bq}}$. The sensitivity is a result of the geometry of the detector, gap between crystals, depth of the crystals and the crystals stopping power.

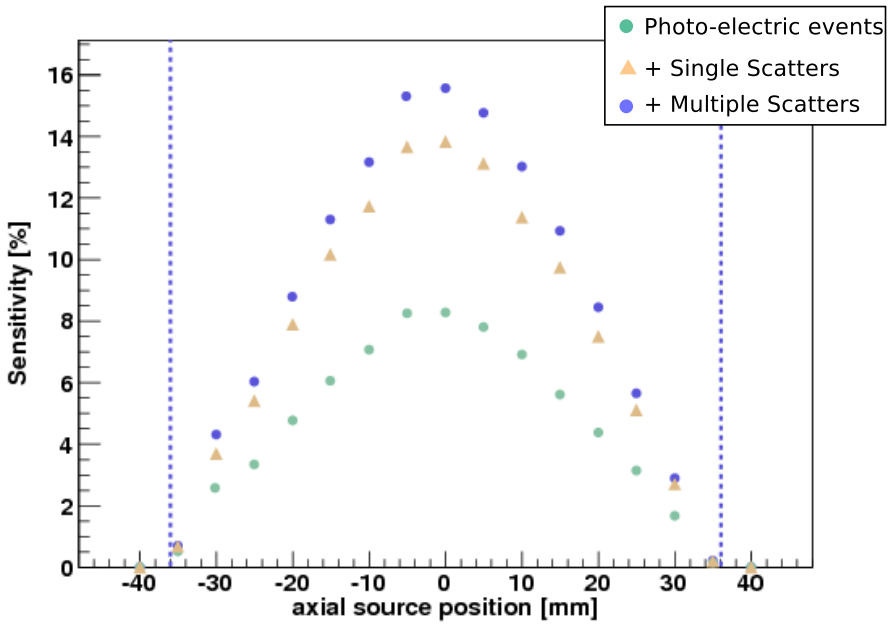


Figure 2.8: Sensitivity plot from COMPET showing the scanners sensitivity as a function of axial position, the different graphs show the sensitivity if single and multiple scatters are included

2.3.3 Count Rate performance

The count rate performance is an indication on how well the detector performs with different event rates. There are mainly three different factors which influence the count rate performance. The detector dead time is the minimum allowed time between two events, this time is usually different for the read-out system as a whole and per channel. If there is an event in the dead-time, the event is lost which will lower the sensitivity. Detector throughput is the maximum event rate the system can handle. This rate is limited either by event handling throughput in the read-out electronics or in the communication channels for data storage. Trigger window width which is the allowed

time difference between two incident photons to be called a true event has an effect on the count rate performance. The wider the trigger window is, the higher is the possibility of having multiple and random events. To visualize count rate performance a NEC curve (Noise Equivalent Counts) is made. This is usually done by measuring the random, multiple and true event rate while a high activity source decays. The NEC curve is calculated as

$$NEC = \frac{T_0^2}{T_0 + S_0 + kfR_0}$$

where T_0 is the count rate of true events, S_0 is the singles count rate, R_0 is the randoms count rate, k is a constant related to the method used for random event estimation and f is the fraction of the entire field-of-view used for randoms estimation.[27]

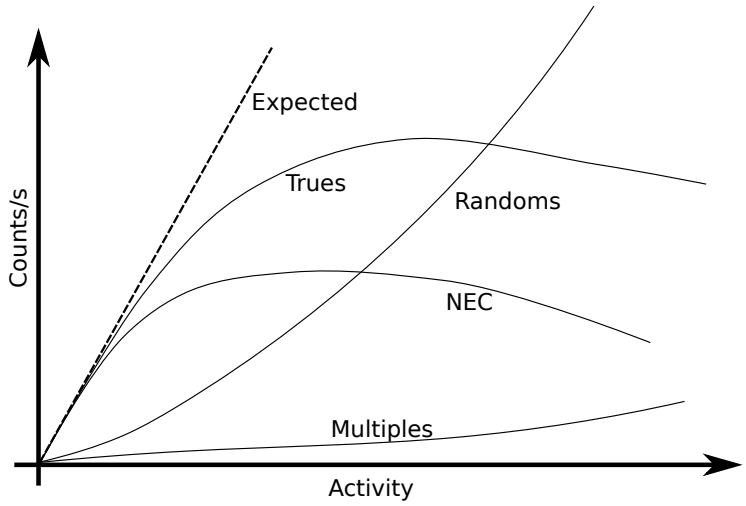


Figure 2.9: NEC curve showing the relation between expected, true,random and multiple event rates with the NEC curve

2.3.4 Parallax Error

Parallax error is the systematic error due to detector geometry and crystal length and occurs when the origin of a coincidence is not in the center of the detector. An incident gamma will in a straight line traverse multiple detector elements before all its energy is deposited. The closer to the detector edge the prompt was, the more elements will be traversed, see figure 2.10. [27]

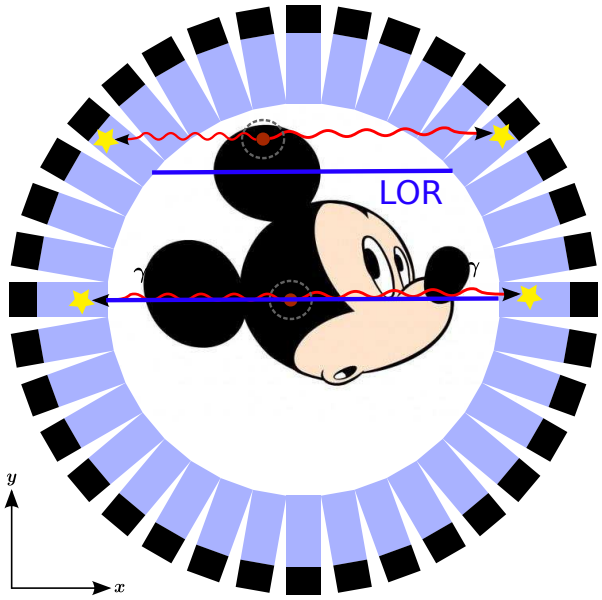


Figure 2.10: Illustration of parallax error where the line of response from the event in Mickey's ear is incorrect due to parallax error and the event in center is correct

2.4 Detector Design Considerations

A scanner should ultimately have a high spatial resolution for distinguishing small concentrations of the radio ligand. The resolution is a function of (among others, but most importantly) crystal size and detector radius. The closer the detector elements are to the subject the better the sensitivity. A large detector with small crystals is expensive (electronics and material cost). To keep the sensitivity high, the crystals needs to be long (for a detector with ring geometry) which again gives a larger parallax error. A scheme to use smaller detectors with long crystals without loosing resolution due to parallax error is to extract Depth-of-Interaction (DoI) from the crystal. This can be done by reading out the crystal in both ends and use light-fraction and/or timing information to establish where in the crystal the event happened.

By having event time resolution in the sub nano second range the possibility of Time-of-Flight (ToF) PET opens. Instead of drawing a line of response, it is possible to have a likelihood curve of where the positron-electron annihilation took place. The position is then calculated with respect to the mid-point in the line of response with $\Delta d = \frac{\Delta t * c}{2}$. A 5 cm resolution would then require a timing resolution of 330 ps.[28]

2.5 Annihilation Coincidence Triggering

An important part of the PET data acquisition is the coincidence triggering. The task is to separate the single events (noise) from the true events. A trigger will reduce the strain on the electronic read-out by lowering the noise rate which it has to process. This is usually done in real-time with synchronous logic. The term coincidence window is the allowed time difference between two separate events to be considered a coincidence. The length of the coincidence window is usually limited to the timing resolution of the entire system, but must in any case be large enough such that positron events with a difference in arrival time due to position will be included. For an event happening 15 cm off center the difference in arrival time will be $T_{diff} = \frac{2*L}{c} = \frac{2*15cm}{3e8m/s} = 1$ ns. If timing uncertainties from the crystal to the electronic read-out is included it is clear that having a too narrow coincidence timing window will reject true events. The length of the coincidence window will be the deciding factor of how many random events will be recorded. The random rate between two channels will be a function of the singles rate in each channel as $R_{random_{i-j}} = 2 * \tau * R_i * R_j$, where τ is the trigger window width, and R is the rate in a single channel.

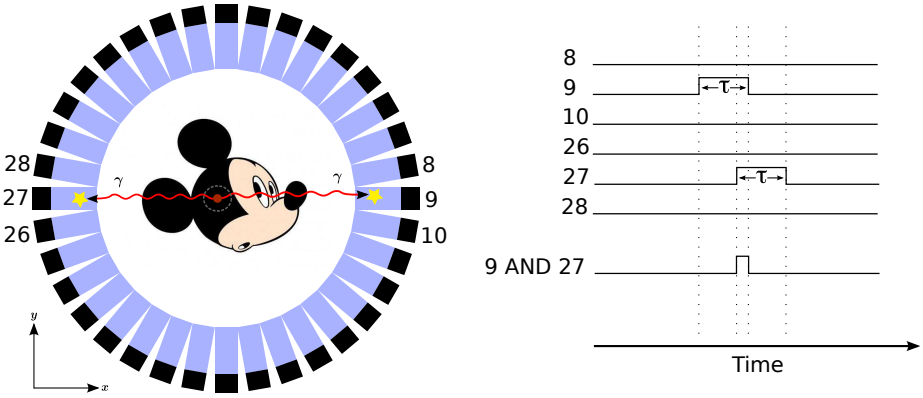


Figure 2.11: Showing two events within a coincidence window

Annihilation coincidence triggering can have geometrical constraints. In figure 2.11 a coincidence between two opposing crystals is shown, but a line of response could in principle be drawn between neighboring crystals. As an effort to reduce the parallax error the coincidence trigger logic can constrain the trans-axial FoV by having a minimum distance between crystals in coincidence effectively rejecting events a certain distance from the center of the detector, see figure 2.12

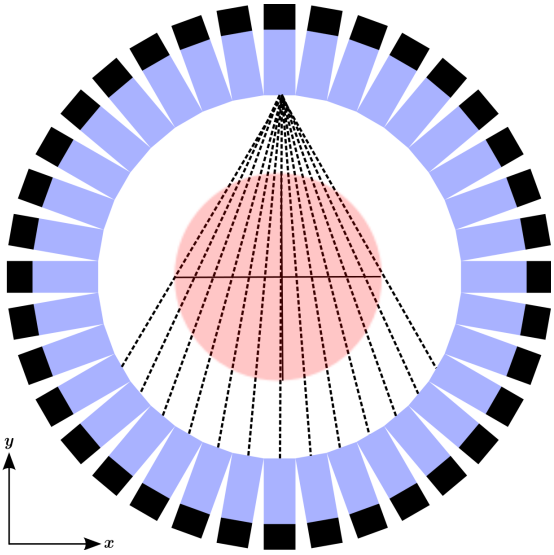


Figure 2.12: Trans axial field of view is determined by which detector elements are allowed to be in coincidence

Figure 2.11 shows one single slice of crystals, while PET scanners usually have several slices together forming a cylinder. Fully 3D acquisition means that each slice can be in coincidence with any other slice. This will increase the scanners sensitivity since the gamma-pair can have an angle to the x-y plane in the scanner. see figure 2.13

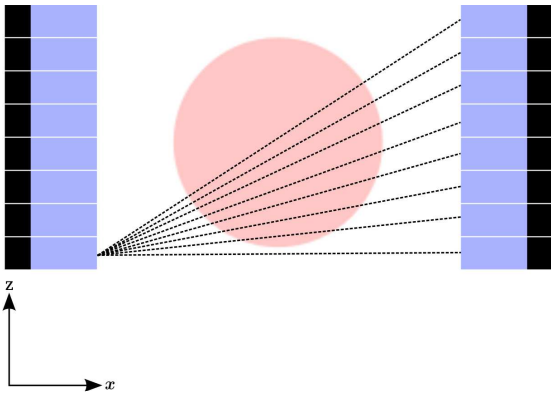


Figure 2.13: Fully 3D coincidence triggering in the axial field of view

Random Event correction

The random events add a relatively uniform background noise to the reconstructed image. The distortion will come in form of suppressed contrast and wrong relation between the image intensity and actual activity at that location. One way of correcting for random events is by a method called Delayed Window. Since the random events are continuously distributed in time the number

of randoms within a trigger window set up by a prompt event is the same as the number of randoms inside trigger window set up arbitrarily. The delayed window method will set up a trigger window at the correct time for the prompt event (making it a true event) and additionally setting up a trigger window a set delay later(usually 10-20 times later than the length of the coincidence window itself). By recording the events inside that window it is possible to do a baseline subtraction. This is possible due to the fact that the events recorded in the delayed window are uncorrelated photons, and the rate of uncorrelated photons is the same in the event trigger window and the delayed window trigger. [28]

2.6 Pre-clinical PET Scanners

Pre-clinical refers to research done before the clinical stage. This research is usually done on animals. To study disease development, in-vivo imaging (e.g. imaging of living and intact subjects) is needed to see the development in the same subject. Some of the first uses of pre-clinical PET scanners on animals was in neurological studies in non-human primates and dogs.[29][12] This lead to the development of specialized systems for small animal imaging. Rodents (rats and mice) have properties which makes them suited for research, where rats are preferred in neuroscience because of its size compared to mice. Due to the relatively small size of the subjects, pre-clinical scanners have high demands on spatial resolution and count rate performance. One example of a pre-clinical specialized system for rodents is the RatCAP. The RatCAP is a wearable PET scanner for rats, which allows the subject to move while capturing the image, which makes it possible to image awake rats.[4] [18]

Figure 2.14: Mediso NanoPET/CT pre-clinical small animal dual modality scanner

Chapter 3

COMPET

3.1 COMPET, A Pre-Clinical PET Scanner

The COMPET project is housed by University of Oslo at the Experimental Particle Physics group and led by Steinar Stapnes and Erlend Bolle. The work is financed by the Norwegian Research Council (NRC) and the Swiss National Fund (SNF).
The aim for COMPET is to create a high sensitivity and high resolution preclinical PET scanner by implementing a novel block detector geometry with axial aligned LYSO crystals interleaved with wavelength shifters[1]. The detector consists of four equal modules where Each of the four modules are made from five stacked layers. One layer consists of 30 LYSO crystals with 24 Wavelength shifters (WLS). The LYSO crystals are $3 \times 2 \times 80\,\text{mm}^2$ and the WLS are $1 \times 3 \times 80\,\text{mm}^2$ in (HxWxL) [20]. See figure 3.4 and figure 3.1.[21] The LYSO crystal will give the energy information of the incident gamma and one of the coordinates, the WLS will detect the cone of light that escapes the LYSO crystal on one side and will give the position to the perpendicular coordinate. This set up gives Depth of Interaction (DoI) information from which layer the event occurred. The idea of using axial crystals interleaved with wavelength shifters was first proposed by the AXPET collaboration.[26][25] The geometry and DoI information will minimize parallax error which is uniform throughout the whole field of view. This enables the detector elements to be placed close to the subject for a higher sensitivity. Simulations have shown a resolution as low as 1 mm with a 15% photon sensitivity with a 50 mm bore opening.

3.2 COMPET Read-Out Chain

3.2.1 Analog Pre-Amplifier

The WLS and the LYSO channels are read out individually by custom SiPMs/MPPCs from Hamamatsu. The SiPM output signal is charge integrated followed by a linear decay from a constant current. The shaper has a rise time of around 1 ns (but the LYSO crystals rise time is around 40 ns) and decay time

[1] A wavelength shifter is a photo fluorescent material that absorbs higher frequency photons and emits lower energy photons

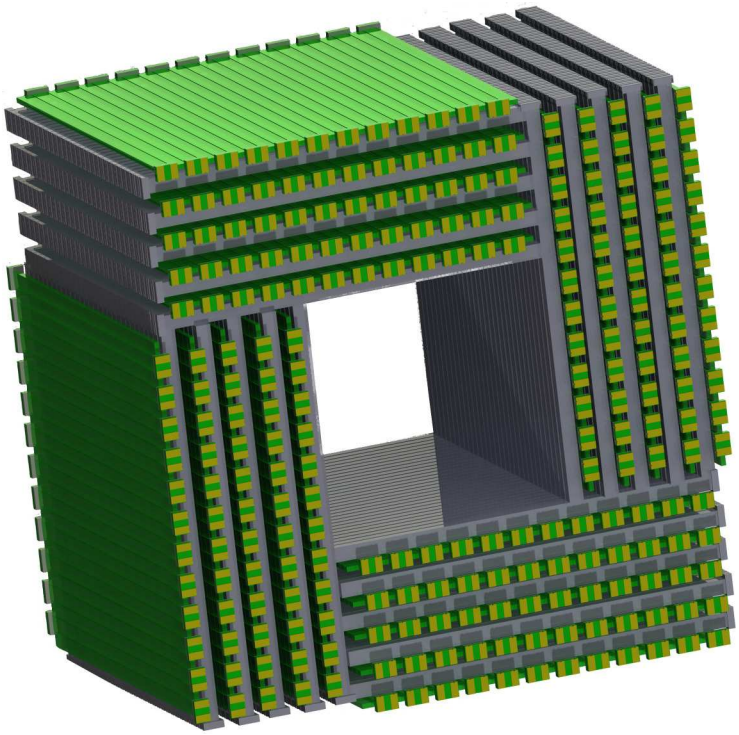


Figure 3.1: CAD drawing of COMPET detector design

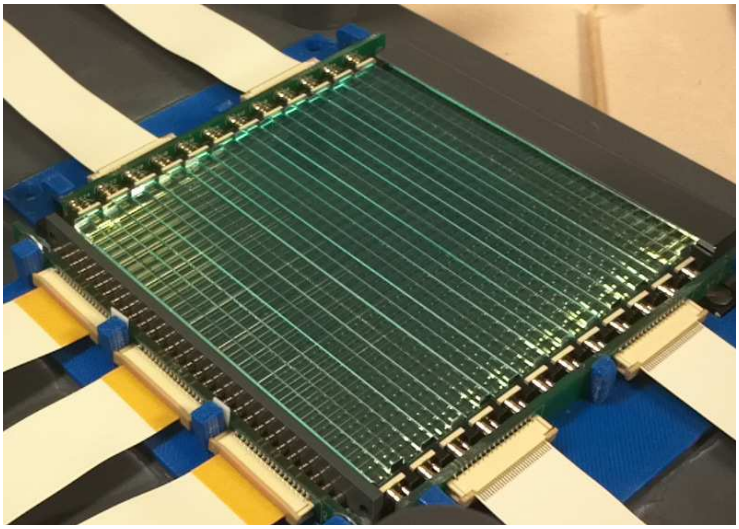


Figure 3.2: One fully assembled layer with 30 LYSO channels and 24 WLS channels

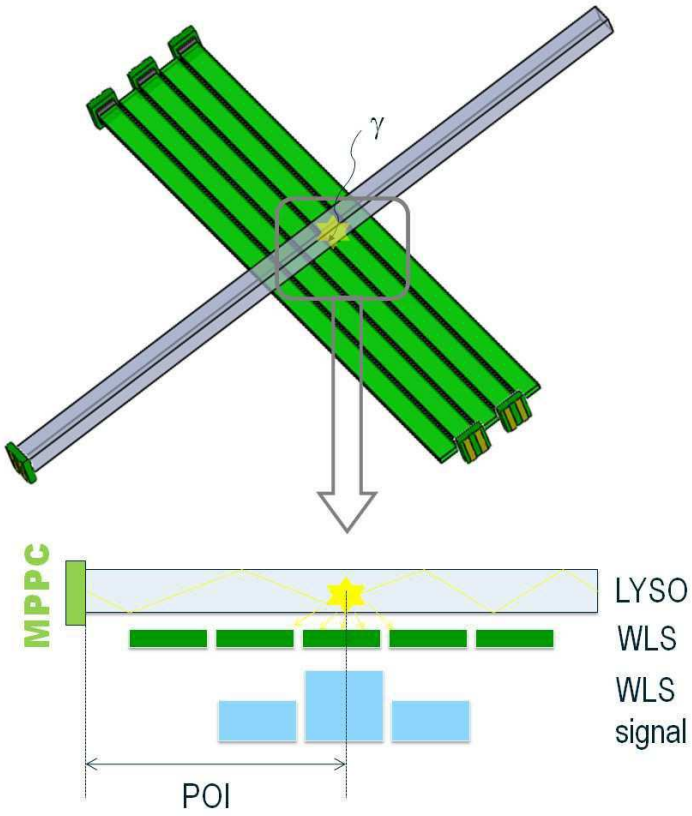


Figure 3.3: A cone of the scintillation light produced in the LYSO enters the WLS and has a certain probability of being absorbed and re-emitted at longer wavelengths, figure from [20]

of several 100 ns [21]. This signal is then discriminated with a threshold of several tens of photo-electron equivalents for the LYSO channels, and a few photoelectrons for the WLS channels. The reason for having a lower threshold on the WLS channels is because of their relatively low light output compared to the LYSO channels. In the end, the output of the pre-amplifier is a Time Over Threshold (ToT) digital signal where the length of the pulse corresponds to the energy deposited in the LYSO or WLS.

3.2.2 Digital Read-Out

The ToT signal is input to a deserializer port on a FPGA. The deserializer samples the signal at 1 Gbps and has a parallel output which is 6 bit wide. The parallel stream is then analyzed for rising edges which indicates a start of a ToT signal and a falling edge which is the end of the ToT. The frames with the rising and falling edges are then input to a parametrization extractor which computes the total time over threshold, start time (with 1 ns resolution), channel number and give the event an event number. When a rising edge occurs in one of the LYSO channels two trigger signals are raised. One which triggers the read-out of the WLS channels on the same card, and another which is to be sent to the Central Trigger Unit (CTU). The event itself is then put on a delay line to wait for a trigger decision which comes a set number of clock cycles later. If the synchronous CTU raises a trigger signal, the events within this trigger window will be read out. . For a full scanner there will be 20 read-out cards where each read-out card will handle one full layer with 30 LYSO channels and 24 WLS channels. [2]

3.2.3 Computer farm

The data produced by the read-out cards is sent to a computer farm by using UDP/IP networking. The UDP streams are written to memory/disk on the computers for off line analysis. The computers also handles the slow control of the read-out cards through UDP networking.

3.3 Central Trigger Unit Definition

3.3.1 Clock Distribution and Reset

The read-out cards need to have synchronized logic clocks to be able to supply meaningful data. Since coincidences are to be found between different cards the coincidence time resolution can never be better than the error due to unsynchronized cards. The read-out cards rely on global counters (for time stamping events) which are incremented on every clock cycle. Because of this, a distributed reset must exist to synchronously reset the read-out cards. A natural task for a centralized trigger is to handle clock distribution and resets as well as trigger logic. The read-out cards have a fine time resolution of 1 ns which is used to extract the ToT information from the pre-amplifier. In that regard a requirement for the clock distribution would be to have all cards synchronized within 0.5 ns. The reset mechanism on the read-out cards are made to be asynchronous assert and synchronous de-assert which means that for the cards to

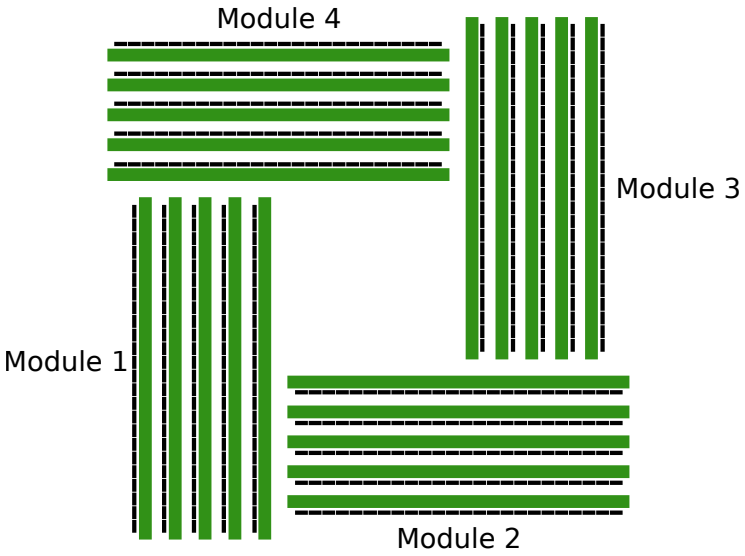


Figure 3.4: 2D view of the detector where the four modules are labeled

be synchronized within the same clock cycle the reset pulse need to be received within 5 ns on all cards when using a 100 MHz system clock.

3.3.2 Trigger Logic

Because of the detector geometry in COMPET, which gives it a uniform parallax error over the entire Field of View (FoV), there is no need in constraining the acceptance angle in the transverse FoV. A trigger can then assume four geometrical modules which can all be in coincidence with each other. COMPET aims to implement 3D event reconstruction, which is the reconstruction of single scattered events and even multiple scatters. This can be done by having a lower energy threshold and summing up events which happened in crystals geometrically close. If the energy of a group of events close together sums up to 511 keV the events probably arouse from a prompt. The line of response can then be drawn from the crystal in that group where the first event interaction took place from Compton kinematics.

The modules and read-out cards are arranged such that there are five cards handling one module, where each card handles a full layer. In the end there will be 20 read-out cards for the full detector. At this stage of the project, only one full layer is assembled and connected. This layer is read-out by five read-out cards. Because the COMPET detector is in a stage where number of layers and read-out cards change rapidly, the centralized trigger must be flexible and easy to configure.
The hardware for the Central Trigger Unit must have enough external IO ports to support trigger inputs, trigger outputs (for a synchronous scheme) and preferably a dedicated reset output for twenty read-out cards. The reset output does not necessarily need be on a dedicated line since it can be coded on the trigger output. An asynchronous solution will need gigabit Ethernet capabilities and

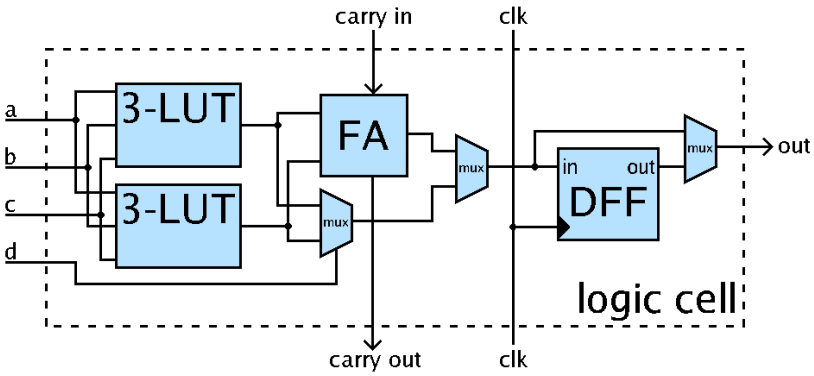


Figure 3.5: schematic overview of a simplified logic block in a FPGA

preferably a MCU in silicone for processing. When the hardware is unable to process the data from the detector because the rate is to high then events will be lost. Lost events mean that the sensitivity of the scanner will be compromised, which is highly unwanted. A clear aim for a Central Trigger Unit is to not be the bottle neck of the system.

3.3.3 Technology Selection

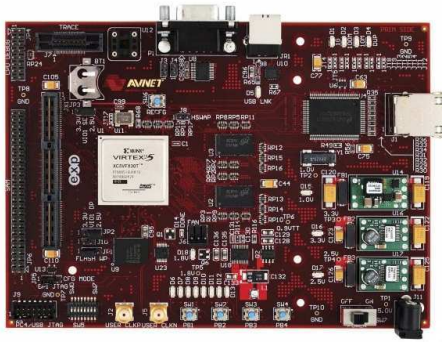
A centralized trigger for COMPET will need parallel processing capabilities. The synchronous solution has to listen to 20 different input channels for event signals and distribute trigger signals back. Since it needs to be done in real-time a micro-controller unit will simply not be fast enough to handle 20 input signals within 10 ns (which is one clock cycle of the system clock). An asynchronous scheme will need an Ethernet Media Access Controller for the PHYS[2] and enough processing power to evaluate the event information from all the cards. Two popular solutions for parallel logic are Application Specific Integrated Circuits (ASICs) or Field Programmable Gate Arrays (FPGAs). NIMs (Nuclear Instrumentation Modules) can be used for PET instrumentation, but a module based system is deemed to be less flexible than custom systems. An ASIC is fully customizable down to transistor level giving superior throughput and timing performance, but has a long development time and are costly for small production runs and once fabricated they are no longer customizable. For a one-off unit like a Central Trigger Unit an ASIC is not efficient in terms of development and production cost.

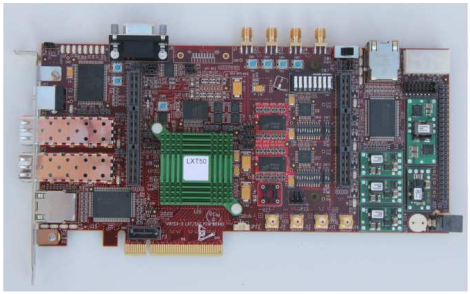
FPGAs contains a matrix of programmable logic blocks with programmable interconnections. A programmable logic block consists in general of Look-Up-Tables (LUTs), a Full Adder (FA), Multiplexers (MUX) and a Data Flip-Flop (DFF), See figure 3.5. These logic blocks can be programmed through vendor specific hardware and software tools. Different languages can be used for programming these devices, specifically Verilog, VHDL (VHSIC Hardware Description Language, where VHSIC is an abbreviation for Very High Speed

[2]Short for 'physical layer' in the OSI network model

Integrated Circuit) and SystemC. A modern FPGA usually also contain more specific resources like Phase-Locked Loops (PPLs), Digital Signal Processing blocks, Micro Controller Unit, high-speed transceivers and analog to digital converters. They are a popular choice in physics experiments because of the parallelism and reconfigurability, and by using pre-fabricated development boards you can have true rapid prototyping.

Development Boards in COMPET The COMPET project own three different development boards listed in table 3.1. The read-out cards are implemented on LXT boards, and the ML505 was used for prototyping them. The FX30T was bought to be used as centralized trigger, but it might not be the best choice. Because a synchronous trigger would need at least two connections for the trigger in and trigger out, and preferably one more for a dedicated reset the amount of external IO lines needed exceeds the available on the V5-FXT board and the ML505. The LXT board has 42 LVDS (low voltage differential signaling) lines and 34 single ended lines which would make it suitable as a synchronous Central Trigger Unit (which is available through an expansion board with four 60 pins male headers). It also have two Ethernet PHYs available, which would mean that slow control and asynchronous trigger information can be separated. However, on the other hand it does not have a dedicated Power PC core which would be suitable for slow control of the Central Trigger Unit. Note that all cards have SMA connectors which can be used for clock distribution.

Xilinx Virtex-5 FXT Evaluation platform V5FXT-EVL30T	Key Features
	- Xilinx Virtex-5 FX 30T - 20 480 FFs and LUTs - 2 480 kb BlockRAM - 4 10/100/1000 Ethernet MAC - 64 DSP48 slices - 8 RocketIO Tranceivers - 1 PCIs Express endpoint - 1 Embedded PowerPC core - 64MB DDR2 SDRAM - 16MB Flash - RS-232, JTAG, USB, - 1 10/100/1000 Ethernet PHY - 30 pins SAM connector - 1/2 EXP connector: 21 LVDS pairs

Xilinx Virtex-5 LXT PCI Express Development Kit V5LX-EVL50-G	Key Features
	- Xilinx Virtex-5 LX 30T - 28 800 FFs and LUTs - 2 160 kb BlockRAM - 4 10/100/1000 Ethernet MAC - 48 DSP48 slices - 12 RocketIO Tranceivers - 1 PCIs Express endpoint - 64MB DDR2 SDRAM - 16MB Flash - RS-232, JTAG, USB, - 2 10/100/1000 Ethernet PHY - EXP connector: 42 LVDS pairs + 34 SE pins

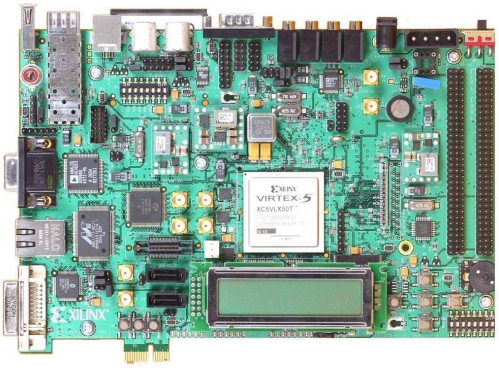
Xilinx Virtex-5 LXT Evaluation platform ML505	Key Features
	- Xilinx Virtex-5 LX 50T - 28 800 FFs and LUTs - 2 160kb BlockRAM - 4 10/100/1000 Ethernet MAC - 48 DSP48 slices - 12 RocketIO Tranceivers - 1 PCIs Express endpoint - 256MB DDR2 SODIMM - 32MB Flash - RS-232, JTAG, USB, Audio Jack In/Out, PS/2, DVI, VGA, SATA - 1 10/100/1000 Ethernet PHY - SAM connector: 16 LVDS-pairs

Table 3.1: the three different development cards owned by COMPET

Chapter 4

Implementation

4.1 System Overview

The FPGA design has been developed in several steps going from a Clock and Reset Unit to include synchronous triggering, Ethernet capabilities, rate measurements, slow control and finally asynchronous triggering. The system description in this chapter will not include any VHDL code but will explain the working principles by written descriptions and block diagrams. The project has been uploaded and maintained with a GIT repository located in "si:/mn/felt/epflab/pet/git/trigger.git" which is restricted to the COMPET project. A read-only repository is available at "git@github.com:kimei/CTU.git".

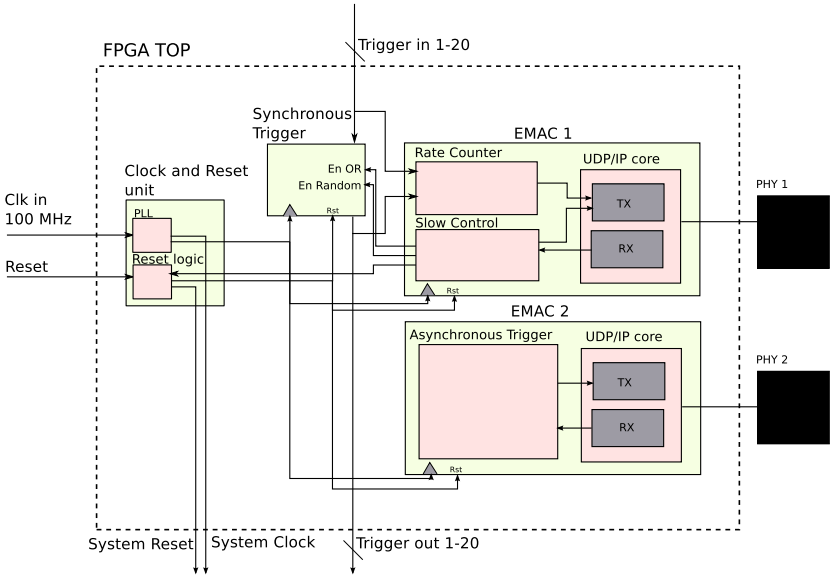


Figure 4.1: Simplified block diagram of the top level entity for the Central Trigger Unit

4.2 Clock and Reset Unit

4.2.1 Motivation

When we want to find coincidences between different read-out cards it is essential that the read-out cards have a common time reference. Synchronization is essential for both the synchronous triggering and for time stamping the events. It is essential that the logic clock on all the read-out cards are in phase with minimal time jitter with a distributed reset to synchronize the global counters.

4.2.2 Clocking methods

The following methods were considered to get a synchronous time reference.

GPS

A GPS sends continuously out time and position information on a 1.575 GHz carrier frequency. It would be possible to get a course time from a GPS and use the on-board clock for fine time. A set-up with GPS-time and fine time is used in smaller experiment such as in "Low-Cost Data Acquisition Card for School-Network Cosmic Ray Detectors" [6]. But just the fact that an open-air GPS antenna is needed makes this solution difficult to implement.

Fan-out

Another solution would be to generate a clock on the central trigger unit and fan it out to the read-out cards differentially with cables of equal length. This is a cheap and effective way of doing it, but would require a lot of cabling on the central trigger unit, which has a limited amount of differential IO ports. See figure 4.2.

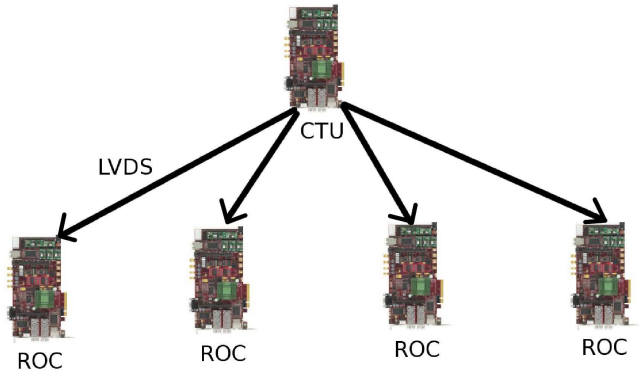


Figure 4.2: Clock distribution with fan-out from the central trigger unit

Daisy Chain

Since all the read-out cards are identical, the time delay from the IO ports to the FPGA should be the same. Therefore it would be possible to daisy chain a clock signal from the central trigger unit through the entire read-out chain while using the same configuration on all read-out cards. See figure 4.3

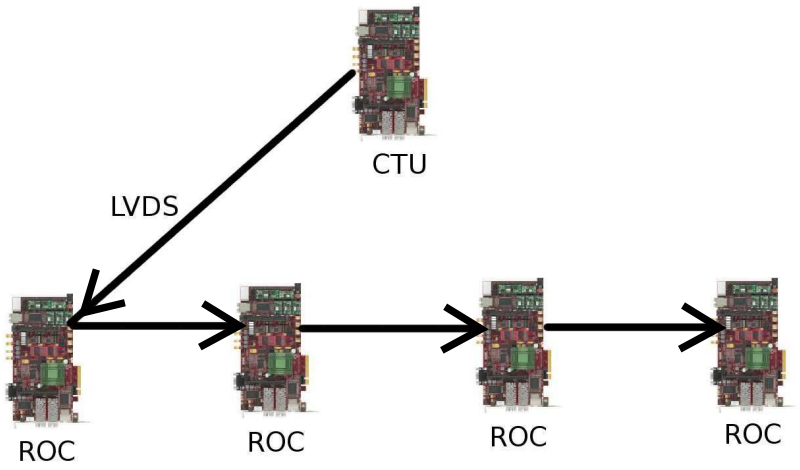


Figure 4.3: Clock distribution by daisy chaining the clock from the central trigger unit

4.2.3 Implementation

Clocking: The daisy-chain approach was chosen because the read-out cards have eight SMA connectors for LVDS signal. Four for input, and four for output. Only two inputs and two outputs are needed for differential signaling, the unused SMA connectors can be used for oscilloscope traces.

Output from Central Trigger Unit: The 100 MHz crystal on the Central Trigger Unit is used as the source clock. This is fed into a Phased Locked Loop (PLL) created by Xilinx CoreGen. The purpose of the PLL is to generate the clocks used internally on the Central Trigger Unit, mainly the 100 MHz master clock and the 200 MHz clk needed by the networking components, and the 100 MHz that is going out to the SMA connectors with LVDS. The PLL makes the output clock more or less unaffected by rapid changes in supply voltages (supply noise rejection), and it helps reduce clock jitter. See figure 4.4.

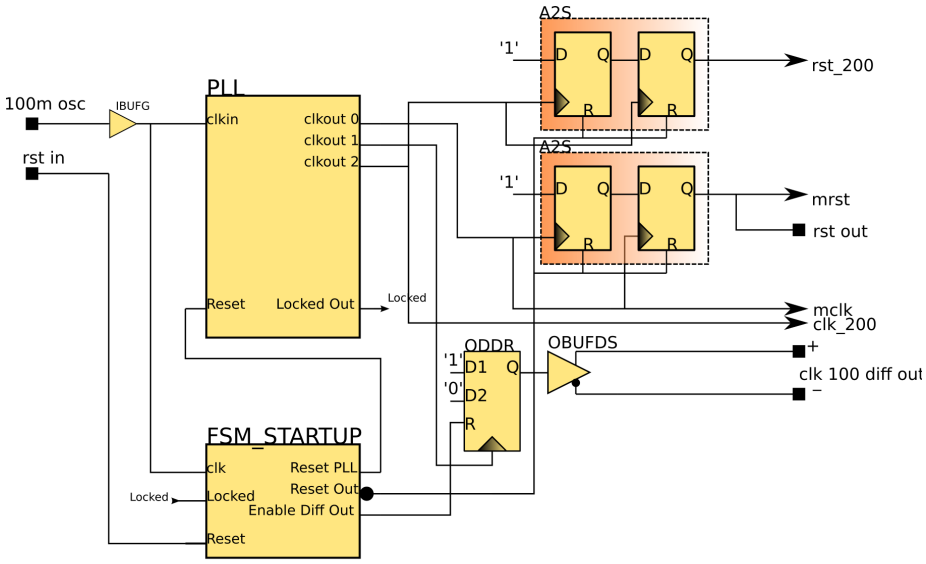


Figure 4.4: Clock and Reset Unit from the Central Trigger unit

The daisy-chain The output from the Central Trigger Unit is sent through 0.25 cm brass cables with SMA contacts in each end with 2.5 v LVDS signals to the first Read-out card. The read-out card has a Clock and Reset unit in the FPGA which was to be adapted to allow external clocking. The read-out card need to input the clock and adjust the phase so it matches the delay from one card to the next. Since the layout of the evaluation board is unknown we can not calculate the delay, it must be measured and adjusted in an iterative process. The phase adjustment is done by passing the clock through a Digital Clock Management block (DCM) from Xilinx, which is basically a Delayed Lock Loop (DLL) which in its simplest form is a tapped delay line, and then sent through a Phase Locked Loop[11]. The adjusted clock is then passed on as the master logic clock and as a clock out to the next read-out card. One of the quirks with using an external clock is that the DCM and PLL need to be reset on startup when a clock is present on the clock input, but we do not want to reset it if it has locked on to a clock. Since the clock is to be sent in a daisy chain we need some logic to make sure that reset signals from the Central Trigger Unit do not affect the DCM and PLL, see figure 4.5. A state machine in the Central trigger unit handles its own PLL and sends out reset signals with intervals matching the worst case time used for the DCM and PLL on the read-out cards in the daisy-chain to get a lock on the clock. see figure 4.5.

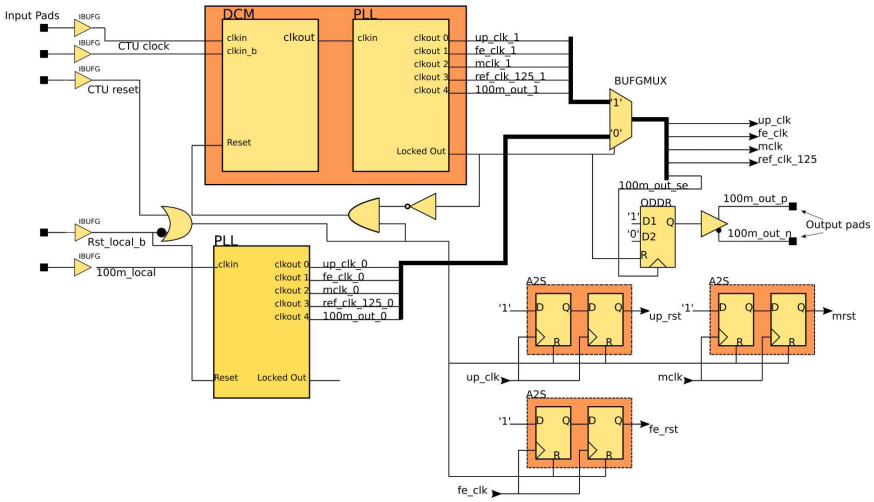


Figure 4.5: Clock and Reset Unit in the Read-Out Card

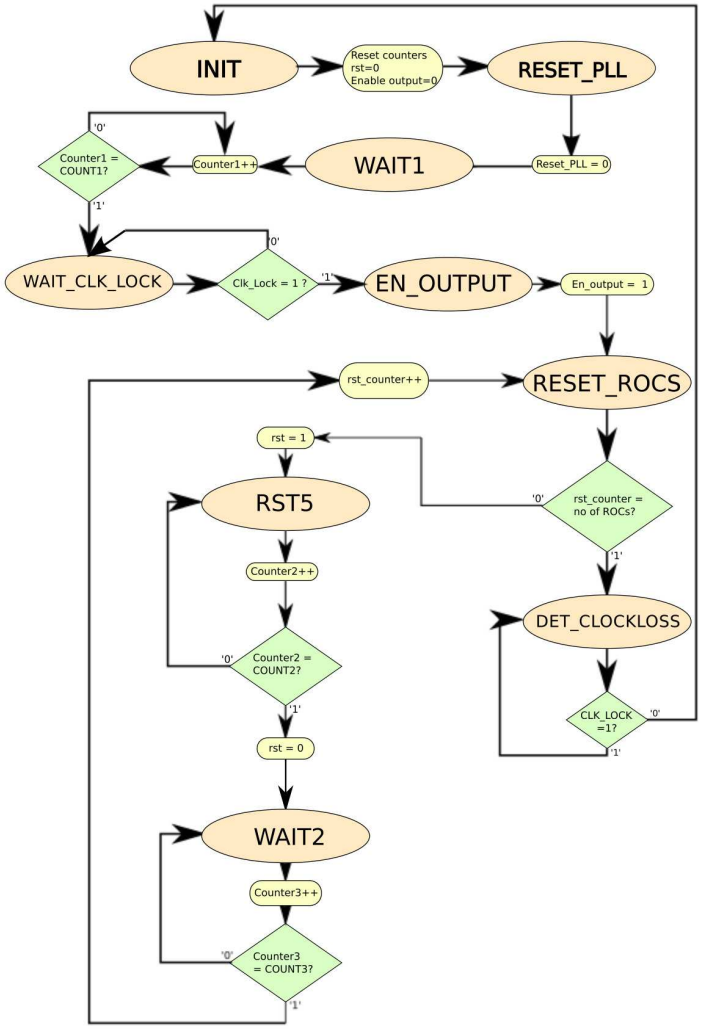


Figure 4.6: State machine in the Central Trigger Unit to handle the start-up procedure for the daisy chained clock (reset signals are active low)

Reset is distributed with a fan-out of 2.5 V LVDS from the Central Trigger Unit to the Read-Out cards. The DCM on the Read-Out Cards make sure that all the cards are in phase, but they are certainly not in the same clock cycle. Therefore the reset to the cards must arrive within the same clock cycle to make sure they are synchronized. This is done by fanning the reset out with cables of equal length. The connection for the reset signal is made on one of the headers on the Xilinx expansion card. The resets on both the read-out cards and the central trigger unit is made to be asynchronous assert and synchronous de-assert, following the design practice used in the read-out card design [2].

4.2.4 Test methodology

Testing the behavioral model of the clock and reset unit is done using a VHDL test-bench. Since the delay from one read-out card to the next is unknown, the easiest solution is to set-up a daisy chain with a number of cards and route the clock-out signal to an extra set of SMA connectors. The delay between each card and in the entire chain is then measured and corrected in the Digital Clock Module. The phase correction should be the same for each card in the read-out chain.

4.3 Synchronous Trigger

4.3.1 Synchronous Coincidence Trigger In COMPET

In a Real-Time system as COMPET a synchronous trigger would be an advantage for several reasons. In a synchronous scheme the need for buffering and storage is low because the waiting time for a trigger decision is known, and short. In COMPET, there are two trigger levels implemented which is not coincidence triggers. The L1 trigger in the pre-amplifier discriminates events with too low energies. The L2 trigger is situated in the read-out card itself and triggers the WLS channels when there is a LYSO event. When a LYSO event occurs, the read-out card also raises an External Trigger Out signal which will go to the Central Trigger Unit. If after a certain amount of clk cycles, the read-out card receives a trigger back, the event is sent out to the computer farm. A prerequisite for this to work is that the read-out cards only handles channels on the same module, which is the case in COMPET. To enable 3D event reconstruction a trigger between two read-out cards will open a trigger window on all the read-out cards. To be enable the read-out of the wavelength shifter events a wide trigger window must applied and timed such that the first event of an event pair is situated in the middle of the trigger window. With a synchronous scheme at 100 MHz the trigger window should be three clock cycles wide with the first triggered event in the second clock cycle. The coincidence window will then be two clock cycles wide and 20 ns on a 100 MHz system clock. The reason for not having a single clock cycle window of 10 ns is to not loose gamma sensitivity from coincident events happening near the edges of a clock cycle. The two clock cycle window must be a sliding window such that a single event will have to central trigger wait until the next clock cycle to make a trigger decision.

4.3.2 Electrical Connectivity

If the synchronous trigger shall work, events which happens within the same clock cycle on the read-out cards must arrive within the same clock cycle on the CTU. A flat ribbon cable has a typical propagation delay of roughly 5 ns/m, so if we have a flat ribbon cable with three pairs (reset, trigger in, trigger out) with the same length within a couple of centimeters propagation delay will not be an issue.

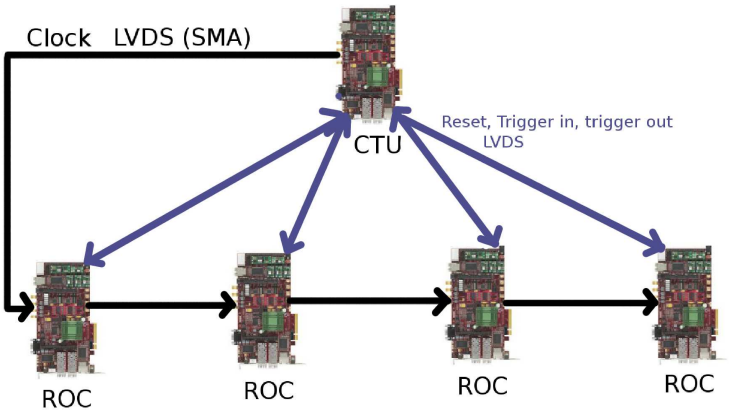


Figure 4.7: Reset, Trigger in/out connection with the daisy-chained clock

4.3.3 Trigger logic

First stage

The trigger inputs from the read-out cards are sent through a Xilinx IBUFDS to go from differential to single ended and synchronized internally by passing it through two flip-flops. The single ended signals are input to a rising edge detector where the output of the rising edge detector is grouped according to which module they belong to and sent through an OR port. The grouping is set by constraints in the Constants.vhd file, where the number of modules implemented is specified with the number of read-out cards in each module. For example, if there are two modules with three cards in module 1, and two cards in module 2, trigger input 1-3 is routed to module 1 and input 4 to 5 is routed to module 2. The trigger logic must be adaptable to different detector set-up since the full scanner is still in the making. The trigger input signals from the read-out cards is three clk cycles wide (10 ns), see figure 4.8.

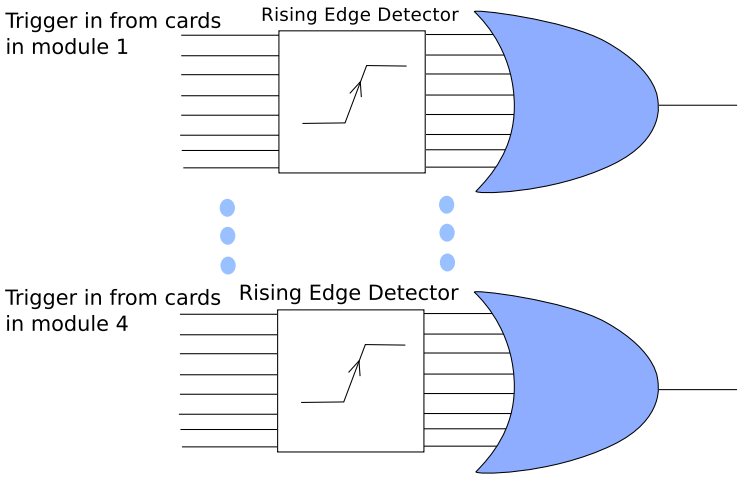


Figure 4.8: The trigger inputs are grouped by their respective modules and is input to a rising edge detector

Second stage

The trigger conditions are:

- Rising edge from two or more modules in the same clock cycle.
- Rising edge from two or more modules one clock cycle apart.

Because a trigger on the previous clock cycle can create a trigger condition a memory element is needed. The chosen solution was to create a 4 bit wide and 2 bit deep shift register and shift the output from the rising edge detector through on every clock cycle, see figure 4.9. A coincidence is then registered if there are more than one '1's in the first column, see figure 4.10, or more than zero '1's in the second column and more than zero '1's in the first column, see figure 4.11. If an event is registered, the logic '1's in the event shift register which created the trigger will be replaced by '0's. This is to prevent that one event can create multiple triggers.

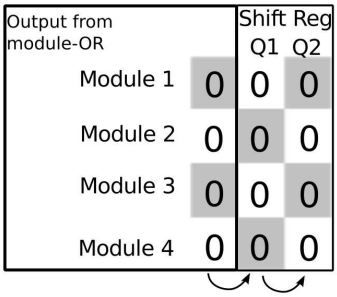


Figure 4.9: The output from the OR-gates in figure 4.8 is data in on the first column of the shift registers

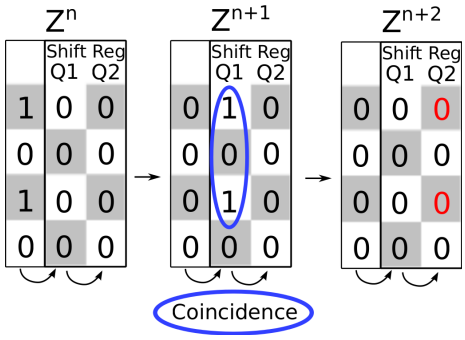


Figure 4.10: Coincidence from two rising edges in the same clock cycle

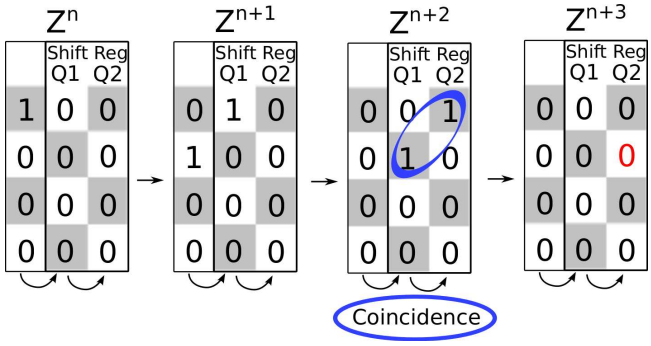


Figure 4.11: Coincidence from two rising edges in two consecutive clock cycles

Third stage

The trigger window need to have a known delay from the first event comes in to the trigger window opens. This seems like a simple constraint, but in reality it requires some logic. Since a coincidence can happen with both events in the same clock cycle or one clock cycle apart the trigger window needs to be set appropriately according to those two cases, see figure 4.12 and 4.13. On the read-out cards, only the LYSO channels can create trigger outputs. To be sure that Wavelength shifter events are triggered as well the trigger window needs to be three clock cycles long. To make up for the trigger window delay, all event information is put on a delay line and discarded or stored based on the trigger input.

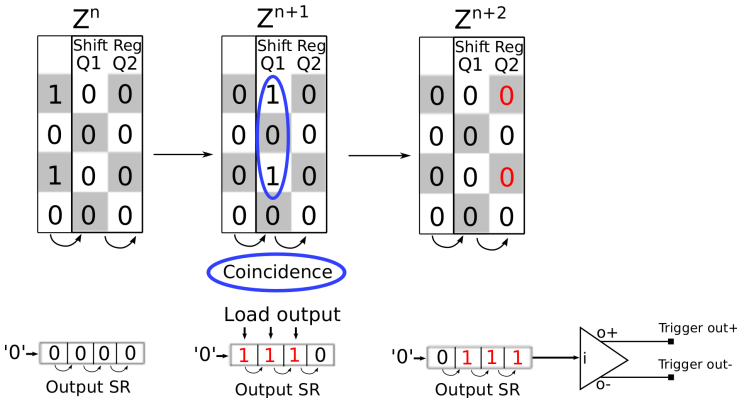


Figure 4.12: Loading the shift register to open the trigger window with a fixed delay

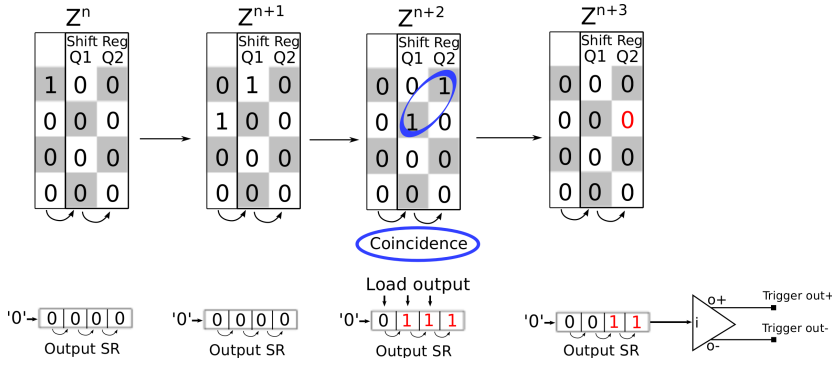


Figure 4.13: Same as figure 4.12 but with a different coincidence criteria

4.3.4 Trigger Modes

Different trigger modes are implemented. Specifically AND-triggering, OR-triggering, random-triggering and delayed window triggering. The OR trigger will create a trigger window on every incoming event. This is most useful to capture intrinsic spectras and source spectras for energy calibration. The AND-triggering is set as default and looks for coincidences between modules. Random triggering will open trigger windows at a set frequency unrelated to actual events. The delayed window opens a trigger window 100 μs after the coincidence occurred. Both random and delayed window trigger is used for random events estimation. Switching between trigger modes is done by the slow control (see appendix A).

4.3.5 Test methodology

Since the relation between trigger input (events) and trigger output (coincidences) are well known, a behavioral simulation is a straight-forward approach. A Python script writes out an ASCII file with input stimuli vectors. The input

stimuli is based on a probability of coincidence within a clock cycle and probability of noise in a clock cycle. It creates coincidences in both the same clock cycle and one clock cycle apart. The input vectors are then read in a VHDL test bench and the resulting simulation is analyzed for correct behavior.

4.4 Asynchronous Trigger

4.4.1 Motivation

The synchronous trigger need to have two LVDS pairs connected for each new read-out card that is added. With the reset mechanism which also need a LVDS pair this amounts to 60 LVDS pairs for a full set-up with twenty read-out cards. This could be solved by adding an extension board which makes the LVDS signals single ended or fanning out the reset and trigger-out signals with a strong driver. Another solution which will improve flexibility and scalability is to make the trigger signals asynchronous and use a 1 Gbps UDP/IP Ethernet network for communication. Since the events are time stamped from the de-serializing input on the read-out cards with a resolution of 1 ns (fine stamp, 10 ns for coarse), it would be possible to have coincidence windows in the nano second range using the sub-clk resolution. Scaling the system would be done by connecting the new cards to the existing network switches.

4.4.2 Networking

Hardware

The VLX5-PCIe development card has two National 10/100/1000 Ethernet PHYs. To stabilize the network load it is possible to have two separated networks, one for control signals and data read-out and the other for trigger signals. To send and receive event and trigger times an UDP/IP stack is implemented in the FPGA based on the open core implementation [24]. To have an idea of the reachable event rate on a 1 Gbps network a calculation is in order. An IP header is between 20 to 60 bytes depending on options set and the UDP header is 8 bytes long. Maximum payload on a UDP packet in the Xilinx Ethernet MAC implementation is 1480 Bytes. For event times of 32 bits this will allow us to send $\frac{1400\,\text{Bytes}}{4\,\text{Bytes}} = 350$ event times in one datagram. On a 1 Gbps we would theoretically be able to send $1\,\text{Gbps}/((1400+60)*8) = 85616\,\text{packets/second})$ which gives us a total of $350*85616 \approx 30\,\text{Mega events/second}$ with 32 bit event times.

Datagram structure

Each packet sent from the read-out cards and central trigger unit contain a 4 byte pre-amble which consists of 2 Bytes of static recognizable data, 1 Byte for a package counter that is incremented for every packet sent to the CTU and 1 Byte for number of event time in this packet, which limits the number to 255 event times per packet. The rest of the packet is 4 byte trigger times in big endian. The trigger time packets sent from the central trigger unit has the same structure, except that trigger times are little endian.

4.4.3 Trigger logic on the read-out cards

When an event occurs the value of a 32 bit 100 MHz global counter is stored in two FIFOs, event time FIFO and UDP Tx FIFO. The output of the parametrization filter, which is Time Over Threshold (energy), channel number and a fine-time stamp (1 ns resolution) is stored in a third FIFO. The contents of the UDP Tx FIFO is sent to the Central Trigger Unit. Trigger decisions are made in the CTU based on the event times sent from all the read-out cards. Trigger times are then broadcasted from the CTU to the read-out cards where they are stored in a fourth FIFO, the trigger time FIFO. The output of the event time FIFO is compared to the trigger time FIFO, and if they are equal, the corresponding output from the parametrization filter is read out and sent to the computer farm. The output of the trigger time FIFO and event time FIFO is also compared to a late running counter (which is in the order of 5 ms late). If the late running counter is equal to the event time FIFO, the event time is discarded among with the corresponding output of the parametrization filter. If the late running counter is equal to the output of the trigger time FIFO, the trigger time is discarded. see fig 4.14

4.4.4 Trigger logic on the Central Trigger Unit

The CTU receives event times (32 bit) from the read-out cards by UDP/IP. Based on the senders IP address and a look up table the received event times get stored in one out of four FIFOs. The placement is based on which module the read-out card belongs to. The output of the four FIFOs are then compared to a late running counter which is synchronized with the late running counter on the read-out cards. If the output of any of the FIFOs are equal to the late running counter, the event time is read-out and discarded. In the case that the output of two (or more) of the four FIFOs are equal, the event time is read-out and sent to a UDP Tx FIFO to be broadcasted back to the read-out cards, see fig 4.15.

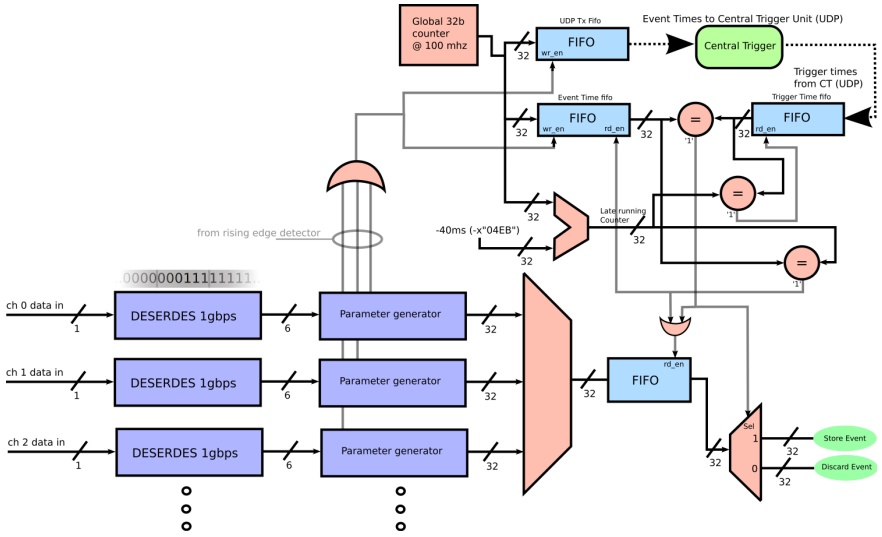


Figure 4.14: Simplified schematic overview of trigger functionality on the readout cards

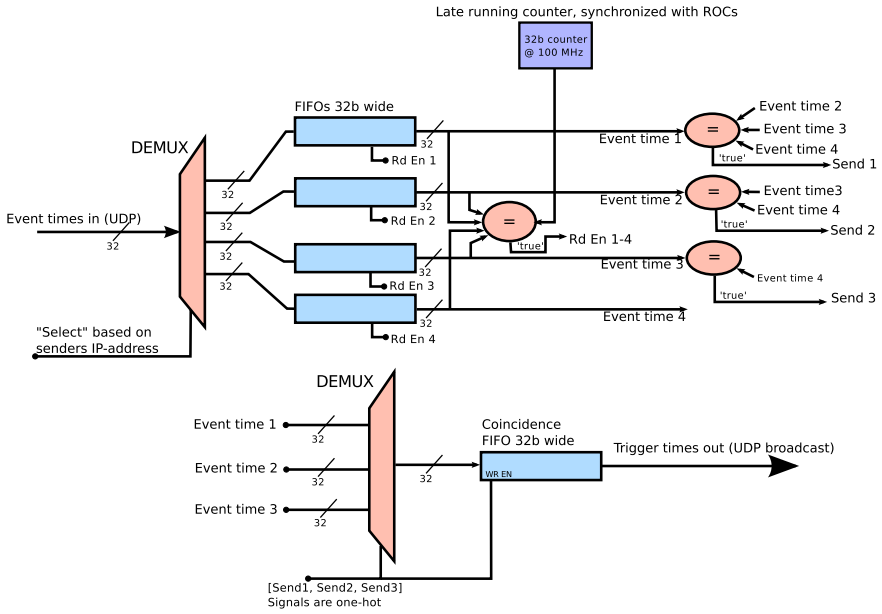


Figure 4.15: Simplified schematic overview of trigger functionality on the central trigger unit

Asynchronous trigger in software

Because debugging is simpler in software it was decided to create the asynchronous trigger in software first. The program was written in C++ using three different POSIX threads.

Thread 1 receives event times from the read-out cards and put them into one out of four FIFOs dependent on which module the read-out card belongs to. UDP communication is done by using the BOOST asio library.

Thread 2 reads the four FIFOs looking for correlating times while holding a semaphore. When an event is found this time is written to a trigger time FIFO. When a certain number of trigger times are written to the trigger time FIFO the semaphore is released.

Thread 3 waits for the semaphore which Thread 2 is holding. When the semaphore is released it reads a certain amount of Trigger times and broadcasts them back to the read-out cards.

4.4.5 Timing and buffering

Estimation of the network latency is needed to estimate how much buffering is needed. We assume that we are using a store-and-forward switch and not a cut-through switch. The difference is that a cut-through switch only reads the first 6 bytes of the package to know the location and not the entire package (and it does not discard corrupted packages). The store and forward latency for a 1400 byte package (which is the largest frame size the Xilinx EMAC can handle) is $L_{\text{sf}} = \frac{1400\,\text{Bytes}}{125\,\text{MB/s}} = 11.2\,\mu\text{s}$ The wireline latency is negligible with the cable length used in this experiment. The average queuing latency will be $L_q = \text{L}_{\text{network load}} * L_{sf}$. It is appropriate to assume a 75% network load, which gives us $L_q = 0.75 * 11.2\mu\text{s} = 8.2\,\mu\text{s}$. The total expected average latency on 75% load is $L_{tot} = L_q + L_{SF} = 11.2 + 8.2\mu\text{s} = 19.4\,\mu\text{s}$. The read-out cards and the central trigger is set to send out event times and trigger times every 0.5 ms or when there are 255 stored event or trigger times. This means that a delay of $0.5\,\text{ms} + 0.5\,\text{ms} + 2 * 19.4\,\mu\text{s} = 1\,\text{ms}$ can be expected for a trigger decision. The late running counter goes 5 ms late on the read-out card and 2.5 ms late on the central trigger unit. With this scheme a FIFO 32 bit wide 16384 slots deep can buffer a rate of $(16384/2\,\text{ms} = 8\,\text{M events/s})$.

4.4.6 Trigger modes

AND triggering is set as default trigger mode, this is done by looking for equal event times in the different modules. OR triggering should effectively send a trigger on every event captured, without sending the same event time twice. This limits the use of a simple loop-back where every event time is sent back as a trigger time. One solution is to buffer every received event time for some period T, if the regular AND trigger finds a coincidence with this event time in this period it is sent back as soon as the coincidence is found. If no coincidence is found in this period T, it is sent back after T seconds. T is set to be 2/3 of the late running counter (3.3 ms). Random triggering is done by sending out the value of a synchronized counter at a set frequency. Delayed trigger windowing is done by adding a constant t_d to every trigger time, and sending that delayed time after t_d seconds.

4.4.7 Limitations

When more than one read-out card is associated to each module FIFO (see fig 4.15) the event times are no longer sorted. Therefore, this implementation needs exactly one read-out card per module, which is fine when there is only one layer per module but will be problematic when each module has five layers. One solution is to add as many FIFOs as there are read-out cards, but there is simply not enough memory on the FPGA to do this.

4.4.8 Trigger Logic improved

An improved solution for functional UDP trigger logic has been designed, but not fully implemented and tested. The preliminary results have been presented and published at the IEEE NSS/MIC (Nuclear Science Symposium/Medical Imaging Conference) poster session and proceedings [14]. The logic on the read-out cards is mostly the same, but instead of sending event times when a certain amount of event times are stored, they are sent at a set frequency. The period between each time all the read-out cards send out event times is a time slot, and trigger times are only found within one time slot and not between two or more time slots.

Trigger Logic On The Readout Cards

Every $100\,\mu$s a UDP package is sent to the central trigger unit containing the event times from the last $100\,\mu$s. A FIFO holds the trigger times that are received as broadcasts from the CTU. The received trigger times are compared to a late running counter which is synchronized on all the readout cards. The value of this late running counter is equal to the system time minus the trigger decision time of $120\,\mu$s when using $100\,\mu$s time slots. If the late running counter is equal to a stored event time, the associated event will be discarded. If an event time is equal to a trigger time, the event will be sent and stored to disk in a computer farm. If a stored trigger time is equal to the late running counter, which means that the trigger time did not coincide with a stored event on this card, the trigger time will be discarded.

Trigger Logic On The Central Trigger Unit

Readout cards send their event times every $100\,\mu$s. Since the readout cards are synchronized they send out packages at the exact same clock cycle, the network switch receives the twenty UDP packages at the same time and sends them one by one to the CTU. Such, the UDP streams are naturally synchronized and package loss can easily be detected. The maximum network load can be set by having a maximum number of event times in one package. It also reduces the memory usage on the CTU since event times only get stored the duration of one time slot. The latency between each package received by the CTU is measured to be $1\,\mu$s with 512 Bytes of payload in each package. This gives the trigger $100\,\mu\text{s} - 1 * 20\,\mu\text{s} = 80\,\mu$s to make trigger decisions from the received event times with $100\,\mu$s time slots.
The event times in the event time packages are all put in a unique FIFO. The sender's IP address decides which FIFO the event times are written to. When

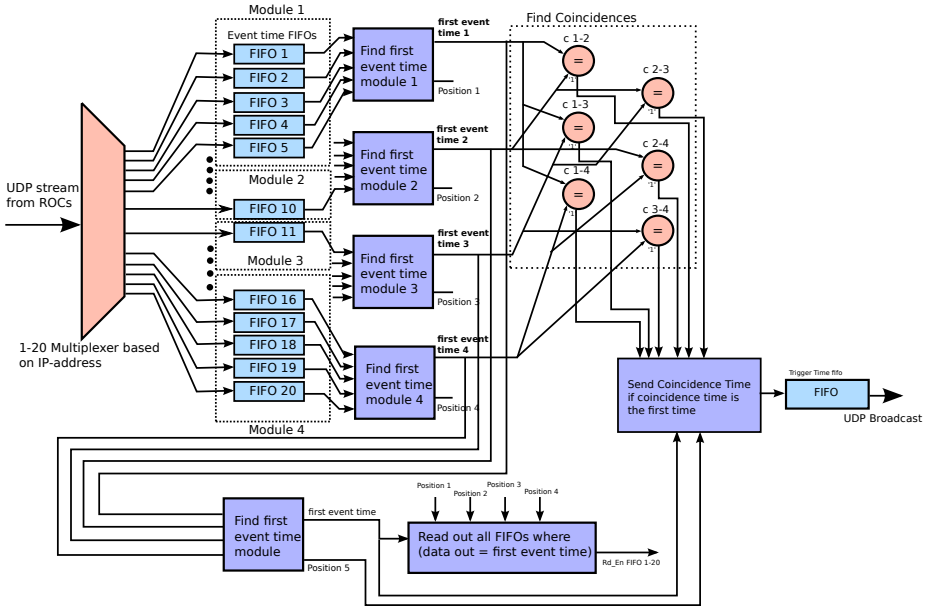


Figure 4.16: Simplified schematic showing the improved trigger logic on the Central Trigger Unit

an event time package has been received from all readout cards marked as active, the FIFOs contains the event times from all the readout cards sorted such that the first event time is on the output of each FIFO and the latest is at the end. The twenty FIFOs are then grouped such that the content of the FIFOs in a group comes from the readout cards in a geometrical module. In COMPET there are four different modules which gives us four groups. There are five readout cards in each module which gives us five FIFOs in each group. At every clock cycle the FIFO with the earliest time on the output is read. This happens at the same time as the earliest event time from each group is compared for coincidences. A coincidence simply means that two or more times are equal. If they are equal, the coincidence time is written to a coincidence time FIFO and all event time FIFOs with this value on the output is read out. When all twenty event time FIFOs are empty, the content of the coincidence time FIFO is sent as a UDP broadcast to all readout cards.

The CTU senses overflows in the counter and treats them logically. In order to adjust to an unknown number of readout cards in each group, the inactive UDP streams can be masked out on the fly, or a statically set trigger mask can be applied. Slow control of the CTU and the readout cards is also done through UDP.

Data structure and timing

The coarse time stamp used for event times and trigger times are each 32 bits. At 100 MHz this gives a dynamic range of 40 seconds, which is highly redundant and will be reduced in the final implementation. The number of bits needed depends on the resolution and length of a time slot. With a timing

resolution of 1 ns and $100\,\mu$s time slots, 14 bits are needed to get a dynamic range larger than $100\,\mu$s. If a high resolution time stamp exists it can be used for trigger information without changing any trigger logic. In the COMPET readout system the time stamp from the 1 Gbps deserializer inputs can, and in the end, will be used for trigger decisions. A theoretical limit on trigger rates is network dependent, and for a 1 Gbps switched network using 32 bits for each event time, the theoretical limit is $1\,\text{Gbps}/32\text{b} \approx 30\,\text{Mcps}$. This assumes that the header size is negligible to the payload size, meaning that high number of events are sent with each package.

The proposed scheme makes use of time frames. In the first implementation a time frame is $100\,\mu s$ and a maximum number of events per time frame is set to 128. This limits the event rate from each card to approximately 1 Mcps, and gives the CTU $80\mu s$ to make trigger decisions, which is redundant since the time to sort singles from coincidences is $N_{\text{max. number of event times in one package}} * T_{\text{clk period CTU}}$. To achieve higher rates the length of a time frame and the maximum number of event times per package must optimized. The longer the time frame, the higher is the need to buffer events on the readout cards. The buffering is implemented with the means of FIFOs, where the largest uses the on-board DDR2 memory.

4.4.9 Test Methodology

Testing asynchronous logic can be a cumbersome task. Since delays are dependent of network load and the ratio of singles versus coincidences, the testing should be done in-situ. By using ChipScope it is possible to assure that trigger times come in correct order while checking that the FIFOs do not fill up. A list of key factors to observe is written below.

- Incoming trigger times arrive sorted.
- Incoming trigger times are always bigger (higher value) than the late running counter.
- None of the FIFOs used for buffering fills up.
- Package losses, observable with the package counter.

4.5 User Interface

4.5.1 Motivation

The Central Trigger needs to have an interface for changing trigger modes, masks and overview of trigger rates and event rates from the read-out cards.

4.5.2 Communication

The Trigger card has a USB port, Serial rs-232 port and two 1 Gbps network PHYs. All suitable for control signals. USB has a maximum cable length of 5 meters, this is because USB uses source termination and voltage mode drivers. Longer cables will allow for reflections to pile up and destroy the

driver. Because a PET scanner is a radiation detector, the chances of a control center further away than 5 meters is very probable. RS-232 with null-modem cable is not exactly bleeding edge technology, and does not offer the best data rates. Some newer computers does not even ship with a serial input. The UDP trigger only use one of the Ethernet PHYs, which means that the second can be connected to the read-out network which is connected to the data farm and control computer. The control signals will go through UDP/IP, which is an connectionless protocol not designed to handle control signals. To have some sort of reliability without implementing TCP, the received datagram is echoed back to the computer which sent it. With this it is possible to control that the sent signal is the same as the received. The computer have to take care of re-sending signals if the sent and echoed does not match.

4.5.3 Masking, trigger modes and reset

For debugging and a general development environment being able to turn on and off cards for trigger input and output is an advantage. In this trigger this is done by sending an appropriate pre-amble, command and a 3 byte long bit string to specify which inputs or outputs to turn on/off. In the synchronous trigger the Trigger input masking is done by bitwise AND with the trigger-in mask. By default this is set to all '1's. Masking the the trigger output is done in the same fashion by masking the trigger windows going out to the cards. Masking trigger input for the asynchronous trigger is done by discarding the event time packet received by disabled read-out cards. Masking the trigger output for the asynchronous trigger is not implemented since the trigger times are broadcasted over UDP/IP. Trigger modes are also set by sending the appropriate commands. The Central Trigger Unit defaults to AND triggering. The Central Trigger can also send out reset pulses to the read-out cards, this reset also resets counters which are synchronized with the read-out cards on the central trigger. The commands and format are described in detail in Appendix A.

4.5.4 Trigger rates

The number of rising edges from each read-out card and the number of coincidences is counted and stored as a 32 bit unsigned integer. Every second these numbers are sent to the computer running the control software. The format is described in Appendix A.

Chapter 5

Results

When most of the results were obtained COMPET consisted of one fully populated layer read out by five read-out cards. The deserializer inputs on the read-out cards were set to capture 6 bits at 500 MHz giving a timing accuracy of 1.6667 ns. In-situ tests were carried out with this set-up unless stated otherwise.

5.1 Timing and Synchronization

5.1.1 Phase Relations From the Clock Distribution

To measure the performance of the daisy-chained clock distribution, the phase relations between the different cards must be measured. Since the clock lines go from a SMA connector to a shielded coaxial cable there are no obvious lands for probing. To have a probe point, the clock output from each card is output to an extra set of SMA connectors which can be connected to a trace on a oscilloscope through SMA-coaxial-BNC connection. By connecting two cards with the same set-up the phase relation between read-out cards can be measured.
The oscilloscope used for the following measurements is a Lecroy Wavepro 715Zi 1.5 GHz 20 GS/s which can measure the phase relation between two traces and histogram the results. The histograms are limited to 1000 samples, but measuring the average phase relation and standard deviation is unlimited in the number of samples. The phase adjustment in the DCM block on each card was set to have a phase adjustment of -194.064 degrees which is -5.391 ns on a 100 MHz clock. The increments can be adjusted in steps of 0.04 ns for a 100 MHz input clock.

Table 5.1: Measured phase relations from Lecroy Wavepro 715Zi with N=10000

Card numbers	Phase relation	Standard deviation
Card 1 - 2	-1.79%	0.96%
Card 2 - 3	0.58%	0.96%
Card 3 - 4	0.27%	0.92%
Card 4 - 5	-1.97%	1.17%
Card 1 - 3	0.164%	1.06%
Card 1 - 4	1.33%	1.12%
Card 1 - 5	0.85%	1.36%

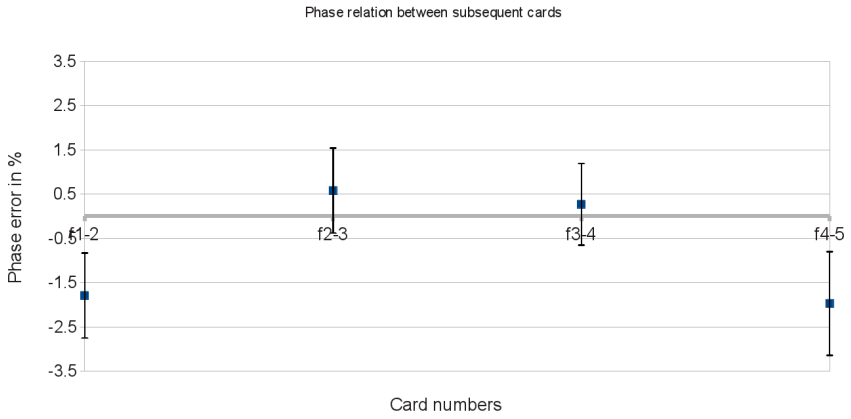


Figure 5.1: Plotted data from table 5.1 showing phase relation between subsequent cards

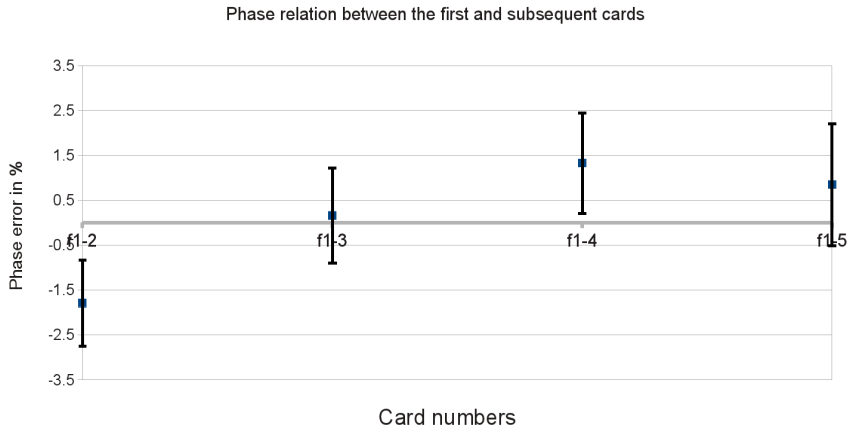
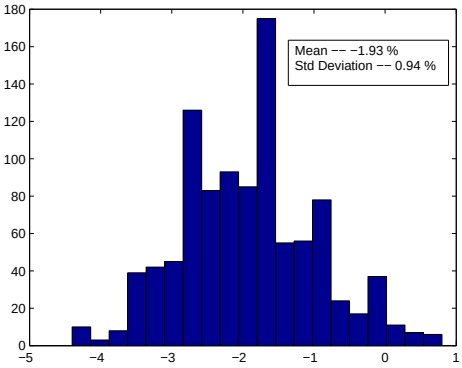
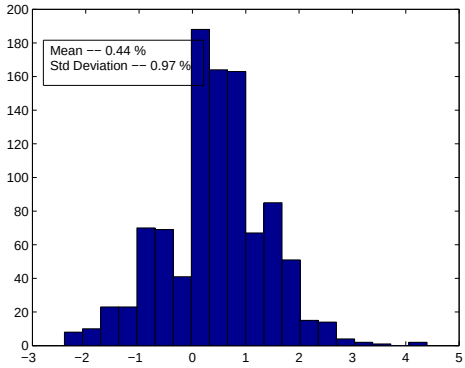


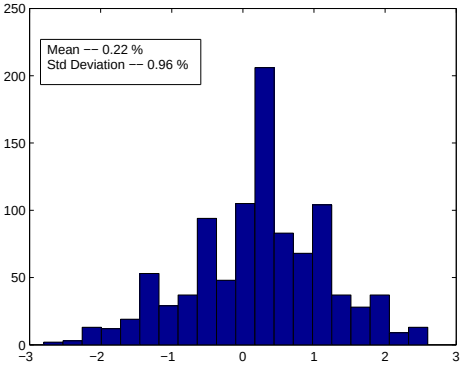
Figure 5.2: Plotted data from table 5.1 showing phase relation between the first and the subsequent cards



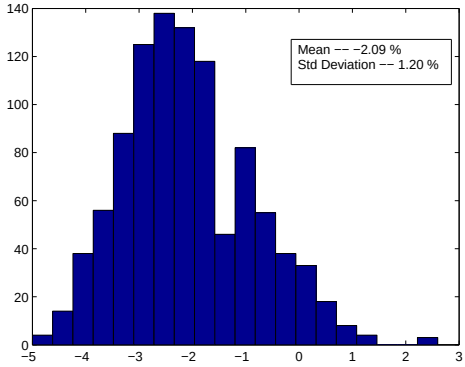
(a) Card 1 to 2



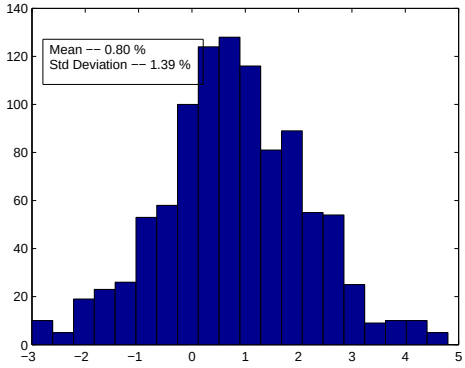
(b) Card 2 to 3



(c) Card 3 to 4



(d) Card 4 to 5



(e) Card 1 to 5

Figure 5.3: Histograms of phase errors between different cards. Histogram data was taken by Lecroy Wavepro 715Zi with N limited to 1000

Table 5.2: Measured time difference from 0 V to -70mV with different peak pulse heights

Pulse peak in mV	Δt from 0 V to -70 mV in nano seconds
-88	28$\pm$5
-102	27$\pm$5
-180	18$\pm$5
-255	16$\pm$5
-300	15$\pm$5
-350	13$\pm$5
-458	10$\pm$5
-616	10$\pm$5
-1100	8$\pm$5

5.1.2 Time-walk and Timing Uncertainties From the Pre-Amplifier

The COMPET read-out system does not yet have a scheme for time-walk corrections which is a major contributor for the timing uncertainties. The main contributor for timewalk in the COMPET pre-amplifier comes from the threshold set on the comparator which creates the Time-over-Threshold signal from the charge integrated pulse with linear decay. To measure the time-walk a test pulse generator is used, but since all pre-amplifiers are in use the measurements must be done in-situ. The measurements were done on the pre-amplifier for the LYSO channels. The two probe points are before and after the comparator which is used as an energy discriminator. The voltage threshold for the comparator is set to -70 mV and all measurements are done with the Lecroy Wavepro 715Zi oscilloscope.
The first measurement was done by triggering a single time and measuring the time difference from the analog signal started to rise (fall) until it reached the threshold of -70 mV. This was done for different pulse height peaks to get an energy vs Δt plot. The measurement error is mostly from the oscilloscope reading to find the beginning of a pulse. Since the rise time is exponential it is difficult to find an exact start for where it left the baseline.

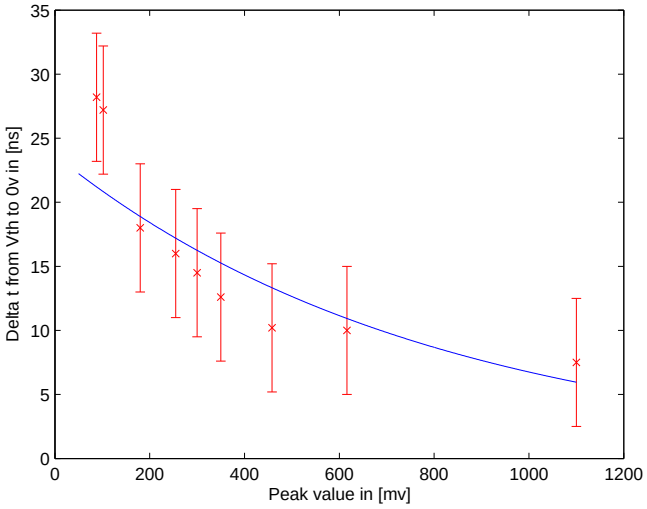


Figure 5.4: Measured time-walk for different peak pulse heights with an exponential line fit with data from table 5.2

The oscilloscope was then set to persistent mode, where it overlays each triggered trace on top of each other. A Na-22 source was put close to the scanner while the analog input signal to the comparator was measured to see the shape of it. The result is shown in figure 5.5 where the 511 keV events can be seen as the lower red to green lines and the Compton spectrum above with a valley in between.

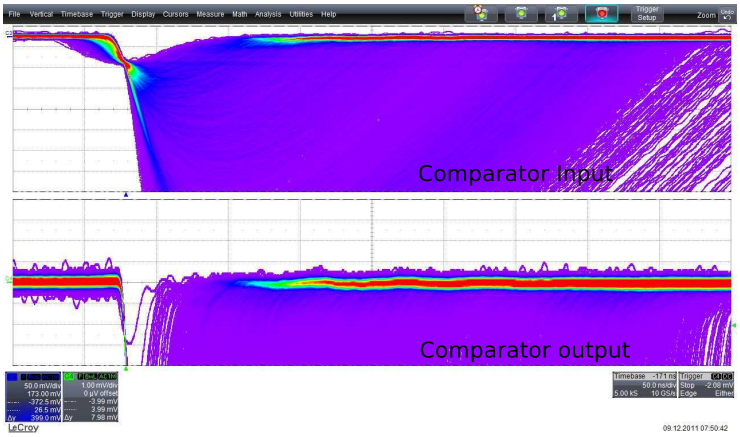


Figure 5.5: Analog input to the comparator and the comparator output traced in persistent mode from Lecroy Wavepro 715Zi

To see the difference between start time in the analog signal and the differential signal from the comparator the oscilloscope was set to trigger on the output of the comparator in persistence mode. What is expected to see then

is a clear line for the rising edge of the comparator output (falling edge in this case since we are probing the negative output) and a smeared out rising edge for the analog signal.

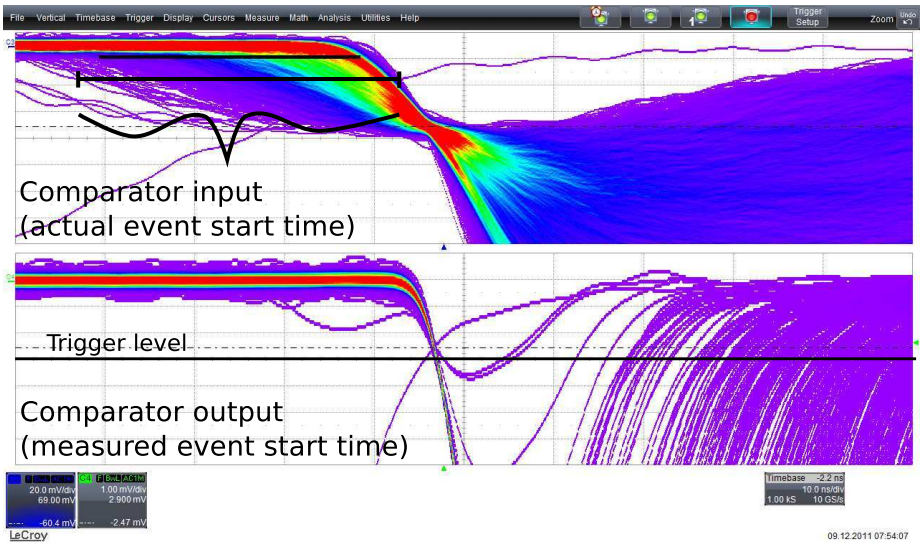


Figure 5.6: The charge integrated pulse with linear decay as comparator input and comparator. Oscilloscope trigger is on the comparator output channel. The use of persistent mode show the relation between the start time of falling edges on the comparator output and the start time of falling edges on the comparator input. from Lecroy Wavepro 715Zi

Figure 5.6 show that there is timing uncertainty from time-walk in the pre-amplifier, the black line drawn on the comparator input gives a coarse representation of the difference in event start time in relation to the measured start time. With no time-walk the falling edges on the comparator input would all be in same place on the x-axis. The length of the black line is around 35 ns. However, it does not give any accurate information about the time-walk itself because jitter is included as well. There is also a underlying noise from the 100 MHz system clock.

5.2 Synchronous Trigger

5.2.1 Simulations

The synchronous trigger has a well defined input/output relation which makes it suitable for behavioral simulations with ModelSim. A test bench would need to provide input vectors for worst case scenarios and preferably random data. Since the synchronous trigger opens a trigger window of three clock cycles and the read-out cards are edge sensitive there must be a dead-time of four clock-cycles. The dead time is what limits the maximum achievable trigger rate to 25 MHz when running on a 100 MHz clock. The single rate from each card is also limited by the fact that the Central Trigger is edge sensitive. The maxi-

mum single rate will then be half of the system clock which is 50 MHz on the 100 MHz clock. A python script was written to read the constants.vhd file to interpret how many modules and read-out cards is currently implemented, and from a chosen possibility of a single event and coincident event, write input vectors for each clock cycle for 1 ms in ASCII format. The test bench reads this list of input vectors and use them for trigger inputs. The output waveforms is then analyzed manually. The simulations all have twenty trigger inputs divided between four modules where input 0-4 is in module 1. from figure 5.7 the behavior of the trigger can be studied. In the simulations the trigger was set to trigger on coincidences between modules. Meaning that input 0-4 can be in coincidence with inputs 5-19, and inputs 5-9 can be in coincidence with 0-4 and 10-19, and so on. The waveforms are colored such that inputs and the resulting trigger window have the same color. The dead time is colored as a shade of green, the trigger inputs which is located within this green area will be ignored. The exception is of course if the second event to form a coincidence is one clock cycle later than the first. The dead-time means that no new coincidences can form within this area. Simulations for the delayed trigger, OR trigger functionality and different patterns can be found in the appendixes. However, the results from the simulations all show that the trigger behaves as expected for different trigger modes and input stimuli.

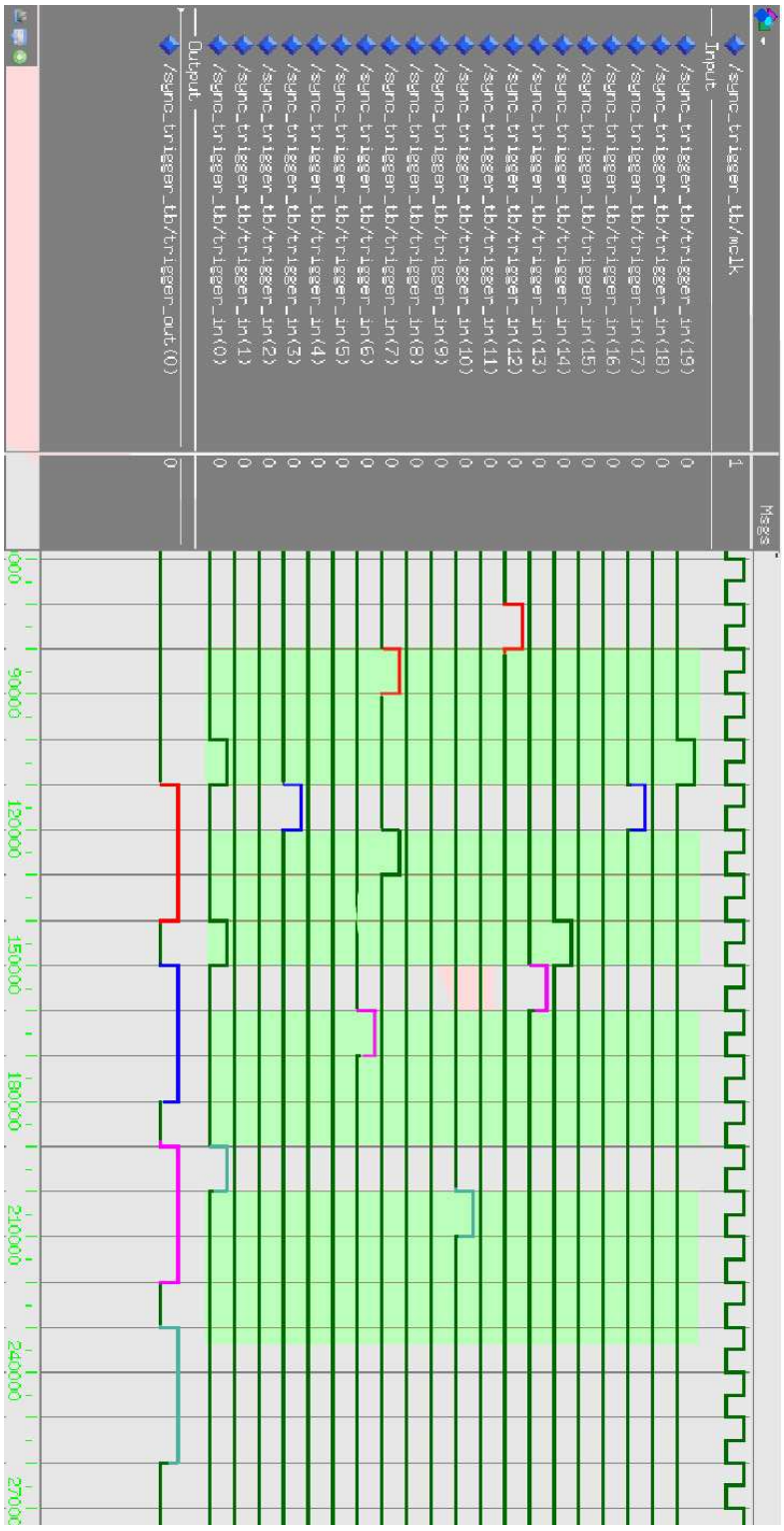


Figure 5.7: Simulation from ModelSIM with twenty read-out cards in four modules in coincidence with random high-rate data

5.2.2 Timing and in situ-tests

The Read-out cards were set to send out test pulses. A trigger signal will be raised from a read-out card every time an internal counter overflows. By using ChipScope[1] in the central trigger the trigger input signals can be analyzed together with the trigger out signal. This will show two things: The read-out cards are synchronized within the same clock cycle viewed from the trigger, and the delay from trigger input until a trigger window opens on the output is the same as in the simulations.

Figure 5.8 shows the delay from trigger input to trigger output is the same as in the simulations. Two of the clock cycles in the delay are from the synchronization through two flip-flips. When the trigger input was captured with Chipscope before this synchronization you could observe that the trigger inputs sometimes stretched over two clock periods. That was to be expected since the trigger inputs and the local clock is not synchronized at this stage. Figure 5.10 shows the behavior of the delayed trigger behaves as expected where it opens a window 127 clock cycles later than the coincidence occurred, figure 5.11 shows the trigger operated in OR mode when two trigger inputs separated by the dead-time arrives. The result is as expected from the simulations as it opens a two trigger windows separated by one clock cycle.

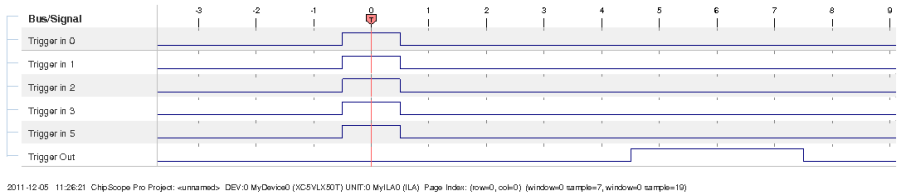


Figure 5.8: Chipscope capture of the trigger input output relation

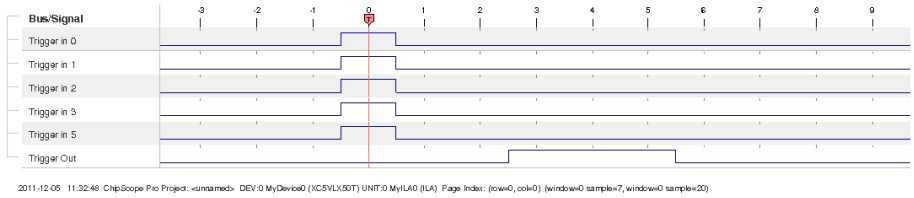


Figure 5.9: Chipscope capture of the trigger input output relation after synchronization with two flip-flops

[1]ChipScope is a JTAG device which acts as an internal logic analyzer while the FPGA is operating. An embedded, software logic analyzer

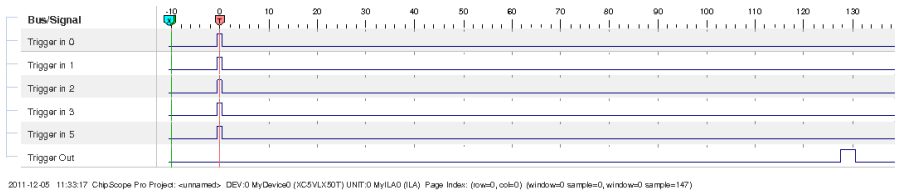


Figure 5.10: Chipscope capture of the delayed trigger

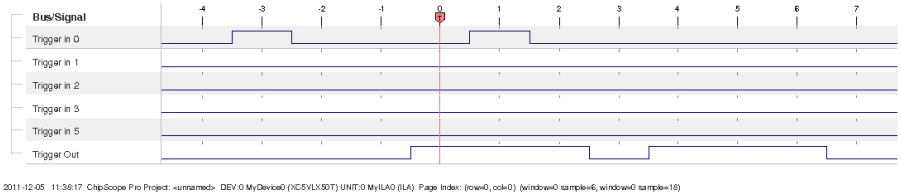
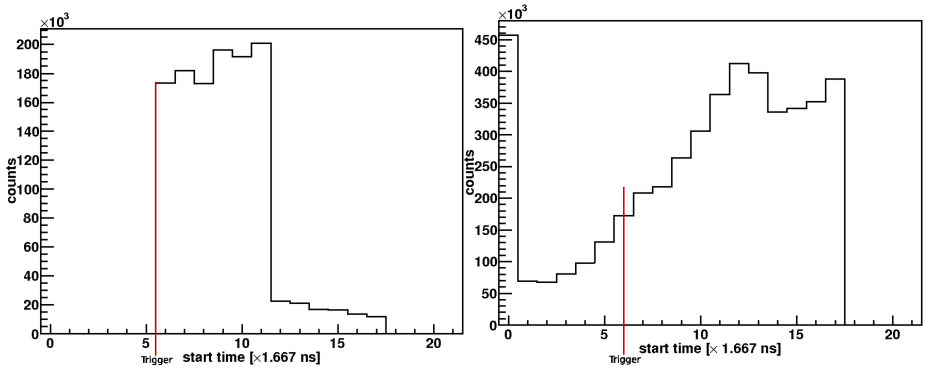
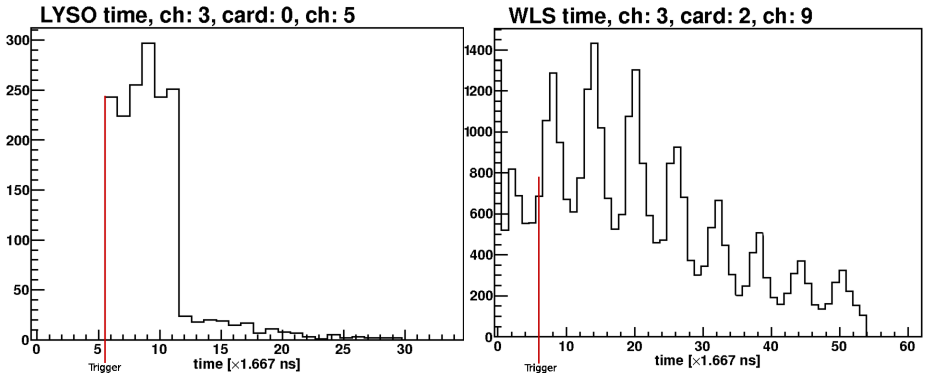


Figure 5.11: Two trigger input pulses separated by the dead-time of three clock cycles

From analyzing the timing information in the event frames the choice of using three clock periods (30 ns) trigger windows can be validated. This can be done by finding where in a frame the rising edge is. One frame from the deserializer consists of 6 bits clocked at 600 MHz, and from this information the start time of the Time-Over-Threshold pulse can be found with a 1.6667 ns resolution. In figure 5.12 this information is extracted and histogrammed in an off-line analysis. The histograms were captured from the off-line analysis program written by Michael Rissi. They show time difference between the trigger signal at 10 ns and the channel signal. From histogram (b) in figure 5.12 the spread of the WLS start time can be seen with respect to the rising edge of the trigger input from the central trigger unit. The histogram seems to form half of a bell curve, which indicates that the time jitter between the WLS and LYSO channels are larger than the trigger window. Histogram (d) shows the WLS start time distribution when the trigger window is extended to 90 ns, here the suspicion is confirmed. A full bell curve can be seen with a underlying noise (from 50 ns an outwards). The spikes seen are believed to be a grounding issue between the read-out cards and the pre-amplifier. The histograms for the LYSO channel shows that the event start times are situated in the middle of the three clock cycle long trigger window.



(a) LYSO channel with 30 ns opened window (b) WLS channel with 30 ns opened window

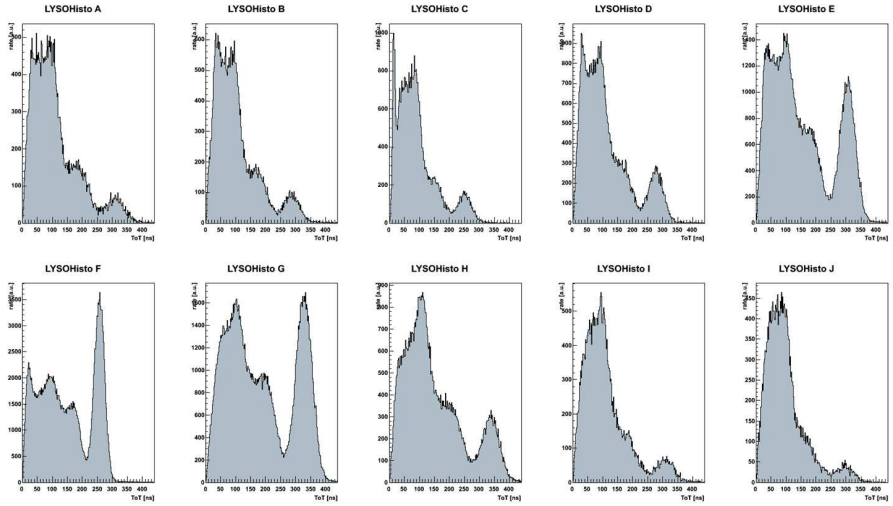


(c) LYSO channel with 50 ns opened window (d) WLS channel with 90 ns opened window

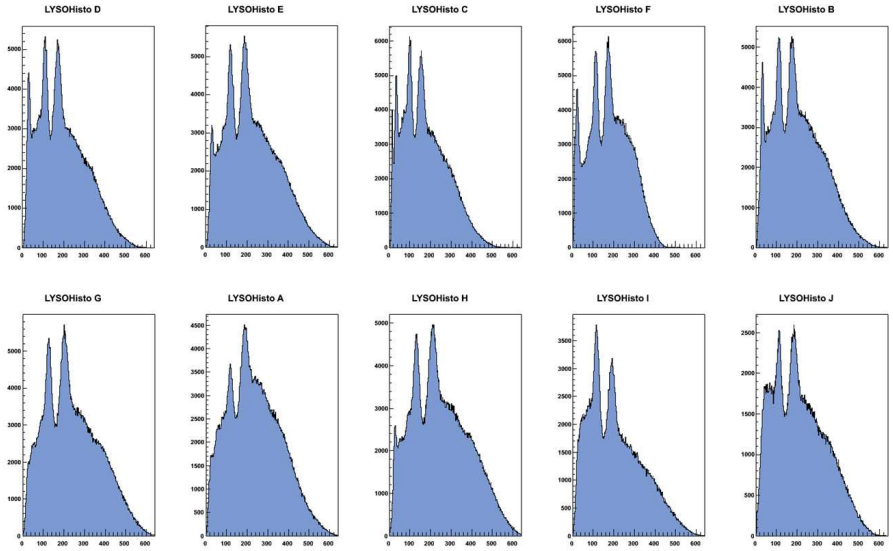
Figure 5.12: Histograms of event start time with respect to rising edge of the trigger captured on the read-out cards for 30 ns windows and 60 ns windows

5.2.3 Intrinsic and Coincident Spectras

In the late summer 2011 10 LYSO channels were set up in coincidence with a tagger crystal. Data was first taken with no source present and the Central Trigger set to OR triggering, and later with a Na-22 source with OR triggering. The Na-22 source has a β^+ decay mode and therefore emits positrons. The positron annihilate with an electron and emits back-to-back gammas of 511 keV. The intrinsic spectra is a bit more complicated, the Lutetium-176 has a β^- decay mode. In addition to emitting an electron from the β^- decay it emits three photons in cascade (89 keV, 203 keV and 306 keV). The three gammas has a probability to escape and interact in the neighboring crystal. This probability is dependent on the LYSO crystals geometry. As a result the β spectrum will be shifted according to the gamma interactions which happens in the same crystal.



(a) Coincident gamma spectra with a Na-22 source (positron emitter)



(b) Intrinsic gamma and beta spectra without a source with OR triggering

Figure 5.13: Energy spectras with AND triggering and a source and OR triggering with intrinsic

5.3 Asynchronous Trigger

The asynchronous trigger as currently implemented works when there is exactly one card for each module in coincidence. This was tested earlier in the project when half of a layer was implemented and one read-out card handled the LYSO channels and another handled the WLS channels. The asynchronous trigger was set-up to work in coincidence between the LYSO and the WLS. The

maximal achievable trigger rate was determined by setting up a computer to listen to the broadcasted trigger times packages sent from the trigger unit. In the trigger time packages there are two integers in the pre-amble which is the package number and number of trigger times this package, and from this the package loss and trigger rate can be estimated. The trigger times was represented by a 32 bit unsigned integer. The read-out cards and the central trigger unit was all located on the same Gigabit switch while the computer was located connected to this switch through a second switch. Up to a trigger rate of 5 Mcps the package loss was negligible, At 20 Mcps the package loss becomes significant. The result shown in figure 5.14

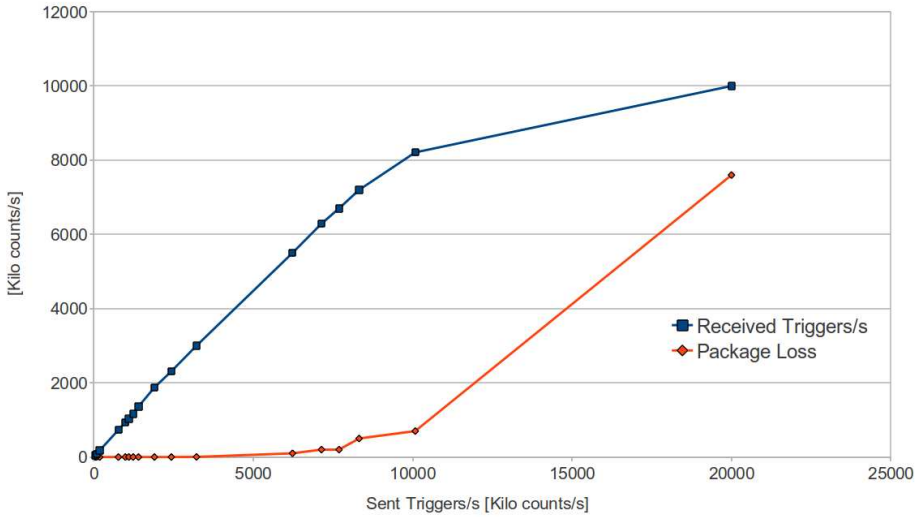


Figure 5.14: Rate measurement with 1 Gbps network using 32 bit trigger times

5.3.1 Preliminary Results for the Improved Asynchronous Trigger

The improved asynchronous trigger will synchronize the UDP stream by the use of time slots. To get the most out of the system it is beneficial that the time slots are as short as possible since there is a limit on how much payload there can be in each UDP package. To find an estimate on how small the time slots can be the time required to receive all the packages must be known. The delay between each package received by central trigger unit can be found using ChipScope connected to the central trigger and look at how many clock cycles it is between each data field when the read-out cards send out data on the same clock cycle. One clock cycle is 10 ns.

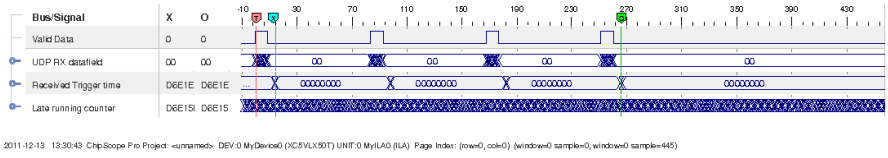


Figure 5.15: Delay between each received data field with 5 Byte data payload and frame size of 47 Bytes

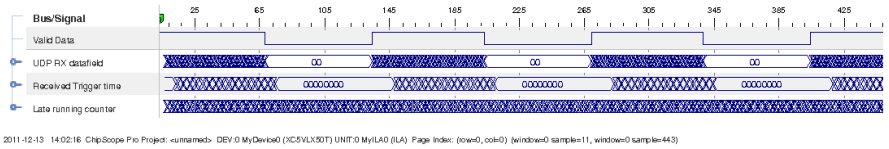


Figure 5.16: Delay between each received data field with 68 Byte payload and frame size of 90 Bytes

The delay between each data field (from figure 5.16 and figure 5.15) is 56 clock cycles when the packages has a 5 Byte payload (47 Bytes on the wire) and is 66 clock cycles when the payload is 68 bytes (frame is then 90 Bytes on the wire). Assuming $1\,\mu$s delay for a 1400 Byte package and twenty cards implemented, the time slots could be as low as

$$T_{\text{Timeslot}} = (20 * (100 + 1400) + 20 * 1400/4) * 10\,\text{ns} = 370\,\mu\text{s}$$

with time slots of $370\,\mu$s the maximum achievable rate from each card and maximum trigger rate will be $\frac{1400}{4} * \frac{1}{370\,\mu\text{s}} \approx 1\,$Mcps assuming 32 bit event and trigger times. It should be noted the event times only need as many bits such that they have a dynamic range equal to the time slot length with the system clock.
’

Chapter 6

Discussion

6.1 Timing and Synchronization

6.1.1 Clock distribution

The read-out cards have the capability of reaching 1 ns time resolution with the current design, although the current implementation is done with a 1.6667 ns time resolution. The goal was therefore to have an order of magnitude lower error from the clock distributions phase shift than the read-out cards timing resolution. When the resolution is 1 ns, the standard deviation for this would be the standard deviation for a square distribution, giving:

$$\frac{1\,ns}{12} = 83\,\text{ps}$$

Tests performed by Jo-Inge Busknes revealed a timing accuracy of 0.3 ns[2]. Although, this accuracy is the accuracy in which the length of the ToT pulse can be measured, and that accuracy is not affected by the clock distribution if the local jitter is the same. Because the PLL used internally is the same, the local jitter should not be effected much by using an external clock instead of the internal. In order to achieve a good coincidence time resolution, we need to improve the relative accuracy of events on different cards where the accuracy on the card itself is 1 ns. A systematic shift can be corrected in off-line post processing of the data, but will distort the trigger window online. From a time stamp with 1 ns resolution, the lowest possible coincidence window is a sliding window with a width of 2 ns. The results in table 5.1 shows that the phase shift between card 4 and 5 is at -1.97% with a standard deviation of 1.17% and is -1.79% with a standard deviation of 0.96% between card 1 and 2. The reason for the big shift at these two places is yet to be explained since the design is equal on all of them. The only notable difference is that card 1 and card 5 are placed in an awkward position close to a high-voltage power supply and the power supply for the read-out cards. A new test should be carried out when all the read-out cards are placed in racks. In figure 5.2 there is no clear trend of the phase relation diverging off from 0%. Since the delay line in the Virtex-5 (called Digital Clock Module) is adjustable in 0.04 ns increments it would be possible to adjust this even more, but the reason for the large variance must be identified.

6.1.2 Cabling and Reset

As more and more cards are added, cabling will be more of an issue. The reset scheme works as it is implemented now, but it would be preferable to have a more standardized cable for the trigger in, trigger out and reset lines. A proposed solution is to use RJ-45 connectors with CAT-5 twisted pair cables. These cables have 4 pairs and the RJ-45 connectors use little PCB land. Twisted pair cables will have a propagation delay because different pairs on the same cable travels different lengths. The propagation delay of a CAT-5e cable is in the order of 5 ns/m which must be accounted for when re-designing the cabling scheme.

6.1.3 Time-walk

The time-walk measurements shows that time-walk has to be considered if the coincidence detection window is to be shortened. Even though the measurements are not as precise as they should be, they show that there is a difference in arrival time due to time-walk in the order of 10-30 ns from high absorption events to low absorption, which is a problem when reconstruction of Compton events is wanted. This can be adjusted for by using pre-amplifiers with constant fraction discrimination or simply lowering the threshold thus allowing for more noise.

6.2 Synchronous Trigger

When the simulations differs from the measurements, either a timing issue or insufficient amount of test vectors exists. For the synchronous trigger the input-output relation is well defined and the amount of test vectors needed to test the worst case scenarios was few. ISE reported no timing violations for the logic, but there was however some undefined noise in the first implementation which was caused by wrong termination for the LVDS lines. This was however fixed before the final testing and was not included in the Results chapter. With the current set-up using three LVDS pairs for each card (trigger in, trigger out and reset) the expansion card on the central trigger unit will run out of IO ports. To overcome this, a patch board with strong buffers to drive a fanned-out reset and trigger-out line can be implemented. Figure 5.12, especially histogram (b) shows that a 30 ns window does not cover all associated wavelength shifter events. The 30 ns trigger window seems to cover the left side of a bell curve, and only covering around half of the correlated LYSO events. Histogram (d) shows the result with a 90 ns trigger window where it is possible to see the bell curve flatten out at around 50 ns. The reason for the uneven spikes in histogram (d) is unknown, but is believed to be a bug in the analysis software or a coupling to the 100 MHz clock. The 30 ns window for the LYSO channels seems to be a reasonable choice with the system as it is today. A wider trigger window only adds more background noise as it can be seen by comparing histogram (a) and (d).

6.3 Asynchronous Trigger

The first version of the asynchronous trigger issues had a poorly chosen design which needed a lot of memory to function properly, and as more cards that were added, the more memory it required. The next version, which is not yet implemented, is a better design with regards to flexibility, and memory usage. However, the pre-liminary tests of such a system (throughput and delays) shows that it is feasible.

6.4 Outlook

- Further adjust the phase correction on the read-out cards and investigate what effect the proximity to the power supply has
- Implement the revised version of the asynchronous trigger and find out if it has the stability and throughput such that the synchronous trigger can be replaced
- If the synchronous design stays with COMPET a patch board for fanning out reset signals to the read-out card from one pin at the FPGA is needed
- In order too use the high resolution time stamp from the deserializer time-walk corrections must be implemented
- Consider implementing PicoBlaze or MicroBlaze for slow control if more functions are wanted. Extending Finite State Machines for each new command can get tiresome

Chapter 7

Conclusion

The aim for this master thesis was to design, implement and test a synchronous and an asynchronous approach for a centralized trigger system for COMPET. To improve timing accuracy the read-out cards were synchronized with the use of a Digital Clock Management unit and a Phase Locked Loop. This proved to give adequate synchronization between each read-out card. There is still room for improvement by adjusting the delays on each card individually, but this will increase system complexity by needing an unique bit file for every read-out card. The synchronous scheme was implemented and tested and proved to be a very stable solution, but with a limited coincidence time resolution and the drawback of having two LVDS pairs for each new read-out card connected. The asynchronous trigger scheme showed promising results, but was in this version limited to a small system. However, the asynchronous triggers ability to have sub-clock coincidence time resolution and the added flexibility shows that this is a viable concept. The pre-liminary results showed that the revised version should be able to handle trigger rates in order of 5 Mcps while keeping the memory usage at an acceptable level.

Bibliography

[1] Nema nu 2-2007 performance measurements of positron emission tomographs.

[2] Jo Inge Buskenes. Readout electronics for a novel animal pet scanner using field programmable gate arrays. Master's thesis, University Of Oslo, 2010.

[3] Bradley P. Thomas et al. High-resolution 7t mri of the human hippocampus in vivo. *Journal of Magnetid Resonance Imaging*, 2008.

[4] C. Woody et al. Ratcap: a small, head-mounted pet tomograph for imaging the brain of an awake rat. *Nuclear instruments and methods in physics research A 527 166:170*, 2004.

[5] Matthias Hofmann et al. Mri-based attenuation correction for pet/mri: A novel approach combining pattern recognition and atlas registration. *The Journal of Nuclear Medicine*, 2008.

[6] S. Hansen et al. Low-cost data acquisition card for schoolnetwork cosmic ray detector. *IEEE Trans.Nucl.Sci. 51 (2004) 926-930*, 2004.

[7] V.R. BOM et al. Comparison of pinhole collimator materials based on sensitivity equivalence. *Physics in Medicine and Biology*, 2011.

[8] Thomas Frach. Digital silicon photomultiplier for low light detection in high-energy physics. CERN Detector seminar.

[9] Hamamatsu. Mppc, http://sales.hamamatsu.com/en/products/solid-state-division/si-photodiode-series/mppc.php, November 2011.

[10] editorial committee Hamamatsu Photonics. *Photomultiplier Tube, basics and applications*. Hamamatsu, 2006.

[11] Xilinx Inc. Clocking debug guide, February 2011.

[12] Rogers GA et al. Ingvar M, Eriksson L. Rapid feasibility studies of tracers for positron emission tomography; high resolution pet in small animals with kinetic analysis. *J. Cereb Blood Flow Metab 11:926-31*, 1991.

[13] E. M. Leiholdt jr J. M. Boone J. T. Bushberg, J.A. Seibert. *The Essential Physics of Medical Imaging*. Williams & Wilkins, 2002.

[14] E. Bolle K.E. Hines, M. Rissi and O. RÃ¸hne. Highly scalable asynchronous trigger system for pet using udp/ip. In *2011 Nuclear Science Symposium and Medical Imaging Conference*.

[15] R. Lecomte. Novel detector technology for clinical pet. *Eur. J. Nucl. Med. Mol. Imaging, vol. 36, no. Suppl. 1, pp. 69-85, 200*, 2009.

[16] Dr. William R. Leo. *Techniques for Nuclear and Particle Physics Experiments*. Springer, 1987.

[17] Hoffman EJ. Levin CS. Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution. *Phys Med Biol 44:781-99*, 1999.

[18] Hoffman EJ. Levin CS. Current trends in preclinical pet system design. *PET Clin 2 125-160*, 2007.

[19] Charles L. Melcher. Scintillation crystal for pet. *The Journal of Nuclear Medicine, vol. 41*, 2000.

[20] D.Volgyes J.G. Bjaalie O.Dorholt K.E. Hines O.Rohne M.Rissi, E.Bolle. Characterization of a high resolution and high sensitivity pre-clinical pet scanner with 3d event reconstruction. *Nuclear Instruments & Methods In Physics Research Section A, doi:10.1016/j.nima.2011.10.042*, 2011.

[21] J.G. Bjaalied J.I. Buskenes O. Dorholt O. Røhne A. Skretting S. Stapnes M.Rissi, E.Bolle. Compet high resolution and high sensitivity pet scanner with novel readout concept: Setup and simulations. *Nuclear Instruments and Methods in Physics Research Section A: doi:10.1016/j.nima.2010.12.097* |, 2011.

[22] Yuri Musienko. State of the art silicone photomultipliers. CERN SiPM workshop, 2001.

[23] Andrew Webb Nadine Barrie Smith. *Introduction to Medical Imaging*. Cambridge University Press, 2011.

[24] Alexandros Stamatakis Nikolaos Alachiotis, Simon A. Berger. Efficient pc-fpga communication over gigabit ethernet. *10th IEEE International Conference on Computer and Information Technology (CIT 2010)*, 2010.

[25] P.Beltrame. The ax-pet demonstrator. design, construction and characterization. *Nuclear Instruments & Methods In Physics Research Section A, 10.1016/j.nima.2011.06.059*, 2011.

[26] P.Beltrame. Demonstration of an axial pet concept for brain and small animal imaging. *Nuclear Instruments & Methods In Physics Research Section A, doi:10.1016/j.nima.2010.07.01*, 2011.

[27] David W. Townsend Michael N. Maisey Peter E. Valk, Dale L. Bailey. *Positron Emission Tomography Basic Sciende and Clinical Practice*. Springer, 2003.

[28] Michael E. Phelps Simon R. Cherry, James A. Sorenson. *Physics in Nuclear Medicin, third edition*. Springer, 2003.

[29] Laforest R. Tai YC. Instrumentation aspects of animal pet. *Annu Rev Biomed Eng, 7:255-85*, 2005.

[30] Osama Mawlawi Waheeda Sureshbabu. Pet/ct imaging artifacts. *The Journal of Nuclear Medicine Technology*, 2005.

Appendix A

Getting started

A.1 Downloading the Project Files

The VHDL project and the asynchronous software trigger are both contained within the same Git repository. During development the Git repository has been located on a UiO server, but the current version is available from a public github repository. To get the latest version from the UiO server:

1. Install Git via your package manager on your Linux system, if you are on a windows system Cygwin can provide you the proper environment
2. In a shell type "git clone si:/mn/felt/epflab/pet/git/trigger.git" and the project while be downloaded to the folder you are standing
3. If you do not have access to the /mn/felt/epflab folder on si, and are unable to gain access you have to create a github account and set-up a SSH key. A guide is located here: help.github.com/win-set-up-git, the guide is made for windows, but since they use Cygwin the guide applies for Linux systems as well. To clone the project from github type "git clone git@github.com:kimei/CTU.git" and the project folders are downloaded to the folder you are currently standing in.
4. The top VHDL project file is located under Trigger/HW/HDL/top_sxt/top_sxt/top_sxt.xise and is compatible with ISE 13.1
5. An ISE license is only required to generate a bit file but not for browsing the project, but a license for ISE at UiO is available on 5370@pordoi.ifi.uio.no.

A.2 Simulations and off-line setting

The simulations can be started from ISE by changing the environment view from implementation to simulation, select which test bench to run and click simulate with Modelsim (modelsim must be installed separately). The test bench for the synchronous trigger needs a bit vector file to read. This can be generated by going to Sync_trigger/sim/test_vectors folder and running the python script generate.py. The output file is named vectors.dat and is placed

in the same folder as the script. The vectors.dat file must be moved to the top_sxt folder for the test bench to find it.
To change the grouping and number of read-out cards currently connected, you have to find the constants.vhd file located under the Shared folder. Here you can set the number of read-out cards in total, number of modules and number of read-out cards in each module.

A.3 Slow control through UDP

The Central Trigger Unit accepts slow control messages which are sent from IP address 192.168.1.64, the IP address of the Trigger is 192.168.1.63 and is connected to PHY 1. The messages should be sent to destination port 5005, the source port is "don't care" (all this is changeable in the instantiation of the UDP core in the EMAC unit). The allowed commands are defined in the receiver_control.vhd file and are ASCII letters:

- 'r' (72h) resets all the read-out cards without resetting the central trigger, the trigger will reply with 02h&'r' if successful
- 's' (73h) enables random triggering at a set frequency of 10 kHz, the trigger will reply with 02h&'s' if successful
- 't' (74h) enables AND trigger mode and disables the random triggering and delayed window triggering (if enabled), the trigger will reply 02h&'t' if successful
- 'u' (75h) enables delayed window random trigger, the trigger will reply with 02h&'u' if successful
- 'o' (6Fh) enables OR trigger mode and disables the random triggering and delayed window triggering (if enabled), the trigger will reply with 02h&'o' if successful
- 'm'& 3 Byte bit vector (6DXXXXXXh) will set a input trigger mask, all '1's in the three bytes enables all the read-out cards. 1111 1110 1111 1111 1111 1111 will disable card number 1 and 1111 1111 1111 1111 1111 1101 will disable card number 18. The trigger card will answer with 02h and the received bit vector if successful
- 'n'& 1 Byte bit vector (6EXh) will set a module mask, where 1111 1101 will disable module 2. The trigger will answer with 02h and the received bit vector if successful
- when the central trigger unit is reset it will start up with AND triggering and all trigger inputs enabled

A.4 Rate Counter

Rate packages arrive every second with information about number of incoming triggers from each read-out card and number coincidence triggers out from

the central trigger. The trigger packages are sent to ip: 192.168.1.64 on destination port: 21846 and source port: 21847. They are all 129 bytes long where the first byte is 01 in HEX. The rest of the message is 32 4 byte integers in small endian where the first integer is the rate of card number 1, second from card 2 etc. The last integer is the coincidence trigger rate from the trigger unit. The received package can be read with the unpack() function in python.

To test this a simple UDP sender can be written in Python as this:

```
import socket

UDPIP="192.168.1.63"
UDPPORT=5005
MESSAGE="r"

sock = socket.socket( socket.AF_INET,
                      socket.SOCK_DGRAM )
sock.sendto( MESSAGE, (UDPIP, UDPPORT) )
```

A receiver to check the received message can be written in Python as:

```
import socket

UDPIP="192.168.1.64"
UDPPORT=21846

sock = socket.socket( socket.AF_INET, # Internet
                      socket.SOCK_DGRAM ) # UDP
sock.bind( (UDPIP,UDPPORT) )

while True:
    data, addr = sock.recvfrom( 1024 ) # buffer size is 1024 bytes
    print "received message:", data
```

A.5 Networking for the asynchronous trigger

Packages sent to the asynchronous trigger must be sent to IP-address 192.168.1.62 on PHY2 on port 16962. The senders IP address is must start from 192.168.1.34 and onwards. IP address ending at 34 will be assigned as card 1 and 35 will be assigned as card 2 and so on. Grouping of the cards is at the moment done in async_trigger.vhd. The datagram must start with a pre-amble with the ASCII characters ":)" followed by a 2 byte integer which is the total number of 4 byte event times in the UDP package. The event times must be BIG-endian. The Central Trigger Unit will send trigger times with the same pre-amble, but trigger times are in SMALL-endian as a UDP broadcast to port 21846.

Appendix B

Synchronous Trigger Simulation output from ModelSIM

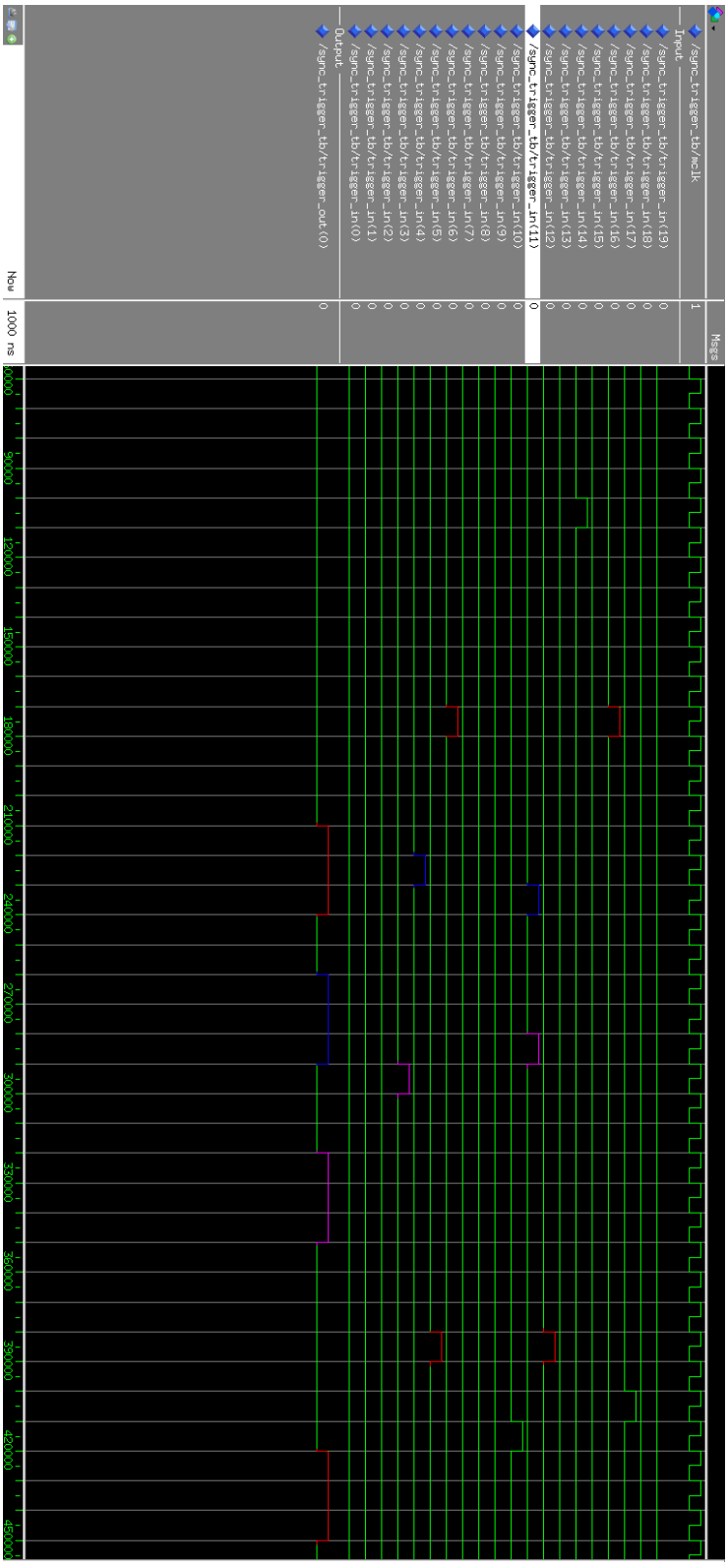


Figure B.1: Simulation from ModelSIM with twenty read-out cards in four modules in coincidence with random lower-rate data

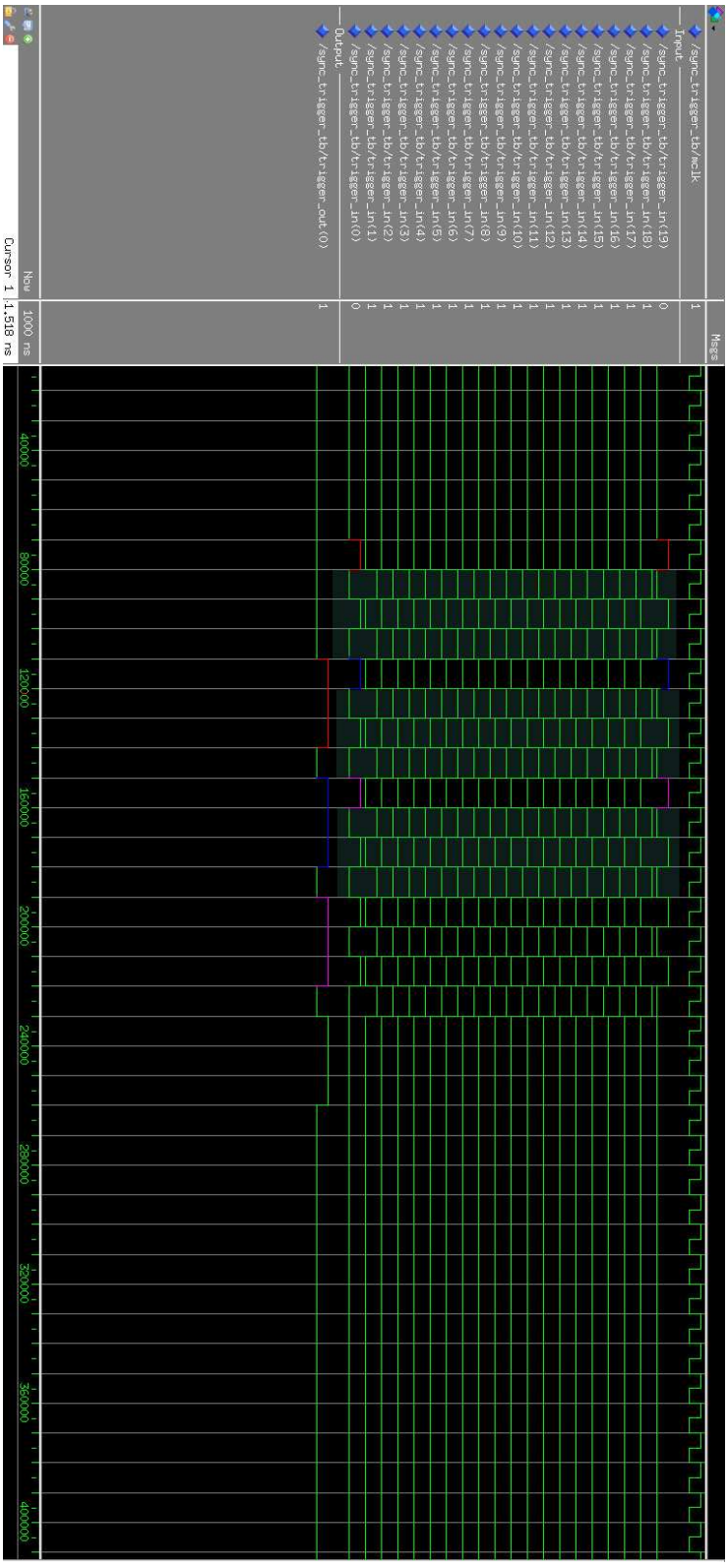


Figure B.2: Simulation from ModelSIM with twenty read-out cards in four modules in coincidence with worst case input data

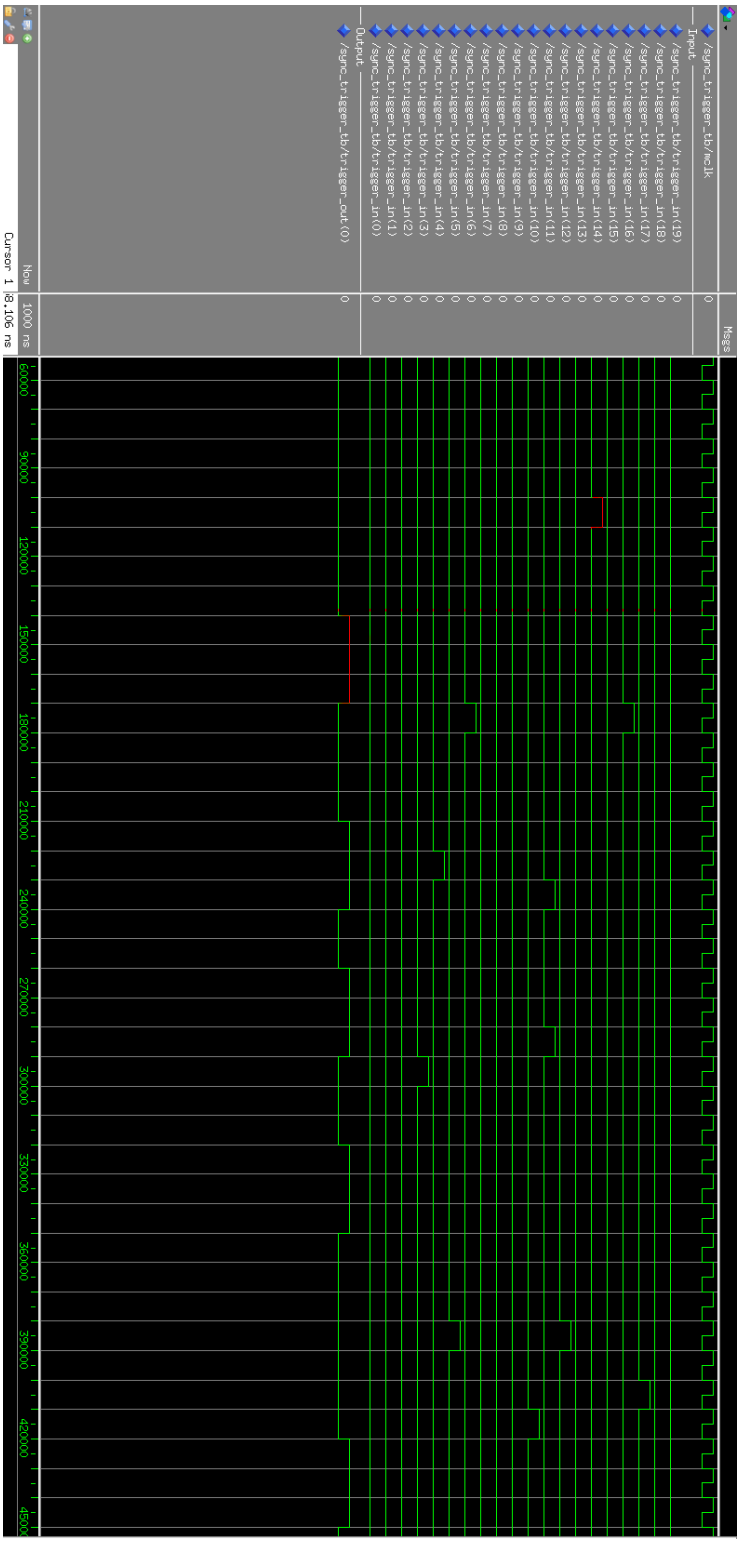


Figure B.3: Simulation from ModelSIM with twenty read-out cards in four modules with OR-triggering enabled

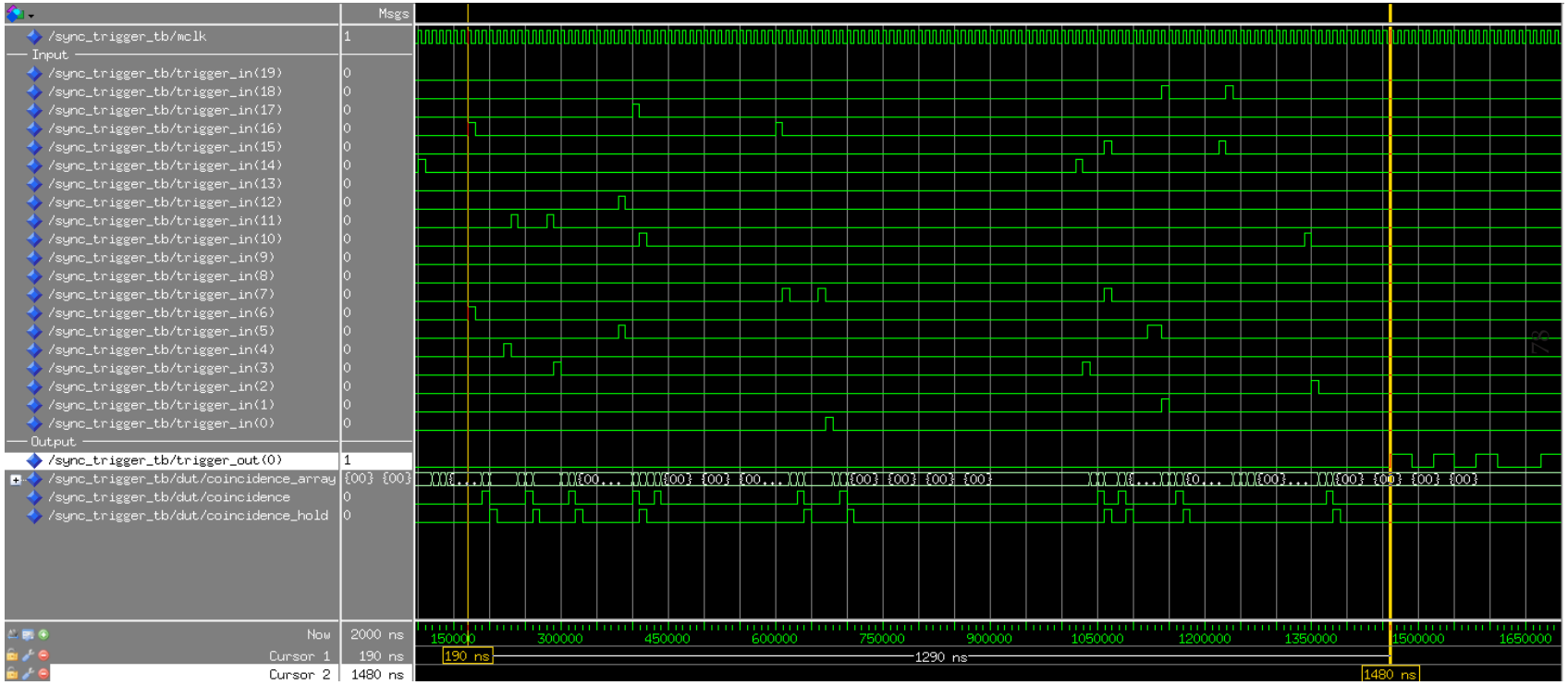


Figure B.4: Simulation from ModelSIM with delayed trigger enabled

Appendix C

Pin-Out for the Expansion Connector

Table C.1: Pin-out for Jx-1

Virtex-5 Pin#	Net Name	EXP Connector Pin # (JX1)		Net Name	Virtex-5 Pin#
H32	EXP1_SE_IO_0	2	1	EXP1_SE_IO_1	R34
G33	EXP1_SE_IO_2	4	3	EXP1_SE_IO_3	P32
-	**2.5V**	6	5	**2.5V**	-
G32	EXP1_SE_IO_4	8	7	EXP1_SE_IO_5	N33
F34	EXP1_SE_IO_6	10	9	EXP1_SE_IO_7	N34
-	**2.5V**	12	11	**2.5V**	**-**
F33	EXP1_SE_IO_8	14	13	EXP1_SE_IO_9	N32
E34	EXP1_SE_IO_10	16	15	EXP1_SE_IO_11	M33
-	**2.5V**	18	17	**2.5V**	-
E33	EXP1_SE_IO_12	20	19	EXP1_SE_IO_13	M32
E32	EXP1_SE_IO_14	22	21	EXP1_SE_IO_15	L33
-	**2.5V**	24	23	**2.5V**	-
D34	EXP1_SE_IO_16	26	25	EXP1_SE_IO_17	L34
D32	EXP1_SE_IO_18	28	27	EXP1_SE_IO_19	K33
-	**2.5V**	30	29	**2.5V**	-
C34	EXP1_SE_IO_20	32	31	EXP1_SE_IO_21	K34
C33	EXP1_SE_IO_22	34	33	EXP1_SE_IO_23	K32
-	**2.5V**	36	35	**2.5V**	-
C32	EXP1_SE_IO_24	38	37	EXP1_SE_IO_25	J34
B33	EXP1_SE_IO_26	40	39	EXP1_SE_IO_27	J32
L19	EXP1_DIFF_CLK_IN_p	42	41	EXP1_SE_IO_28	H34
K19	EXP1_DIFF_CLK_IN_n	44	43	EXP1_SE_CLK_IN	H14
-	**GND**	46	45	**GND**	-
A33	EXP1_SE_IO_30	48	47	EXP1_SE_IO_29	H33
B32	EXP1_SE_IO_31	50	49	EXP1_SE_CLK_OUT	R33
-	**GND**	52	51	**GND**	**-**
K1	EXP1_MGT_RX_DIFF_p20	54	53	EXP1_MGT_TX_DIFF_p21	L2
J1	EXP1_MGT_RX_DIFF_n20	56	55	EXP1_MGT_TX_DIFF_n21	K2
-	**GND**	58	57	**GND**	**-**
G11	EXP1_DIFF_p18	60	59	EXP1_SE_IO_32	P34
G12	EXP1_DIFF_n18	62	61	EXP1_SE_IO_33	T33

-	**GND**	64	63	**GND**	**-**
E12	EXP1_DIFF_p16	66	65	EXP1_DIFF_p19	L10
E13	EXP1_DIFF_n16	68	67	EXP1_DIFF_n19	L11
-	**GND**	70	69	**GND**	**-**
T10	EXP1_DIFF_CLK_OUT_p	72	71	EXP1_DIFF_p17	F13
T11	EXP1_DIFF_CLK_OUT_n	74	73	EXP1_DIFF_n17	G13
-	**GND**	76	75	**GND**	**-**
D12	EXP1_DIFF_p14	78	77	EXP1_DIFF_p15	K11
C12	EXP1_DIFF_n14	80	79	EXP1_DIFF_n15	J11
A13	EXP1_DIFF_p12	82	81	EXP1_DIFF_p13	D11
B12	EXP1_DIFF_n12	84	83	EXP1_DIFF_n13	D10
-	**3.3V**	86	85	**3.3V**	**-**
K8	EXP1_RCLK_DIFF_p10	88	87	EXP1_DIFF_p11	F10
K9	EXP1_RCLK_DIFF_n10	90	89	EXP1_DIFF_n11	G10
-	**3.3V**	92	91	**3.3V**	**-**
B13	EXP1_DIFF_p8	94	93	EXP1_DIFF_p9	H10
C13	EXP1_DIFF_n8	96	95	EXP1_DIFF_n9	H9
-	**3.3V**	98	97	**3.3V**	**-**
F11	EXP1_DIFF_p6	100	99	EXP1_DIFF_p7	J10
E11	EXP1_DIFF_n6	102	101	EXP1_DIFF_n7	J9
-	**3.3V**	104	103	**3.3V**	**-**
E9	EXP1_DIFF_p4	106	105	EXP1_DIFF_p5	M10
E8	EXP1_DIFF_n4	108	107	EXP1_DIFF_n5	L9
-	**3.3V**	110	109	**3.3V**	**-**
F9	EXP1_DIFF_p2	112	111	EXP1_DIFF_p3	M8
F8	EXP1_DIFF_n2	114	113	EXP1_DIFF_n3	L8
-	**3.3V**	116	115	**3.3V**	**-**
G8	EXP1_DIFF_p0	118	117	EXP1_DIFF_p1	N10
H8	EXP1_DIFF_n0	120	119	EXP1_DIFF_n1	N9
-	**GND**	122	121	**GND**	**-**
-	**GND**	124	123	**GND**	**-**
-	**GND**	126	125	**GND**	**-**
-	**GND**	128	127	**GND**	**-**
-	**GND**	130	129	**GND**	**-**
-	**GND**	132	131	**GND**	**-**

Table C.2: Pin-Out for JX-2

Virtex-5 Pin#	**Net Name**	**EXP nector # (JX2)**	**Con- Pin**	**Net Name**	**Virtex-5 Pin#**
V32	EXP2_SE_IO_0	2	1	EXP2_SE_IO_1	AD32
V34	EXP2_SE_IO_2	4	3	EXP2_SE_IO_3	AE34
-	**2.5V**	6	5	**2.5V**	-
AB32	EXP2_SE_IO_20	32	31	EXP2_SE_IO_21	AH32
AB33	EXP2_SE_IO_22	34	33	EXP2_SE_IO_23	AJ34
-	**2.5V**	36	35	**2.5V**	-
AC32	EXP2_SE_IO_24	38	37	EXP2_SE_IO_25	AK33
AC34	EXP2_SE_IO_26	40	39	EXP2_SE_IO_27	AK34
AH15	EXP2_DIFF_CLK_IN_p	42	41	EXP2_SE_IO_28	AK32
AG15	EXP2_DIFF_CLK_IN_n	44	43	EXP2_SE_CLK_IN	AG21
-	**GND**	46	45	**GND**	-
AD34	EXP2_SE_IO_30	48	47	EXP2_SE_IO_29	AM33
AC33	EXP2_SE_IO_31	50	49	EXP2_SE_CLK_OUT	AJ32
-	**GND**	52	51	**GND**	-
AG10	EXP2_DIFF_p20	54	53	EXP2_DIFF_p21	AN14

AG11	EXP2_DIFF_n20	56	55	EXP2_DIFF_n21	AP14
-	**GND**	58	57	**GND**	-
AF11	EXP2_DIFF_p18	60	59	EXP2_SE_IO_32	AL34
AE11	EXP2_DIFF_n18	62	61	EXP2_SE_IO_33	AL33
-	**GND**	64	63	**GND**	-
AG8	EXP2_DIFF_p16	66	65	EXP2_DIFF_p19	AN13
AH8	EXP2_DIFF_n16	68	67	EXP2_DIFF_n19	AM13
-	**GND**	70	69	**GND**	-
AJ7	EXP2_DIFF_CLK_OUT_p	72	71	EXP2_DIFF_p17	AP12
AJ6	EXP2_DIFF_CLK_OUT_n	74	73	EXP2_DIFF_n17	AN12
-	**GND**	76	75	**GND**	-
AF8	EXP2_DIFF_p14	78	77	EXP2_DIFF_p15	AM12
AE9	EXP2_DIFF_n14	80	79	EXP2_DIFF_n15	AM11
AE8	EXP2_DIFF_p12	82	81	EXP2_DIFF_p13	AL11
AD9	EXP2_DIFF_n12	84	83	EXP2_DIFF_n13	AL10
-	**3.3V**	86	85	**3.3V**	-
AD10	EXP2_RCLK_DIFF_p10	88	87	EXP2_DIFF_p11	AK11
AD11	EXP2_RCLK_DIFF_n10	90	89	EXP2_DIFF_n11	AJ11
-	**3.3V**	92	91	**3.3V**	**-**
AC8	EXP2_DIFF_p8	94	93	EXP2_DIFF_p9	AK8
AB8	EXP2_DIFF_n8	96	95	EXP2_DIFF_n9	AK9
-	**3.3V**	98	97	**3.3V**	**-**
AA5	EXP2_DIFF_p6	100	99	EXP2_DIFF_p7	AJ9
AB5	EXP2_DIFF_n6	102	101	EXP2_DIFF_n7	AJ10
-	**3.3V**	104	103	**3.3V**	**-**
AB6	EXP2_DIFF_p4	106	105	EXP2_DIFF_p5	AH9
AB7	EXP2_DIFF_n4	108	107	EXP2_DIFF_n5	AH10
-	**3.3V**	110	109	**3.3V**	**-**
AA8	EXP2_DIFF_p2	112	111	EXP2_DIFF_p3	AF9
AA9	EXP2_DIFF_n2	114	113	EXP2_DIFF_n3	AF10
-	**3.3V**	116	115	**3.3V**	**-**
AB10	EXP2_DIFF_p0	118	117	EXP2_DIFF_p1	AC10
AA10	EXP2_DIFF_n0	120	119	EXP2_DIFF_n1	AC9
-	**GND**	122	121	**GND**	**-**
-	**GND**	124	123	**GND**	**-**
-	**GND**	126	125	**GND**	**-**
-	**GND**	128	127	**GND**	**-**
-	**GND**	130	129	**GND**	**-**
-	**GND**	132	131	**GND**	**-**

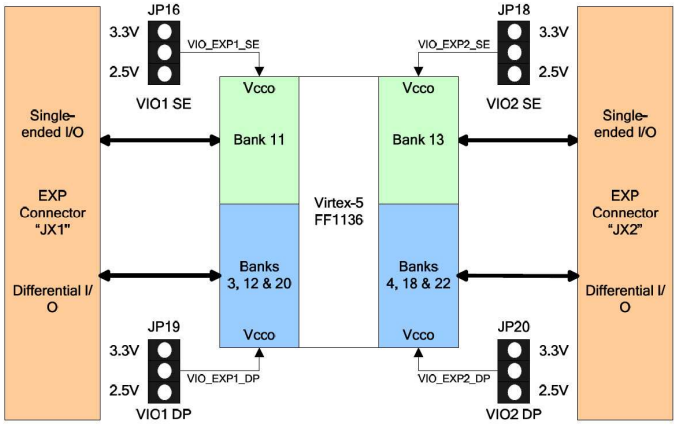


Figure C.1: Block diagram of JX 1 and 2s relation to the FPGA input/output banks

Appendix D

IEEE NSS/MIC Conference Record Paper

Highly scalable asynchronous trigger system for PET using UDP/IP

K.E. Hines, M. Rissi, E. Bolle, and O. Røhne

***Abstract*—Asynchronous coincidence triggering in PET can be realized by using UDP over 1 Gbps Ethernet. This will bypass problems related to point-to-point connections and tree-structures in real-time coincidence trigger units. By using UDP over Ethernet, the trigger system is easy to scale, and off-the-shelf components can be used. However, due to unknown and network load dependent delay times of UDP communication packages, and due to possible package losses, the system must react tolerantly towards these occurrences. In our FPGA based readout system, event packages (consisting of an event time, energy, and channel number) are stored in a FIFO. The event times are sent to a Central Trigger Unit (CTU) through a UDP/IP link where trigger decisions are made and broadcasted back to the readout cards. The broadcasted trigger times are compared to existing event times to decide if the event package is discarded or sent to a computer farm. Since the UDP packages with event times and trigger times are self-contained there is no need for sorting algorithms usually associated with UDP networking. The UDP/IP stack is implemented solely in hardware without using large amount of resources.**

I. Introduction

CURRENT PET systems with thousands-to ten thousands of crystals quickly grow in complexity and cost when real-time trigger systems with point-to-point connections or tree-like structures are used. In order to reduce the complexity, we are investigating an easy to scale trigger system for our COMPET preclinical PET scanner, [2], [3] with off-the-shelf components based on UDP/IP.

II. COMPET Geometry and Readout

The COMPET scanner consists of four modules each with five layers of LYSO crystals ($3 \times 2 \times 80\,\mathrm{mm}^3$) interleaved with wavelength shifters ($1 \times 3 \times 80\,\mathrm{mm}^3$), see fig. 1.

The scintillation light is registered with Geiger Mode Avalanche Photodiodes (GAPDs). Their signal is amplified and shaped such that it shows a fast rise time and a linear decay. Using a discriminator, the resulting digital signal has a length proportional to the incident scintillation light. This signal is fed to the deserializing input of a Xilinx Virtex 5 FPGA, where its length and rise time are extracted, see fig. 2.

The event package containing Time-Over-Threshold (energy), time stamp (event time), channel and event number is stored in a FIFO. If the time stamp of the event package is equal to a trigger time received from the Central Trigger Unit (CTU) the event package is stored. If no coincidence trigger for a certain event is received within the maximal turn-around time of the trigger network, the corresponding event will be discarded. A full system will consist of twenty readout cards and one CTU reading out a total of 480 wavelength shifter channels and 600 LYSO channels. The synchronized system clock is distributed as daisy chain from the CTU with phase correction in each node. The CTU is also responsible for synchronous system reset, such that the readout cards are synchronized within 200 ps.

Manuscript received November 14, 2011

K.E. Hines (kimei@fys.uio.no),M. Rissi, E. Bolle and O. Røhne are with the Universitetet i Oslo, Norway.

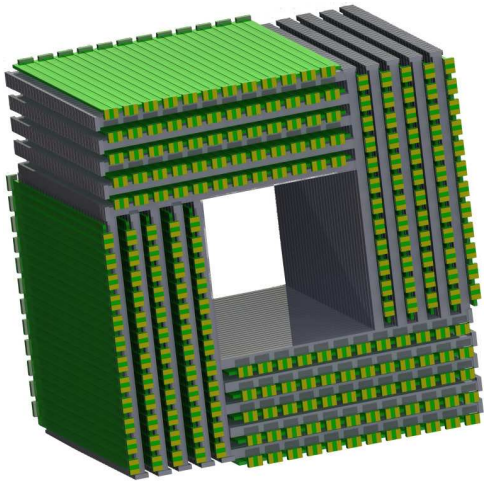


Fig. 1. COMPET block geometry

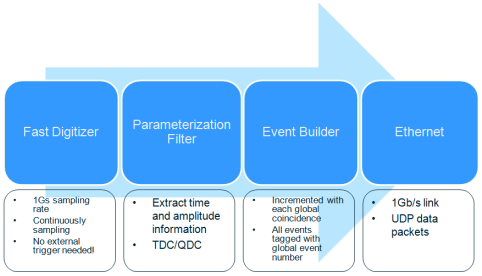


Fig. 2. The data flow inside the readout FPGA. From the ToT pulses, the energy and event time is extracted, and the events are sorted by event time. Figure from [3]

III. Implementation

A. Trigger Logic In The Readout Cards

Every $100\,\mu$s a UDP package is sent to the central trigger unit containing the event times from the last $100\,\mu$s. A third FIFO holds the trigger times that are received as broadcasts from the CTU. The received trigger times are compared to a late running counter which is synchronized on all the readout cards. The value of this late running counter is equal to the system time minus the trigger decision time of $120\,\mu$s when

using $100\,\mu$s time slots. If the late running counter is equal to a stored event time, the associated event will be discarded. If an event time is equal to a trigger time, the event will be sent and stored to disk in a computer farm. If a stored trigger time is equal to the late running counter, which means that the trigger time did not coincide with a stored event on this card, the trigger time will be discarded, see fig. 3.

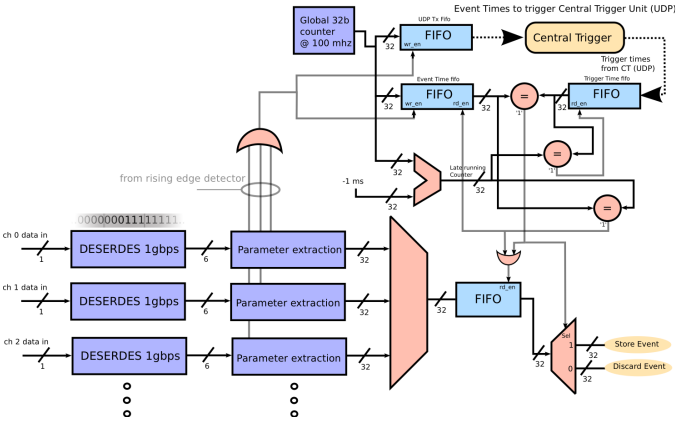


Fig. 3. Simplified schematic showing the trigger logic on a readout card

B. Trigger Logic On The Central Trigger Unit

Readout cards send their event times every $100\,\mu$s. Since the readout cards are synchronized they send out packages at the exact same clock cycle, the network switch receives the twenty UDP packages at the same time and sends them one by one to the CTU. Such, the UDP streams are naturally synchronized and package loss can easily be detected.

The maximum network load can be set by having a maximum number of event times in one package. It also reduces the memory usage on the CTU since event times only get stored the duration of one time slot.

The latency between each package received by the CTU is measured to be $1\,\mu$s with 512 Bytes of payload in each package. This gives the trigger $100\mu\text{s} - 1 * 20\,\mu\text{s} = 80\,\mu\text{s}$ to make trigger decisions from the received event times with $100\,\mu$s time slots.

The event times in the event time packages are all put in a unique FIFO. The sender's IP address decides which FIFO the event times are written to. When an event time package has been received from all readout cards marked as active, the FIFOs contains the event times from all the readout cards sorted such that the first event time is on the output of each FIFO and the latest is at the end. The twenty FIFOs are then grouped such that the content of the FIFOs in a group comes from the readout cards in a geometrical module. In COMPET there are four different modules which gives us four groups. There are five readout cards in each module which gives us five FIFOs in each group. At every clock cycle the FIFO with the earliest time on the output is read. This happens at the same time as the earliest event time from each group is compared for coincidences. A coincidence simply means that two or more times are equal. If they are equal, the coincidence time is written to a coincidence time FIFO and all event time FIFOs with this value on the output is read out. When all twenty event time FIFOs are empty, the content of the coincidence time FIFO is sent as a UDP broadcast to all readout cards.

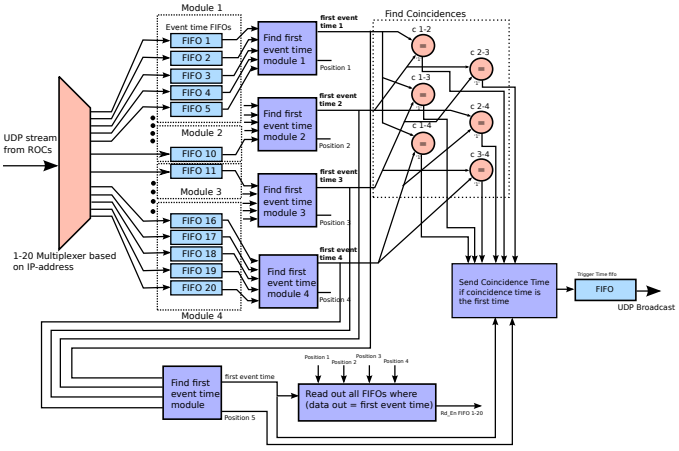


Fig. 4. Simplified schematic showing the trigger logic on the Central Trigger Unit

The CTU must sense overflows in the counter and treat overflows logically. In order to adjust to an unknown number of readout cards in each group, the inactive UDP streams can be masked out on the fly, or a statically set trigger mask can be applied. Slow control of the CTU and the readout cards is also done through UDP.

C. Data structure and timing

The coarse time stamp used for event times and trigger times are each 32 bits. At 100 MHz this gives a dynamic range of 40 seconds, which is highly redundant and will be reduced in the final implementation. The number of bits needed depends on the resolution and length of a time slot. With a timing resolution of 1 ns and $100\,\mu$s time slots, 14 bits is needed to get a dynamic range larger than $100\,\mu$s. If a high resolution time stamp exists it can be used for trigger information without changing any trigger logic. In the COMPET readout system the time stamp from the 1 Gbps deserializer inputs can, and in the end, will be used for trigger decisions. A theoretical limit on trigger rates is network dependent, and for a 1 Gbps switched network using 32 bits for each event time, the theoretical limit is $1\,\text{Gbps}/32\text{b} \approx 30\,\text{Mcps}$. This assumes that the header size is negligible to the payload size, meaning that high number of events are sent with each package.

The proposed scheme makes use of time frames. In the first implementation a time frame is $100\,\mu s$ and a maximum number of events per time frame is set to 128. This limits the event rate from each card to approximately 1 Mcps, and gives the CTU $80\mu s$ to make trigger decisions, which is redundant since the time to sort singles from coincidences is $N_{\text{max. number of event times in one package}} * T_{\text{clk period CTU}}$. To achieve higher rates the length of a time frame and the maximum number of event times per package must optimized. The longer

the time frame, the higher is the need to buffer events on the readout cards. The buffering is implemented with the means of FIFOs, where the largest uses the on-board DDR2 memory.

IV. Results

To test the network capabilities, the CTU was connected to two readout cards. The two readout cards sent random times at a specific rate, and the times were always the same from both cards so that it would be treated as a coincidence in the CTU. Since the trigger times are broadcasted on the network, a computer was set up to listen to the broadcast messages and extract the package number to detect lost packages and the number of trigger times present in the package. We observed that the rate was consistent with what the readout cards sent out to around 8 Mcps, and at 10 Mcps and above the network saturated and packages were lost, see fig. 5.

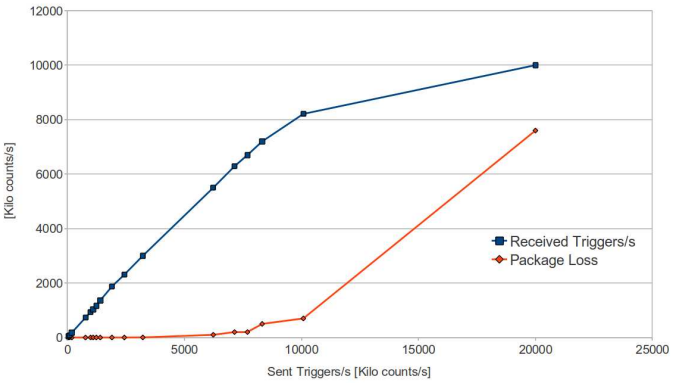


Fig. 5. Graph showing relation between trigger rate and package loss

The set-up gave us a chance to measure the delay between each package received on the CTU when they were sent at the same time from the readout cards. The delay was found the be 1 μs for packages containing 128 event times.

V. Conclusion

Using UDP for an asynchronous trigger system for PET has proven to be feasible. A throughput of 10 Mcps has been observed using 32 bit event times on a 1 Gbps UDP/IP switched network. Synchronization of UDP streams can be done by using timeslots. The implementation was done using off-the-shelf network components.

Acknowledgment

We want to thank the epflab at the University of Oslo for the excellent working condition. The support from the Norwegian Research Council (NRC) is acknowledged.

References

[1] N. Alachiotis, S. A. Berger, A. Stamatakis, *Efficient PC-FPGA Communication over Gigabit Ethernet*, The Exelixis Lab, Exelixis-RRDR-2010-4, TU Munich, February 2010.

[2] E. Bolle, M. Rissi et al. *COMPET high resolution and high sensitivity PET scanner with novel readout concept: Setup and simulations*, Nucl.Instr. and Meth. A (2010),doi:10.1016/j.nima.2010.12.097.

[3] E. Bolle, M. Rissi et al. *COMPET - a Preclinical PET scanner implementing a Block Detector Geometry*,Conference Record IEEE NSS/MIC, 2010.