FPGA based data acquisition system and digital pulse processing for PET and SPECT

Abdelkader Bousselham

Stockholm University
Department of Physics
2006
Thesis for the degree of Doctor of Philosophy in instrumentation physics
Department of Physics
Stockholm University
Sweden

© Abdelkader Bousselham 2006
©
ISBN write it here— Printed by Universitetsservice US AB, Stockholm
Abstract

write abstract here
Acknowledgements

write Acknowledgements here—___________. 
3 Digital signal processing for PET and SPECT
  3.1 Introduction .................................................. 23
  3.2 Free running sampling ....................................... 23
  3.3 Signal timing and energy detection ......................... 24
    3.3.1 Trigger detection ........................................ 25
    3.3.2 Rise point estimation .................................... 25
    3.3.3 Linear extrapolation ..................................... 25
    3.3.4 Digital constant fraction discriminator (DCFD) ....... 27
  3.4 Optimal filtering algorithm ................................ 28
    3.4.1 Matched filter ............................................. 28
    3.4.2 The reference pulse $f$ and the noise covariance matrix $R$ 30
  3.5 Optimal filtering for non stationary noise .................. 31
    3.5.1 The reference pulse and the noise covariance matrix ... 31
    3.5.2 The algorithm ............................................. 32

4 Digital Data Acquisition systems for PET and SPECT .......... 33
  4.1 Introduction .................................................. 33
  4.2 Decentralized data processing system ....................... 33
  4.3 Digital pulse processing implementation ................... 34
    4.3.1 Application-specific integrated circuits (ASICs) ....... 35
    4.3.2 Digital signal processors (DSPs) ........................ 35
    4.3.3 Field programmable gate arrays (FPGA) .................. 36
  4.4 Embedded system design in an FPGA ......................... 37
    4.4.1 Technical reasons for using FPGAs in system design .... 38
    4.4.2 Design partitioning ....................................... 38
    4.4.3 Soft and Hard Processors .................................. 39
  4.5 Embedded operating system .................................. 40
  4.6 Embedded system design software tools ...................... 41
  4.7 Embedded DAQ systems in Nuclear Medicine ................. 42

5 SPECT related projects ........................................... 43
  5.1 The SU KS cylindrical SPECT camera ....................... 43
  5.2 Design and implementation of the new data acquisition system 45
    5.2.1 System design ............................................. 45
    5.2.2 Trigger generation and data acquisition ................. 45
    5.2.3 Pulse processing ......................................... 47
  5.3 Implementation of the new data acquisition System ........ 48
    5.3.1 The FPGA implementation ................................ 48
5.3.2 The DSP implementation ................. 49
5.3.3 Host PC implementation .................. 51
5.4 Debug, calibration and evaluation .......... 52
5.4.1 Calibration of PMTs ....................... 53
5.4.2 The first picture .......................... 53

6 PET related projects .......................... 55
6.1 The Biocare project .......................... 55
6.2 Experimental setup .......................... 55
6.3 Digital timing measurement ................. 56
6.3.1 Implementation ............................ 60
6.3.2 Software ................................. 60
6.3.3 Data analysis: Extracting the timing information ... 60
6.4 Digitizing by using the oscilloscope ............ 61
6.4.1 Pulse and noise analysis ................. 61
6.4.2 The timing algorithm for non-stationary noise .... 61
6.5 Ethernet based distributed data acquisition system .... 62
6.5.1 An FPGA based data acquisition module ....... 62
6.5.2 The server and control clients ............... 64
Accompanying papers & Author’s contribution

I  A. Bousselham & C. Bohm,
   Sampling pulses for optimal timing,
   Nuclear Science Symposium Conference Record, 2005 IEEE.

II A. Bousselham, P. Stenström, C. Bohm, S. Larsson.
   Evaluation of Cylindrical SPECT Camera
   Nuclear Science Symposium Conference Record, 2004 IEEE.

Additional published work

I  P. E. Ojala, A. Bousselham, L. A. Eriksson, A. Brahme, C. Bohm
   Influence of Sensor Arrangements and Scintillator Crystal Properties on the 3D Precision of Monolithic Scintillation Detectors in PET
Chapter 1

Nuclear Medicine Instrumentation

1.1 Medical background

The unprecedented development in many areas of medicine during the last 50 years is to a large degree due to the availability of new diagnostic instruments. From the beginning of medical imaging in 1895 when Roentgen produced the first X-ray image, many methods and techniques have been developed, which were adapted from or inspired by nuclear and particle physics instrumentation. A large part of medical imaging techniques concerns the viewing of anatomical structures inside the body. X-ray computed tomography (CT) [1] and magnetic resonance imaging (MRI) [2] are sophisticated techniques of this type which yield high resolution images of anatomical structure parameters. However, in medical research and in the diagnosis of many medical conditions it is often necessary to use functional information. It is thus highly desirable to acquire images of physiologic function to complement images of the anatomy. Certain biologically active pharmaceuticals concentrate in different organs of the human body. When these are chemically labeled with specific radioactive materials and administered to a patient, its three dimensional distribution can be deduced from the externally recorded gamma ray emission pattern. The activity distribution and its changes with time, from the injection and on, are then
used to derive functional information. This type of imaging is known as nuclear medicine imaging [3]. Nuclear medicine techniques are applicable in the diagnosis of a wide variety of diseases. They can be used for tumor detection, imaging of abnormal body functions or to quantitatively measure the heart or the brain function. For example, since blood flow in the brain is tightly coupled to local brain metabolism and energy use, the 99mTc-HMPAO (hexamethylpropylene amine oxime) blood flow tracer can be used to assess brain metabolism regionally [4]. This information can then be used in an attempt to diagnose and differentiate the different causes of dementia. Depending on the type of the radio isotope labeling, nuclear medicine techniques can be split into: gamma or positron emission methods

1.2 The Gamma Camera

The first gamma camera was developed by Hal Anger in 1957 [5]. His original design is still used. This device gives projective images where it is impossible to determine what is in front and what is in back from the image alone. However, some of this ambiguity could in principle be restored by combining information from several projective images. When imaging with a gamma camera the patient is injected with a small amount of a gamma emitting tracer chosen to accumulate in specific regions. The radioisotope emits a single gamma photon with energies typically between 80 and 350 keV. 99mTc (140keV), which is the most frequently used isotope, can be produced locally in a radio chemical generator that must be replaced every week at a moderate cost. The low costs associated with gamma camera systems is an important reason for its wide spread.

The components making up the Gamma camera (Figure 1.1) are the collimator, the scintillator, the front end electronics (FEE) and the data acquisition system (DAQ).

The collimator is made of a gamma ray absorbing material (lead or tungsten), which acts to select a given direction of photons incident to the camera. In a parallel hole collimator only photons traveling in a direction parallel to the collimator holes will reach the scintillator detector. In fact, most of the photons are stopped in the collimator.
The collimator defines the geometrical field of view of the camera and determines both, the spatial resolution and the sensitivity of the system. The latter two are conflicting properties, increasing the precision reduces the sensitivity and vice versa.

Behind the collimator the gamma-rays are usually detected by a large single thallium-activated sodium iodide (NaI(Ti)) scintillator crystal, typically about 50 cm in diameter. The interaction of the gamma-ray with the scintillator crystal results in a large number (about 6000) [6] scintillation photons emitted isotropically.

The emitted photons are detected by an array of photomultiplier tubes (PMTs) which are optically coupled to the surface of the crystal. A PMT consists of two elements, a photo cathode coupled to an electron multiplier. The photo cathode ejects electrons when stimulated by light photons. This happens in about 25th the photo cathode).

The electron multiplier consists of an arrangement of dynodes that serves both as efficient collection geometry for the photoelectrons and an amplifier that greatly increases the number of electrons. After amplification, a typical scintillation pulse will give rise to 10^7-10^10 electrons [6], sufficient to generate a strong charge signal that can be collected at the anode.

At the photomultiplier output an amplifier, filter, shaper and line-driver are used to adapt and transport the pulse to the data acquisition system.
were the two coordinates of the interaction position are extracted from the amplitude distribution (i.e. the center of gravity) of the PMT signals while the total energy is obtained from their sum. The total sum allows discrimination between different isotopes (if several isotopes are used simultaneously) or between scattered and direct photons. The data are then sent to a computer for processing into a readable image showing the spatial distribution of the uptake in the organ.

### 1.3 Computed Tomography

Computed tomography (CT) is a medical imaging method where digital processing is used to generate a three-dimensional image volume from a set of two-dimensional projections, taken uniformly around a single axis of rotation. The principles of tomographic reconstruction have been known by mathematicians since early 1900 (Radon), but were rediscovered by physicists during the 50s. Accurate tomographic imaging techniques benefited greatly from the digital revolution due to the computing power needed. The first practical application was made in 1960’s. Then in the 1970's there was an explosion of activity with several techniques being developed simultaneously, most notably X-ray computed tomography (CT) [7] and positron emission tomography (PET). CT was invented in 1972 by Godfrey Hounsfield, who later along with Allan Cormack was awarded the Nobel Prize (1979) for their development of tomography in medicine and science.

The computed tomography technique has applications in non-medical fields as well [8], for example in astronomy where images taken at different times from different angles are combined to form a more accurate composite image. Simple back projection algorithm was the first algorithm used (in the 40s) for image reconstruction, where it was performed using analog methods. In the simplest form of back projection, each projective image is smeared along it’s the direction of projection [9]. Since you don’t know where along this direction an object seen in the projection is located, you let it contribute to all the possible points. By combining several back projections you get a three dimensional representation of the object. Simple back projection produces blurred images. However, the blurring can be removed by applying
digital processing algorithms (filtering). Depending on the application, different filters can be applied [9]. The maximum likelihood expectation maximization (ML-EM) and filtered backprojection (FBP) algorithms are the most used reconstruction methods [10].

1.4 Single photon emission computed tomography (SPECT)

SPECT imaging is performed by rotating a gamma camera around the object (or patient) and acquiring projections from multiple angles (Figure 1.2). A tomographic reconstruction algorithm is then applied to the full set of the 2D projections in order to reconstruct a 3D image[11]. This image volume can be sliced in any direction to determine the position and the concentration of the radionuclide distribution. How-
ever the use of a collimator results in low sensitivity especially if one wants high resolution. In order to increase the sensitivity and hence the scanning time, many SPECT systems are equipped with multiple detector heads. Systems with two or three Anger type gamma cameras 180°, 120° or 90° apart with small or large field of view are most common. These systems have improved the sensitivity, the resolution and reduced the scanning time considerably compared to single camera systems. The parallel collimator can be replaced by a collimator with diverging holes or by a pinhole collimator which allows geometric enlargements and can obtain high resolution. This is useful when imaging small objects.

The variable amount of attenuation experienced by the photons emitted from the organ of interests when they pass through the body as well as the scattering of photons in detector crystal can lead to significant underestimation of activity inside the body. This problem can be solved by incorporating a map of attenuation coefficients from a transmission scan by X-ray CT into the reconstruction algorithm. Iterative reconstructions algorithms [12] which are used to reduce the image degradation are of growing importance.

SPECT based on conventional gamma cameras suffer from limitations such as image non-uniformity, image distortion and degradation of the position resolution towards the edge of the camera.

1.5 Compton cameras

In SPECT one tries to avoid Compton scatter since it has a detrimental effect on the position resolution. However, in a Compton camera it is the kinematics of Compton scatter that is used as an electronic collimator to reconstruct the radioisotope distribution. In the most basic form, the Compton camera consists of two detectors, scatter and absorber, separated by a known distance, the first with very good energy and spatial resolution. When the gamma photon enters the camera, it interacts with the first detector and scatter into the second detector (Figure 1.3). However, there is no need for a sensitivity limiting collimator.
By using the two detectors in coincidence, the total energy, the energy loss in detectors and the direction of travel from the first detector to the second can be extracted for each event. This information in conjunction with the Compton equation can be used to derive a cone of all possible directions of the incident gamma ray consistent with the recorded track.

\[ \cos \theta = 1 - m_0c^2 \left( \frac{1}{E_\gamma} - \frac{1}{E_{re}} - \frac{1}{E_\gamma} \right). \]  \hspace{1cm} (1.1)

Where \( \theta \) is the cone semi-aperture and the Compton scatter angle; \( E_{re} \) is the recoil electron energy in the front detector; \( E_\gamma \) is the source energy; and \( m_0c^2 \) is the rest mass of an electron (511keV). A large number of scattering events from a point source of gamma rays will define multiple cones which intersect at the location of the source (Figure 1.4). In practice, a filtered back projection technique is used to recover the isotope distribution.

### 1.6 Positron emission tomography (PET)

Positron emission tomography use biologically active tracers labeled with position-emitting isotopes. Following the decay, the emitted positron slows down and annihilates with electrons in the tissue producing two
back to back emitted 511-keV photons. The detection of these two photons in coincidence defines a line along which the emission point is located. By accumulating many annihilation events and then sorting them according to line directions (sinograms), a full set of projections can be acquired. From this the spatial distribution of the isotope can be derived, using tomographic methods. Since the electronic coincidence detection already limits the position of the source to a line, there is no need for a sensitivity limiting collimator. This is one of the great advantages of PET. Figure 1.5 is a schematic of the principle of the PET technique.

1.6.1 Types of coincidence events

Coincidence events in PET can be classified as: true, scattered and random [13] (Figure 1.6). True coincidences occur when both photons from an annihilation event are detected by detectors in coincidence, neither photon undergoes any form of interaction prior to detection, and no other event is detected within the coincidence time-window. There are two effects that cause the annihilation point to deviate from the associated line of response connecting the detectors. The positron travels a distance before annihilation and photons are not emitted exactly back to back. These effects provide a lower limit of achievable resolution.
1.6 Positron emission tomography (PET)

Figure 1.5: A schematic of the PET principle

Figure 1.6: Types of coincidence events in PET
A scattered coincidence occurs when at least one of the detected photons has undergone a Compton scattering event prior to detection causing the direction of the photon to change. The event will be then assigned to the wrong line of response. Scattered coincidences add a background to the true coincidence distribution causing the isotope concentrations to be overestimated. One way to reduce the number of scattered events by eliminating those that have suffered large energy losses, for this the system needs a good energy resolution.

Random coincidences occur when two photons belonging to different events reach the detectors within the coincidence time window. The rate of random coincidences increases roughly with the square of the activity in the field of view (FOV) and linearly with the size of the coincidence time window. The latter should be kept as small as possible. The number of scattered and random events detected depends on the volume and attenuation characteristics of the object being imaged, as well as the geometry of the camera. Estimated random and scatter events are subtracted in the reconstruction process, but even if the subtraction is accurate, they still add statistical noise to the data.

### 1.6.2 Detector configurations

Different detector configurations have been developed for commercial PET scanners [14]. For high sensitivity cameras the detectors are mounted on multiple circular or polygonal rings. Early detectors consisted of a single scintillator coupled to a single PMT. Today the most common configuration used for commercial scanners and small animal PET systems is a block detector configuration. The block detector (Figure 1.7) consists of a 2D array of single crystals cut from a large cubic crystal where the cuts are filled with reflective material to optically isolate the single crystals from each other.

The array is coupled to 4 PMTs with light sharing. By comparing the signal in A and C to the signal in B and D, one can determine which crystal row is activated. Similarly, by comparing A+B to C+D the column is identified. Some PET systems use Position Sensitive PMTs (PSMT) for more accurate positional and energy information [15].
1.6.3 Two dimensional (2D) and three dimensional (3D)

PET scanners can be designed to operate in two dimensional (2D) or 3D mode. In 2D mode thin septa of lead or tungsten separate each crystal ring and coincidences are only recorded between detectors within the same ring or adjacent rings (Figure 1.8a). This reduces the contributions from scatter and random coincidences, however, with consequent reduction in the overall sensitivity [16]. Image data are allocated to different detector planes, direct planes or cross planes with coincidences between adjacent planes. Each plan is reconstructed by itself.

In the 3D mode, the septa are removed, and the coincidences are recorded between any ring combinations (Figure 1.8b). This results in a substantial sensitivity increase, however, at the expense of an increased scatter fraction and increased randoms [16]. Here the entire image volume is reconstructed as one unit, which is very computation intensive, especially when all the detectors are modeled in detail.
1.7 Time of flight PET (TOFPET)

An extension of PET is TOFPET (Time-Of-Flight Positron Emission Tomography). In ordinary PET, it is impossible to determine where on the line between the two detectors the annihilation took place; the annihilation is equally probable to have occurred along the full extension of the line between the detectors. However, in TOFPET, in addition to the coincident gamma photons detected inside the time window, the time difference between the arrivals of coincident photons is measured and used to estimate a more probable location of the annihilation point along the line [17]. Incorporating time of flight information into the image reconstruction algorithm gives more weight to the more probable locations of the emission point for each event [17]. This reduces statistical uncertainty in the reconstructed image and thus one obtains better image quality. It also reduces the effect of random coincidences. However, the direct improvement of the image resolution is small.

1.8 SPECT versus PET

The use of the collimator in SPECT results in a tremendous decrease in the sensitivity and the efficiency compared to PET where collimation is performed electronically. High sensitivity improves the signal to noise ratio which improves the image. Although SPECT imaging sensitiv-
ity is much less than PET, the high cost of PET scanners, the need of the accelerator close the examination place, the availability of low cost SPECT pharmaceutical and the practical and economic aspects of SPECT instrumentation makes this technique of functional imaging attractive for a lot of clinical studies specially for the brain and the hear

1.9 Detectors for PET and SPECT

1.9.1 Scintillator crystals

The gamma photon interacts within the scintillator mostly via Compton and photoelectric effects. The photon deposits its energy in one location by photoelectric effect (preferred in gamma cameras) or in different positions by Compton interaction. The absorbed energy causes the created electron in the crystal to make a transition to higher energy state, from which it may decay after a characteristic time by emitting lower energy photons that are detected by a photodetector [18]. Photoelectric and Compton cross-sections are a function of the density (\(\rho\)) and the effective atomic number (\(Z_{\text{eff}}\)) of the crystal. A high density favors the interaction of the photon in the crystal, whilst a higher \(Z_{\text{eff}}\) value increases the number of photoelectric occurrences with respect to Compton scattering. Therefore, high \(Z_{\text{eff}}\) crystals are to be preferred in most cases. The light yield and the decay time are important physical properties of the crystal. A high light output (number of photons per MeV) implies a better energy resolution, hence high positioning accuracy. Short decay time allow high counting rates without risk of pile-up. Furthermore the scintillation photon wavelength has to match the properties of the photodetectors. NaI(Tl) is the scintillator of choice for imaging with gamma camera due to it is light output (41000 photons/Mev), which allows an energy resolution (FWHM) in the range 9-11% at 140 keV[18]. The drawback of NaI(Tl) for high energies is its low stopping power. CsI(Tl) and YAP:Ce are suitable crystals for high energy gammas [19].

There is a trend towards SPECT systems based on compact and dedicated gamma cameras for specific clinical applications and for small
animals (molecular imaging). Some of the proposed devices use new semiconductor detectors (Ge, CdTe or CdZnTe) where the gamma rays are converted directly to digital electronic signals [20]. Continuous or pixellated scintillator crystals are coupled to an array of solid state photodiodes or to Position Sensitive Photo-Multiplier Tubes.

### 1.9.2 Detector materials for PET

There are three important practical features affecting the PET detector performances, the attenuation coefficient for the 511-keV photon, the light output, and the speed (decay time of the scintillator). The attenuation coefficient determines the detector sensitivity, the energy resolution is improved with the light output and fast crystals allow high count rates and a narrow coincidence window and thus reduce the random rate. NaI(Tl) were used in early PET detectors. It has high light output and it is inexpensive, but its stopping power is small compared to other crystals. Due to its large attenuation coefficient BGO was widely used in many commercial PET systems. The disadvantages of BGO compared to other crystal materials are due to its low light output, long decay time, and the fact that its emission is centered at 500nm where most PMTs are less sensitive [21]. LSO is the crystal of choice in most of today’s PET scanners; it has short decay time, high light yield, and high density [22]. Table 1.1 summarizes the characteristics of the crystals used for PET and SPECT.

### 1.9.3 Position-sensitive photomultiplier (PSMT)

A PSPMT is position sensitive which means that photons impinging at different sites on the photocathode will give rise to pulses of varying amplitudes at the anodes outputs. Typically the output signal is read using independent multiple anodes where each anode is connected to an individual independent electronic chain or by means of current or charge-dividing center-of gravity where the anodes are connected through a resistive network and position signals are read from the two ends for both X and Y direction. For each event, time of interaction is provided by the last dynode. A PSPMT coupled to a thin scintillator
such as CsI(Tl) or NaI(Tl) is an attractive approach to achieve good spatial and energy resolutions [23].

1.9.4 Silicon photodiodes

Photodiodes are semiconductor devices with a pn junction or pin structure where light absorbed in a depletion region generates electrical carriers and thus a photocurrent. Such devices can be very compact, fast, highly linear, and exhibit high quantum efficiency. The device most used for gamma ray imaging is the Avalanche Photodiode (APD) [24]. The avalanche photodiode is a semiconductor-based photodiode which is operated with a relatively high reverse voltage (typically tens or even hundreds of volts), sometimes just below breakdown. In this regime, carriers (electrons and holes) excited by absorbed photons quickly get accelerated in the strong internal electric field, so that they can generate secondary carriers, as it is done in photomultipliers. The advantage of APDs is the high signal to noise ratio due to high gain, and the faster response relative to PMTs and photodiodes. APDs can be produced in arrays and used in very compact systems. A disadvantage is the limited amplification (especially with PIN diodes), the poor energy and timing resolution.

1.9.5 Semiconductor Detectors

Semiconductor detectors are promising alternatives to scintillators for gamma ray imaging due to their superior energy resolution. They can easily be pixelated and read out directly. The most attractive semiconductor material is cadmium zinc telluride (CdZnTe). CdZnTe has very good absorption characteristics due to its high density (5.8 g/cm³) and large effective atomic number (50). The attenuation length of 140 keV gamma-rays is only 3 mm and the photofraction 81 % [25]. The absorption of a 140 keV produces approximately 3*10⁴ photons, which is a good charge yield, hence an excellent energy resolution. Due to the wide band gap it has a high resistivity which results in low noise level. Solid state detectors offer high segmentation, hence higher spatial resolution. They are ideal for the first detector of a Compton camera and they are promising candidates for future PET generations.
1.9.6 Silicon Photo Multipliers

Silicon Photomultiplier (SiPM) is a new type of Geiger-mode avalanche photodiode that shows promise for use with scintillators materials. SiPM consists of many (103 mm$^2$) silicon micro pixels, which are independent photon micro-counters, each of which acts like an independent and identical APD biased above the breakdown voltage in order to create a Geiger avalanche. The SiPM output signal is a sum of the signals from a number of pixels. The photon detection efficiency of the SiPM is at the level of photomultiplier tubes. The device has very good timing resolution (30 ps for 10 photoelectrons) [26]. It has high gain (106) at low bias voltage (50 V), it is insensitive to magnetic fields and has very good temperature stability. These characteristics mean that this novel device combines the main advantages and benefits of normal APDs (size, low voltage operation, robustness) with the main advantages of PMTs (i.e. high gain, gain stability) in a single silicon device and may prove useful for many applications.
Chapter 2

Detector System architecture

2.1 Introduction

The nuclear medicine detector described in the previous chapter is just a part of the full detector system. A gantry is needed for mounting the detector in a way that allows the necessary detector motion. A couch is needed to position the patient. A data processing system is required for data acquisition, control and monitoring. Databases are needed to store the results, keep track of system configuration and experimental conditions. Figure 2.1 is a schematic of detector system architecture. The system components and functions for nuclear medicine instrumentation are often similar to those in particle physics experiment.

2.2 Data acquisition

The main data flow is associated with the data acquisition process. In particle physics detectors the initial data rate is often very large requiring successive data reduction to reach manageable amounts of data for data storage. The reduction is achieved by refining and selecting relevant data. The selection is often performed in multilevel trigger system [27]. Modern detectors are dead time free(Figure 2.2), which means that pipeline memories and synchronous trigger processors are required in the first level trigger. The trigger and the memory work
here as a "bucket brigade", entering data in one end at regular intervals, and exiting decision and delayed data in the other end. The data will be pushed successively through the processor and the memory. The "data bucket" reaches the output of the memory at the same time as the processor finishes processing a copy of its content, reaching a decision whether to transfer the "bucket" to the second level or not. The first trigger must be very fast to minimize the pipeline size. It is usually implemented as massively parallel digital logic (systolic arrays) [28]. Level 2 and 3, on the other hand, are asynchronous, sometimes using farms of regular (PC-type) computers. Here the "data bucket" are queued and then sent to the first available computer in the farm, one processor is temporary allocated to each bucket. The buffer memory must be sufficiently large, to hold all pending bucket. In the event building stage relevant data is combined in data sets that are stored for off line processing. The event data structure must contain or point to all relevant information that is used in the record externally produced data with time stamps so that they can be corrected with the image data in the subsequent data analysis. Similar functions are required in nuclear medicine detector systems.
2.3 Control and monitoring

The main role of the control and monitoring is to assure safe and reliable operation of the system, by controlling different operational parameters of the experiment and monitoring the stability and reliability of the system operation and recording changes related the time trends. To fulfill this task the host has to communicate with a large number of devices. Very complex systems such as particle physics experiments or nuclear medicine detectors need complex control and monitoring functionality. In such systems the devices are logically grouped in several subsystems, which are monitored separately.

2.4 Detector control system

In particle physics or medical imaging applications, the detector control system has different tasks to do at different stages of the system operation. At the startup the detector control system may load and initialize the firmware and set-up the different parameters required by the system (such as different voltages, maximum temperature and power). While the system is running, the control system has to supervise the system by checking temperatures, voltages, cooling system and so on. An important task of the control system is to quickly react to faults for example by shutting down the power and initializing a recovery con-
2.5 System operating modes

Generally a detector system operates in one of the three different modes: test, calibration and production. In the test and calibration mode the systems does not need high computational resources. However, it requires software with user interactive capabilities to let the user interact with the system to change different parameters or run different configurations and record the system response. In the production mode, minimized software part is left for the control and monitoring system, but most of the resources are dedicated to the processing and the communication to the sub-system or host computer. The test mode is used to verify correct system operation and to identify failing parts. Self test is often a part of the normal startup procedure. Otherwise the test mode is used at regular intervals or whenever there is an indication of a possible malfunction. During calibration different procedures are followed to determine calibration constants that are necessary for accurate quantification. In the production mode the system is configured for maximal throughput. Monitoring processes supervise data quality and the detector control system alarms when potentially dangerous conditions occur.

2.6 Data handling in nuclear medicine instrumentation

Imaging instruments in nuclear medicine are complex systems. They need data bases to store configuration data such as run parameters and configuration soft and firmware to be loaded into the system at appropriate places. One has many options in running SPECT or PET, and there is many parameters to set before starting the system, for example in imaging with SPECT one need to specify the number of pixels, the number of frames, collimator type, energy window, number of projection, time of each projection and other parameters. The data
base must also contain calibration parameters along with their interval of validity, so that raw data files can be reconstructed with proper parameter values.

The nuclear medicine investigation starts with entering patient data into a patient base and to specify the type of study intended and its parameters. In state of the art PET and SPECT one deals with high data rates during the data takings, especially when the analog signals are sampled at high clock rates. The pulse processing units trigger when they discover pulses, and perform data reduction by extracting the relevant data parameters (fine time of arrival, energy and data quality) for further processing. The time coarse is recorded as a time stamp. In PET one then sorts pulse data according to time stamp to identify coincidences (i.e. pairs or events that occur within a short time interval). In SPECT it is necessary to correlate the time stamps with instantaneous collimator positions.

The event builder assembles event data structures and in list mode stores them sequentially. Keeping the parameter history allows reconstruction of all raw data. External information such as information about the heart and the respiratory cycles, patient motion and stimuli that are part of the motion protocol are also stored together with timestamps to allow later correction with the image data. An alternative to list mode is to enter the data in projection histograms. This reduces the data storage size drastically but it eliminates the possibility to robin the data in a later reconstruction. A relational data base helps to keep track of all parameters and data, provided it does not slow down the data acquisition.
Chapter 3

Digital signal processing for PET and SPECT

3.1 Introduction

Most of the nuclear medicine detectors produce electric pulses that should be recorded, processed and stored for later image reconstruction. Until recently this has always been done with analogue electronics. During the last few years, digital signal processing has increasingly replaced analog hardware solutions. As in many other fields this also takes place in the area of nuclear medicine detectors.

A straightforward way of digitization is based on a free-running ADC [29]. Here, the acquired digital samples hold the information of interest. There is no need for further analog hardware such as discriminators, shapers or coincidence detection circuits and hard-wired pulse-shaped analyzers. In this chapter the technique of free-running sampling and the necessary signal processing algorithms for determining the detector signal time and energy information are described.

3.2 Free running sampling

In free running sampling mode analog digital converter (ADCs) are clocked synchronously at the speed depending on the frequency con-
tent of the pulse. The ADC’s data are sent continuously to a ring buffer or shift register. Trigger signals cause the buffers to transfer data waveforms corresponding to a given time information (time frame) for pulse processing. Before it is converted by the ADC, the detector signal is amplified and filtered by low pass filter. However, the filter is necessary to avoid aliasing. A schematic diagram is shown in Figure 3.1.

### 3.3 Signal timing and energy detection

For both PET and SPECT the image resolution depends on the energy resolution. To achieve high resolution images with PET, the image reconstruction algorithms require also very precise information about the interaction time of a photon in the detector, ideally with a resolution of less than 1 ns (in time of fight PET much less). Without any signal processing, this would require a very high sampling rates and resolution. To be able to use cost efficient ADCs, the sampling frequency and the resolution have to be lowered using signal processing algorithms to
extract the optimal time and energy and close the gap between the requested time resolution and the interval of the ADC sample points. The signal processing can be divided into two steps:

1. Detection of a signal pulse and
2. Determination of the rise point of the signal with high timing resolution.

### 3.3.1 Trigger detection

The first step in the signal processing chain is the distinction between noise and a pulse from the detector. This can be achieved by comparing each sample of the data stream to a threshold value. When the data value is greater than the threshold level the trigger information is obtained. As the pulse amplitude corresponds to the energy of the detected photons, this is equal to an energy threshold for the photon detection. Usually a filter is applied before the detection to improve the discrimination between signal and noise. Unfortunately, due to the variation of the actual maximum amplitude of the signal, this method cannot always be used for timing since it introduces a timing error ("time-walk") that may be of several sampling intervals long (Figure 3.2).

### 3.3.2 Rise point estimation

### 3.3.3 Linear extrapolation

After the trigger detector has detected a pulse in the data, the exact rise point of the signal has to be computed. A simple solution is based on the linear extrapolation of the rising edge of the pulse. Therefore, two sample points on the rising edge are connected via a virtual line which is then somewhere crossing the zero line of the signal. The extrapolated cross point, which is independent of the signal amplitude, can then be seen as the rise point of the pulse (Figure 3.3).

The calculation formula can be derived from the theorem of intersecting lines:
Figure 3.2: Timing error of threshold detection

Figure 3.3: Linear extrapolation method
3.3 Signal timing and energy detection

\[
\frac{SA}{SB} = \frac{SC}{SD} = \frac{AC}{BD} \quad (3.1)
\]

\[
SC = \frac{AC \times CD}{BD} - \frac{u(t) \times \Delta t}{u(t + \Delta t) - u(t)} \quad (3.2)
\]

For different amplitudes, the calculated time index of the point S can then be used as an approximation for the rise point of the analog signal, although the two points do not match. However, this quite simple method has certain disadvantages: It does not solve the time-walk problem introduced by the fixed trigger threshold. In contrast, it depends on the assumption that the reference points A, B respectively a, b of signals with different amplitudes but equal timing, are always taken from the same position on the rising edge of the signal. Otherwise the calculated rise point will be completely wrong, due to the nonlinear signal shape. The algorithm works only for high signal to noise ratio with good results, as the error from the noise is introduced at the two points A and B in the calculation and has therefore a big influence on the result.

3.3.4 Digital constant fraction discriminator (DCFD)

The DCFD is digital version of the analog constant fraction discriminator technique [30]. The original signal is delayed by time D and a copy of it is inverted and multiplied by a factor C, with 0 \leq C \leq 1. The two signals are then added. This process when optimized, transforms the unipolar pulse into a bipolar pulse. The bipolar pulse crosses the time axis at constant fraction of the height of the original pulse (Figure 3.4). The crossing time is linearly interpolated if it occurred between time steps. The advantage of this method is that the trigger is independent on the signal peak height. The accuracy of the interpolation depend on where the zero-crossing occurs within the sample interval (phase sensitivity), this is especially true when the intervals are long. The result is also affected by the mount of noise present.
3.4 Optimal filtering algorithm

Optimal filtering (OF) is an algorithm which reconstructs the time and the amplitude of an analog signal from its digital samples [31]. We consider the signal pulse from the detector at the input of the ADC $Af(t - T)$, where $A$ is the amplitude, ideally is proportional to the energy and $T$ is the time which is related to the time of occurrence. The measurement of $A$ and $T$ are affected by an error due to the noise of the detector and the readout electronics. The main purpose of the algorithm is to estimate the parameters $A$ and $T$. The shape of the signal $f$ and the statistical description of the noise are supposed to be known.

3.4.1 Matched filter

The best estimate of the amplitude $A$ and the time $T$ of the signal in presence of given noise is obtained using the weighted least square method [32] and are those that minimize the error

$$\varepsilon = Y^T R^{-1} Y.$$ (3.3)
3.4 Optimal filtering algorithm

Where $Y = S - E$ is the deviation between the experimental samples and the values of the fitting curve (reference pulse) at the sampling instants. The vector $Y^T$ is the transpose and $R^{-1}$ is the inverse of the noise covariance matrix, containing the information of the noise autocorrelation function.

Because the problem is not linear with respect to the parameter $T$, we cannot use the equation (3.3) directly. The least squares method is therefore used iteratively in linearized form around successive guess values of the parameters $A$ and $T$. The algorithm starts with the first guess values $A_0$ and $T_0$, and gives the corrections $\Delta T$ and $\Delta A$, using the equation:

$$|\Delta A| = (G^T R^{-1} G)^{-1} G^T R^{-1} Y,$$

(3.4)

where the matrix $G$ contains the derivative of the fitting curve with respect to $A$ and $T$ [10]. The curve $E(t)$, the vector $Y$ and the matrix $G$ are then updated by using the new values $A_1 = A_0 + \Delta A$ and $T_1 = T_0 + \Delta T$. The procedure can be iterated until $\Delta A$ and $\Delta T$ are considered negligible, and therefore:

$$(G^T R^{-1} G)^{-1} G^T R^{-1} Y = 0.

(3.5)$$

At the end of the iterative process, the curve $E(t)$, corresponding to the last parameters $\overline{A}$ and $\overline{T}$ is the best fit of the sampled signal.

For the two parameters $A$ and $T$ equation (3.4) can be written as

$$[(G^T R^{-1} G)^{-1} G^T R^{-1}] A Y = W_A Y = 0,$$

(3.6)

$$[(G^T R^{-1} G)^{-1} G^T R^{-1}] T Y = W_T Y = 0,$$

(3.7)

and as $Y = S - E$, this equivalent to

$$W_A (S - E) = 0, W_T (S - E) = 0.$$

(3.8)

It can be shown [8] that equations (3.8) are equivalent to

$$\overline{A} = W_A S, 0 = W_T S.$$

(3.9)

Whose scalar form is:

$$\overline{A} = \sum_{i=1}^{N} w_A(t_i) s(t_i), 0 = \sum_{i=1}^{N} w_T(t_i) s(t_i).$$

(3.10)
To synthesize the full curve at any time, we should substitute the vector $S$ with the fitting curve $E(t)$,

$$A(\lambda) = \sum_{i=1}^{N} w_A(t_i) s(t_i - \lambda). \quad (3.11)$$

In a similar way we can derive $T(\lambda)$.

### 3.4.2 The reference pulse $f$ and the noise covariance matrix $R$

The knowledge of the reference pulse $f$ and the noise covariance matrix $R$ is essential to reach the optimum resolution. First the reference pulse $f$ is obtained by sampling a number of pulses (from 1000 to 10000), the pulse waveforms acquired were then scanned to find a representative starting reference pulse. Disregarding the noise autocorrelation at this time, $G$ can be calculated using the optimal filter method described above with the pulses acquired. The pulses are then resampled to make them appear at the same time as $f$, adding them on top of each other and normalizing the result gives a new estimate of $f$. This pulse was then used as the starting reference pulse, and the above procedure repeated once more.

Using the residues after the noise-free fit above gives the remaining effective noise present in the system. The noise autocorrelation can then be estimated as

$$R_{ss}(i) = \frac{1}{N} \sum_{j=1}^{N} e_j e_{j+1}, \text{ with } i = 0, \ldots, N - 1. \quad (3.12)$$

Where $e_j$ is the residue of the $j$th sample in the fit. The elements of $R^{-1}$ are then obtained as $R_{ij}^{-1} = R_{ss}(i - j)$.

The optimal filter method described above is derived with the assumption that the noise is stationary [33], this means that if we assume a data set with $n$ data samples per pulse and $m$ pulses and each pulse can be written as a vector $x_i$. The noise can be described by a covariance matrix $R_{ij} = \langle x_i x_j \rangle$, where $\langle \rangle$ represents an average over the $m$
3.5 Optimal filtering for non stationary noise

If the noise is stationary, \( \langle x_i x_j \rangle \) is invariant with respect to time translations; it depends only on \( i - j \). In the case where the noise is non stationary a new method should be developed.

3.5 Optimal filtering for non stationary noise

A general method for the non stationary noise case can be derived from the least square method. If we start from the equation (3.3), and by inserting the expression of \( Y = S - aE \) we get:

\[
\varepsilon = (S - aE)^T R^{-1} (S - aE) = S^T R^{-1} S - 2a E^T R^{-1} S + a^2 E^T R^{-1} E. \tag{3.13}
\]

By minimizing the error with respect to the amplitude we get:

\[
\frac{d\varepsilon}{da} = 0 \Rightarrow a = \frac{E^T R^{-1} S}{E^T R^{-1} E}. \tag{3.14}
\]

Inserting the value of \( a \) in the equation (3.13) we get:

\[
\varepsilon = S^T R^{-1} S - \frac{(E^T R^{-1} S)^2}{E^T R^{-1} E} = S^T R^{-1} S - (CS)^2. \tag{3.15}
\]

To find the best time we need to position the sampled data \( S \) to the reference pulse \( E \). The optimal position is the one which minimize \( \varepsilon \).

The starting point in this method is the reference pulse and the noise covariance matrix

3.5.1 The reference pulse and the noise covariance matrix

A number of pulses (the more the better) are recorded and scanned for pile-up (only good pulses should be used). The pulses are then aligned using the constant fraction method and averaged to get the reference pulse. If the needed precision is more than the sampling time then the pulses are Sinc-interpolated first. By subtracting each normalized pulse from the reference pulse we construct the noise ensemble and the covariance matrix is then calculated.
3.5.2 The algorithm

Once the reference pulse and the covariance matrix are calculated for the precision required (number of sub samples Figure (3.5)), the second step is to calculate the constants $C_s$ by using the same number of samples as the sampled data and for different positions (time shift). All these calculations have to be done once, off-line, tabulating the constants $C_s$ and the $R_s^{-1}$ for different time shift. When a pulse arrives, $\varepsilon$ is calculated in real time, using the equation (3.15) for all the sets of $C_s$ and $R_s^{-1}$.

At the end a binary search algorithm is used to find the index of the minimum. The arrival time of the pulse is then the sample comb plus or minus the index of the minimum (depending on whether the shift is right or left). Finally the amplitude is calculated using the equation (3.14).
Chapter 4

Digital Data Acquisition systems for PET and SPECT

4.1 Introduction

Most existing (PET/SPECT) scanners are built around analog subsystems implemented via discrete circuits or with large scale integration components to reduce power consumption, space, noise and cost. This technology yields good results in dedicated systems but offers little flexibility for sophisticated signal processing and it is costly to upgrade. Advances in flexibility and size of modern field programmable gate array (FPGAs) suggest replacing many of the analog electronics by digital logics (Figure 4.1), enabling a new paradigm where more optimal statistical approaches to the gamma event detection are possible [33]. The aim of this chapter is to present FPGA based methods for digital data acquisition and control.

4.2 Decentralized data processing system

In a decentralized data processing system most tasks are shared between multiple local processing modules (nodes). The connection topology of the modules depends on the application. A common topology is where the modules are connected to each other for trigger distribution
4.3 Digital pulse processing implementation

The development of digital signal processing implementations has been driven by the following consideration; utilizing data parallelism, allowing application-specific specialization, while keeping functional flexibil-
ity and minimizing Power consumption. Each implementation option includes different trade-offs in terms of performance, cost, power and flexibility. While application-specific integrated circuit (ASICs) and programmable digital processors (PDSPs) remain the implementation mechanisms of choice for many digital signal processing applications, new system implementation are often based on field programmable gate arrays (FPGAs).

4.3.1 Application-specific integrated circuits (ASICs)

ASICs have a significant advantage in area and power and for many high-volume designs the cost-per-gate for a given performance level is much less than that of high speed FPGA or PDSP. Unfortunately, the inherently fixed nature of ASICs limits their reconfigurability and the long design cycle may not justify its improved performance benefits for low-volume or prototype implementation, unless the design would be sufficiently general to adapt to different applications such as Medipix [35].

4.3.2 Digital signal processors (DSPs)

DSPs have features designed to support high-performance, repetitive, numerically complex sequential tasks. Single-cycle multiply-accumulate, specialized execution control, on cheap memory and the execution of several operations with one instruction are the features that accelerate performances in sate of art DSPs [36]. The peak performance of the DSP relies heavily on pipelining. However, parallelism in DSP is not very extensive; DSP is limited in performance by the clock rate (Figure 4.3), and the number of useful operations it can do per clock. The TMS320C6202 processor (which we have used extensively) has two multipliers and 200MHz clock, so can achieve 400M multiplies per second, which is much less compared to the field programmable gate arrays.

The programmable DSP chip manufacturers are Texas Instruments, with the TMS320C2000, TMS320C5000, and TMS320C6000 series of chips; Motorola, with the DSP56300, DSP56800, and MSC8100 (Star
Until fairly recently, FPGAs lacked the gate capacity to implement demanding DSP algorithms and didn’t have good tool support for implementing DSP tasks. They were also been perceived as being expensive and power hungry. All this may be changing, however, with the introduction of new DSP-oriented products from Altera [37] and Xilinx [38]. High throughput and design flexibility have positioned FPGAs as a solid silicon solution over traditional DSP devices in high-performance Digital signal processing applications. FPGAs provide more raw data processing power than traditional DSP processors due to the possible massive parallelism (Figure 4.4).

Since FPGAs can be reconfigured in hardware, they offer complete hardware customization while implementing various DSP applications. FPGAs also have features that are critical to DSP applications, such as embedded memory, DSP blocks, and embedded processors. Recent FPGAs technology provides up to 96 embedded DSP blocks, delivering 384 18 x 18 multipliers operating at 420 MHz. This equates to over 160 giga multiplies per second throughput, a performance improvement of over 30 times what is provided with the fastest DSPs. This configuration leaves the programmable logic elements on the FPGAs available to implement additional signal processing functions and system logic,
including interfaces to high-speed chips such as RapidIO and fast external memory interfaces like DDR2 controllers. With up to 8 Mb of high bandwidth embedded memory, FPGAs can in certain cases eliminate the need for external memory.

4.4 Embedded system design in an FPGA

A modern digital system design consists of processors, memory units, and various types of input/output peripherals such as Ethernet, USB, and serial RS232 port. In addition to the major components, large amounts of custom logic circuitry are often needed. In the traditional approach, when designing such systems, each component is included as a separate chip and the custom logic circuits are designed with separate integrated circuit chips. The advanced capabilities of today’s integrated circuit technology have led to embedded systems, by implementing many of the components of the system within a single chip (system on chip (SOC)) using FPGAs [39, 40].

Typical FPGA-based embedded systems use FPGA devices as their processing unit, external memories to store data and FPGA configurations, and I/O interface components to transmit and receive data. FPGA-based embedded systems provide a high degree of flexibility required in dynamically changing environments and allow high processing rates.
4.4.1 Technical reasons for using FPGAs in system design

FPGAs are a good choice for implementing digital systems because they can:

- Include embedded processor cores. These could be hard microprocessor implemented as a dedicated predefined block such as PowerPC in Xilinx FPGAs or soft microprocessor IP blocks such as Microblaze.
- Include an embedded multipliers, adders and multiply and accumulate (MAC) blocks, which are useful for building massively parallel and/or pipelined high speed processors (systolic arrays).
- Support complex clocking schemes due the embedded delay- locked loops (DLL) and phase- locked loops (PLL).
- Offer a large storage capacity in embedded block RAMs, in addition to the distributed look-up table memories.
- Offer large logic capacity, exceeding 5 million system gates.
- Offer a large number of general purpose input/output pins (up to 1000 or more) and Support high speed serial protocols.
- Support a wide range of interconnection standards, such as double data rate (DDR SRAM) memory and PCI.
- Include special hard-wired gigabit transceivers blocks such as RocketIO in Xilinx FPGA, which enable gigabit serial connectivity between buses, backplanes and subsystems.

In addition to the above features, FPGAs provide a significant benefit as "off-the-shelf" chips that are programmed by the end user and can be reprogrammed as many times as needed to make changes or fix errors. However there are also disadvantages, such as high component cost.

4.4.2 Design partitioning

Almost any portion of an electronic design can be realized in hardware by using the FPGA resources or in software by using a microprocessor. In embedded system design using FPGAs, there is a considerable flexibility in deciding which parts of the system should be implemented
in hardware and which parts should be implemented as software running in soft or hard embedded processor (Figure 4.5) [41]. One of the main partitioning criteria is how fast the various functions need to perform their tasks. Nanosecond logic mandates to be implemented in the FPGA fabric. Millisecond logic which is generally for implementing interfaces such as reading switches or flashing LEDs would use much hardware logic, for example huge counters that are needed to generate long delays. Thus it is better to implement these functions in software. The microsecond logic is reasonably fast and can be implemented either in software or hardware.

### 4.4.3 Soft and Hard Processors

Two types of processors are available for use in FPGA devices: hard and soft. A hard processor is a CPU core immersed directly into the main FPGA fabric to implement high performance embedded applications. One, two and even four core implementation in a single FPGA are currently available. For example Xilinx fabricated The IBM PowerPC 405 core within the latest FPGA devices (Virtex-4 and Virtex-II Pro). A flexible alternative is to use a soft processor. A soft processor is an Intellectual Property (IP) core that is written in a hardware description language (HDL), and implemented along with the rest of the system.
by using the logic and memory resources in the FPGA fabric. The performance depends on the configuration of the processor and the target FPGA architecture and speed grade.

Key benefits of using a soft processor include configurability to trade off between price and performance, and easy integration with the FPGA fabric. One advantage of using soft processors is that resources on the FPGA are consumed for processors only when these components are actually needed in the system. The number of processors on a single FPGA is only limited by the size of the FPGA. The Xilinx MicroBlaze soft processor use between 900 and 2600 Look-Up Tables (LUTs), depending on the configuration options and can run up to 100 MHz [40]. MicroBlaze includes several configurable interfaces that allow one to connect his own custom peripherals and co-processors, as well as peripherals provided by Xilinx and third party supplier.

4.5 Embedded operating system

An embedded operating system is an operating system for embedded systems. These operating systems are designed to be very compact and efficient, containing only those functionalities used by the specialized applications they run [42], and forsaking functionalities that non-embedded computer operating systems provide. They are frequently also real-time operating systems (RTOS). Real-time operating system is intended for real-time applications. A RTOS does not necessarily have high throughput; rather, a RTOS provides facilities which, if used properly, guarantee that system deadlines can be met generally (soft real-time) or deterministically (hard real-time). A RTOS will typically use specialized scheduling algorithms in order to provide the real-time developer with the tools necessary to produce deterministic behavior in the final system. A RTOS is valued more for how fast it can respond to a particular event than for the amount of work it can perform over time. Key factors in an RTOS are therefore minimal interrupt and thread switching latency.

Embedded operating systems include:

- eCos (embedded Configurable operating system) [43] is an open source, royalty-free, real-time operating system for embedded systems.
eCos is highly configurable and allows the operating system to be customized to precise application requirements, delivering the best possible run-time performance and an optimized hardware resource footprint. eCos was designed for devices with small memory footprints. It can be used on hardware that doesn’t have enough RAM to support big embedded operating systems.

- **OSE (The Operating System Embedded)** [44] is a real-time embedded operating system created by the Swedish firm ENEA. OSE uses signaling in the form of messages passed to and from processes in the system. Messages are stored in a queue attached to each process.

- **Embedded Linux**. Embedded Linux [45] has these advantages compared to other embedded Operating Systems: Open source, Small-footprint (around 2MB for a minimal installation), No royalty costs, Mature and stable (over ten years of age and used in many devices) and Well supported.

In general one need an operating system that accomplishes the task with minimal foot print.

### 4.6 Embedded system design software tools

In Order to create an embedded system, the design process consists of the creation of the system hardware and the development of software that runs on the processors included in the system. FPGA manufacturers provide a suite of automated tools to facilitate the task of designing both parts of this design flow [46, 47]. For creating the hardware circuitry, the tools allow the user to customize the hardware logic in the system, by making use of pre-designed building blocks (intellectual Property (IP)) for processors, memory controllers, digital signal processing circuits and various communication modules. IP blocks can be developed by the user, obtained from the tool vendor or from a thirty party. The software allows easy instantiation of these sub-circuits and can automatically interconnect them on the FPGA chip. Design of these components seamlessly integrates with the tool set used to create the custom logic circuits, which are also implemented in the FPGA. The Electronic Design Automation tools generate memory maps for
the system, allowing the processor(s) to access the system’s hardware resources. A software platform consisting of a collection of software drivers and, optionally, the operating system are available for the user to build the application software. The software image created consists only of the Portions of the library actually used in the embedded design. Application software development is supported with the typical toolsets expected by software programmers, including compilers, debuggers and operating system support.

4.7 Embedded DAQ systems in Nuclear Medicine

As it has been described previously, modern data acquisition systems for nuclear medicine devices such as PET or SPECT should handle the high count rates and the increased number of channels required to read out state of art detectors, and they should be compact in order to build small compact dedicated devices. Embedded systems in the FPGA technology described above can make this feasible [48, 49]. The FPGA fabric is well suited for implementing low level triggers by using logic resources, and implementing the pulse processing algorithms using dedicated blocks such as MACs and RAMs. The embedded processors (PowerPc or MicroBlaze) are easy to program using high level languages (C or C++), thus they are good choice for implementing the control and monitoring processes. With the small foot print operating system available one can implement a small server in the embedded processors for remote control and monitoring.
Chapter 5

SPECT related projects

5.1 The SU KS cylindrical SPECT camera

In order to improve the resolution, sensitivity and to overcome mechanical stability and field uniformity problems associated with conventional rotating SPECT cameras, a SPECT camera with a single cylindrical Nal (TI) scintillator and a rotating collimator has been designed and constructed in our group in collaboration with researcher from Karolinska Hospital (K.S) [50]. The SU-KS SPECT consist of a cylindrical monolithic Nal(TI) scintillator (180mm long, 305mm inner diameter, and 12.5 mm thickness). The scintillator is optically coupled via a 12.5 mm thick cylindrical light guide to 72 hexagonal photomultiplicier tubes organized in four rows of 18 densely packed 60 mm PMTs (Figure 5.1). The photomultiplier tubes has the same curvature as the light guide and are mounted directly on its convex surface for maximum light collection. In order to protect the scintillator from humidity and to shield it from light, a 1mm thick aluminium cylinder has been used. The collimator rotates inside the aluminium cylinder. The main advantage of this configuration is that the detector can act closer to the object, span a larger solid angle and have a constant distance to the object. The camera is optimised for doing SPECT studies on the brain and may allow PET studies as well especially when it equipped with fast data acquisition system.
A data acquisition system [51] was designed and partially built, consisting of 24 modules, where each module handled three channels. The DAQ module consisted of two PCB boards; a ADC/trigger board and a DSP/Firewire board. The ADC/trigger board consisted of three-channels 8-bit 20 MHz ADC for free running clock sampling, a Xilinx 4000 FPGA for implementing the pulse detection and the trigger algorithm, and a FIFO for buffering data. The boards communicated the trigger signal to adjacent modules via RS485 drivers over a backplane. The DSP/Firewire board consisted of a TMS320C50 40 MHz DSP for acquiring data from the FIFO through a piggyback connector, for implementing the pulse processing algorithm and for running the Firewire drive routines. It also contained the Firewire link and physical chips. The choice of the Firewire as the data transfer network was made because it was considered that Firewire popularity in consumer electronics would guarantee well supported inexpensive components.

The evaluation of the SU-KS SPECT data acquisition (DAQ) system [51] showed that the performance of the system was seriously limited by the chosen DAQ design. The system suffered from count rate limitations, due to the low performance of the digital signal processor DSP (TMS320C50), the 8 bit wide data bus connecting the DSP to the FIFO, the band width limit and the speed of the firewire board, the speed limit of the RS485 drivers and receivers, and the number of channel per module [51]. In order to overcome these limitations and to
5.2 Design and implementation of the new data acquisition system

5.2.1 System design

The main concept in the realization of the system was that the entire signal processing and pulse analysis should be accomplished digitally. In order to handle the detection and acquisition of simultaneous events independently at different locations and hence reduce the overall dead time of the camera, the system was based on a decentralized modular acquisition architecture.

In the new data acquisition system six PMT channels per module were considered instead of three in the previous one, thus reducing the number of the acquisition modules required for the whole system to 12 (instead of 24 in the previous one), decreasing the number of trigger signals connecting the modules and the number of the involved modules per event. The modules are each responsible for 6 channels (A, B, C, D, E, F) arranged in triangular pattern.

The triangles are arranged side by side and cover the 4 by 18 arrangement of the channels. Two orientations are needed up and down where down is a 180 rotated version of up (Figure 5.2). Each triangular module is connected via 12 bidirectional links to its left and right neighbours. Each channel is globally labelled in row-column fashion and the complete arrangement of modules and channels is shown below (Figure 5.3).

5.2.2 Trigger generation and data acquisition

The light that is generated from an interaction event in the scintillation is seen by several PMTs around the interaction point. In order to minimize the information loss, it was made possible to choose different
The data from the detected gamma photon flows from the PMT through the corresponding module; the first step is the acquisition where a free running 8-bit 100 MHz ADC is used to digitize the analogue signal to a number of samples. A trigger decision is made when the signal exceeds a certain programmable level or when the channel receives a trigger pulse from one of its neighbouring PMT channels. Once the trigger decision is made the pulse data is time stamped and written together with a trigger identifier (who triggered the channel, self trigger or which neighbour channel) to an intermediate memory. When a certain amount of data is accumulated in the modules memory, the corresponding module alerts the digital pulse processor to read the entire memory content.

Each channel is responsible for the acquisition of its own waveform and the distribution of the generated or received trigger pulse. A synchronization unit coordinates the acquisition of all the channels involved.
5.2 Design and implementation of the new data acquisition system

in the event, assuring that all channels involved in the event acquire their waveform data at the same time stamp and get the correct trigger identifier code.

5.2.3 Pulse processing

Because of the requirement for high frequency implementation and the number of channels per modules, the most critical part of the system is the digital pulse processing. Within the digital signal processor, by application of different algorithms on digital samples, the pulse amplitude and the time of arrival are determined and the pulse pile-up and the Muon events (high energy events from cosmic rays) are detected. The simple algorithm for determining the pulse height is to take the peak value of the pulse minus the baseline value, but this is not good measure because of the noise, especially for small pulses. In order to determine the pulse height and the time of arrival, an optimal filtering technique was adopted [52]. The method, which was an iterative least square method, was developed for signals of known shape, fitting the sampled pulse to of a reference pulse. The method has been described in chapter 4.
5.3 Implementation of the new data acquisition System

The idea was to split the work between the FPGA and the DSP. The trigger logic and the acquisition were implemented in the FPGA, the pulse processing, firewire routines and routines for configuration of the FPGA and controlling the programmable parameters were implemented in the DSP.

5.3.1 The FPGA implementation

The architecture of the data flow inside the FPGA is presented in Figure (5.5). The FPGA used is Xilinx Virtex 100 [53], this FPGA include 10 dedicated block memories of 4096 bits each, 4 delay lock loop (DLL) for clock-distribution and delay compensation and 100K system gates. This FPGA can handle clock frequencies up till 200 MHz. The design was partitioned to different modules; each module was designed and tested separately. The VHDL language was used for describing the modules, Modelsim simulation tool for simulations and Leonardo spectrum for the implementation. The FIFO was implemented within the FPGA by using the dedicated block RAM, and the delay unit was implemented by using the SRL16 shift register look-up tables [53]. The Compare unit is a simple comparator, comparing the incoming data to a trigger value (level register) and generate trig signal to the synchronize unit when the data is higher than the level.

The synchronize unit is responsible for the synchronization of all the modules and generation of the trigger identification code \((\text{trigID})\). It receives the trigger signal from the comparator of the channel and from all six neighbours channels, if the trigger is from the inside it delays the generation of the Sync signal one clock cycle and distribute the trigger to the neighbour channels, and if it is from the outside it generate the sync signal. When the write unit receives the sync signal it generates the shift signal to the shifter for one clock cycle in order to shift the \(\text{trigID}\) and the value of the timer first. The write signal is generated (for length of the window clock cycles) to the FIFO for writing. The FPGA and the programmable registers (window, pre-trig
5.3 Implementation of the new data acquisition System

5.3.2 The DSP implementation

The software that enables the DSP to function in an IEEE 1394 compliant mode, had been developed in our group. It works as an IEEE 1394 stack, allowing application running in the DSP to send and receive data through the bus via a provided application programming interface (API). The run time kernel (DSP/BIOS) [54] from Texas Instrument has been used for scheduling threads and setting up interrupts. The on-chip direct memory access (DMA) is used to transfer data from the FIFO to the internal memory of the DSP. The DMA is configured to work in repetitive mode, when the FIFO is almost full it sends an interrupts to the DSP, the DMA then move a block of data in burst mode to a buffer in the internal DSP memory and send an interrupts to the CPU, the CPU act by setting-up the DMA to do the next transfer to a new buffer and start the pulse processing algorithm to extract the pulse amplitude of each channel, format the data and call the API send
Figure 5.6: Different threads running on the DSP, lines from the code are removed purposely in order to show the important features.

to send the results to the host PC. A dual-buffering scheme was used to ensure that the CPU does not operate on an incorrect set of data; the DMA moves the data from the FIFO to one buffer while the CPU operates on the other buffer.

Figure (5.6) display the most important threads of the program implemented in the DSP. The program starts with the main() thread, where it initializes and sets parameters of all peripherals (Xbuss, DMA, Multi-channel serial port) as well as registers required for proper functioning of the system. The DMA – ISR() thread starts running when the DMA signals that it finished moving the data block from the FPGA to the DSP data memory. It start by setting-up the destination register
of the DMA, check the start or stop command if were received form
the host PC and post the pulse processing thread for running. In The
pulse – processing() thread, first, the destination buffers are swiped,
thus, the DMA will continue moving data to the previous buffer while
the DSP is working in the current one. As the data arrive as block (of
16 event, each event has a time stamp and six channels, each channel
with a trigger ID and 16 data samples) the processor extract the time
stamp of the event and the trigger ID of each channel and send the
samples to the optimal filter algorithm to extract the energy. When
all the channels are processed, a transmitting buffer is allocated, filled
with the results and sent to host PC by posting the transmitting thread
TX1394(). While the Pulse processing thread is running, the interrupt
from the DMA is disabled, thus, the thread will run to completion. At
the end of the thread the condition register of the DMA is cleared and
the interrupt is enabled.

5.3.3 Host PC implementation

The major part of the host PC software was programmed using the
graphical programming tool Labview. The Labview main program im-
plements the functions for downloading the FPGA configuration bit
file, setting-up the different parameters and the sorting algorithm. The
sorting algorithm is responsible for collecting all he data from the dif-
ferent modules, assemble the data sets which belong to one event and
store them on the hard disk for later processing or butter them to make
them available for the positioning algorithm for real time visualization.

The sorting algorithm was implemented in two threads, the producer
and the consumer. The producer thread was implemented using the
C++ programming language by using the firewire APIs from the firewire
board supplier. This thread has a higher priority, and its main task is
to wait for data and sort it on the basis of which board is the source.
The data from the same board are pushed to a global buffer. The
consumer thread is busy emptying the global buffers and building the
events. Channels with the same time stamp and trigger ID different
than 00 (00 means that the channel did not receive any trigger) are
considered as a single event. The last process is the positioning, the
simple anger logic algorithm has been implemented to extract the (x,
y) position for each projection. Event consisting of position, energy and projection angle is saved for the image reconstruction algorithm.

5.4 Debug, calibration and evaluation

Different methods and tools were used for debugging the firmware, Software, separated boards and a subsystem consisting of three boards. The functionality of the firmware was tested by a functional simulation (ModelSim simulation tool) and post layout simulation by using the returned VHDL code from the LeonardoSpectrum Synthesizer describing low level component, and the sdf-timing file containing information of the timing of each component used. After implementation the hardware was tested with real data from the detector by using the ChipScope Pro [55] tool from Xilinx. The Code Composer Studio (CCStudio) Integrated Development Environment (IDE) form Texas Instruments [56], was used in the embedded software developing process (writing the application, compiling, linking, optimization, profiling and debugging). In designing a multithreaded real time application, one has several problems to solve. Assigning priorities to threads is a critical task and the time consuming peace of code should be optimized. The profiler from the CCStudio was used to find and optimize the time consuming sections.

Different configurations of the priority threads assignments were tested. The task execution with execution graph of the optimal configuration is shown in Figure 5.7. The bus reset has the first priority in order to reset the bus immediately whenever a node has been added to the firewire bus or when problems occurred. The transmitting ($TX_{1394}$) and receiving threads ($RX_{1394}$) should have the second priority to prevent losing data. $Sent - Queue - Scanner$ is to prevent the system from crash, by checking if all the buffers are not full, and should run before attempting to send a new data block. The pulse processing thread has the third priority and it is assured to run to completion, because the $TX_{1394}$ is posted from this thread.
5.4 Debug, calibration and evaluation

5.4.1 Calibration of PMTs

The amplification gain of PMTs is not the same. In order to have a good positioning, each channel must be normalized. By moving a technetium (99mTc) source in front of each channel and recording the maximum energy by a multi-channel analyzer implemented in Labview (Figure 5.8), we were able normalized all channels.

5.4.2 The first picture

After several tests with point sources and Phantom, a SPECT of a rat (Figure 5.8) heart was performed with 99mTc-Mibi- Cardiolite. Because only a subsystem consisting of 12 PMTs was working, we had to rotate the rat in front of the working sector. The center of rotation was at 5 cm from the parallel collimator. We took Projections from 32 angles, one minute for each projection.

List mode data was generated and sent to the Karolinska group for image reconstruction. Figure 5.8 is the picture of the rat heart produced.
Figure 5.8: the rat in the SPECT Scanner
Chapter 6

PET related projects

6.1 The Biocare project

Within the Molecular Imaging for Biologically Optimized Cancer Therapy (Biocare) project work, the goal of the Work Package 1 is to develop a high resolution ultra sensitive large field of view whole body PET-CT camera and a diagnostic treatment unit for molecular tumor imaging and response monitoring. The Karolinska institute and Stockholm university group focus on the design of a PET detector front-end using early digitization with a free-running clock that can be adapted to the two PET detector configurations under consideration: $LSO + PMT$ and $LSO + APD$. The task of the front-end is to deliver optimum time and energy resolution in real time for high sensitivity PET. The front-end will consist of a low pass filter for anti-aliasing, an analog-digital converter, a digital processing module and a link to transfer the results to a digital coincidence unit. The processor module and the link are most conveniently implemented in a high performance FPGA.

6.2 Experimental setup

The first experimental set-up consisted of two PMTs, each coupled to a $4x4x20\ mm\ LSO$ crystal wrapped with Teflon tape. In order to detect coincident signals, the two PMTs are arranged face to face with gamma
source (Na$^{22}$) in between (Figure 6.1). Two different PMTs have been used: Photonis XP2020 [57] and Hamamatsu R9779 [58].

The first step was to build an analog reference setup for time measurement in order to check the detector set-up and to allow comparisons between analog and digital timing. A schematic diagram of the set-up is shown in Figure (6.2). The experiment has been repeated for different PMTs and the results were consistent. The coincidence peak of the time distribution obtained from the set-up where the two XP2020 PMTs are coupled to 4x4x20 mm LSO is shown in Figure (6.3).

### 6.3 Digital timing measurement

Here we kept the detector set-up unchanged, feeding the anode signals from both PMTs to the free running ADCs of a commercial data acquisition board; the BenADDA-USB board from Nallatech (Figure 6.4) [59]. The key features of the BenADDA are:

- **On-board Xilinx Virtex-II FPGA.**
6.3 Digital timing measurement

Figure 6.2: The analog set-up for time measurement
Figure 6.3: The coincidence peak of the time distribution (from the analog set-up)
Figure 6.4: The BenADDA board and a schematic of the data flow

- Two independent analogue capture channels.
- 2x 14-Bit Resolution ADC, with up to 105MSPS sampling rate each.
- 3rd order anti-aliasing filter on analogue inputs.
- Differentially clocked ADCs.
- Xilinx Spartan-II PCI FPGA, pre-configured with PCI, USB and board control firmware.
- 8MB of SRAM memory, in two independent banks.
- Multiple clocking options: internal and external.
- Connection to PC via a USB port or via PCI.
6.3.1 Implementation

After the anti-aliasing filter the data from the two PMT’s anodes were sampled by the 14 bit free running analog digital converter at sampling rate of 105 MHz. The detection algorithm implemented on the FPGA is relatively simple and follows the same approach described in the SPECT project. The sampled data from the ADC were compared sample by sample to a programmable trigger level, when the data of one of the channels is higher than the level trigger then a one clock cycle trigger signal is generated. If at the next clock cycle a trigger signal from the other channel is found, a window is opened for programmable number of clock cycles (samples) in order to cover the width of the pulse along with a few pre-samples to restore the baseline. During the opening time of the coincidence window the delayed data of both channels is merged in 32 bit words and written to a FIFO. When the FIFO is full it generates an interrupt signal to the host computer.

6.3.2 Software

A development API for C/C++, supplied with the board, has been used for building a Dynamic loading library (DLL) for the control and configuration of the FPGA, this allows data to be transferred between the board and host PC. The processing algorithm for data analysis and a user interface was developed using Labview. During the start-up the program calls the DLL for configuring the FPGA and for writing to different control registers. After that it enters a loop waiting for a block of data. When data arrives, it sent it to the processing unit to extract the timing information.

6.3.3 Data analysis: Extracting the timing information

For extracting the timing information of each event we used the same optimal filtering algorithm developed for the SPECT (see chapter 4 and paper II). The algorithm was implemented in a PC using the C programming language. After calculation the arrival time of the two
pulses belonging to same event, the difference is stored in a histogram. We were not able to get a clear coincidence peak for the time distribution. The result has been interpreted as due to the sampling rate which is too low to get a sufficient number of samples in the rising edge of the pulse.

6.4 Digitizing by using the oscilloscope

In order to solve the problem stated above we used a Locroy 8300 A, with 3 GHz analogue bandwidth, 20 GS/s sample rate and 8 bit resolution for sampling the pulses. The detector setup was the same. The anode signals were fed directly to the oscilloscope and the gate signal obtained from the analogue setup (Figure 6.1) was used for triggering the oscilloscope. We stored 10000 events and sent the data file to a PC for processing. We used the same processing algorithm for different sampling rates and we found that when we have more than 3 points in the rising edge the problem of the coincidence peak is resolved but the timing resolution obtained was less than the analogue one even for very high sampling rates.

6.4.1 Pulse and noise analysis

To understand why the matched filter method was not efficient for the data obtained from the digital set-up even with high sampling rates, we decided to perform a full study on pulses and noise from different PMTs with different bases and to develop a simulation model to study different features on the data. The results are presented on paper IV and the main conclusion is that the noise is non stationary which means that the conditions for using matched filters efficiently are not fulfilled.

6.4.2 The timing algorithm for non-stationary noise

A timing algorithm for non-stationary noise has been derived using the least square method. Due to the heavy calculations, the algorithms have been implemented using the Matlab tool. We studied the timing
resolution of different PMTs and from the simulated data. The results were compared to the analogue reference and to results obtained from the CFD method (see Paper IV). We have shown that the timing resolution calculated from the developed algorithm is at least 30ps better than the analogue one.

It is important to find the least sample rate and the least number of ADC bits that could be used without losing too much on the resolution. This information helps to find a way to implement the algorithm in real time with reasonable cost. This is why the effect of the sampling rate and the numbers of bits on the timing resolution has been evaluated with both the experimental and the simulated data.

6.5 Ethernet based distributed data acquisition system

To address the problem of processing the high count rate associated with many PET and SPECT systems, and to implement the digital signal processing algorithms (described above) for high resolution timing in real time, we decided to push the performance of the digital data acquisition system even further. The performance of recent state-of-art FPGAs, the availability of very high speed ADCs and the experiences gained from previous projects (particle physics and SPECT) were the motivation for developing a new distributed data acquisition system.

The system architecture is shown in Figure (6.5). The distributed modules are connected to a server implemented on Host PC through 10-100-1000 Mbit Ethernet via a switch. The server has access to a database, where different configuration parameters and raw data or projection data from the detectors are stored. Different clients can connect to the server in order to control and monitor the system.

6.5.1 An FPGA based data acquisition module

An FPGA based data acquisition module has been developed for implementing the data acquisition and the pulse processing algorithms described above. The module is fully digital; the only analog part
6.5 Ethernet based distributed data acquisition system

Figure 6.5: The architecture of distributed data acquisition system built around system on FPGA Ethernet based controllers

is the anti-aliasing filter just before the ADCs. After the high speed ADCs, a sate-of- the art Xilinx FPGA was used for implementing the system on a single chip (Figure 6.6). It is an auto-calibration system [58] running in two modes: calibration and production. The calibration mode is assured by the network based controllers implemented in MicroBlaze (Paper III) and the client implemented in LabVIEW (in the host PC). In the production mode the pulse processing algorithm is a systolic array implemented on the FPGA fabric, it consists of a generic pulse processing blocks, each block contains a multiply and accumulate unit (MAC), a block RAM and logic unit (paper IV). The number of the pulse processing blocks and the number of MACs in each block is generic and are determined by the sampling rate, FPGA speed, and the timing resolution required.

The on chip system was implemented in a Xilinx Virtex-II Pro (XC2VP7) on a development board from Memec, FF672 Rev 2, were the MAC was implemented using dedicated 18x18 multipliers. For high resolution timing the systems was implemented in Xilinx Virtex 4 FPGA (XC4VSX35-FF668) on development board ML402, were the Xtreme DSP48 slice operating at 500 MHz (appendix A) was used for implementing the MAC blocks. A National Semiconductor 1GHz 8-bit ADC (081000) was used as a data source. A high speed general purpose board
64 PET related projects

Figure 6.6: System on FPGA. Trigger logic, Pulse processing algorithm, memory and Microprocessor are all implemented on single chip based on Virtex 4 FPGA (XC4VSX35-FF668) has been designed, and tested recently in our group [60], and it is planed to be used with high speed ADCs from Atmel (AT84AS004 ) with sampling arte up till 2 GHz [60] for the system prototype.

6.5.2 The server and control clients

From the prototyping and evaluation phase to the production mode of PET/SPECT system, one usually uses different development tools. For example one may start with Labview for control and monitoring in the evaluation phase and finish with optimized piece of software developed in C or other language for the production mode or working with different tools at the same time by assigning each one the task which it can do best. In order to overcome the complexity associated with the development of embedded controller, standardized middleware communication modules have been designed and implemented in the server. By doing this, we made the communication between DAQs modules and the server standard and independent of the client module used for control and monitoring (paper V). Labview and Java clients have been developed and tested with the SPECT subsystem described in the previous chapter.
Bibliography


[27] FPGA

[28] FPGA


