Summer 7-26-2018

CMOS Radioactive Isotope Identification with Multichannel Analyzer and Embedded Neural Network

Samuel Murray
University of Nebraska-Lincoln, smurray5@unl.edu

CMOS RADIOACTIVE ISOTOPE IDENTIFICATION WITH MULTICHANNEL ANALYZER AND EMBEDDED NEURAL NETWORK

by

Samuel Murray

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professors Sina Balkir and Michael Hoffman

Lincoln, Nebraska

August, 2018

CMOS RADIOACTIVE ISOTOPE IDENTIFICATION WITH MULTICHANNEL ANALYZER AND EMBEDDED NEURAL NETWORK

Samuel Murray, M.S.

University of Nebraska, 2018

Advisors: Sina Balkir and Michael Hoffman

A radiation detection and identification system is designed and implemented to perform gamma ray spectroscopy on radioactive sources and identify which isotopes are present in the sources. A multichannel analyzer is implemented on an ASIC to process the signal produced from gamma rays detected by a scintillator and photomultiplier tube and to quantize the gamma ray energies to build a histogram. A fast, low memory embedded neural network is implemented on a microcontroller ASIC to identify the isotopes present in the gamma ray histogram produced by the multichannel analyzer in real time.

DEDICATION

In loving memory of William Slaughter and Jeanette Terrill Murray

ACKNOWLEDGMENTS

I wish to thank my advisors, Dr. Sina Balkir and Dr. Michael Hoffman, for training and mentoring me throughout my degree program. Their guidance and teaching is much appreciated, and has made graduate school rewarding and exciting. I also thank Joseph Schmitz for his immense aid in analog and digital chip design and for his friendship. Furthermore, I wish to express gratitude to Daniel Rogge and Mahir Gharzai for their insights and assistance in research, and for their companionship in the lab. I am grateful for my supportive and loving family, especially for my grandfather Joseph Murray, who has always pushed me to learn and grow, and for whom I have the utmost respect. Finally, I thank God and Jesus, to whom I give all credit for the work I perform.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Measuring the energy of the gamma rays emitted from radioactive sources and identifying which isotopes are present within a system is useful for many applications, such as detecting radiation in the earth [1] or in building materials [2], performing surveillance on ports of entry that potentially traffic illegal and dangerous radioactive materials [3, 4], measuring the astrophysical properties of interstellar objects [5], performing medical tests such as cardiovascular and thyroid screening [6], etc. The need therefore arises for a complete, compact, real-time system that collects gamma rays from a nuclear source, quantizes the energy of each gamma ray, and identifies which, if any, radioactive isotopes are present in the source.

Such a system is presented in this thesis. The system consists of a gamma ray detector that feeds a signal to the analog front-end of an ASIC. The analog circuitry processes the signal from the detector, converts it to a digital value corresponding to the gamma ray energy, and stores the value in the ASIC's memory. An external supporting microcontroller reads the gamma ray energy data from the ASIC and

processes it with a computational intelligence algorithm that identifies the radioactive isotopes present in the nuclear source.

The main goal of this work is to provide a complete and compact radiation detection system and to implement a fast, low memory isotope identification algorithm for low resolution systems using computational intelligence techniques. Systems to collect gamma rays and identify the isotopes in the nuclear source already exist, such as in [7]. However, the computational intelligence algorithm presented here is novel for its ability to be executed on a small microcontroller using little memory and with little execution time.

The following sections introduce important concepts behind the top-level design of the system: gamma ray spectroscopy, radiation detection, analog processing via a multichannel analyzer, and a computational intelligence algorithm for isotope identification.

1.2 Gamma Ray Spectroscopy

Many radioactive isotopes emit gamma rays during nuclear decay. Every isotope emits a unique spectrum of gamma ray energies from which it can be identified. Therefore, it is desirable to obtain the gamma ray spectrum of an unknown substance in order to identify it.

Gamma ray spectroscopy involves the detection of individual gamma rays and the subsequent quantization of their energies into a histogram. The energy of each detected gamma ray is measured and added to a histogram. Each bin in the histogram represents the number of gamma rays the detector has captured with an energy that falls within a certain range. The bin energy ranges are linearly spaced throughout the energy spectrum starting with a minimum value (often around 10 to 20 keV) and

ending at a maximum value large enough to detect the most energetic gamma rays expected in the system (perhaps around 2.0 MeV, depending on the application).

1.3 Radiation Detection System

The radiation detection system presented in this thesis begins with a known or unknown source of radiation that emits gamma rays. The gamma ray energies of interest typically range from a few keV to several MeV. A sensor, called a detector, is then used to capture incident gamma rays and convert them into an electrical signal. Then, a multichannel analyzer (MCA) implemented on an ASIC collects the electrical signals, quantizes them in a manner proportional to the original gamma ray energy, and constructs a histogram from the energy values. A microcontroller on another ASIC configures the MCA and retrieves the gamma ray histogram from it. The microcontroller then runs computational intelligence software to identify which, if any, isotopes are present in the source. A block diagram of the system can be found in Figure 1.1. The dashed box indicates the system implemented in this work.

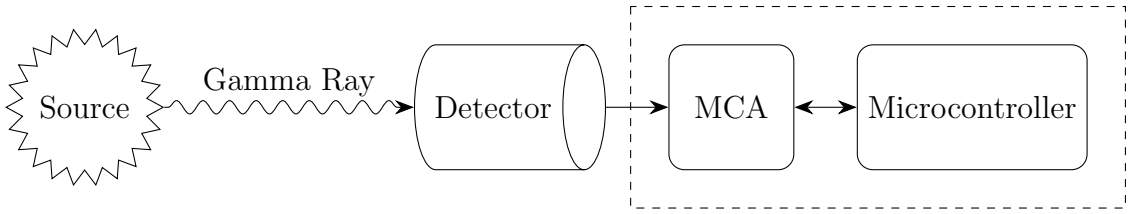

Figure 1.1: Radiation Detection System Block Diagram

1.3.1 Detectors

Detectors to convert gamma rays into electrical signals come in many forms. A scintillator crystal with a photomultiplier tube (PMT) is a common sensor used for

this purpose [8]. A gamma ray photon has three main routes of interacting with the scintillator: photoelectric absorption, Compton scattering, and pair production.

1.3.1.1 Photoelectric Absorption

During photoelectric absorption, the gamma ray photon transfers all of its energy to a single electron bound to an atom, called a photoelectron, in the scintillator crystal. The photoelectron is ejected from the crystal lattice with a kinetic energy equal photon energy minus the electron binding energy (the binding energy is usually small in comparison, and is often neglected). The photoelectron quickly loses its kinetic energy, releasing many visible light photons in the process, which are collected by the cathode of the PMT. The sum of the energies of the visible light photons are about equal to the original gamma ray energy [8].

1.3.1.2 Compton Scattering

During Compton scattering, the gamma ray is deflected by an electron, causing the gamma ray to change direction and lose some of its energy, which is transferred to the electron. The electron gains the lost photon energy in the form of kinetic energy. The amount of energy transferred depends on the scattering angle, meaning the electron can take on a continuum of energy values less than the initial photon energy. However, certain scattering angles are more probable than others, leading to a well-defined probability distribution for the scattered electron energy. The Compton scattering effect favors electron energies somewhat lower than the photon energy, with a gap between the initial photon energy and the most probable electron energy. This leads to a cliff-shaped energy spectrum artifact known as a Compton edge. The energetic electrons quickly bleed off their kinetic energy, releasing visible light photons that maintain the same total energy as the original electron [8]. Figure 1.2 depicts an

example of a photopeak at bin 29 (representative of the initial photon energy) and a Compton edge around bin 19, generated by a Cs-137 gamma ray source. The x-axis is binned into linearly spaced energy ranges (arbitrary units).

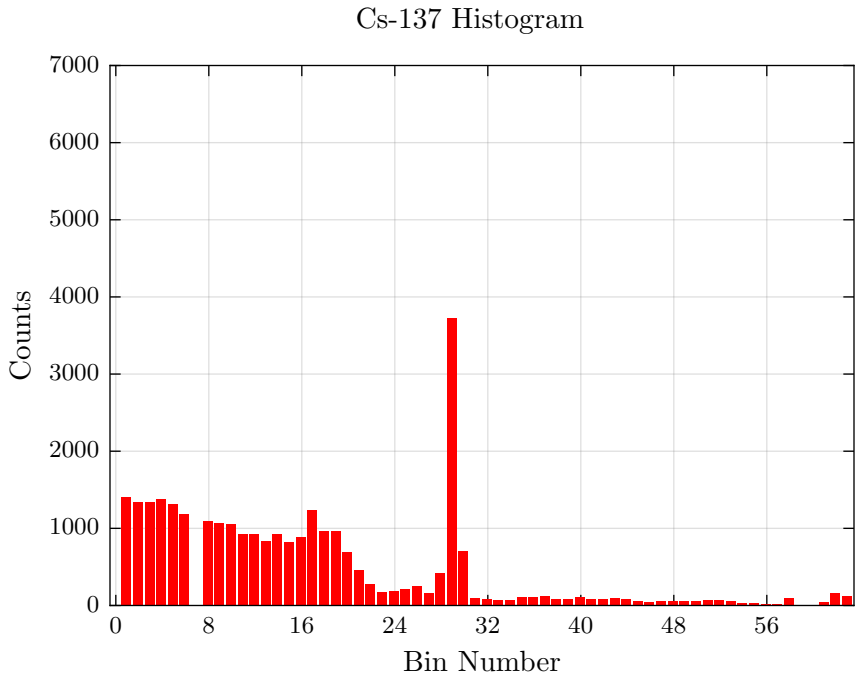

Figure 1.2: Cs-137 Gamma Ray Histogram (with Photopeak and Compton Edge)

1.3.1.3 Pair Production

Pair production occurs only when the gamma ray energy is greater than twice the rest mass energy of an electron $2m_ec^2 = 1.02$ MeV. The gamma ray photon transforms into one electron and one positron, using twice the electron rest mass energy to do so. Any leftover photon energy is split between the electron and positron as their kinetic energies, and the photon itself is annihilated. The electron and positron are quickly slowed, releasing their kinetic energy in the form of many visible light photons which have a net energy equal to the original gamma ray energy minus $2m_ec^2$. When the positron loses most of its kinetic energy, it annihilates with another electron, releasing

two photons at the rest mass energy. These photons then undergo photoelectric absorption or Compton scattering, provided they do not escape the scintillator. Many radioactive isotopes do not emit gamma rays with energies large enough to produce significant pair production events, so the effect is sometimes ignored [8, 9].

All three effects convert each gamma ray into high energy electrons. The electrons quickly lose their energy and emit a cascade of many low energy, visible light photons via fluorescence. The net energy of the visible light photons is equal to the original gamma ray energy in ideal scintillators. Real scintillators, however, always have losses.

1.3.1.4 Photomultiplier Tubes

A PMT attached to the end of the scintillator converts the visible light photons to low energy electrons using a photocathode exhibiting the photoelectric effect. The photocurrent generated by the photocathode is far too weak to be measured electronically with acceptable noise immunity, so the electrons must be amplified before interacting with the downstream processing electronics. The PMT accelerates the electrons through several stages of high voltages separated by dynodes. After being accelerated to a high kinetic energy within a stage, each electron strikes the next dynode electrode and frees many more low energy electrons from the dynode metal. The new electrons are accelerated through the next stage, and the process continues for several stages. Each stage multiplies the electron population, yielding large current amplification with relatively little noise added in. The anode of the PMT outputs charge pulses where the energy of the incident gamma ray is proportional to the total charge contained in the pulse [8].

1.3.2 Multichannel Analyzers

The charge pulses emitted from the anode of the PMT are large enough to be processed with solid state electronics without severe noise-related issues. An MCA is generally used to perform this task [8]. The MCA linearly converts the amount of charge contained within the pulse (proportional to gamma ray energy) to a voltage, converts the voltage to a digital value, selects the corresponding histogram bin in memory, and increments the bin. Each bin in the histogram is therefore represented by an integer number of counts, the number of times a gamma ray has been detected within a particular energy range.

The MCA contains an analog front-end that processes the signal from the PMT and prepares it for event-driven analog to digital conversion. Its design is based on the architectures from [10–12]. The front-end begins with a charge sensitive amplifier (CSA) to integrate and convert the charge pulses into voltages, and is adjustable to accommodate for a wide range of gamma ray energies and count rates. A noise floor comparator determines if the output of the CSA is large enough to convert to a digital value, filtering out very small charge pulses and noise. A peak detector holds the output of the CSA while a 6-bit analog to digital converter (ADC) converts the CSA output voltage to a digital value proportional to the energy of the incident gamma ray. A digital control system manages the analog front-end and stores the ADC results in memory, creating a gamma ray histogram discrete in energy. Once the noise floor comparator detects a pulse large enough to quantize, it triggers the ADC to begin a conversion.

The MCA in this work is designed to balance conversion linearity, count rate, space, and simplicity. It also contains the memory that stores the counts in each bin. It requires an external microcontroller, which configures its settings and collects the

gamma ray histogram from it. The microcontroller uses the histogram bins as inputs to a neural network to identify which isotopes, if any, are present in the radiation source.

1.4 Computational Intelligence

Once the gamma ray spectrum has been collected, it is often desirable to identify which radioactive isotopes are present in the histogram. Classical methods have been developed for this purpose, utilizing curve-fitting and derivatives. However, classical methods involve much effort, and have low adaptability to changing conditions.

In order to allow for adaptability in changing conditions, as well as reduce the level of design complexity, computational intelligence can be used for isotope identification rather than or in addition to classical methods. Notably, the multilayer perceptron neural network (MLPNN) is particularly useful for classifying radioactive isotopes using their gamma ray histograms, as in [13–15]. The MLPNN can be trained to recognize the unique attributes that specific isotopes introduce to a gamma ray spectrum. The MLPNN is given a training set of input histograms with output vectors representing the abundance of each isotope in the sample from which the histogram was generated. For example, the output vector $\hat{\mathbf{y}}$ is of the form

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_{n_o} \end{bmatrix} \tag{1.1}$$

where each $\hat{y}_i$ represents the relative abundance of each desired isotope and n_o is the

number of outputs in the vector (i.e., the total number of desired isotopes). After the training process is complete, the MLPNN can infer the abundance of each isotope in a new histogram. However, it cannot identify isotopes that it is not specifically trained to identify, and may mis-categorize untrained isotopes as trained ones.

The MLPNN is a particularly good tool for isotope identification because of its classification ability and architecture. MLPNNs take n_i numerical inputs and produce n_o numerical outputs. In this case, the inputs are the (normalized) gamma ray histogram bins, while the outputs are the relative abundances of several isotopes, including background radiation. Ideally, the sum of all the outputs would equal exactly 1, but it usually is only close to 1. In between the inputs and outputs are several layers comprised of interconnecting neurons. Each neuron uses only the outputs of the previous layer as its inputs. The layer that interacts with the MLPNN inputs is called the input layer, and while the one that interacts with the MLPNN outputs is called the output layer. The layers in between are known as hidden layers, and there can be any whole number of them present. In three layer MLPNNs, there is only one hidden layer with n_h neurons. A diagram of the MLPNN neuron interactions can be found in Figure 1.3.

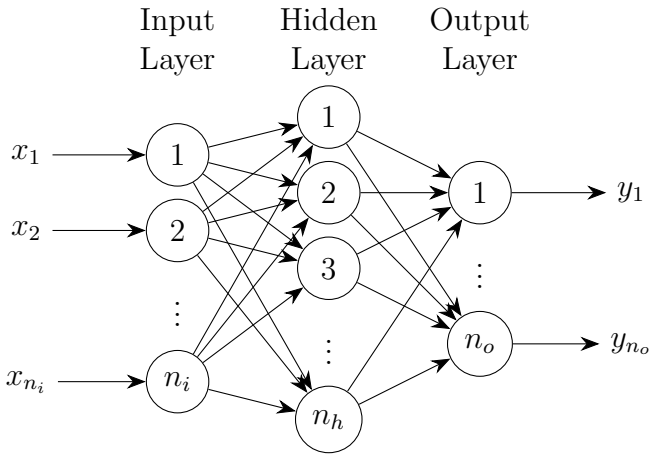

Figure 1.3: MLPNN

Each neuron senses every neuron from the previous layer via synaptic weights. The weights are simply multipliers that control how much influence each input has over the output of the neuron. Each neuron's input is multiplied by its associated synaptic weight and summed with the others. The sum is then sent through the activation function to the output of the neuron. The activation function serves to prevent a single neuron from dominating the output of the MLPNN, and is often a nonlinear function, such as the logistic sigmoid function. The weights in the neurons determine their sensitivity to the inputs, meaning that an individual neuron often has a specific characteristic or attribute that activates it. It is sometimes necessary to bias a neuron's activation function at a certain level, so a bias term can optionally be added to the sum in the neuron. A diagram of an MLPNN neuron is shown in Figure 1.4.

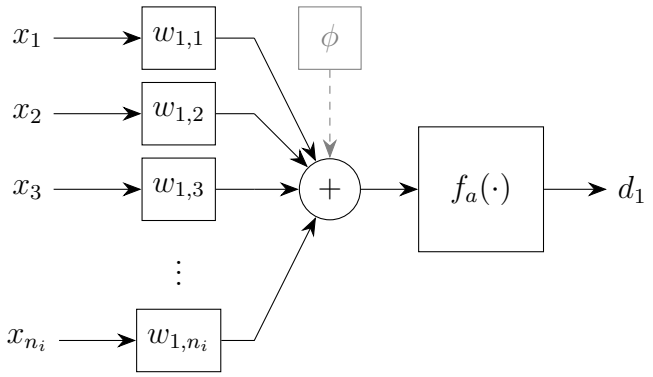

Figure 1.4: MLPNN Neuron

1.4.1 Feedforward Operation

The feedforward operation is the process of propagating the inputs through each layer of neurons to the outputs. Each layer is a step in the feedforward process that uses every output of the previous layer as the inputs to each and every neuron in the successive layer. The input vector to the layer $\mathbf{x}$ is first multiplied by the synaptic

weights in the layer. Since each neuron in the layer uses the output of every neuron in the previous layer, the synaptic weights of a layer can be expressed as a matrix $\mathbf{W}$ where each row vector in the matrix represents each neuron's synaptic weights. The multiplication of the layer input vector with the synaptic weights matrix yields the activation vector $\mathbf{a} = \mathbf{W}\mathbf{x}$.

Next, the activation function is applied to the activation vector on an element-wise basis to produce the decision vector $\mathbf{d} = f_a(\mathbf{a})$. In this case, the activation function for every layer is the logistic sigmoid function, expressed by:

$$f_a(x) = \frac{1}{1 + e^{-x}} \tag{1.2}$$

The decision vector represents the outputs of the layer.

The inputs to the MLPNN are referred to as the input vector, also notated as $\mathbf{x}$, and the outputs similarly as $\hat{\mathbf{y}}$. During feedforward, each layer takes its inputs, calculates the activation vector, and calculates the decision vector. The input layer of the MLPNN serves only to pass on the inputs to the next layer. It can be viewed as having the layer input vector $\mathbf{x}$ be equal to a single element from $\mathbf{x}$, a single synaptic weight equal to 1, and a linear activation function $f_a(x) = x$.

1.4.2 Backpropagation Training Algorithm

The goal of training the MLPNN is to adjust the values of the synaptic weights (and bias terms) until they can produce a desirable output from a set of inputs. The set of inputs and desired outputs used during training is called the training set. MLPNNs are expected to perform well with data not included in the training set after they have been trained.

A common method for adjusting the synaptic weights within MLPNNs is called the

backpropagation algorithm, which is based on the work in [16]. In backpropagation, the goal is to minimize the error in the MLPNN output in relation to the desired output from the training set. The overall error is calculated using the square law:

$$E = \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{y}}\|^2 \tag{1.3}$$

where $\mathbf{y}$ is the desired output and $\hat{\mathbf{y}}$ is the feedforward output. The square law is desirable because it is easily differentiated, focuses on larger errors, and de-emphasizes smaller errors. To minimize the error, the error vector must be *opposite* the direction of the error derivative, which is done with a multiplication by -1. Hence, the error vector can be calculated from the overall error with:

$$\mathbf{e}_{\hat{\mathbf{y}}} = -\frac{\partial E}{\partial \hat{\mathbf{y}}} = -\frac{\partial}{\partial \hat{\mathbf{y}}}\frac{1}{2}\left(\mathbf{y} - \hat{\mathbf{y}}\right)(-1) = \mathbf{y} - \hat{\mathbf{y}} \tag{1.4}$$

Once the error vector is calculated, the error in the vectors and weights matrix within each layer can be calculated in reverse order, beginning with the output layer. First, the decision vector for the i-th layer is calculated. For the output layer, the decision vector $\mathbf{d}$ is the output of the MLPNN $\hat{\mathbf{y}}$, and therefore the decision error vector $\mathbf{e}_{\mathbf{d}}$ for the output layer is the MLPNN error vector $\mathbf{e}_{\hat{\mathbf{y}}}$. For all other layers, the decision error vector is:

$$\mathbf{e}_{\mathbf{d}i} = \frac{\partial E}{\partial \mathbf{d}_i} = \frac{\partial E}{\partial \mathbf{y}_i}\frac{\partial \mathbf{y}_i}{\partial \mathbf{d}_i} = \mathbf{W}_{i+1}^{\mathrm{T}}\mathbf{e}_{\mathbf{a}(i+1)} \tag{1.5}$$

noting that for all layers except the output layer, the decision vector $\mathbf{d}_i$ is also the input vector for the next layer $\mathbf{x}_{i+1}$. Next, the activation vector error for the i-th

layer is calculated:

$$\mathbf{e}_{\mathbf{a}i} = \frac{\partial E}{\partial \mathbf{a}_i} = \frac{\partial E}{\partial \mathbf{d}_i}\frac{\partial \mathbf{d}_i}{\partial \mathbf{a}_i} = \mathbf{e}_{\mathbf{d}i}\frac{\partial \mathbf{d}_i}{\partial \mathbf{a}_i} = \mathbf{e}_{\mathbf{d}i} \odot \left(\frac{\partial}{\partial \mathbf{a}_i} f_{\mathrm{a}i}(\mathbf{a}_i)\right) \tag{1.6}$$

where $f_{\mathrm{a}i}$ is the activation function for the layer. Finally, the weights error matrix is calculated:

$$\mathbf{E}_{\mathbf{W}i} = \frac{\partial E}{\partial \mathbf{W}_{i+1}} = \frac{\partial E}{\partial \mathbf{a}_i}\frac{\partial E}{\partial \mathbf{W}_i} = \mathbf{e}_{\mathbf{a}i}\mathbf{d}_{i-1}^{\mathrm{T}} = \mathbf{e}_{\mathbf{a}i}\mathbf{x}_i^{\mathrm{T}} \tag{1.7}$$

noting that $\mathbf{d}_{i-1}$ is equivalent to $\mathbf{x}_i$ for all layers except the input layer.

This process is repeated for each layer until all layers have been backpropagated. Then, the delta weights matrices are calculated for each layer with:

$$\Delta\mathbf{W}_i = \gamma_g \mathbf{E}_{\mathbf{W}i} + \gamma_m \mathbf{E}_{\mathbf{W}i,\mathrm{prev}} \tag{1.8}$$

where γ_g and γ_m are the learning gain and momentum gain, respectively, and $\mathbf{E}_{\mathbf{W}i,\mathrm{prev}}$ is the error in the weights matrix from the previous iteration of the backpropagation algorithm. The learning and momentum gains are used to control the speed that the backpropagation algorithm converges on minimum error and its stability. Finally, the weights for each layer are updated with:

$$\mathbf{W}_{i,\mathrm{next}} = \mathbf{W}_i + \Delta\mathbf{W}_i \tag{1.9}$$

An iteration of the backpropagation algorithm is run for each input/output vector pair in the training set in random order, making one epoch. After each iteration, the quality of the iteration is assessed by computing the mean-square error:

$$\mathrm{MSE} = \frac{1}{n}\sum_{k=1}^{n}(y_k - \hat{y}_k)^2 \tag{1.10}$$

where y_k and $\hat{y}_k$ are associated elements from the desired output vector $\mathbf{y}$ and the actual output vector $\hat{\mathbf{y}}$, and n is the total number of outputs in $\mathbf{y}$. Many epochs are run, and after each epoch, the algorithm determines it is time to stop if either the average MSE for the iterations within the epoch is below some desirable threshold or if the maximum number of epochs have been computed. The output of the backpropagation algorithm is the set of computed synaptic weights matrices from each layer, which are now trained.

The trained MLPNN can be used with new input data not included in the training set to identify the isotopes within.

The training parameters of the MLPNN include: the number of input, hidden, and output neurons, the learning and momentum gains, the desired MSE, the maximum number of epochs, and the number of samples in the training set.

1.4.3 Isotope Identification

The training set output vectors used in this work train the MLPNN to output the relative abundances of several pre-chosen isotopes with respect to the background radiation of a particular setting. The background radiation can include PMT radioactivity, cosmic rays, local natural radiation sources, and other sources that are not desired to be tracked. The output is therefore a vector of abundances for each desired isotope and also the background, normalized such that the sum of the vector should be around 1.

While the relative abundances are useful data to have, it is often desirable to also have binary yes/no answers as to if each isotope is present. Therefore, a receiver operating characteristic (ROC) is created for each desired isotope by analyzing the training set data and determining the abundance threshold for each isotope above

which the risk of a false positive is 5% or less.

The next chapters explain each system in more detail, as well as present the results and offer a conclusion. In Chapter 2, the analog front-end and ADC are described. Chapter 3 outlines the architecture of the digital control system. Chapter 4 explains how the neural network is implemented on the microcontroller. Chapter 5 details the results from testing the analog circuits, digital core, and neural network. Finally, Chapter 6 concludes the thesis.

Chapter 2

Analog Design

The analog circuits are the core of the radiation detection system, providing the means to measure the charge pulses incoming from the PMT. This chapter describes the analog designs and their purposes, and continues on to provide simulation data for each main analog system.

2.1 Analog Front-End

The analog portion of the MCA ASIC is focused on converting the charge pulses from the PMT into a measurable voltage and then to a digital value proportional to the energy of the incident gamma ray. The analog front-end consists of a charge sensitive amplifier (CSA), a peak detector (PD), an analog to digital converter (ADC), a noise floor comparator (NFC), and some supporting circuitry.

The flow of the analog front-end, depicted in Figure 2.1, begins with the CSA. Gamma rays striking the scintillator are changed into charge pulses by the PMT, which then flow into the input of the CSA. The CSA outputs a voltage proportional to the amount of charge in each pulse which feeds into the noise floor comparator and

the peak detector. The noise floor comparator determines if the pulse had enough charge to warrant an analog to digital conversion of the charge and sends its decision to the digital control system. The peak detector holds the highest output level of the CSA until it is reset, giving the downstream ADC time to convert the peak level into a digital value. The ADC sends the digital value, proportional to the gamma ray energy, to the digital control system for storage.

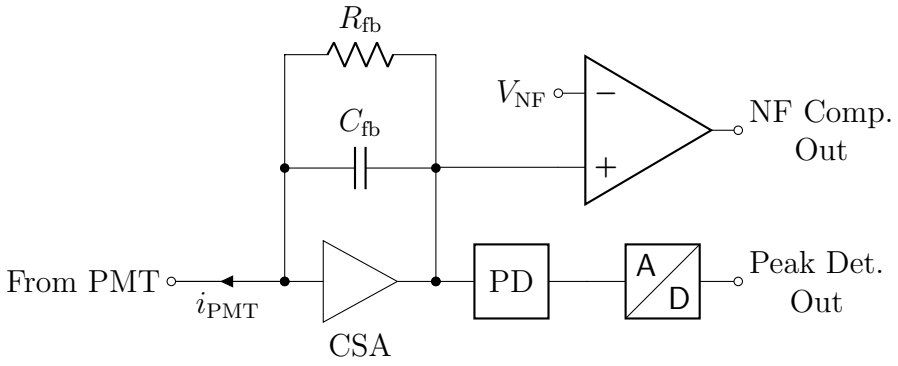

Figure 2.1: Analog Front-End

The supporting circuitry mainly provides biases and references for the main analog components. A constant-g_m cascode bias generator provides biases to all of the analog circuits. Several very low frequency digital to analog converters (DACs) provide voltage references for the CSA feedback resistance voltage, the CSA baseline adjustment voltage, and the noise floor voltage. Another DAC provides linearly spaced reference voltages for each of the flash ADC comparators.

The bias generator and the DACs are controlled and adjusted by the digital control system. Each bias line is multiplexed such that the digital control system can select the bias generator, a DAC, or an external voltage as the source of the bias. Each reference is multiplexed similarly with a DAC or an external voltage. The digital control system can turn off the analog front-end entirely by railing the biases.

2.1.1 Charge Sensitive Amplifier

The charge sensitive amplifier is the first circuit in the analog front-end, and is the primary circuit in the sensor. The charge pulse from the PMT is injected into the CSA, integrated, and converted into a voltage. Thus, the output voltage is proportional to the total charge in the pulse. In an ideal design, the output voltage relates to the input charge pulse with $V_{\text{Out}} = Q_{\text{pulse}}/C_{\text{fb}}$. The CSA design, depicted in Figure 2.2, is a modified version of the CSA in [10].

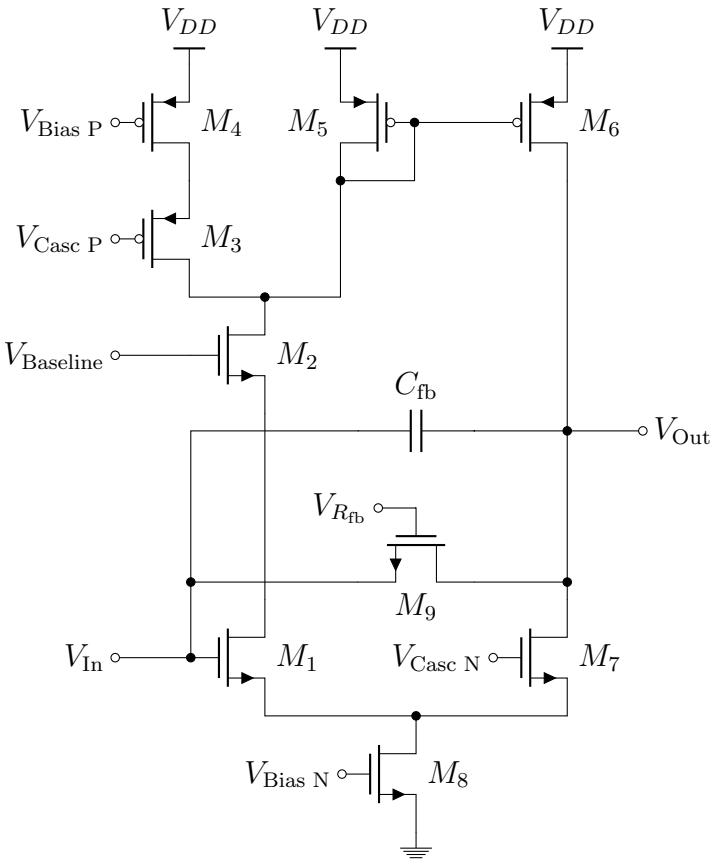

Figure 2.2: CSA

The CSA makes use of a digitally adjustable feedback capacitor across an opamp to perform the integration. The charge pulse from the PMT accumulates on the feedback capacitor, yielding a potential difference. A feedback resistor, implemented

with the MOSFET M_9 in the triode region, allows the capacitor to relax back to equilibrium after the charge pulse has occurred. The effective feedback resistance is varied with the gate to source voltage of M_9, allowing faster or slower decay times.

The opamp part of the CSA is a type of folded cascode amplifier. It functions by redirecting current to or from its output branch, using the output resistance of the cascode transistor M_7 to yield a voltage. The current through M_8 remains static at a given bias level $V_{\text{Bias N}}$, and is shielded from current and voltage fluctuations by M_1 and M_7. The current mirror created by M_4 and M_5 force a large current through M_2 when there are no charge pulses, leaving little current to flow through M_5. The current through M_5 is mirrored by M_6, which determines the output voltage from the active load M_7. Therefore, the output voltage is low when there are no charge pulses.

When a charge pulse arrives, it creates a voltage difference across the feedback capacitor, which in turn creates a small increase in the gate voltage of M_1. This increases the current through M_1, which is drawn from M_5. M_6 mirrors and amplifies the increased M_5 current, causing the output voltage to increase.

The CSA output voltage is dependent on the value of the variable feedback capacitor $V_{\text{Out}} = Q_{\text{pulse}}/C_{\text{fb}}$. The feedback capacitor is adjustable so that the CSA can be sensitive to pulses from tens of keV to several MeV. The decay time of the pulses is tuned with the feedback resistance in M_9.

M_2 serves to adjust the baseline voltage that the output settles to after a pulse. A voltage drop of V_{GS_2} is set by the current traveling through M_1, so the baseline voltage is equal to $V_{\text{Baseline}} - V_{GS_2}$.

2.1.2 Peak Detector

The peak detector is immediately downstream from the CSA. It has two states: track maximum, and reset. During the track maximum mode, the peak detector tracks and holds the maximum CSA output. If the CSA output is not at maximum, the peak detector holds the CSA maximum. In reset mode, the peak detector output falls back to a minimum value regardless of the CSA output. The peak detector schematic, based on the design in [17], can be found in Figure 2.3.

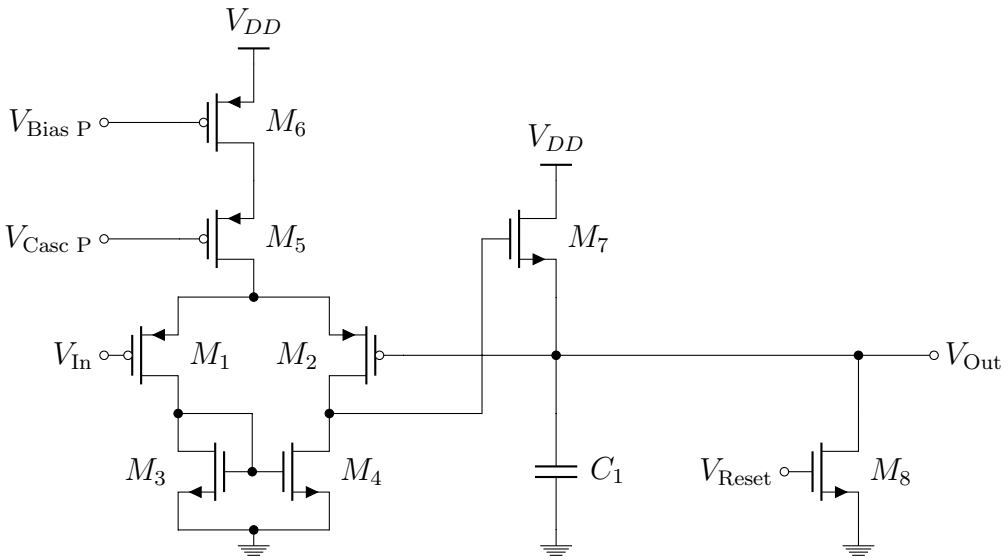

Figure 2.3: Peak Detector

Transistors M_1 to M_6 make a differential amplifier. If the output of the CSA V_{In} rises above the voltage on the capacitor (the maximum CSA value), the amplifier triggers M_7 to deposit more charge on the capacitor until the voltage across the capacitor is equal to V_{In}. However, if V_{Out} is greater than V_{In}, the differential amplifier switches off M_7, and the capacitor holds its maximum voltage. The digital controller electronics can turn on M_8 at any point to drain the capacitor and reset its voltage back to a minimum value.

2.1.3 Noise Floor Comparator

The noise floor comparator sends a signal to the digital controller electronics indicating whenever the output of the CSA is above a threshold, called the noise floor. Ideally, the PMT outputs no ac current if there are no gamma rays interacting with it. In reality, noise causes small ac fluctuations in the PMT output, which the analog circuits might interpret as small x-ray pulses. Furthermore, low energy background radiation will create more ac fluctuations. To avoid these undesirable effects, the noise floor threshold is set above the maximum value of the noise pulses and low energy background pulses. The controller only allows pulses above the noise floor to undergo further processing; otherwise, it ignores them. This significantly reduces the number of pulses the analog and digital electronics must process, which decreases power consumption and decreases the probability of pileup events.

The noise floor comparator used here is implemented with the current mirror opamp from [17]. The schematic is provided in Figure 2.4.

2.2 Analog to Digital Converter

The analog to digital converter (ADC) converts the voltage held by the peak detector into a digital value. The ADC is implemented as a 6-bit flash ADC, chosen for its speed and simplicity. The ADC contains 64 comparators that are fed reference voltages by a resistive ladder. The peak detector fans out to every comparator, which does not adversely affect performance because the fan-out capacitance is much smaller than the hold capacitor of the peak detector. The comparators with reference voltages greater than the peak detector voltage output logic low values, and those with lower reference voltages output logic high values. A priority encoder takes the 64 comparator values and condenses them into a 6-bit value. The digital core

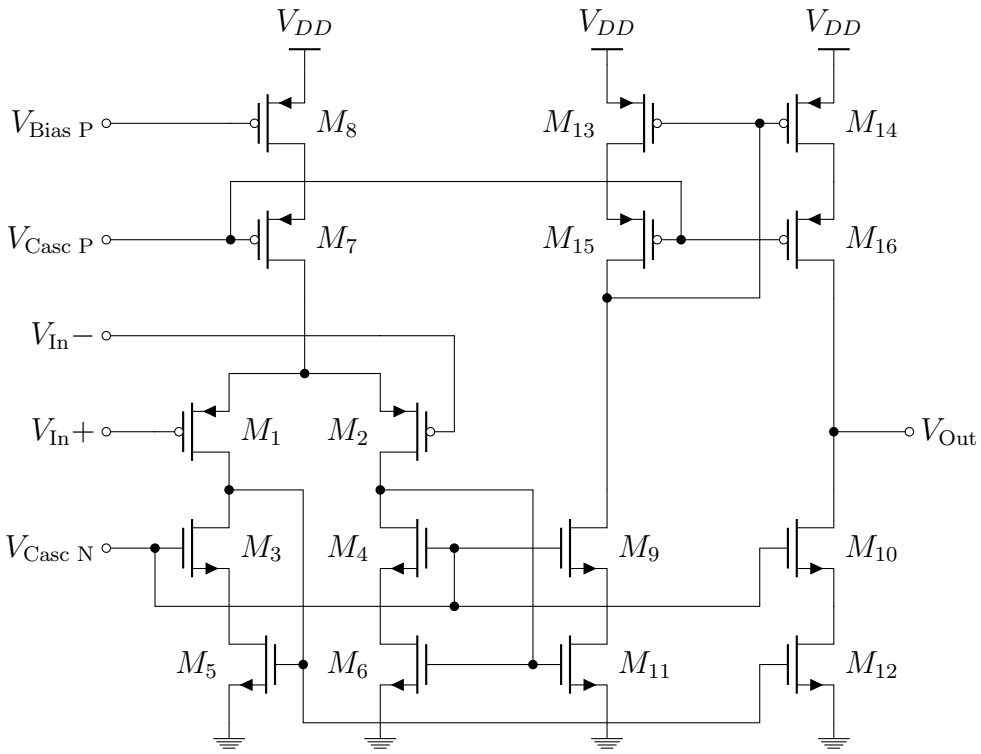

Figure 2.4: Noise Floor Comparator

then increments the bin associated with the priority encoder output by one count. Voltages that are greater than all reference voltages trigger increment special overflow bin (technically, a 65^{th} bin).

The comparator design in the flash ADC is implemented with the two-stage latched comparator from [17], depicted in Figure 2.5. The comparator has two modes: track, and latch. In track mode, the comparator output is a relatively small ac signal, free to change as the input signal does. When the ADC needs to make a conversion, the comparator quickly enters latch mode, which engages the positive feedback, saturating the output to the voltage rails. Further changes to the input signal have no effect on the output until the comparator enters track mode. The comparators feature a preamp stage to help the latching stage with noise immunity and decision speed. The feedback transistors are much stronger than the preamp, ensuring immunity from

input changes once latch mode is entered.

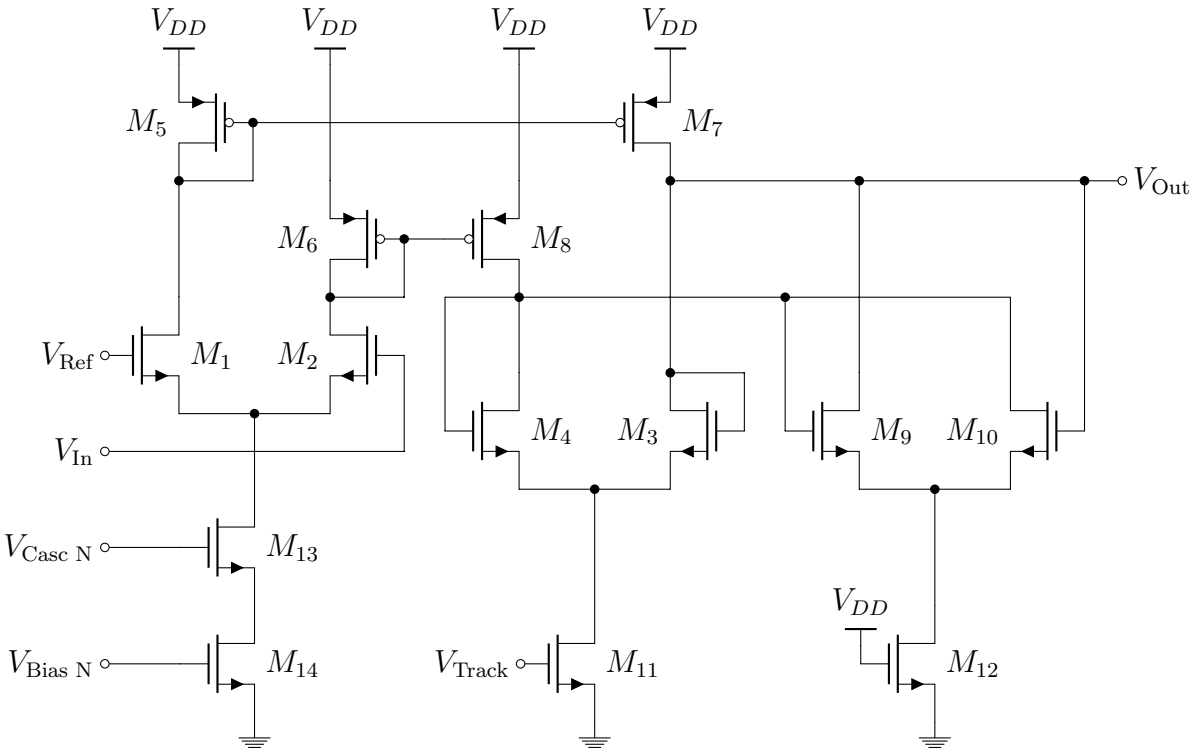

Figure 2.5: ADC Comparator

2.3 Simulation

The analog front-end circuits are simulated in the Cadence Analog Design Environment using model libraries for the 0.5 μm process and the spectreS simulator. Whenever possible, they are simulated using the analog extracted views, which take into account the parasitics inherent to the physical layouts.

2.3.1 CSA

The CSA is the main sensory circuit in the analog front-end, and is also the most important. It needs to make linear conversions from charge to voltage while being adjustable for a wide range of pulse heights and durations.

The gain, phase margin, gain margin, and corner frequency change as V_{baseline} changes, adding an extra level of configurability and process variation resistance to the CSA. As V_{baseline} increases, the open loop gain increases, but the stability degrades and the corner frequency is decreased. Table 2.1 lists data that shows the relationship between these parameters.

V_{baseline} (V)	Gain (dB)	PM	GM (dB)	f_c (kHz)
2.10	24.43	84.67°	25.90	922.3
2.15	34.02	48.36°	22.36	1,060
2.20	38.48	36.78°	19.94	1,034
2.30	48.08	25.02°	15.67	740.5
2.50	53.15	23.04°	14.10	553.5
3.00	54.52	23.13°	14.62	500.1
4.00	57.62	23.30°	15.82	372.7
5.00	64.67	23.68°	16.54	173.2

Table 2.1: CSA Open Loop Simulated Parameters

The CSA has a maximum gain of 64.67 dB. The phase margin drops dangerously close to 0° for some baseline voltage values, though this actually does not present a problem during testing: the CSA can be configured to have an adequate phase margin. The gain margin is sufficiently large for all CSA configurations. Figure 2.6 shows the dependence of the gain, the corner frequency, gain margin, and phase margin on the configurable value of V_{baseline}. Figure 2.7 provides the simulated open-loop frequency response of the CSA at an example baseline voltage.

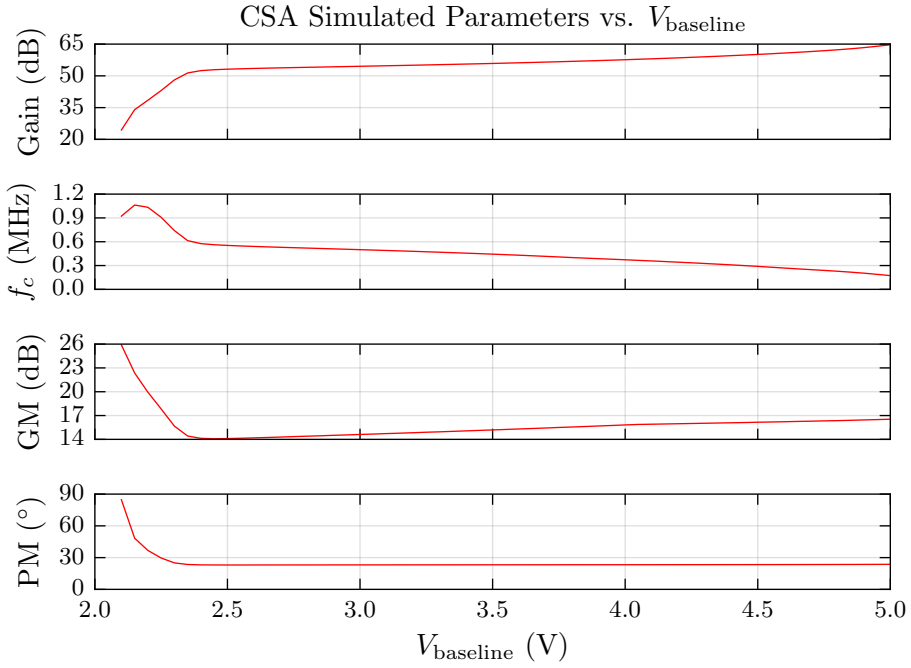

Figure 2.6: CSA Simulated Parameters Depending on V_{baseline}

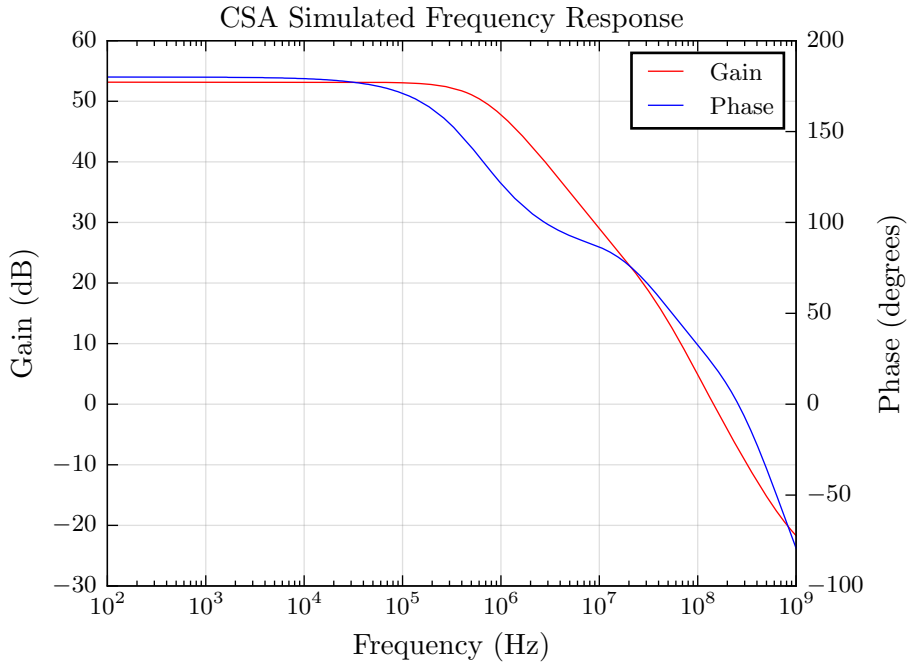

Figure 2.7: CSA Simulated Open-Loop Frequency Response ($V_{\text{baseline}} = 2.5$ V)

The closed-loop CSA is simulated with a mock 10 pC charge pulse, as shown in Figure 2.8. The mock charge pulse is of the form $\alpha t e^{-\beta t} \mathrm{u}(t)$, with α and β chosen to resemble real gamma ray charge pulse data.

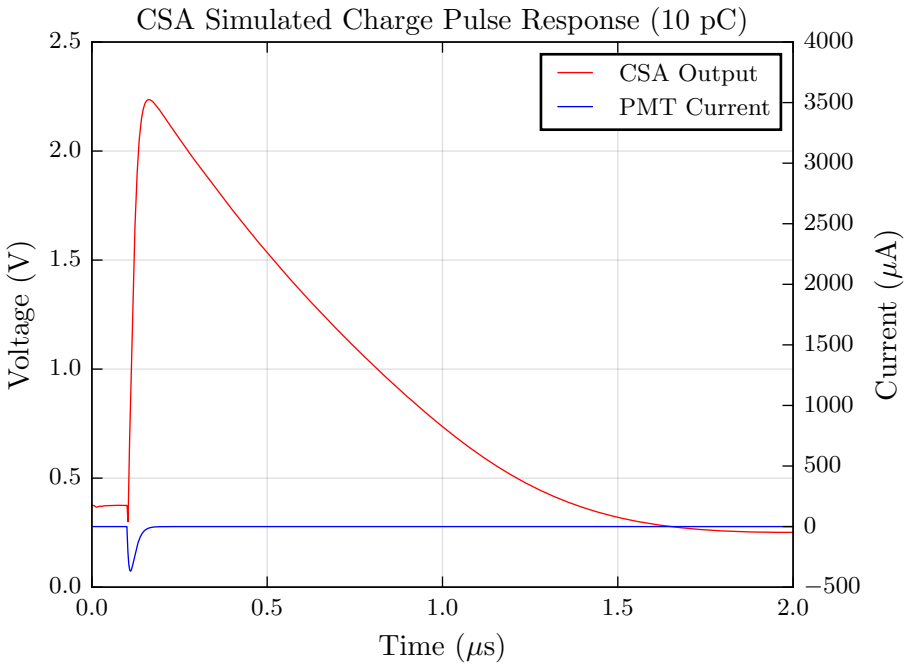

Figure 2.8: CSA Simulated Closed-Loop 10 pC Pulse Response ($V_{\text{baseline}} = 2.5$ V)

The closed-loop CSA is simulated further with different charge pulse values, ranging from 5 pC to 70 pC. The feedback capacitance and resistance are chosen such that the CSA should not saturate to the V_{DD} power rail. Table 2.2 provides the simulation data.

The CSA remains highly linear throughout its operating range, never exceeding 1.2% error. The CSA linearity is plotted in Figure 2.9 using the data from Table 2.2 and a linear regression.

Charge (pC)	Output Swing (V)	Error
5	0.310	1.11%
10	0.624	0.29%
15	0.939	0.10%
20	1.255	0.03%
25	1.570	-0.01%
30	1.885	-0.05%
35	2.200	-0.09%
40	2.516	-0.11%
45	2.830	-0.14%
50	3.146	-0.13%
55	3.464	-0.07%
60	3.785	0.07%
65	4.095	-0.08%
70	4.425	0.24%

Table 2.2: CSA Simulated Linearity Data

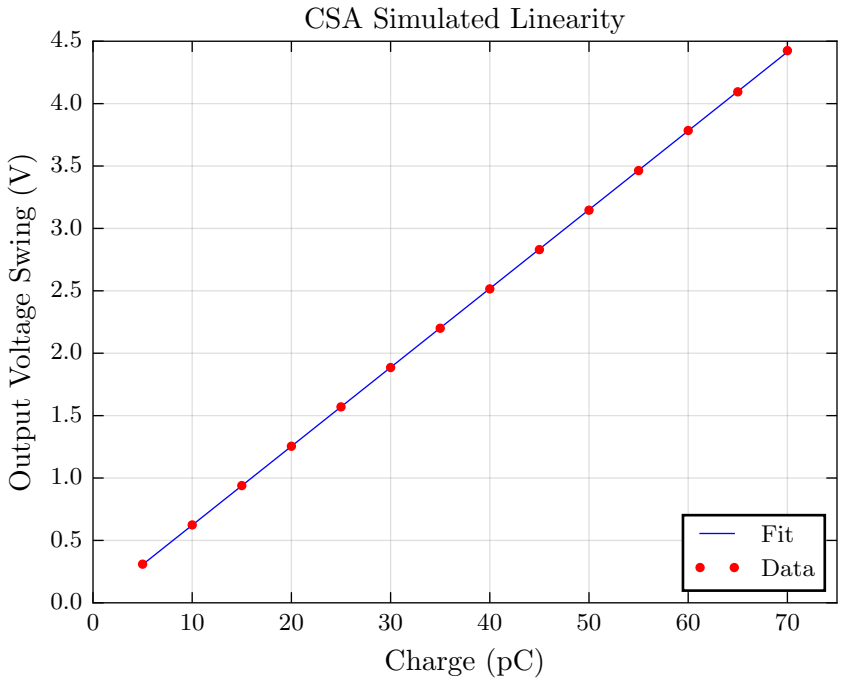

Figure 2.9: CSA Simulated Linearity

2.3.2 Peak Detector

The peak detector, being the interface between the CSA and the ADC, must hold the maximum value it senses from the CSA without allowing any decay. It must not distort the maximum CSA output, must work for all CSA pulse heights and durations, and must quickly reset its output back to the baseline when triggered. The simulated peak detector is given as an input a series of triangle waves of increasing amplitude as a substitute for an actual CSA output. The results can be found in Figure 2.10.

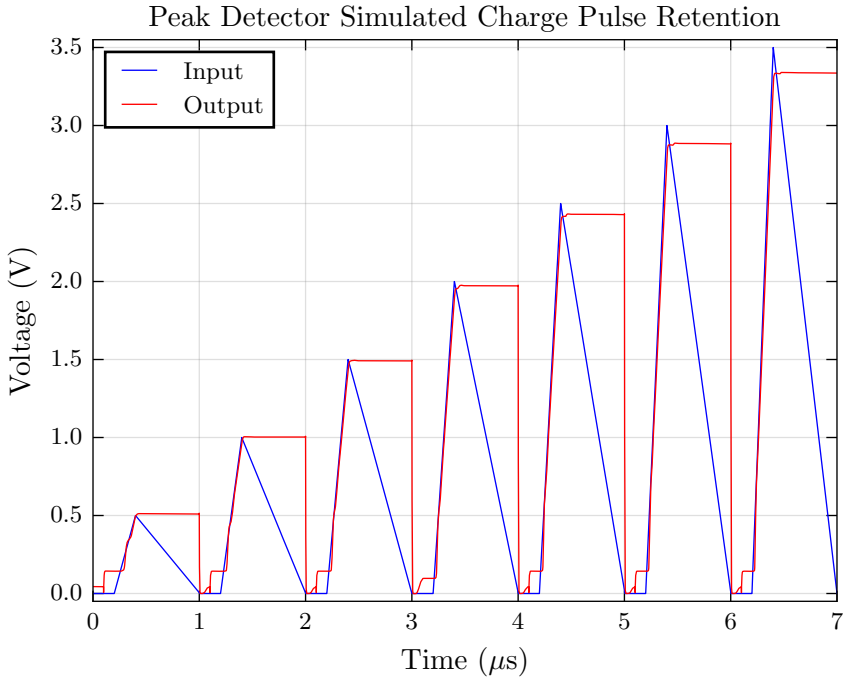

Figure 2.10: Peak Detector Simulated Charge Pulse Retention at Various Magnitudes

The smaller peaks are easy for the peak detector to track, and it does so with very little distortion. However, as the CSA max voltage increases, the peak detector begins to distort the true maximum value. This is in part due to the sharp peaks on the triangle waveforms. An actual charge pulse will linger a little longer at the peak, helping the peak detector. The distortion is also due in part to the speed of

triangle waveforms, which is faster than most actual charge pulses. This simulates a worst-case scenario for the peak detector tracking ability. Table 2.3 provides data on the peak detector tracking error as the CSA maximum output increases.

CSA Output (V)	Pk. Det. Output (V)	Error
0.5	0.509	1.8%
1.0	1.003	0.3%
1.5	1.492	-0.5%
2.0	1.972	-1.4%
2.5	2.429	-2.8%
3.0	2.883	-3.9%
3.5	3.336	-4.7%

Table 2.3: Peak Detector Peak Tracking Simulation Data

The peak detector can reset its output back to the baseline in under 10 ns, which is insignificant compared to the time it takes a pulse to decay back to the baseline.

2.3.3 ADC

The ADC must be linear and fast to allow for the accurate measurement of high count rate sources. Therefore, each comparator within the ADC must trigger when the input voltage is very close to the reference voltage. Furthermore, the comparators must settle quickly to a digital output.

The ADC comparators are simulated for their speed and accuracy. In the simulated worst-case scenario, when the reference voltage is $V_{\text{ref}} = 600$ mV, the conversion time is 96.8 ns. Furthermore, at this level, the comparator switches between true and false at $V_{\text{In}} = 595$ mV, which is 5 mV less than the reference voltage. As there are only 64 comparators spanning a range of at least 3 V, this is an acceptable value.

The ADC reference DAC simulation yields good results, indicating high linearity. However, the simulator uses ideal resistor models and does not factor in process varia-

tions or layout considerations (such as resistor matching), which makes the simulation results unhelpful except to show that the reference DAC is functional.

Chapter 3

Digital Design

The radiation detection system uses two custom and distinct components for processing gamma ray signals: the MCA ASIC, and the microcontroller ASIC. The MCA collects the gamma rays and creates the histogram, while the microcontroller configures the MCA and performs the neural network processing on the histogram data from the MCA. Currently, they are both necessary to perform radiation detection, but future work could combine the two into a single system on a chip.

This chapter describes the roles and functions of both digital cores and how they interact with one another.

3.1 MCA Digital Core

The digital control system in the MCA has four main functions. First, it governs the event driven timing of the peak detector and the ADC, ensuring that every valid pulse is processed as quickly and accurately as the analog circuits are capable of. Second, it stores the digital energy value for every gamma ray detection event in an on-chip RAM memory in histogram form. Third, it provides a means to configure

and adjust the settings requited by the analog circuits. Finally, it provides a means of communication with external devices to read the histogram data stored in RAM or to adjust the settings.

The digital control system is written entirely in VHDL, synthesized to a netlist of logic gates via the Cadence RTL compiler (RC), and translated to hardware in a 3 metal layer, 0.5 μm CMOS process with the Cadence Encounter Digital Implementation (EDI) software. Figure 3.1 depicts a rendering of the MCA digital core generated with EDI. The main regions of the layout are distinguished and labeled.

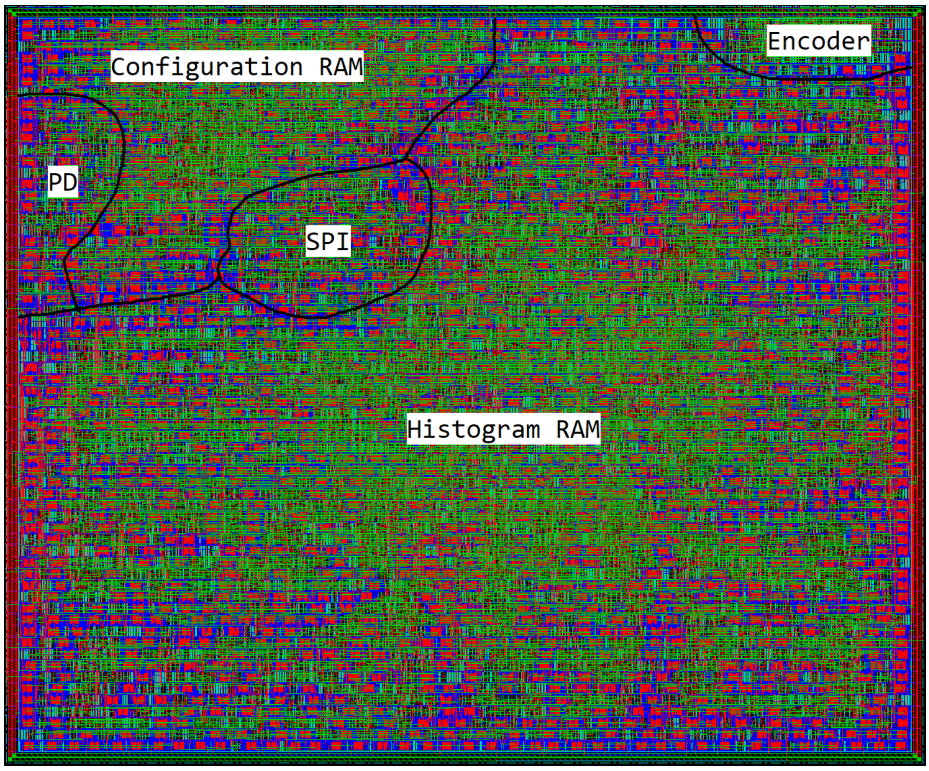

Figure 3.1: MCA Digital Core (Rendering from EDI)

3.1.1 Detection Processing and Timing

The analog front-end in the MCA is event driven, meaning that it only activates when an incident gamma ray stimulates it to do so. Recalling from Chapter 2, the

CSA is always on, looking for incoming charge pulses. The noise floor comparator, sensing the output of the CSA, determines when there is a charge pulse large enough to process, and passes its decision along to the digital control system. The digital control system then begins the process of quantizing the charge pulse by activating the peak detector, which holds the maximum value of the CSA during a pulse event. The control system waits for the peak detector output to stop increasing, indicating that the charge pulse is ready to be quantized, using a timer. Then, the control system triggers the ADC to begin a conversion, waits a very small amount of time for the ADC output to be valid, and stores the data in its own RAM memory. The system then waits until the noise floor comparator indicates the CSA output has fallen back to the baseline before it allows another conversion to be made, reducing the risk of pulse pileup (an undesirable phenomenon where another pulse begins before the first has decayed back to the baseline).

3.1.2 Histogram Memory

The MCA memory stores the gamma ray histogram as a vector of 16-bit words. Each word is addressable and corresponds to a bin, which represents a range of energies. Once the ADC converts a charge pulse into a digital value, the control system uses the ADC value as a memory address (or bin number) and increments the associated bin by 1. Thus, each word represents the number of times a gamma ray has been detected within the associated energy range. If more than $2^{16} - 1$ counts are recorded in a bin, the bin overflows, resets back to zero, and begins counting up again.

The memory storing the histogram is implemented with an SRAM of many D flip flops, and takes well over half the area of the 3 mm by 3 mm MCA chip. A better RAM architecture could have decreased the amount of area needed, but no RAM IPs

were available to the designer.

3.1.3 Configuration

The digital control system is also equipped to configure the various analog circuits in the front-end, as well as some settings for the timers governing the detection processing. Another vector of 8-bit words is used to store the configuration settings, implemented as another SRAM of D flip flops. The configuration RAM is much smaller than the histogram RAM, but still takes significant area. The configuration RAM controls the analog front-end biases, CSA feedback capacitor and resistor values, noise floor reference, baseline reference, state machine mode, ADC reference DAC, optional bias DACs, and several GPIO pins.

3.1.4 Communication

A simple SPI interface is included on the MCA digital control system so that the external microcontroller can communicate with the MCA. The microcontroller acts as a SPI master and polls the MCA whenever it needs to send or receive information to or from it. Every SPI transmission is 16 bits in length, beginning with a 2-bit command. The four commands are: write to configuration RAM, read from configuration RAM, read from the histogram memory, and a no-op. After the command, the remaining bits are occupied by an address parameter and optionally a data parameter (if writing to the configuration RAM). The slave responds with the answer to the master's query during the next 16-bit transmission. The response contains the data requested by the master.

The entire histogram memory can be accessed by the microcontroller via SPI commands, and all the configuration settings can be read and set through the SPI

interface as well. This gives the microcontroller complete control over the MCA. The histogram can be transmitted very efficiently and quickly for fast updates. However, the MCA does not have the capability to contact the microcontroller at will; it can only reply to the queries made by the microcontroller. Therefore, the microcontroller must take care to ensure the MCA does not have any overflowing bins by polling it regularly.

3.2 Microcontroller Digital Core

The microcontroller is tasked with setting up the MCA, polling the gamma ray histogram from it, and performing the MLPNN computational intelligence algorithm to identify which isotopes are present in the histogram. It is also able to communicate with a computer for easy control and data collection, but a computer is not necessary for the microcontroller to operate.

The microcontroller features a 16-bit processor written in VHDL, synthesized with RC, and translated into a layout with EDI. It is fabricated using a 0.18 μm process with 8 metal layers, and is 1.5 mm by 1.5 mm in size. The processor is based on the TI MSP430 architecture and instruction set, and features a 4 KiB boot ROM, a 32 KiB RAM, and various peripherals, including SPI and GPIO. The microcontroller has no FLASH memory or EEPROM of its own, and uses an external FLASH chip to store software and other large pieces of data. The standard TI GCC compiler is used to compile software for it, which is written to the FLASH memory and loaded by the bootloader upon reset or power on.

3.3 Simulation and Timing

3.3.1 MCA

The speed of the digital control system within the MCA partially determines the maximum number of counts per second (cps) it can process. The MCA maximum clock speed is simulated at 64.5 MHz. For each incoming pulse, the system requires at least 3 clock cycles, the maximum ADC comparator settling time, and the time it takes the pulse to decay back under the noise floor before another pulse may be admitted. At 64.5 MHz, 3 clock cycles takes 15.5 ns, and the maximum ADC comparator settling time is 96.8 ns. The pulse duration can be as short as hundreds of nanoseconds to as long as hundreds of microseconds, depending on the CSA feedback resistor and capacitor and the scintillator type. However, for the sake of this computation, assume that the scintillator and CSA configuration result in pulses similar to the pulse in Figure 2.8, which takes about 1.5 μs to return to baseline. Therefore, the total time it takes to process a pulse is about 1.61 μs. If a pulse train of back-to-back pulses were to arrive at the CSA (not a realistic scenario), it could process 621,000 counts per second.

A more realistic measure of the count rate involves the probability of two or more radiation detection events occurring simultaneously [8, 18]. The MCA is prone to paralyzable dead time, meaning that if a second pulse occurs before the first has finished, every other pulse that occurs before the dead time expires will extend the dead time. The effective MCA count rate, or the rate at which the MCA successfully processes pulses, is given with the expression:

$$R_{\text{MCA}} = ne^{-n\tau_p} \tag{3.1}$$

where n is the rate at which the detector collects pulses and τ_p is the time it takes to process each pulse, a fixed value. The theoretical maximum MCA count rate occurs when $n = 1/\tau_p$, or 621,000 detection events per second, and is therefore $R_{\text{MCA}} = 228$ kcps.

3.3.2 Microcontroller

The microcontroller maximum clock speed simulates at around 45 MHz. The MLPNN, with 64 input neurons, hidden bias terms, 15 hidden neurons, and 5 output neurons, requires 1,050 multiplies.

Chapter 4

Artificial Neural Network

The multilayer perceptron neural network implemented in software on the microcontroller is designed to use the gamma ray histogram gathered by the MCA to identify which isotopes are contained within the radiation source. However, many microcontrollers are quite computationally limited. Common limitations of microcontrollers include (in a non-exhaustive list): relatively slow clock rates ($<$ 100 MHz), small memory and execution bandwidths (8- or 16-bit), small RAM memories ($<$ 100 kB), and the lack of a floating point unit (FPU). To overcome these challenges, the embedded MLPNN presented in this work is designed for use with a low resolution gamma ray histogram using only fixed point arithmetic, small amounts of program, RAM, and FLASH memories, and few hidden and output neurons. The result is a fast, low memory embedded MLPNN with acceptable accuracy and identification rates.

4.1 Embedded Implementation

The three main goals for the embedded MLPNN, fast operation in real-time, low memory, and acceptable accuracy, is accomplished using several techniques. The first

is done by using a small number of input neurons and reducing the number of hidden neurons. This is shown by the total number of matrix multiplies required to perform the feedforward operation:

$$\text{Matrix Multiplies} = n_h(n_i + n_{hb}) + n_o n_h \tag{4.1}$$

where n_{hb} is 0 if there are no hidden bias terms, and 1 if there are. Reducing n_h significantly reduces the number of multiplies, while reducing n_i also does to a slightly lesser extent. This MCA has a 6-bit ADC which yields a 64-bin histogram, and every bin maps to one and only one input neuron, as in [15]. It should be noted that other MCAs may have many more bins, in which case the histogram should be rebinned to have fewer bins in order to fulfill this condition. Second, the number of hidden neurons must be minimized while still producing acceptable results. There is not a definite way besides trial-and-error to determine what the minimum value for n_h should be, but in general, $n_h > n_o$.

Another way to improve speed and reduce the memory footprint is to use only fixed point arithmetic. Floating point numbers usually consume 32 bits each or more, and all operations on them must be emulated if no FPU is included on the microcontroller, at a very significant time cost. On the other hand, fixed point addition and subtraction are usually atomic operations (or almost atomic), and typically use fewer bits than floating point. Furthermore, many microcontrollers include a hardware multiplier, a logic circuit that multiplies fixed point numbers quickly. All synaptic weights, vectors, inputs, and outputs of the embedded MLPNN presented here use fixed point numbers exclusively. However, fixed point numbers cannot dynamically scale their magnitudes, which causes a cascade of errors to permeate through the neural network, described in [19].

The activation function, which in floating point systems is usually the logistic sigmoid function, also must be fixed point. Therefore, a fast logistic sigmoid approximation is used in its place.

Finally, the weights are stored in an external FLASH memory to conserve program space and RAM.

4.1.1 Q Numbers

The weights, vectors, inputs, and outputs of the embedded MLPNN are stored as fixed point numbers known as Q numbers. A Q number is a representation of a decimal value using a fixed point variable, notated as $Q_{a,b}$. The value of a represents the number of integer bits in the Q number, while b represents the number of fractional bits. Signed Q numbers include an additional sign bit, while unsigned do not and are implied to be nonnegative. Signed Q numbers have a range of $[-2^a, +2^a - ((2^b - 1)/2^b)]$ and have a precision of $1/2^b$. Unsigned Q numbers have a range of $[0, +2^a - ((2^b - 1)/2^b)]$. Errors occur if an operation attempts to set a Q number outside its allowed range or if greater precision is needed. The implications of Q numbers can cause problems if the system is not designed carefully.

Table 4.1 lists each data structure in the embedded MLPNN, what type of Q number it is, the number of elements it contains, and the number of bytes of RAM it uses.

Data Structure	Symbol	Q Type	Elements	Bytes
Unnormalized Input Vector	$\mathbf{x}$	Unsigned $Q_{0,24}$	n_i	$3 \times n_i$
Hidden Weights	$\mathbf{W}_1$	Signed $Q_{8,15}$	$n_h \times n_i$	$3 \times n_h \times n_i$
Hidden Decision Vector	$\mathbf{d}_1$	Signed $Q_{0,15}$	n_h	$2 \times n_h$
Output Weights	$\mathbf{W}_2$	Signed $Q_{8,15}$	$n_o \times n_h$	$3 \times n_o \times n_h$
Output Vector	$\hat{\mathbf{y}}$	Signed $Q_{0,15}$	n_o	$2 \times n_o$

Table 4.1: Embedded MLPNN Q Number Data

Note that the activation vectors are missing because the software computes the decision vector of each layer as soon as it computes a new value of the activation vector, which saves RAM memory.

Two Q numbers can be added or subtracted from one another the same way as any fixed point number as long as both addends have the same b value. Two Q numbers can be multiplied, but the resultant b value will be the sum of the b values from the multiplicands $b_1 + b_2$. Often, the resultant of the multiplication is bit shifted and rounded so that it fits in a smaller data type.

4.1.2 Logistic Sigmoid Approximation

Many floating point MLPNN systems utilize the logistic sigmoid function $f_a(x) = 1/(1 + e^{-x})$. However, fixed point systems cannot directly compute the exponential without converting to floating point, performing the calculation, and converting back to fixed point, which is a very computationally expensive operation. A logistic sigmoid approximation that takes advantage of bit shifting and powers of two is used instead [20]. The approximation is two back-to-back parabolic functions that follow the shape of a sigmoid in general:

$$\hat{f}_a(x) = \begin{cases} 0 & \text{for } x \leq -4 \\ \frac{x^2}{32} + \frac{x}{4} + \frac{1}{2} & \text{for } -4 < x \leq 0 \\ \frac{-x^2}{32} + \frac{x}{4} + \frac{1}{2} & \text{for } 0 < x < +4 \\ 1 & \text{for } x \geq +4 \end{cases} \tag{4.2}$$

The approximation is implemented with two bit shifts, two additions, one multiply, one to three comparisons, and a possible subtraction/negation. The algorithm is

significantly faster than using floating point operations, especially without an FPU. A depiction of the logistic sigmoid approximation can be found in Figure 4.1.

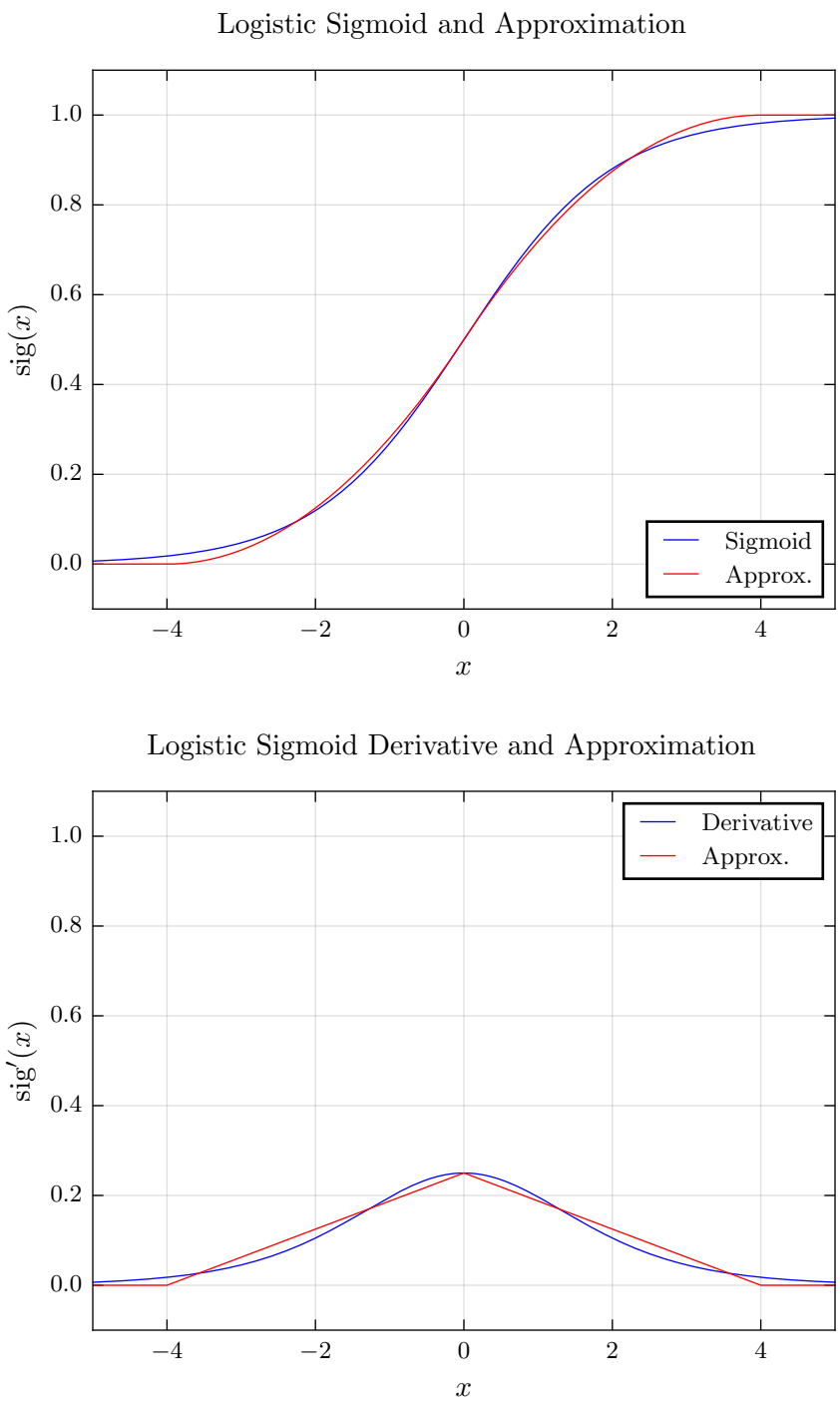

Figure 4.1: Logistic Sigmoid and its Fixed Point Approximation

While the approximation works well for this application, there is an important

difference between it and the logistic sigmoid. First, the approximation saturates to 0 if $x \leq -4$ and to 1 if $x \geq +4$. The logistic sigmoid only approaches 0 and 1 as x approaches negative and positive infinity, respectively. The error is small and irrelevant in feedforward mode, but can become a problem during backpropagation, where the derivative of the approximation is needed. If x lies outside the range $-4 < x < +4$, the derivative will be zero, which can make the backpropagation algorithm freeze.

The embedded MLPNN uses the logistic sigmoid approximation for both the hidden and output layer activation functions.

4.1.3 Receiver Operating Characteristic

The embedded MLPNN outputs a vector of $Q_{0,15}$ numbers that indicate the relative abundances of the desired isotopes and background, but often a binary decision is needed for if a particular isotope is present or not. To provide a binary decision, an ROC is created from the training set data for each isotope. A threshold value for each isotope is extracted from the ROCs that would result in less than a 5% chance of signaling a false positive and greater than a 95% chance of signaling a true positive. The threshold values are stored and compared to the embedded MLPNN outputs to make the binary decision for each isotope. Figure 4.2 provides an example of an ROC for Cs-137. The x-axis marks the MLPNN abundance output for Cs-137, while the y-axis indicates the probability of that outcome occurring. The red curve represents the CDF for a true negative, while the green curve is the CDF for a true positive. The lime dashes at the bottom indicate MLPNN outputs generated by sources with the desired isotope included, while the orange dashes indicate MLPNN outputs generated by sources without the desired isotope. The blue highlighted region marks the possible

range of thresholds that satisfy the aforementioned conditions, and the blue dashed line depicts the chosen threshold.

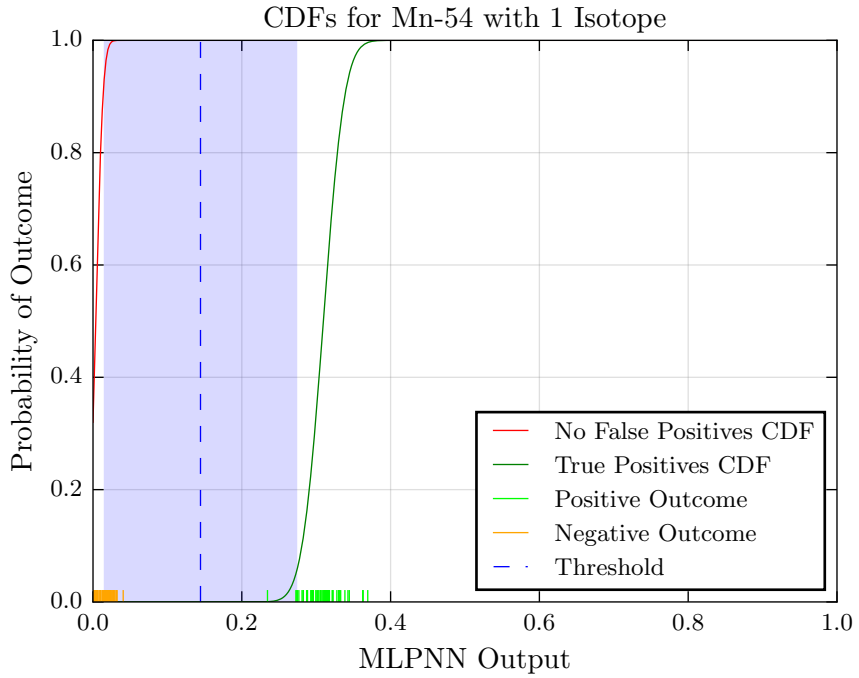

Figure 4.2: Example Receiver Operation Characteristic CDFs

4.2 Training

The training for the embedded MLPNN is done on a remote computer to save memory and time on the microcontroller, and to improve accuracy. The training is done with floating point operations (which is not a concern, as mainstream computer processors have good FPUs). Once the training is complete, the synaptic weights are converted to fixed point numbers and loaded onto the FLASH memory for the microcontroller to use.

The design parameters used for the training are listed in Table 4.2.

Parameter	Symbol	Value
Input Neurons	n_i	64
Hidden Neurons	n_h	15
Output Neurons	n_o	5
Learning Gain	γ_g	0.75
Momentum Gain	γ_m	0.75
MSE Target	–	10^{-5}
Max Epochs	–	10,000

Table 4.2: Embedded MLPNN Design Parameters

4.2.1 Training Set Preparation

The MLPNN training incorporates histograms containing five isotopes: Cs-137, Co-60, Ba-133, Mn-54, and Na-22. Of the five, Na-22 is (rather arbitrarily) selected to be an undesirable isotope, while the rest are selected to be desirable and are given an output neuron. The background radiation is also desirable and given an output neuron. Doing so simulates a noise source manifested as an extra radiation source in the system that the neural network should ignore. The MCA is used to collect many histograms of all combinations of the five isotopes, plus many without any of them to gather information about the background radiation. The histograms are normalized such that the bin with the greatest number of counts in each histogram is equal to 1. The normalized histograms represent the input vectors of the training set.

Next, the relative abundances of each desired isotope are calculated for each histogram and used as the output vectors of the training set. First, the background is subtracted from the histograms using the data from the histograms with no isotopes. Then, if there is only a single isotope known to be in the histogram, its abundance is equal to the total counts in the background-subtracted histogram divided by the total counts in the unmodified histogram. The background abundance makes up the remainder such that the output vector sum is 1. However, if more than one iso-

tope is known to be in the histogram, each one must be isolated by subtracting the counts from other isotopes to calculate its abundance. This is done by subtracting all but one of the background-subtracted, single-isotope histograms from the desired background-subtracted, multi-isotope histogram. The counts that are left over are a good approximation of the contribution the remaining isotope made to the histogram. The abundance of the remaining isotope is calculated by summing the left over counts and dividing them by the total counts in the unmodified multi-isotope histogram.

4.2.2 Backpropagation

The training set data is presented to a floating point backpropagation algorithm running on a modern computer and is executed until the error is sufficiently reduced. The receiver operating characteristic is generated by analyzing the outputs of the trained neural network when stimulated by each histogram and computing the desired binary threshold. The synaptic weights and ROC binary thresholds are converted to fixed point and loaded on the FLASH, and are ready to be used with new data.

Chapter 5

Testing and Results

This chapter describes various tests that were performed on each component of the radiation detection system, whence the results were obtained. First, the fabrication of the ASICs is described, followed by a brief description of the system functionality. Then, the tests involving the main analog circuits are presented, including the CSA, peak detector, and ADC. Next, the embedded MLPNN accuracy and cost are detailed, followed by the power consumption of the system.

All radiation testing performed on the MCA was done using a lanthanum bromide scintillator attached to a PMT with a cathode bias of 900 V.

5.1 Fabrication

The MCA ASIC was fabricated using ON Semiconductor's C5N 0.5 μm, 3 metal layer process. Each die measures 3 mm by 3 mm, and the physical layout consumes almost all of the available area on the chips. The layout contains 80,593 MOSFETs, 422 resistors, 273 capacitors, and 37,943 nets. A photograph of the MCA ASIC can be found in Figure 5.1.

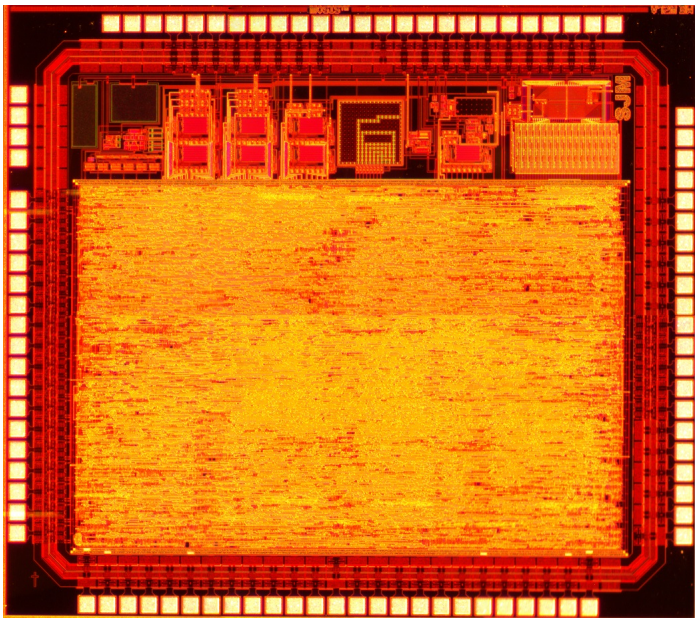

Figure 5.1: MCA ASIC Die Photograph

A testbench PCB was designed, fabricated, and assembled to test the MCA. It included necessary components such as regulators, communication channels, and a PMT connector. It also included several components for testing and backup, such as a DAC to test the on-chip ADC and several other DACs to provide external bias voltages and references, should the on-chip biases fail. A rendering of the MCA PCB can be found in Figure 5.2.

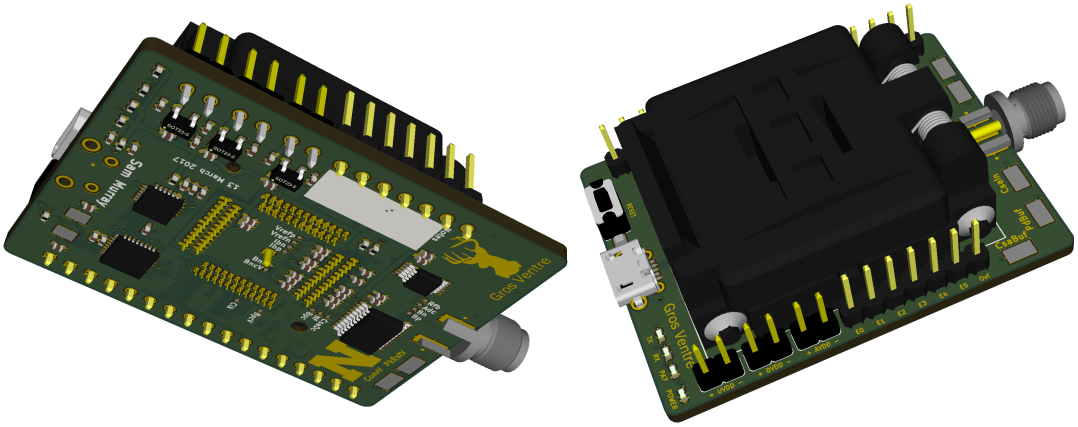

Figure 5.2: MCA PCB Rendering

A software package was written to provide human interface GUI to control the MCA. The GUI was used for testing the functionality of the MCA, configuring its parameters, and collecting histogram data from its internal memory.

The microcontroller ASIC was fabricated using IBM's 8RF 0.13 μm, 8 metal layer process. Each die measures 1.5 mm by 1.5 mm, and the physical layout consumes the entire available area on the chips. The microcontroller ASIC also includes an RF front end and a different CSA and ADC architecture for radiation detection, but only the processor was used for the work in this thesis. Figure 5.3 depicts a die photograph of the microcontroller ASIC.

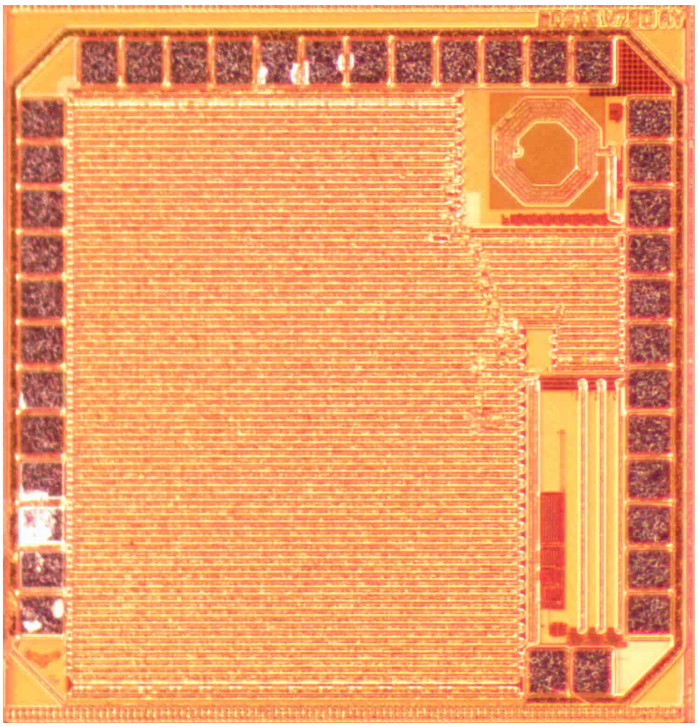

Figure 5.3: Microcontroller Die Photograph

A testbench PCB was also designed for the microcontroller ASIC to interface with the MCA PCB. Software was written for the microcontroller to create a medium between the MCA and a host computer and to implement the embedded MLPNN.

5.2 Functionality

The MCA and microcontroller were tested using the PCBs and software, and were able to produce measurable results. After much tweaking of the MCA settings, valid gamma ray histograms were produced. Figure 5.4 provides some example histograms that were collected from the MCA. The histograms have the same y-axis scale to show the relative strengths for each isotope.

Cs-137 has a single photopeak at 662 keV and a Compton edge associated with the peak. Co-60 has two photopeaks at 1,173 keV and 1,333 keV, two Compton edges, and very weak single and double escape peaks from pair production (not visible). Ba-133 has a relatively low energy spectrum with photopeaks at 80 keV and 356 keV and a Compton edge to go with the larger peak. Mn-54 has a rather weak single photopeak at 835 keV.

Unfortunately, bin 7 cannot accumulate any counts in memory because the bin has excessive DNL, and always displays zero counts.

5.3 Analog Front-End

5.3.1 CSA

The CSA was tested with real gamma ray charge pulses emitted from a Cs-137 source. The CSA output drives both the peak detector and a buffer. The buffer exits the MCA ASIC via an IO pad so that the output of the CSA can be measured on an oscilloscope. However, the buffer introduces some degree of nonlinearity to the CSA output, particularly for larger voltages, along with a dc offset. Unfortunately, probing the CSA cannot be used to measure the linearity or bandwidth of the CSA itself. This limits the tests that can be performed on the CSA that produce quantifiable

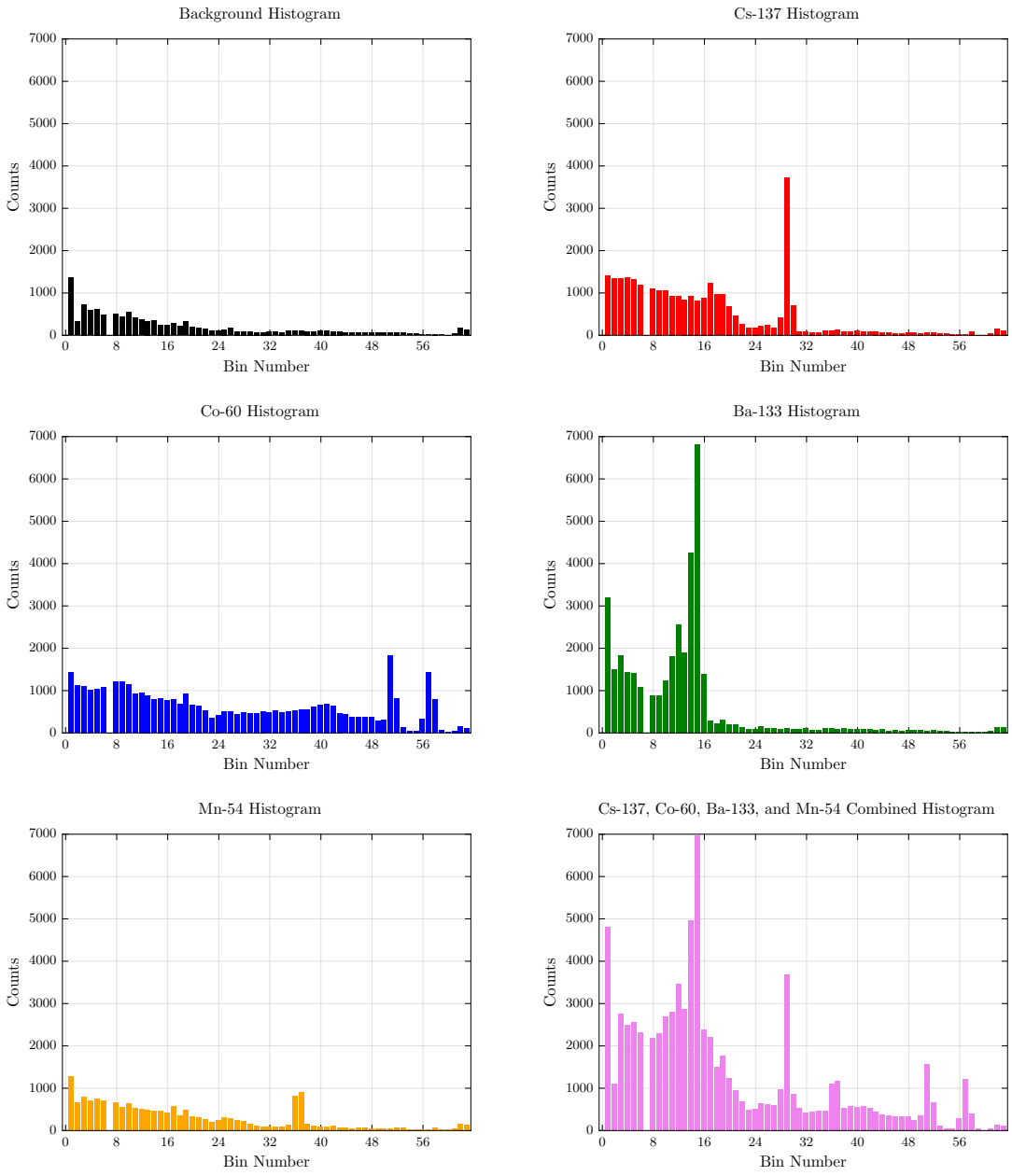

Figure 5.4: Example Histograms Collected from Radiation Detection System

results. However, the function of the CSA can be readily examined with an oscilloscope measurement, such as in Figure 5.5, which shows a 662 keV pulse from Cs-137 as detected by the CSA.

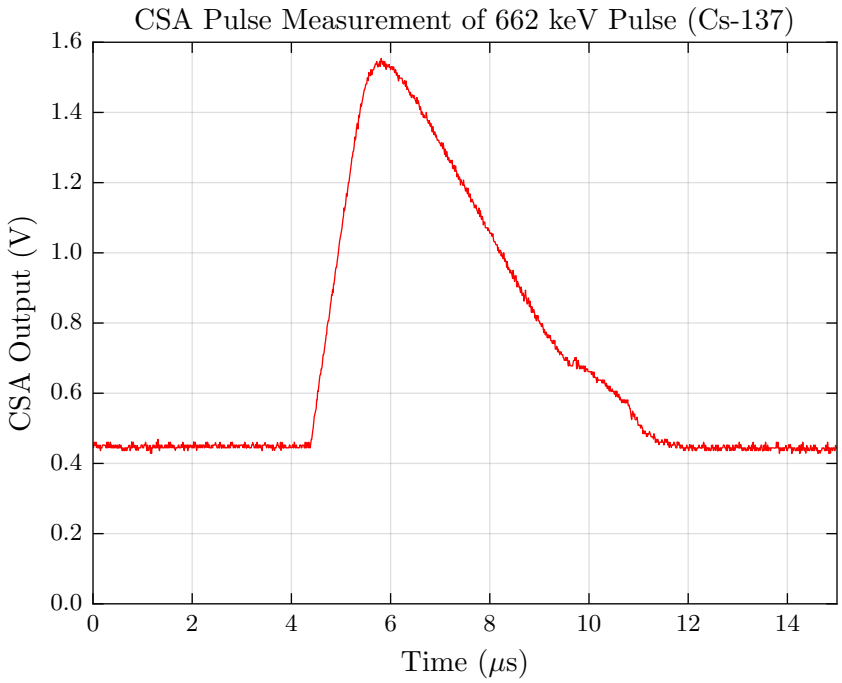

Figure 5.5: CSA Pulse Measurement

By visual inspection of the oscilloscope captures, the CSA was able to integrate the incoming charge pulses reliably. The feedback capacitor and triode FET could be configured to fine-tune the height and duration of the pulses. The peak detector caused some charge stealing from the CSA output (look carefully at Figure 5.5 at the 8.8 μs mark), slightly distorting the output. This was irrelevant, however, because the ADC measures the peak detector output, not the CSA output.

Notably, the duration of the charge pulses is much longer than the simulator models suggested. Referring back to Figure 2.8, the simulator predicted an ordinary pulse duration somewhere around 1.6 μs. However, the testing results, as depicted in Figure 5.5, suggest average pulse durations around 7 μs. This results in the count rate

being over four times slower than predicted in the simulator. Choosing a different $V_{R_{\text{fb}}}$ and C_{fb} value could shorten the pulse tail while retaining the height, but the first part of the pulse cannot be hastened. The difference between the simulated result and the test result is likely due to the simulated pulse model being shorter than what it would be in reality.

5.3.2 Peak Detector

The peak detector has the same nonlinear buffer attached to it for viewing its output on an external oscilloscope, and thus cannot be tested for linearity. However, useful information can still be extracted from scope captures, as exemplified in Figure 5.6.

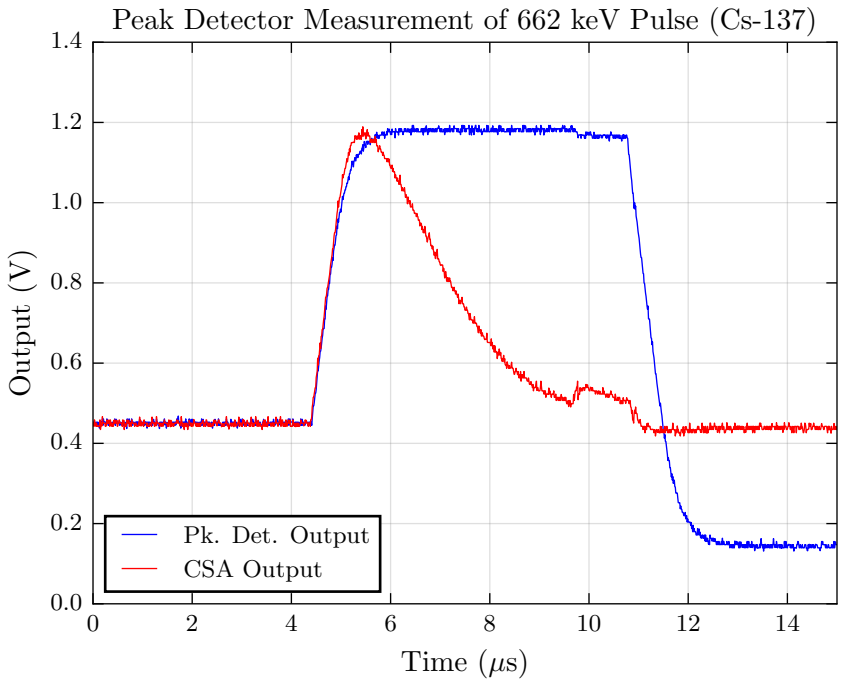

Figure 5.6: Peak Detector Pulse Measurement

The peak detector tracked the CSA output well, leveling off at the maximum CSA pulse voltage. A small amount of charge was drained off the output of the peak

detector when the CSA output dropped below the noise floor (at 8.8 μs), leading to a small voltage decrease. This does create a small amount of bias, as the ADC converts the voltage *after* the small charge is taken from it. The stolen charge is nearly identical for all peak detector voltages, meaning that the ADC sees only a dc shift, which can easily be accounted for with the MCA settings.

The peak detector was tested and measured to settle back to the baseline upon being reset at a rate of 1.008 V μs^{-1}. Peaks as high as 3 V then would take as long as 3 μs to fall back to the baseline. The digital control system has a timer that can be configured to disallow any more pulses from being processed while the peak detector settles back to the baseline. This also severely limits the maximum count rate to a value much below the theoretical value of 228 kcps. If the actual typical pulse duration and the actual peak detector settling time are taken into account, the MCA dead time is about $\tau_p = 8.7$ μs per pulse. Equation 3.1 therefore predicts a maximum count rate of 42.3 kcps.

5.3.3 ADC

The ADC was tested for linearity by sweeping its input with an external DAC voltage and reading its binary output value. The results of the sweep can be found in Figure 5.7. The green fill in the graph indicates sections that the ADC flagged an overflow value, indicating that the input voltage was greater than its positive reference voltage. Note that the DNL of bin 7 is -1, meaning that its DNL is so severe that the ADC never outputs it.

The ADC linearity sweep was used to calculate the differential nonlinearity (DNL) of the ADC, found in Figure 5.8. Note that some bins approach a DNL value of 1 LSB.

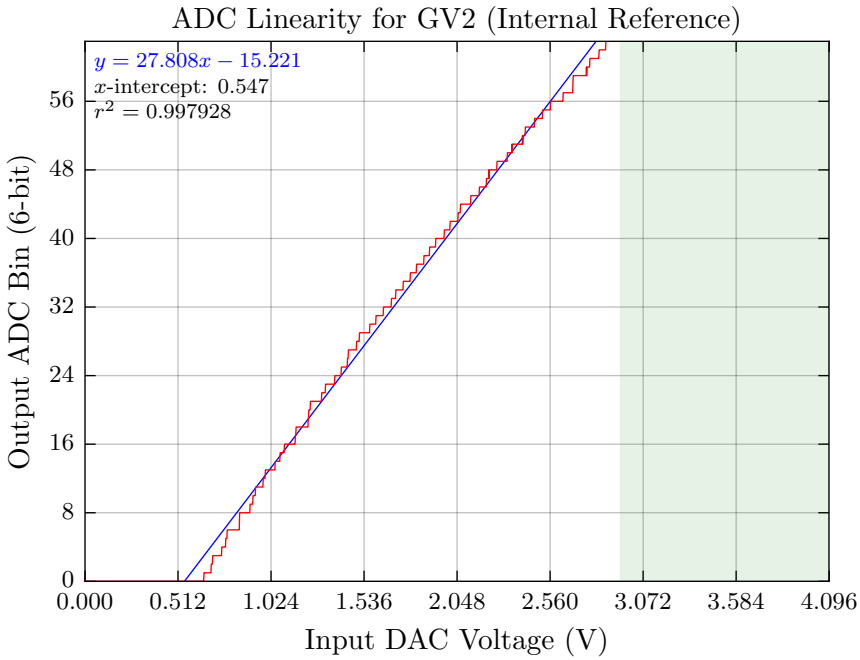

Figure 5.7: ADC Linearity Measurement

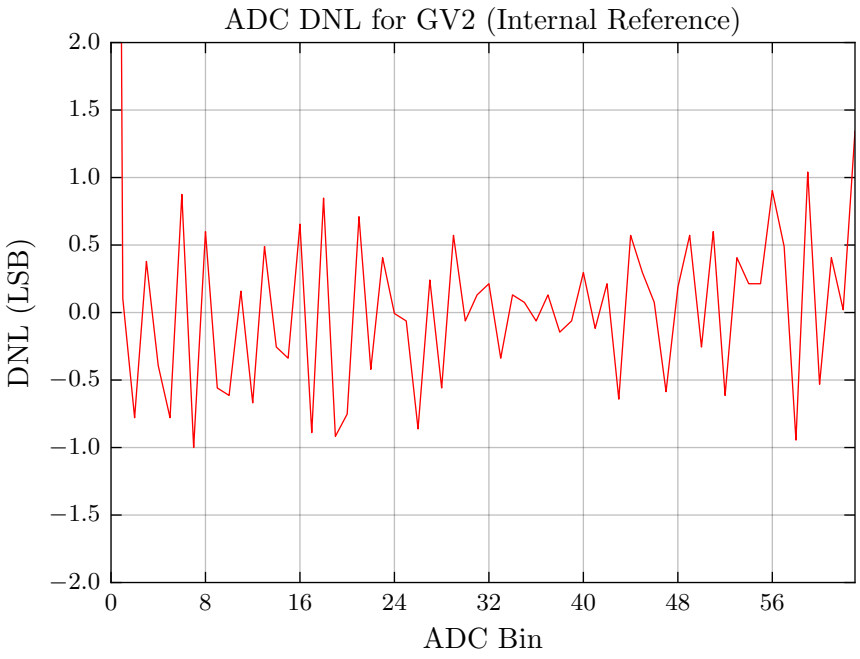

Figure 5.8: ADC Differential Nonlinearity

The speed of the ADC could not be tested directly. However, it was able to produce accurate results with the digital control system allowing it less than 0.75 μs to perform its conversions, suggesting that the simulation data was relatively accurate.

5.4 Neural Network

After the embedded MLPNN was trained, it was tested with real data collected with the MCA separately from the training set. Histograms with combinations of the five tested isotopes (Cs-137, Co-60, Ba-133, Mn-54, Na-22), including one with no isotopes and just the background radiation, were collected every 2 seconds, yielding 64 histograms for each combination, or 2,048 total histograms. The MLPNN had 64 input neurons, 15 hidden neurons, and 5 output neurons. It also used hidden bias terms for each element in the hidden activation vector.

5.4.1 Identification Rate

The expected output vector for each test histogram was determined in the same way as the training set in Section 4.2.1, calculating the relative abundances of each isotope to the background. Next, the training set histograms were normalized and presented to the embedded MLPNN, which produced its output vectors. The output vectors were compared to the expected output vectors. The MSE statistics for the test histograms can be found in Table 5.1.

After 10,000 epochs and about an hour and forty minutes of computer execution time, the training set MSE was minimized to 3.67×10^{-5}, as shown in Figure 5.9. However, after the first 4,000 epochs, only small decreases in the error were observed. The maximum number of epochs could have been reduced to save training time, but training time was not of great importance for this project.

MSE Statistic	Value
Median MSE	1.241×10^{-4}
Mean MSE	2.598×10^{-4}
Min MSE	1.863×10^{-4}
Max MSE	2.900×10^{-4}

Table 5.1: Embedded MLPNN MSE Testing Statistics

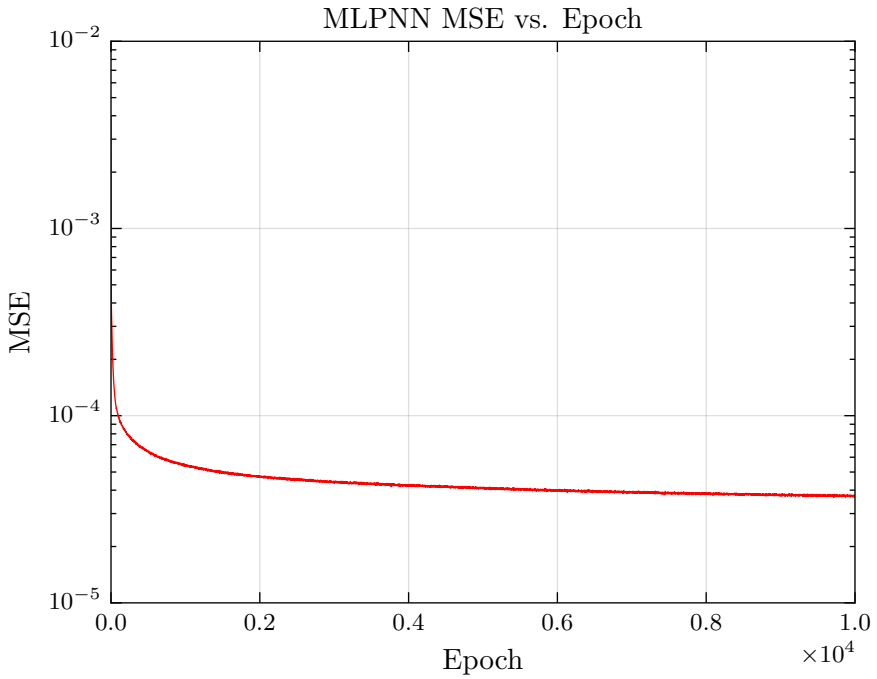

Figure 5.9: MLPNN MSE vs. Epoch

90.65% of the test histograms were below an MSE threshold of 5×10^{-4}. Furthermore, 98.35% of the test histograms correctly identified which isotopes were present in their respective radioactive sources using the ROC binary decision threshold values. After the first 10 seconds of histogram collection time (with the sources generating up to 1,000 cps, or fewer then 10,000 total counts), the identification rate stayed above 95%, as shown in Figure 5.10.

When the MLPNN and ROC were unable to correctly identify the isotopes in the

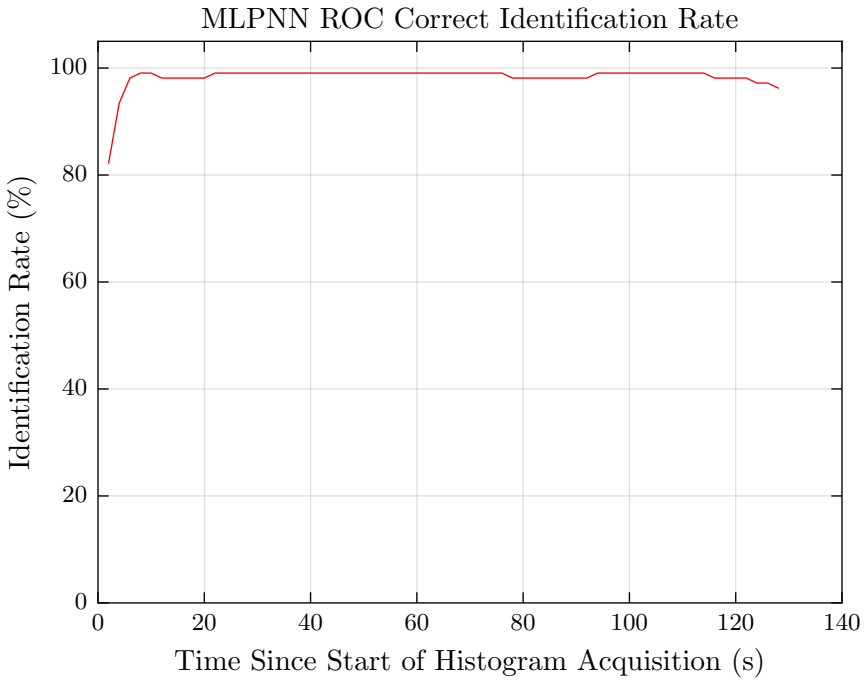

Figure 5.10: MLPNN ROC Identification over Time

system, the source usually contained Mn-54 or just background radiation only. It often would determine that a histogram with only background radiation contained Mn-54. This is because Mn-54 was the least radiative source tested, meaning that its histogram is most akin to the background histogram. The signal to noise ratio for Mn-54 is lower than the others, and the ROC threshold needed to declare Mn-54 is contained in the histogram is also quite low. Therefore, the false alarm rate for Mn-54 is much higher than the other isotopes. Almost all other misidentified histograms contained Mn-54.

5.4.2 Memory Consumption

Table 5.2 lists the amount of memory each MLPNN data structure required, based on the design parameters from Table 4.2.

Data Structure	Symbol	Memory (bytes)	RAM	FLASH
Unnormalized Input Vector	$\mathbf{x}$	192	×	
Hidden Weights	$\mathbf{W}_1$	2,880		×
Hidden Decision Vector	$\mathbf{d}_1$	30	×	
Output Weights	$\mathbf{W}_2$	225		×
Output Vector	$\hat{\mathbf{y}}$	10	×	

Table 5.2: Embedded MLPNN Memory Consumption

Both synaptic weights matrices are stored in the FLASH memory, consuming 3,105 bytes of FLASH in total. To save RAM memory, the matrices are not loaded into RAM in their entirety before being used. Instead, the embedded MLPNN recalls each element of the matrices from RAM as it needs it, one at a time. However, the unnormalized input vector, the hidden decision vector, and the output vector need to be present in RAM during the entire process, consuming 232 bytes of RAM in total.

The low memory footprint of the embedded MLPNN means that many separate "profiles" of synaptic weights matrices can be stored in the FLASH memory at once for different situations. The microcontroller can choose one based on its configuration and setting for a specific task.

5.4.3 Execution Time

Under the testing criteria with 15 hidden neurons, 1,050 multiplies needed to be performed to generate the output of the MLPNN. The MLPNN process was clocked using one of the timers included in the microcontroller. It took 1,916,208 cycles to generate the MLPNN outputs and ROC binary decisions, or less than 1,825 cycles per multiply. With the system clock running at a normal operating speed of about 22.6 MHz, it takes the MLPNN less than 85 ms to run. Some time and power savings could be made if the synaptic weights were moved from the external SPI FLASH module to

the microcontroller RAM at the expense of having fewer RAM resources for additional MLPNN profiles or other program functions. Using a hardware multiplier would also decrease execution time and power consumption.

5.5 Power

While the system is not specifically meant for low power applications, it is still worthwhile to specify the power consumption for the two ASICs.

5.5.1 MCA Power

The MCA ASIC has two power domains: digital (DVDD) and analog (AVDD). Both power domains operate at 5 V provided by two separate regulators.

The current through each power domain was measured to calculate the power. The analog supply power was steady at 46.0 mW for a typical configuration and bias point, but would rise or fall with different bias settings.

The digital supply power, however, was more variable, but independent of configuration. At a clock speed of 16 MHz, the digital supply power averaged 164.5 mW, but could dip down to 140 mW or up to as much as 200 mW. It should be noted that the digital core in the MCA does not incorporate clock gating, meaning that every flip flop is clocked on every clock cycle, leading to great power inefficiencies.

In total, the MCA consumes an average of 210.5 mW.

5.5.2 Microcontroller Power

The microcontroller ASIC has only one relevant power domain, digital (VDD), which operates at 1.2 V. The power consumption, measuring in at 3.002 mW, was tested

while the microcontroller was running a serial command line forth interpreter, which takes up the most processing time by far. Entering sleep states when no action is taking place would significantly reduce the power consumption of the microcontroller. However, the amount of power it consumes when it is awake is not significant when compared to the power the MCA consumes, rendering the effort of implementing a sleep state not worthwhile.

Chapter 6

Conclusion

This section wraps up the thesis, beginning with future work both planned and unplanned for another fabrication, and ending with some closing remarks.

6.1 Future Work

The radiation detection system could be improved in several meaningful ways to boost its accuracy and speed. Indeed, several of the following improvements are scheduled to be implemented on a new version of the MCA ASIC, but will not be fabricated in time to present in this thesis.

6.1.1 CSA Buffer

The CSA buffer on the MCA ASIC provides the only means to measure the output of the CSA with an external oscilloscope. However, the buffer introduces nonlinearity, particularly for large CSA voltages, and has its own frequency response independent of the CSA. While it is adequate for verifying the functionality of the CSA, it cannot be used to measure the linearity or the frequency response of the CSA.

The future chip removes this problem with a CSA capable of driving a pad and an oscilloscope probe, eliminating the need for a buffer. The CSA will be able to be rigorously tested for linearity, charge pulse response, and frequency response.

6.1.2 Peak Detector Return to Baseline

After a charge pulse has been processed, the peak detector output returns to the baseline by shunting its internal capacitor to ground, discharging it. The simulator indicates that the peak detector can discharge the capacitor in under 10 ns, but experimental results show a much longer duration up to 3 μs. Indeed, for very large pulses, the peak detector sometimes does not return back to the baseline, but settles at some voltage above the baseline. It becomes a problem if the peak detector settles above the noise floor level. If a small, valid peak is detected while the peak detector is above the noise floor level, the system may not process it, or may place it in a higher bin than it should be.

This problem can be addressed by increasing the reject time on the MCA digital control system, at the expense of the count rate. The future chip will implement a larger shunting FET to allow the charge on the peak detector capacitor to dissipate more rapidly.

6.1.3 ADC DNL

The ADC on the MCA ASIC has DNL that adversely affects the bin widths within the gamma ray histogram. Ideally, the DNL would be zero for each bin, leading to identically spaced energy ranges. However, the DNL is significant enough to inflate or deflate the bins, resulting in an inaccurate gamma ray histogram.

The DNL is caused by matching problems within the ADC reference and the

ADC comparators. Unfortunately, there is nothing to be done about the comparators without changing the ADC architecture. The comparators are locally matched, but not globally matched, meaning that comparators near one another are matched, but not to the entire block of them. There is not sufficient chip area to match the comparators, so they must remain as they are.

The ADC reference is realized as a resistive ladder with outputs at each rung leading to an individual comparator V_{ref} input. The rung unit resistors are also locally but not globally matched due to area limitations. Originally, the rung unit resistance was a small value, which is much more prone to process variations than a large value. For example, a small change in length of a small resistor yields a larger percent error than the same change in length on a large resistor. The simulator used for the MCA ASIC was unable to help diagnose this problem because it treats all resistors as ideal resistors without consideration for process variations. The future chip will incorporate larger rung unit resistors to combat this problem.

6.1.4 Bin 7 Always Zero

Excessive DNL in bin 7 prevents the MCA from accumulating pulses into that particular bin. Therefore, whenever bin 7 is read, it displays zero counts. The future chip will fix this problem so that the DNL in bin 7 is decreased in magnitude.

6.1.5 High Count Rate (Bit 8 Always Set)

An undetected error in the EDI layout of the MCA digital control system rails bit 8 of every histogram bin register set to logical `1`. Therefore, the actual number of counts in each bin may be 256 counts less than what is stored in the register. In order to get credible data from the MCA, no bin can be allowed to accumulate over

255 counts, or else the exact number of counts may not be accurate. However, this requires polling the MCA very frequently, which drastically reduces the count rate from the expected 228 kcps.

The future chip will implement a new EDI generated layout of the MCA digital control system. The new layout will include a fix for this problem as well as some extra features intended to decrease the time it takes to poll the bin registers.

6.1.6 MCA Digital Power

The MCA digital control system consumes 77% of the total system power, largely because the RAM was implemented in VHDL without clock gating. Every D flip flop in the RAM is clocked for every MCA clock cycle, running at 16 MHz. The power efficiency has much room for improvement by simply adding a clock gate to the RAM blocks. However, the routing density for the digital control system is about at its maximum, which may prevent even a simple clock gate from being added. Effort will be made on the future MCA chip to implement clock gating.

6.1.7 Fixed Point Multiplier

The microcontroller ASIC was implemented with a buggy hardware multiplier. The multiplier is only able to multiply positive integers, and produces errors if either of the multiplicands are negative integers. Indeed, the bug is severe enough to halt the processor if presented with a negative number. The problem was addressed by adding a compiler flag to use software multiplication rather than hardware, which makes multiplications take longer to execute, but sidesteps the faulty hardware multiplier. Future versions of the microcontroller will address this issue.

6.1.8 System on a Chip

The radiation detection system currently requires two ASICs (the MCA and the microcontroller) and several external supporting ICs. The MCA is fabricated on an old, power hungry 0.5 μm process, while the microcontroller is fabricated on a more recent (but not new) 0.13 μm process. Modern processes consume much less power for similar or better results. If combined onto a single chip, the system could be faster, more accurate and precise, much more compact, and more power efficient.

6.2 Conclusion

A complete and compact radiation detection system was designed and implemented using a scintillator and photomultiplier tube, an MCA ASIC, and a microcontroller ASIC. The MCA creates a gamma ray histogram by quantizing detected gamma ray energies and converting them into digital values using an analog front-end and a digital control system and RAM memory. The microcontroller ASIC reads the histogram from the MCA and can identify the isotopes in the radiation source with high reliability once it has collected about 10,000 total counts using a fast, low memory embedded MLPNN.

Bibliography

[1] P. Chiozzi, P. D. Felice, A. Fazio, V. Pasquale, and M. Verdoya, "Laboratory application of NaI(Tl) γ-ray spectrometry to studies of natural radioactivity in geophysics," *Applied Radiation and Isotopes*, vol. 53, no. 1, pp. 127 – 132, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969804300001238

[2] X. Lu, F. Wang, X. Jia, and L. Wang, "Radioactive analysis and radiological hazards of lime and cement fabricated in China," *IEEE Transactions on Nuclear Science*, vol. 54, no. 2, pp. 327–332, April 2007.

[3] W. L. M., W. A. H., B. Manas, and F. S. E., "Preventing the importation of illicit nuclear materials in shipping containers," *Risk Analysis*, vol. 26, no. 5, pp. 1377–1393, October 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1539-6924.2006.00817.x

[4] B. D. Geelhood, J. H. Ely, R. R. Hansen, R. T. Kouzes, J. E. Schweppe, and R. A. Warner, "Overview of portal monitoring at border crossings," in *2003 IEEE Nuclear Science Symposium. Conference Record (IEEE Cat. No.03CH37515)*, vol. 1, Oct 2003, pp. 513–517 Vol.1.

[5] J. D. Kurfess, W. N. Johnson, R. A. Kroeger, and B. F. Phlips, "Considerations for the next compton telescope mission," *AIP Conference*

Proceedings, vol. 510, no. 1, pp. 789–793, 2000. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.1303306

[6] Y. Eisen, A. Shor, C. Gilath, M. Tsabarim, P. Chouraqui, C. Hellman, and E. Lubin, "A gamma camera based on CdTe detectors," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 380, no. 1, pp. 474 – 478, 1996, proceedings of the 9th International Workshop on Room Temperature Semiconductor X- and γ-Ray Detectors, Associated Electronics and Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168900296003646

[7] C. Schrage, N. Schemm, S. Balkir, M. W. Hoffman, and M. Bauer, "A low-power directional gamma-ray sensor system for long-term radiation monitoring," *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2610–2618, July 2013.

[8] G. Knoll, *Radiation Detection and Measurement*, 4th ed. Wiley, 2010.

[9] N. Schemm, "A single-chip ultra-wideband based wireless sensor network node," Ph.D. dissertation, University of Nebraska–Lincoln, December 2010.

[10] N. Schemm, S. Balkir, and M. W. Hoffman, "A 4-μW CMOS front end for particle detection applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 2, pp. 100–104, February 2010.

[11] J. A. Schmitz, M. K. Gharzai, S. Balkir, M. W. Hoffman, and M. Bauer, "A low-power 10-bit multichannel analyzer chip for radiation detection," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[12] J. A. Schmitz, D. Rogge, M. K. Gharzai, S. Balkir, M. W. Hoffman, and M. Bauer, "A low-power radiation detection system for portable, long-duration

monitoring," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018, pp. 1–4.

[13] P. E. Keller and R. T. Kouzes, "Gamma spectral analysis via neural networks," in *Nuclear Science Symposium and Medical Imaging Conference, 1994., 1994 IEEE Conference Record*, vol. 1, Oct 1994, pp. 341–345 vol.1.

[14] P. E. Keller, L. J. Kangas, G. L. Troyer, S. Hashem, and R. T. Kouzes, "Nuclear spectral analysis via artificial neural networks for waste handling," *IEEE Transactions on Nuclear Science*, vol. 42, no. 4, pp. 709–715, Aug 1995.

[15] M. Kamuda, J. Stinnett, and C. J. Sullivan, "Automated isotope identification algorithm using artificial neural networks," *IEEE Transactions on Nuclear Science*, vol. 64, no. 7, pp. 1858–1864, July 2017.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[17] T. Carusone, D. Johns, and K. Martin, *Analog Integrated Circuit Design*, 2nd ed. Wiley, 2011.

[18] L. Abbene and G. Gerardi, "High-rate dead-time corrections in a general purpose digital pulse processing system," *Journal of Synchrotron Radiation*, vol. 22, no. 5, pp. 1190–1201, Sep 2015. [Online]. Available: https://doi.org/10.1107/S1600577515013776

[19] J. L. Holt and J. N. Hwang, "Finite precision error analysis of neural network hardware implementations," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 281–290, Mar 1993.

[20] B. Cyganek and K. Socha, "Computationally efficient methods of approximations of the S-shape functions for image processing and computer graphics tasks," *Image Processing & Communications*, vol. 16, no. 1-2, pp. 19–28, 2011.