

# Chapter 7

## Data Acquisition and Processing Systems



Mark Ellis

**Abstract** A data acquisition system is required to interpret the electrical signals produced by radiation detectors when a gamma ray, neutron or other radiation type interacts with it. The data acquisition system can take on many forms to match the wide variety of radiation detector types available and the different ways in which the signals from these detectors can be processed to give information on the detected radiation. First presented is a detailed overview of a variety of analog data acquisition topologies ranging from simple radiation event counters through to pulse shape discrimination. A comparison is then made with the digital counterparts of these data acquisition topologies, highlighting the relative merits and shortcomings of both analog and digital implementations. Next, the important factors relating to how data is processed, stored and transferred through different stages of a data acquisition chain is discussed and suggestions given on how to avoid bottle-necks and ensure balanced data flow throughout the system. Also discussed are the key challenges associated with the design and configuration of data acquisition systems for active interrogation (AI) environments such as dealing with the high event rates, capturing the time development of active radiation signatures, and techniques for working with pulsed interrogating sources.

### 7.1 Introduction

An electronic system is required to interpret the electrical signals produced by radiation detectors when a  $\gamma$  ray, neutron or other radiation type interacts with it. The common term for such electronic systems is *data acquisition* (or simply *DAQ*) and there are many different designs of such DAQ to match the wide variety of radiation detector types available and the different ways in which the data from these detectors can be processed to give information on the detected radiation. What

---

M. Ellis (✉)

Atomic Weapons Establishment, Aldermaston, Reading, Berkshire, UK  
e-mail: [mark.ellis@awe.co.uk](mailto:mark.ellis@awe.co.uk)

was once a relatively straight forward component of a radiation detection system, data acquisition has, in the last decade or so, grown into a significant area of rapid research and development in its own right. As radiation detectors have become more sophisticated in terms of complex detection modes and physical arrangement, so too have the data acquisition systems required to support them.

### ***7.1.1 Gamma-Ray and Neutron Detection***

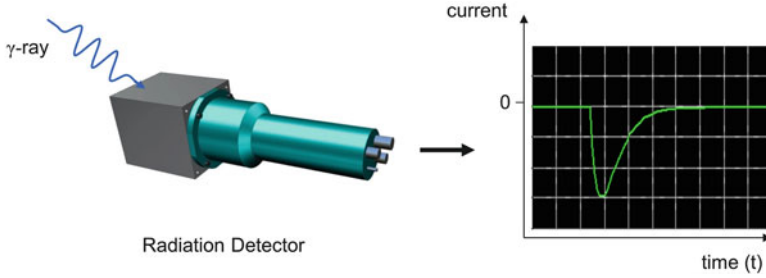
Gamma rays are high energy photons (wavelengths typically less than 10 pm), which can interact directly with the electrons of an atom. Although a large number of possible interaction mechanisms are known for  $\gamma$  rays in matter, only three major types play an important role in radiation measurements: photoelectric absorption, Compton scattering and pair production [1]. In all cases, photon energy is transferred to electron energy, which can be directly or indirectly detected in an electronic circuit.

Neutrons are uncharged particles and so cannot be detected through interactions with the electric field from the electrons within an atom. Instead, neutrons are detected through interactions with the nuclei of atoms. There are two primary interaction types: absorption reactions, where the neutron is absorbed and charged particles (and photons) are emitted, and proton recoil reactions, where the neutron elastically scatters with the nuclei. Both of these reactions produce charged particles that deposit their energy within a detection medium that can be detected with an electronic circuit [1–3].

### ***7.1.2 Overview of Data Acquisition***

In practice, a data acquisition circuit will typically be required for each radiation detector element in a given system. When a radiation particle interacts with a detector element, a certain amount of charge is liberated. The result is a short duration (typically tens of nanoseconds up to several microseconds in duration) current pulse at the detector element output (the definition of current being charge per unit time).

The detector systems of interest for AI (and indeed for most neutron and  $\gamma$ -ray detector systems) will be configured in such a way that the amount of charge liberated is directly proportional to the radiation energy deposited. The function of the DAQ circuit is to first detect the radiation event and then measure certain properties such as the radiation energy. The current pulse seen at the detector output when, for example, a  $\gamma$  ray interacts with the detector will be similar to that shown in Fig. 7.1.



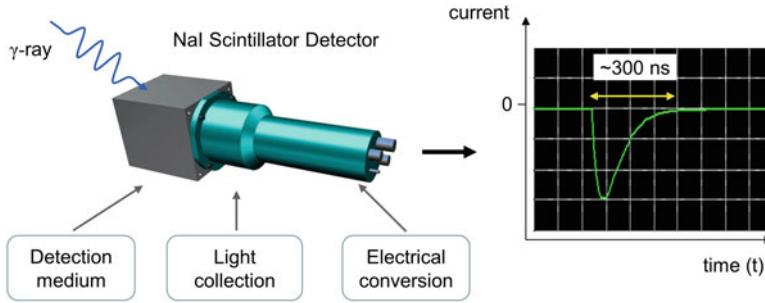
**Fig. 7.1** Current pulse from detector resulting from a  $\gamma$ -ray interaction

The traditional method for measuring the particle energy or counting events within a given energy range has been the analog DAQ chain, which is a series of analog electronics modules configured to detect and record the events; analog data acquisition is described in Sect. 7.2. The last decade or so has seen the emergence and prevalence of digital data acquisition, which has brought about a step change in the way that radiation detector signals are acquired and processed, offering more flexibility in the design of radiation detection systems; digital data acquisition is introduced and detailed in Sect. 7.3. An important part of any data acquisition system is the way in which data is formatted, stored and transferred to the different stages of the data acquisition chain. The key areas related to data handling in both analog and digital DAQ systems are covered in Sect. 7.4. Lastly, Sect. 7.5 is dedicated to the challenges associated with the design and build of DAQ systems for AI.

## 7.2 Analog Data Acquisition

As described in Sect. 7.1, a radiation detector is designed to produce a short current pulse at its output when a radiation particle is detected. Each type of radiation detector (be it semiconductor-based, a scintillator, proportional counter etc.) will have its own way of creating this current pulse; which typically involves accelerating liberated electrons across a potential difference. An example of this pulse generating mechanism for a scintillation detector (such as a sodium iodide radiation detector), is shown in Fig. 7.2.

A typical photomultiplier tube (PMT) for a sodium iodide scintillator might contain 8 or 10 dynodes and offer an electron multiplication factor of the order of  $10^6$ ; a voltage bias of several hundred volts is typically required across the dynode chain in order to achieve this high multiplication factor. In a typical setup (as described above) one might expect to observe something in the region of 1 nC of charge at the output of the photomultiplier tube for a 662-keV  $\gamma$  ray that deposits all of its energy in the scintillator. This is a quantity of charge that can be readily measured by an electronic circuit (the data acquisition system). What has been



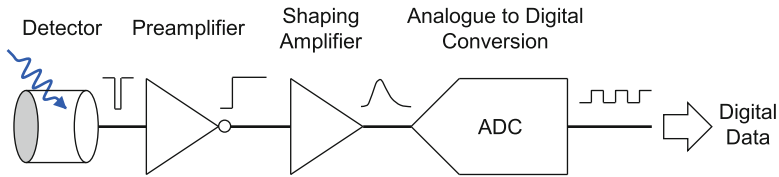
**Fig. 7.2** Particle detection and current pulse generation in a scintillator detector

described here is the current pulse the current pulse generation process for a sodium iodide scintillator but the process would be identical for, say, a liquid scintillator on the detection of a neutron or  $\gamma$  ray. A semiconductor-based detector such as High-purity germanium (HPGe) or a proportional counter such as a  $^3\text{He}$  tube would generate the current pulse via a different mechanism [1], but the general form of the current pulse will be the same as that shown in Fig. 7.2.

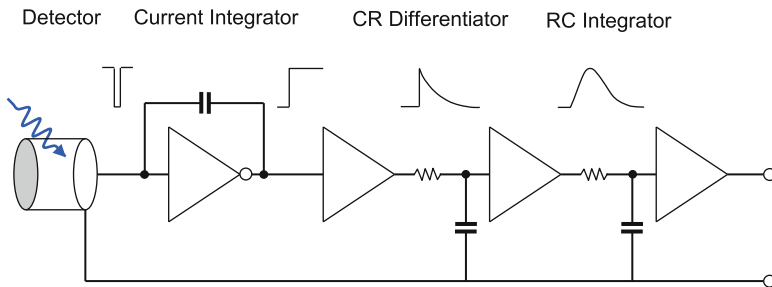
### 7.2.1 Detection Mode

There are three general modes of detector operation called pulse mode, current mode and mean square voltage mode. For AI (and similar applications) it is highly desirable to operate in *pulse mode* (as depicted in Fig. 7.2) as this is the only one of the three modes that preserves information on the amplitude and timing of individual radiation events [1]. For the remainder of this chapter, the focus will be on pulse mode detection since the vast majority of AI DAQ systems operate on this basis.

But there may be instances where the other modes may be necessary or sufficient. *Current mode* detection produces a continuous signal that is a time average of the individual current bursts and this average current depends on the product of the particle interaction rate and the charge per interaction [1]; current mode can be useful in some scenarios such as when the event rate is too high to get satisfactory results in pulse mode but where an indication of particle event rate is needed. *Mean square voltage (MSV) mode* is an adaptation of current mode where the direct-current (DC) component of the current mode signal is blocked and the signal of interest becomes the fluctuation of the current mode signal about its mean value. The MSV signal is proportional to the particle interaction event rate and, importantly, proportional to the square of the charge produced in each event. The MSV signal is thus useful when making measurements in mixed radiation fields where the charge produced from radiation events from one particle type is different from the charge produced from another particle type [1].



**Fig. 7.3** Traditional analog data acquisition chain



**Fig. 7.4** Charge integration and pulse shaping

## 7.2.2 The Data Acquisition Chain

The traditional method for detecting and acquiring information on a radiation particle that interacts with a radiation detector is an analog DAQ chain. This has the steps of charge integration, pulse shaping and then detecting the peak height of the resulting waveform from which the energy deposited by the interacting particle can be inferred; see Fig. 7.3. This method has been the mainstay of neutron detection and  $\gamma$ -ray spectroscopy for many years. The data acquisition chain shown in Fig. 7.3 is, however, just one of a number of possible analog DAQ chains, all of which are described later in Sects. 7.2.4–7.2.6.

### 7.2.3 Charge Integration and Pulse Shaping

The preamplifier and pulse shaping amplifier stages shown in Fig. 7.3 are collectively referred to as charge integration and pulse shaping. This is illustrated in more detail in Fig. 7.4.

The charge output from the detector must be collected (integrated) for the full duration of the charge pulse; it is the integrated charge that is proportional to the particle energy. Hence the full and correct name for this stage in the chain is the *charge-sensitive preamplifier* (CSP). The CSP integrates the charge pulse and

converts it to a voltage signal that can be accepted by the stage that follows. The CSP will usually have some gain to compensate for the (typically) low charge output from the detector.

The function of a pulse shaper is to simplify certain amplitude measurements, improve the Signal-to-Noise Ratio (SNR) of the pre-amplified signal and to bring the signal back to the baseline more rapidly, ready to accept the next pulse. The simplest implementation of this is a high-pass filter followed by low-pass filter (as shown in Fig. 7.4). The basic implementation in electronic components is a CR filter network (differentiator) followed by an RC filter network (integrator). Shaping amplifiers today will employ much more sophisticated shaping stages such as  $n$ th-order integrators, but the principle remains the same.

The peak of the shaped signal is smoother than the peak of the pre-amplified signal and this allows for a more accurate amplitude measurement. The SNR is improved by tailoring the frequency response of the shaping amplifier to favor the signal over the noise. The SNR improvement that can be achieved depends on the specific implementation of the shaping filter but in general an improvement in SNR comes at the expense of the duration of the shaped pulse. When configuring a DAQ system, it is typically a trade-off between noise reduction and maximizing the radiation count rate.

### 7.2.4 Peak Sensing Analog-to-Digital Conversion

The peak height of the pulse from the shaping amplifier is proportional to the energy of the interacting particle in the detector (as described in Sect. 7.2.3). A common implementation to capture this peak height value is to use a unit known as a *peak sensing analog-to-digital convertor* (ADC). The peak sensing ADC data acquisition chain is shown in Fig. 7.5. The first stage of the peak sensing ADC is to hold the peak value of the voltage pulse from the shaping amplifier. The second stage of the peak sensing ADC is a fast ADC, which converts this held voltage level into a digital number that can be stored on a computer or other digital storage device. Following a suitable calibration (establish the relationship between the recorded ADC values and deposited energy by using radiation sources of known energy), the digital number from the ADC can be scaled to give a value in units of energy.

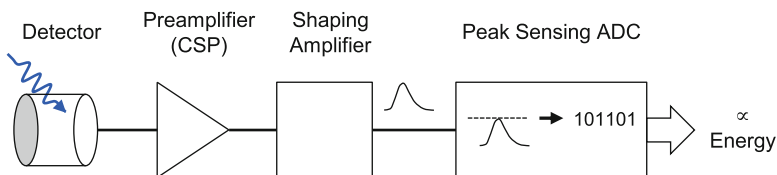


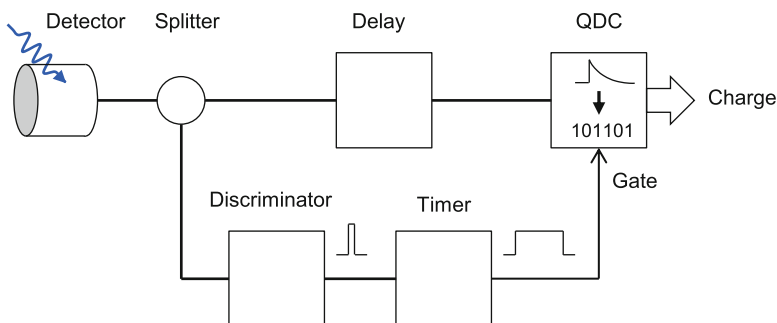
Fig. 7.5 Data acquisition chain based on a peak sensing ADC

The required resolution (number of digital bits used in the conversion) of the fast ADC will depend on the energy resolution of the radiation detector. A sodium iodide detector has an energy resolution in the region of 7% at 662 keV and so an 8-bit fast ADC is deemed sufficient (up to an energy range of, say, 7 MeV) to provide adequate energy value granularity in the digital domain to match the energy resolution that the sodium iodide detector is able to produce; less than 8 bits would be insufficient to faithfully capture the energy value of a detected particle and more than 8 bits would yield no additional information on the acquired energy value (but there may be a cost premium associated with a higher specification ADC). In contrast, a high-purity germanium (HPGe) detector has an energy resolution in the region of 0.3% (2 keV) at 662 keV and so a 14-bit fast ADC will most likely be necessary (depending on the energy range of interest) to provide adequate energy value granularity in the digital domain. For example, a 14-bit ADC will provide 16,384 ( $2^{14}$ ) digital steps, which, spread over an energy range of, say, 7 MeV gives a digital step size of 0.43 keV. A digital step size of 0.43 keV is deemed sufficient compared to the HPGe energy resolution of 2 keV at 662 keV; but one needs to bear in mind that the required digital step size will be less than 2 keV for lower energy  $\gamma$  rays (since the energy resolution, in keV, of the HPGe detector will vary as the square root of the deposited energy [1]).

The ADC stage is usually followed by a multichannel analyzer to get a visual display of the detected radiation events (see Sect. 7.2.7).

### 7.2.5 Charge-to-Digital Conversion

The Charge-to-Digital Converter (commonly referred to as a QDC) is a unit designed to directly integrate the pulse of current from the detector to give a charge value (which is itself proportional to the energy deposited by the radiation particle). This QDC-based data acquisition chain is shown in Fig. 7.6. The QDC unit is made up of two stages; the first stage integrates the current pulse to give a single charge



**Fig. 7.6** Charge-to-digital conversion data acquisition chain

value, which is converted to a voltage level via a Charge-to-Amplitude Converter (QAC); in the second stage, a fast analogue-to-digital converter is used to convert this voltage level into a digital number that can be stored on a computer or other digital storage device.

The QDC requires a charge integration window (a time period over which to integrate the detector current pulse), which is provided by the timer. The start signal for the timer is provided by the discriminator (which can be either a leading edge triggered or a constant-fraction discriminator; see Sect. 7.3.4), which provides a timer start signal when an input pulse is detected. The delay unit (prior to the QDC) is necessary to align the current pulse with the charge integration window, as there is an inherent delay introduced by the discriminator relative to the current pulse. The notable difference between the QDC DAQ chain and the peak sensing ADC DAQ chain (described in Sect. 7.2.4) is the absence of a charge sensitive preamplifier, which can be particularly advantageous where radiation event timing is of particular importance or where the shape of the current pulse must be preserved (the charge sensitive preamplifier destroys pulse shape information and also modifies the pulse timing characteristics). Also, without the long decay time associated with the charge sensitive preamplifier (typically a decay time constant in the region of 50  $\mu$ s), the undesirable effects of pulse pile-up (see Sect. 7.5.2) are also minimized. If the amount of charge per pulse produced by the detector is small then a current sensitive linear fast amplifier will be needed to amplify the current pulse to a level that can be readily accepted by the DAQ electronics; for most photomultiplier-based scintillator detectors, this pre-amplification is not required.

The QDC stage is usually followed by a multichannel analyzer (MCA) to get a visual display of the detected radiation events (see Sect. 7.2.7).

## 7.2.6 Counter-Timer/Scaler

Some applications or experiments only require a simple count of the radiation events within a certain energy range and the Counter-Timer (commonly referred to as a Scaler) DAQ chain (shown in Fig. 7.7) might be suitable and sufficient.

The CSP and shaping amplifier stages are exactly the same as that described in Sect. 7.2.4 for the peak sensing ADC DAQ chain but here the shaping amplifier output is fed to a *single channel analyzer* (SCA) unit. The SCA produces an output

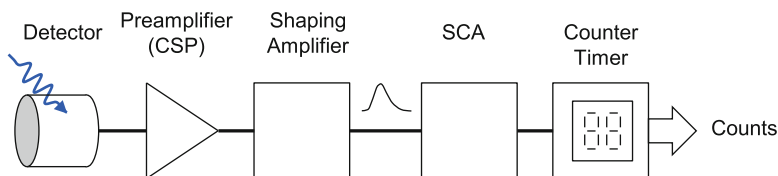


Fig. 7.7 Counter-Timer/Scaler data acquisition chain



logic pulse if the peak amplitude of the pulse from the shaping amplifier falls within a preconfigured pulse-height window. The lower range of the pulse-height window is set with a *lower level discriminator* (LLD) and the upper range set with an *upper level discriminator* (ULD); the LLD and ULD will typically be manual control knobs on the SCA unit. In its simplest form, the *counter-timer* will provide a visual readout of the number of detected events via an LED or similar display. Most likely the counter-timer would also have its own time-base, which allows it to display a count-rate.

When properly configured and energy calibrated, the counter-timer will increment each time a radiation particle, within a given energy range, is detected and will display either the absolute number of detected events or an event rate. It is not unusual for the SCA and counter-timer to be housed in the same unit.

A distinct advantage of the counter-timer DAQ chain is that it can be set up and run without the need for a host computer.

### 7.2.7 Multichannel Analyzer

The multichannel analyzer is a hardware unit that can be used at the back end of the data acquisition chain to visualize the energy (or indeed any quantity that can be represented as an analog voltage level) and frequency of radiation events. The MCA is connected to a computer or some other device that provides a visual display.

The MCA is primarily intended for use with a data acquisition chain of the type shown in Fig. 7.5, which uses a shaping amplifier followed by a peak sensing ADC but can be applied to many other types of DAQ chain. In a practical setup the shaping amplifier would be one hardware module and the MCA would be another hardware module; in this practical setup the MCA module encompasses the functionality of a Peak Sensing ADC unit but also has an interface to a computer. The interface to the computer can take on many forms; the most favored interface for modern systems is USB (Universal Serial Bus) but the MCA could, for example, be a computer add-in card that connects via PCIe (Peripheral Component Interface express). The MCA will be designed to work with a visualization and analysis program running in software on the host computer. The MCA setup is shown in Fig. 7.8.

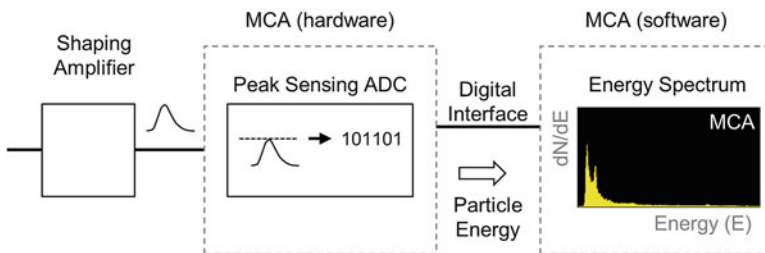


Fig. 7.8 Multichannel analyzer implementation

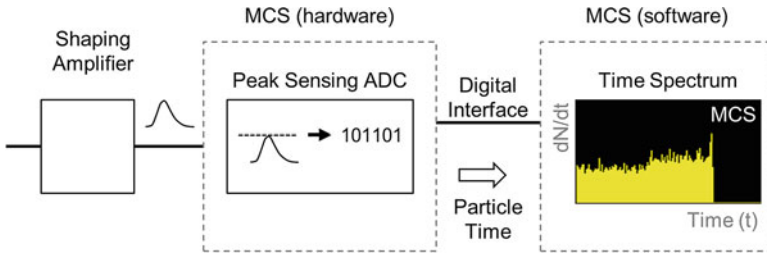
The displayed output of the MCA is a representation known as an *energy spectrum*, which is a histogram of particle energy versus the number of counts ( $N$ ) per energy interval (often referred to as an energy *bin* or energy *channel*). The process is as follows: The MCA accepts individual pulse events from the shaping amplifier; the maximum amplitude of those pulses is converted to an energy value and those energy values are added one by one to the histogram (energy spectrum). So the energy spectrum is built up over time as more radiation events are acquired. An example energy spectrum can be seen towards the right of Fig. 7.8.

On completion of a data acquisition run, the energy spectrum can be saved as a simple histogram file so it can be displayed or post-processed in another software package. In a simple MCA implementation, information on the energy and time of arrival of each event is thrown away once it has been included in the energy spectrum; but in recent years MCA implementations have been adapted to run in a mode known as *time stamp list mode*, where some information on each radiation event is retained. The time stamp list mode is discussed later in Sect. 7.3.8.

## 7.2.8 Multichannel Scaler (MCS)

The *multichannel scaler* (MCS) is a hardware unit that can be used at the back end of the data acquisition chain to visualize the time profile of radiation events. The MCS is connected to a computer or some other device that provides a visual display. The MCS is primarily intended for use with a data acquisition chain of the type shown in Fig. 7.5, which uses a shaping amplifier followed by a Peak Sensing ADC. In a practical setup the shaping amplifier would be one hardware module and the MCS would be a separate hardware module; in this practical setup the MCS module encompasses the functionality of a peak sensing ADC unit but also has an interface to a computer. Similar to the MCA, the MCS interface to the computer can take on many forms; the most favored interface for modern systems is USB (universal serial bus) but the MCS could, for example, be a computer add-in card that connects via PCIe (peripheral component interface express). The MCS will be designed work with a visualization and analysis program running in software on the host computer. The MCS setup is shown in Fig. 7.9.

The displayed output of the MCS is a representation known as a *time spectrum*, which is a histogram of particle event times versus the number of counts ( $N$ ) per time interval (often referred to as a time *bin*, time *channel* or dwell time). The MCS will have a set number of time channels, governed by the amount of storage memory in the unit; typically this will be a memory store addressed by a 16-bit bus, which equates to 65,536 ( $2^{16}$ ) time channels. The dwell time can be chosen to suit the measurements being undertaken with a typical MCS unit having a dwell time range of nanoseconds up to many 100s of seconds. The overall experiment time is the product of the dwell time and the number of assigned channels, which could be from a few microseconds up to several years. An example time spectrum can be seen towards the right of Fig. 7.9.



**Fig. 7.9** Multichannel scaler implementation

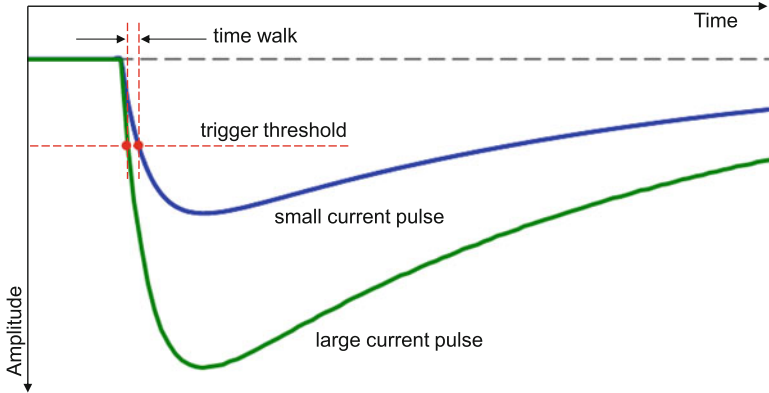
On completion of a data acquisition run, the time spectrum can be saved as a simple histogram file so it can be displayed or post-processed in another software package.

Time spectra are particularly important for AI applications since we are often interested in the rate of decay of some induce signature. Adaptations of the MCS for AI are discussed later in Sect. 7.5.4.

### 7.2.9 Triggering

A trigger is the signal that tells the data acquisition electronics to start acquiring data. The trigger could come from an external source (for example, an associated particle tag signal from a neutron generator) or it could be generated from the radiation current pulse itself (known as self-triggering). When operating in a pulse detection mode (see Sect. 7.2.1), for the data acquisition topologies described in Sects. 7.2.3–7.2.8, it is necessary to have one trigger per radiation event, so self-triggering is the usual mode of operation. Although not explicitly stated or shown in Fig. 7.5, the peak sensing ADC can usually be configured to operate in either a self-trigger mode or a gated mode, where a gate signal (with a width that encompasses the pulse from the shaping amplifier) must be supplied to the peak sensing ADC unit. Whether or not the ADC is operated in a self-triggering or gated mode is very much down to the design and needs of the experiment or measurement. If it is possible to generate an accurate external trigger then this is normally the preferred trigger solution.

Particular attention is given to the self-trigger mode because the trigger scheme used can have a significant effect on the quality and accuracy of the data acquired. As the name suggests, when self-triggering, the trigger must be generated from the detector output signal itself. An example of where a self-trigger is necessary is the QDC data acquisition chain shown in Fig. 7.6. The two most common schemes for self-triggering are Leading Edge Discrimination (LED) and Constant Fraction Discrimination (CFD).

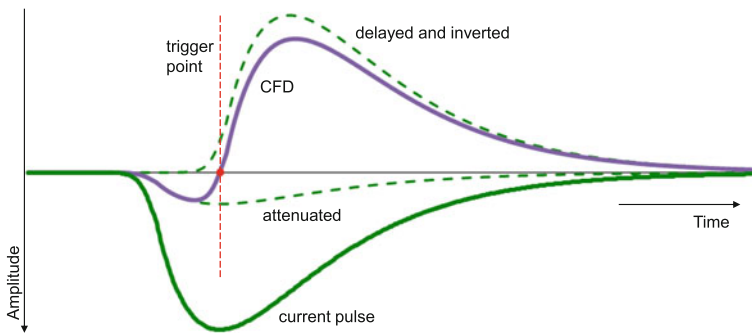


**Fig. 7.10** Illustration of the leading edge trigger method and how the trigger time is amplitude dependent

In the LED method, a suitable amplitude threshold is set and a trigger is generated if the signal crosses this threshold. The threshold is typically set to a level that is sufficiently clear of any electrical or other noise in the system. The LED trigger method is reasonably straight forward to implement and is the mode typically used in oscilloscopes and many commercially available multichannel analyzers. But the LED method does have its drawbacks particularly if signal timing is of importance. Consider the case of two different current pulses from the detector, one of low amplitude and one of high amplitude, as shown in Fig. 7.10. Both pulses have the same leading edge time constant. With a fixed trigger threshold level it can be seen that the two current pulse signals cross the threshold at slightly different times resulting in what is known as *time walk*; the time that the trigger occurs relative to the start of the pulse is amplitude dependent. If the timing of the radiation event is not critical or the amplitude range of pulses is small then time walk may be perfectly tolerable. However if the arrival time of events is of importance (for example in time-of-flight measurements or for certain pulse shape discrimination schemes) then the LED trigger method is inadequate and a more accurate trigger scheme is required.

*Constant fraction discrimination* [4, 5] is a more sophisticated trigger scheme that is designed to deal with the issue of time walk and generate a more time-accurate trigger time than the LED method. The CFD is designed to trigger at the optimum fraction of the pulse amplitude for any given pulse amplitude. In the CFD method, an attenuation-summation operation is performed on the current pulse from the detector to produce a bipolar with a zero-crossing point. The attenuation-summation operation (with a fraction setting of 0.2) is illustrated in Fig. 7.11.

The current pulse is first attenuated to a fraction of its initial amplitude. The current pulse is also inverted and delayed. The delay is chosen to make the fraction point on the leading edge of the delayed pulse line up with the peak amplitude of the attenuated pulse. The two altered pulses are summed to produce a bipolar pulse.



**Fig. 7.11** Illustration of the constant fraction discrimination trigger method and how the trigger time is amplitude invariant

The zero-crossing point of the bipolar pulse is used to generate a trigger signal that is amplitude invariant and has negligible time walk.

LED and CFD stages are typically available as separate DAQ modules but may also be built into other analog pulse processing DAQ modules.

### 7.2.10 Dead Time

The dead time of a detector system is defined as the minimum amount of time that must separate two events in order that they are recorded as two separate pulses [1]. For the vast majority of radiation detection systems used for radiological and nuclear security applications there will be a dead time associated with the pulse processing electronics (where the duration of the shaped pulse sets the minimum pulse separation before pile-up occurs). There is also a dead time associated with signal conversion for storage, as the detection system will require a finite amount of time to perform analog-to-digital conversion and to store data to memory, during which time no further pulses can be accepted.

There are two commonly used models to describe the ideal behavior of a counting system known as paralyzable and non-paralyzable response. The behavior of the two response types are virtually the same at low count rates. In a non-paralyzable system, an event that occurs during the dead time period is simply lost; so with an increasing event rate the detection system will reach a saturation count rate equal to the inverse of the dead time. In a paralyzable system, an event that occurs during the dead time period will not only be lost but will restart the dead time; so with an increasing event rate the detection system will reach a saturation point where it will be totally paralyzed and unable to record any events at all [6].

Dead time in the pulse processing electronics, as well as pileup, can lead to a loss of events; it follows that the magnitude of these losses increases with increasing count rate.

In summary, one should avoid measurement conditions under which dead time losses are high in order to minimize the errors that occur in making dead time corrections. When losses are greater than 30–40%, the calculated true count rate becomes very sensitive to small changes in the measured rate and the assumed system behavior [1].

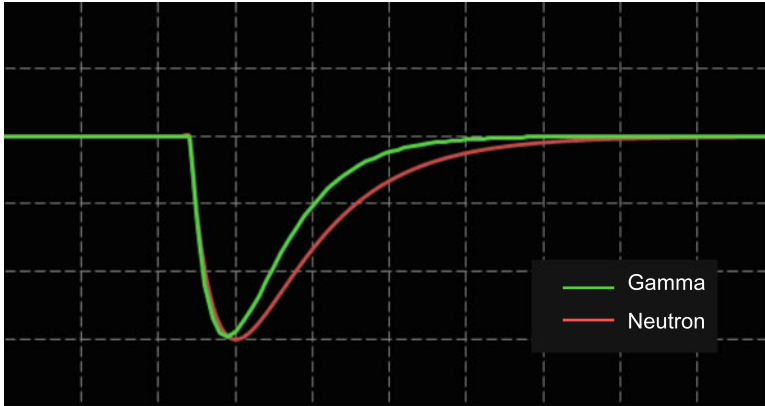
### 7.2.11 Analog Pulse Shape Discrimination

It is fair to say that all of the widely used neutron detection materials have some sensitivity to  $\gamma$  rays. In some cases the neutron response is the wanted signal and so the  $\gamma$ -ray response is considered a nuisance. In some cases both the neutron and  $\gamma$ -ray responses are important in measurements. In either case, it is necessary to separate the neutron events from the  $\gamma$ -ray events.

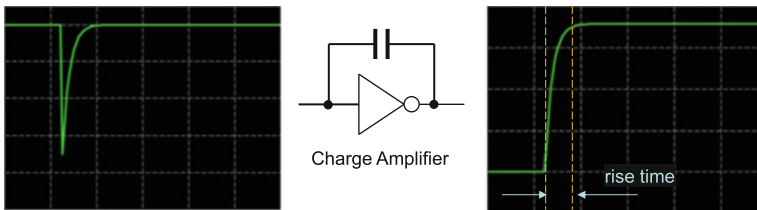
For neutron detectors that employ one of the three main nuclear absorption reaction materials ( $^3\text{He}$ ,  $^{10}\text{B}$  and  $^6\text{Li}$ ), neutron interactions and  $\gamma$ -ray interactions deposit different amounts of energy [2]. So, discrimination of neutron and  $\gamma$ -ray events by energy for these detector types is both feasible and adequate. The DAQ chains described previously in Sects. 7.2.3–7.2.8 are all suitable for neutron capture-based detectors and examples of such detectors include  $^3\text{He}$  tubes, boron-lined tubes and some LiF/ZnS scintillators. Neutron detectors based on elastic scattering (such as PSD liquids and PSD plastics) do not enjoy a separation of neutron and  $\gamma$ -ray events by energy deposition and so some other method must be used to discriminate the two radiation types. There are some scintillator materials, such as certain types of liquid hydro-carbons, where neutrons and  $\gamma$  rays give up their energy in different ways as they traverse the detection medium and this manifests as a difference in the shape of the light pulse emitted by the scintillator; in these materials, neutrons produce light pulses with a longer tail (slower decay constant) when compared with  $\gamma$  rays [7]. The cartoon of Fig. 7.12 illustrates how the PMT current pulse might look for a  $\gamma$  ray and neutron when viewed on an oscilloscope. In practice the observable difference in the  $\gamma$ -ray and neutron pulse shapes will be quite small but this difference has been exaggerated in Fig. 7.12 for the purposes of illustration.

This pulse shape difference can be used as a way of separating neutron and  $\gamma$ -ray events using a technique commonly referred to as *pulse shape discrimination* (PSD). Indeed, with a suitable choice of scintillator material, PSD can be used to separate many different radiation types (for example, beta particle and  $\gamma$ -ray separation in a CsI(Tl) crystal [8] and much of what follows in this section can be equally applied to discriminating radiation types other than neutrons and  $\gamma$  rays. For AI the key focus is usually on neutrons and  $\gamma$  rays so this will be the focus here.

There are several well established methods for performing PSD with analog electronics, the most common of which are listed below:



**Fig. 7.12** Illustration of the pulse shapes resulting from a  $\gamma$  ray and neutron as might be viewed on an oscilloscope



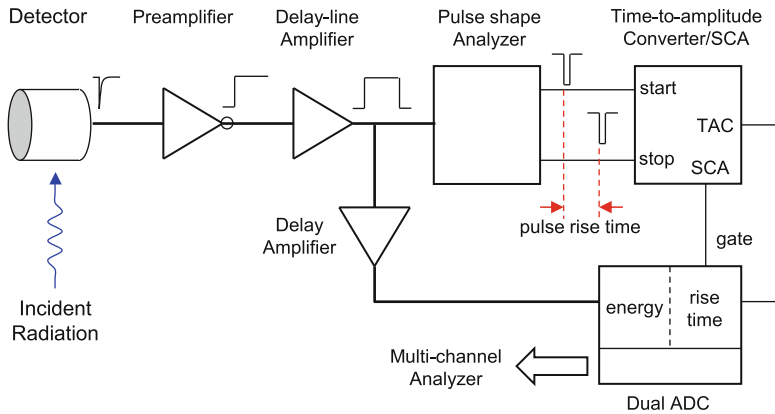
**Fig. 7.13** Illustration of the rise time discrimination method for pulse shape discrimination, as might be viewed on an oscilloscope

- Rise time discrimination
- Zero cross-over/constant-fraction discriminators
- Charge comparison
- Constant time discrimination

In the *rise time discrimination* (RTD) PSD method, timing measurements are made on the rising edge of a charge integrated detector pulse [9]. The rising edge of the charge integrated pulse is a function of the entire time development of the current pulse from the detector, which will be different for neutrons and  $\gamma$  rays. The RTD concept is illustrated in Fig. 7.13.

The amount of time taken for the charge integrated detector pulse to rise from zero to full amplitude (in practice the time interval is measured between 10% and 90% of full amplitude, or some other suitable percentage) will be different for neutrons and  $\gamma$  rays and thus this time duration measurement can be used to discriminate between the two radiation types.

One implementation of a DAQ chain to perform RTD PSD is shown in Fig. 7.14. Referring to Fig. 7.14, a *delay-line amplifier* (DLA) is used to shape the preamplifier pulse to return the signal to the baseline ready to accept the next pulse. By



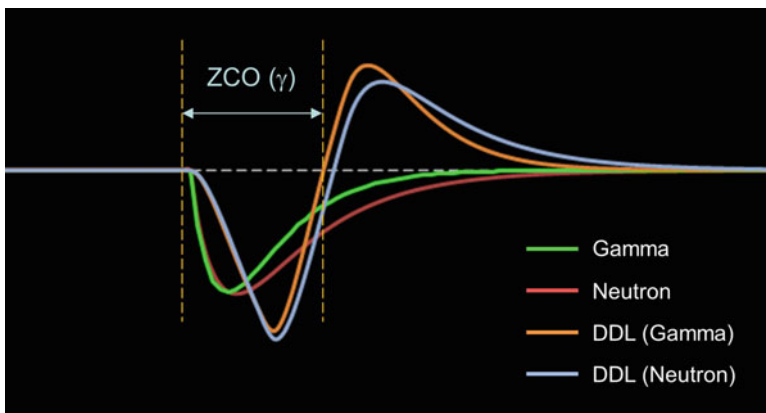
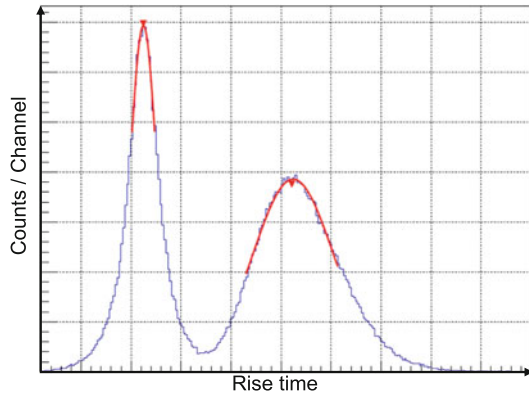
**Fig. 7.14** Rise time discrimination method pulse shape discrimination data acquisition chain

definition, the rise and fall time of the DLA is symmetrical and in principle the rise time can be measured on either edge of the DLA output signal. In practice the rise time is measured on the falling edge of the DLA output signal because the timing measurements are taken at fractional levels of the maximum amplitude, and the measurement is much simpler to make if the maximum amplitude is known beforehand. The *pulse shape analyzer* (PSA) generates a *start* signal for the *time-to-amplitude converter* (TAC) when the falling edge of the DLA output signal falls below a certain fraction of the maximum amplitude (usually 90%). The PSA generates a *stop* signal when the DLA output signal falls below a smaller fraction of the maximum amplitude (usually 10%). The TAC produces an output voltage level that is proportional to the time interval defined by the start and stop signals, which in this case is a measure of the rise time of the charge integrated pulse from the detector. The TAC output can be fed to an ADC and multichannel analyzer (see Sect. 7.2.7) to produce a histogram of the rise time, which will have a form as shown in Fig. 7.15. With properly configured hardware, the  $\gamma$  rays and neutrons will fall into two distributions determined by their rise time differences.

In practice, there is a wide statistical variation in the pulse shapes for both  $\gamma$  rays and neutrons and separation can only ever be performed to some value of statistical significance. Moreover, the amount of separation varies with energy; separation is better at higher energy due to better statistics resulting from the higher number of scintillation photons generated by the radiation interaction. It follows that the amount of separation is also dependent upon the characteristics of the scintillator material and in particular the number of photons produced per keV of deposited energy. The example of Fig. 7.15 is the rise time histogram that can be expected for a xylene-based liquid scintillator detector at an energy deposition in the region of 100 keVee (electron equivalent energy). The width of the  $\gamma$  distribution is in the region of a few nanoseconds and the neutron distribution roughly an order of magnitude greater.



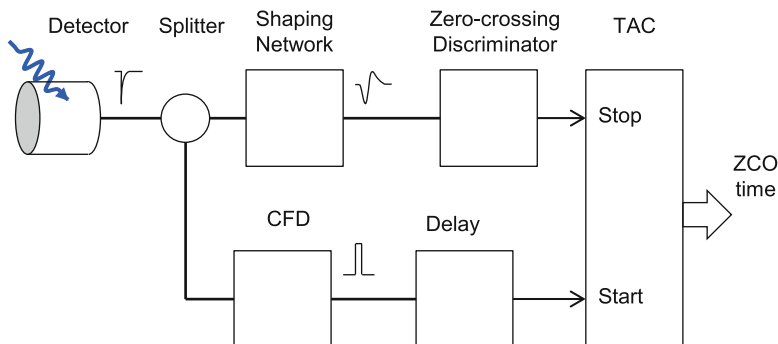
**Fig. 7.15** Example of a pulse shape discrimination rise time histogram (typical of a liquid or PSD plastic scintillator)



**Fig. 7.16** Illustration of the zero cross-over method for pulse shape discrimination

The delay amplifier output can be fed to a second ADC and MCA to produce an energy histogram. It is also possible to combine the energy value and rise-time value for each event to produce a two-dimensional histogram of energy versus rise-time.

In the *zero cross-over* (ZCO) method the individual pulses from the detector are passed through an appropriate shaping network (such as a CR-RC-CR network or a double delay line shaper) to create a bipolar pulse [10, 11]. The time at which the bipolar pulse crosses zero is an amplitude invariant function of the detector pulse shape and rise time [12]. The ZCO concept is illustrated in Fig. 7.16. The time interval between the start of the detector pulse and the zero cross-over point of the bipolar pulse will be different for neutrons and  $\gamma$  rays and thus this time interval measurement can be used to discriminate between the two radiation types. The two Double Delay-Line (DDL) traces in Fig. 7.16 are the results of the two traces shown in Fig. 7.12 after they have each undergone charge integration and fed through two stages of a delay-line circuit; the zero cross-over time for the  $\gamma$ -ray pulse is labelled in Fig. 7.16 and it can be seen that the zero cross-over time for a neutron will be longer.



**Fig. 7.17** Illustration of the zero cross-over method for pulse shape discrimination

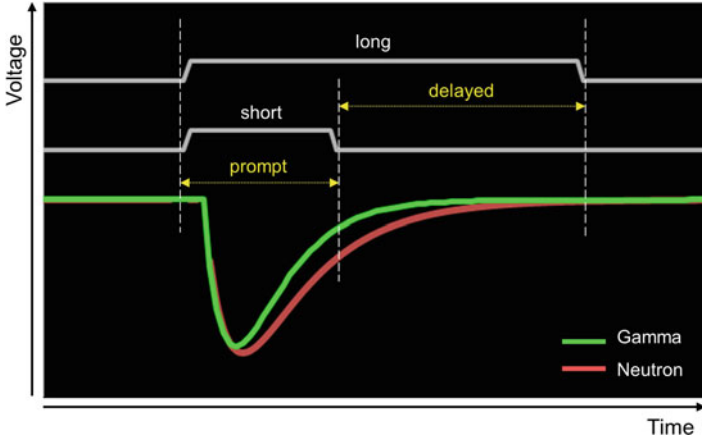
One implementation of a DAQ chain to perform ZCO PSD is shown in Fig. 7.17. The constant-fraction discriminator (CFD) generates an amplitude invariant trigger, which serves as a start signal for the Time-to-Amplitude Converter (TAC). The shaping network contains the circuitry to create the bipolar signal, which would typically be a charge integration stage followed by a double delay line (but there are many variations on this circuitry). The shaping network is followed by a zero-crossing discriminator, which generates the stop signal for the TAC. The TAC produces an output voltage level that is proportional to the time interval defined by the start and stop signals. A delay is often needed after the CFD to bring the start signal into the same time range as the stop signal.

The TAC output can be fed to an ADC and multichannel analyzer (see Sect. 7.2.7) to produce a histogram of the zero cross-over time, which will look similar to the form shown in Fig. 7.15 (generated by the rise time discrimination PSD method). The addition of a QDC stage (see Sect. 7.2.5) would allow one to generate an energy value and a ZCO value for each event making it possible to produce a two-dimensional plot of energy versus ZCO.

In the charge comparison method every detector pulse is integrated over a short time window (short gate) and a long time window (long gate). The relative amounts of the charge in these integrated gate periods determine whether the event is a neutron or a  $\gamma$  ray [13]. The charge integration periods of the charge comparison PSD method are illustrated in Fig. 7.18. The prompt and delayed time periods relate to the prompt and delayed fluorescence from the scintillator [1].

A single PSD parameter that can be used as a measure of neutron and  $\gamma$ -ray separation can be calculated as the ratio of the short gate charge to the long gate charge; or more commonly the ratio of the delayed charge component (see Fig. 7.18) to the overall charge (the long gate charge).

In the constant time discrimination (CTD) method the amplitude of every detector pulse is taken at some constant time relative to the start (or trigger point)



**Fig. 7.18** Illustration of the time gates associated with the charge comparison pulse shape discrimination method. The prompt and delayed time periods relate to the prompt and delayed fluorescence from the scintillator

of the pulse. The amplitude of the pulse at this constant time value will be different for neutrons and  $\gamma$  rays since they have different pulse shapes. Although the CTD method is relatively simple to implement in analog electronics this PSD method is particularly susceptible to time jitter and the pulse triggering scheme used (see Sect. 7.2.9) so should be used with caution.

Similar DAQ chain implementations exist for the charge comparison method and the constant time discrimination method each of which can be used to generate a histogram plot similar to that shown in Fig. 7.15 to quantify separation between neutrons and  $\gamma$  rays. The amount of separation that can be achieved will depend on a number of factors including the detection material itself, the PSD method used as well as the particular DAQ components chosen and the way in which they are configured.

PSD in the analog domain can be quite complex in terms of the hardware required and the expertise needed in setting it up. All of the PSD methods described here typically require a fair number of individual DAQ modules, which can rapidly become quite cumbersome especially when one moves beyond a handful of detection channels. The hardware complexity and high module count associated with analog PSD is a serious limitation when building the multichannel detector arrays associated with many AI systems; one is forced down the route of designing custom DAQ units just to keep the electronics to a reasonable size. The hardware complexity problem has been addressed quite elegantly with the introduction of digital DAQ and, in general, PSD can be much simpler to implement in digital hardware; see Sect. 7.3.7.

## 7.3 Digital Data Acquisition

Analog data acquisition systems, as described in Sect. 7.2, are perfectly suitable and adequate for simple radiation detection systems that have a small number of detectors but the DAQ system becomes rapidly cumbersome and costly as the number of detector channels is increased.

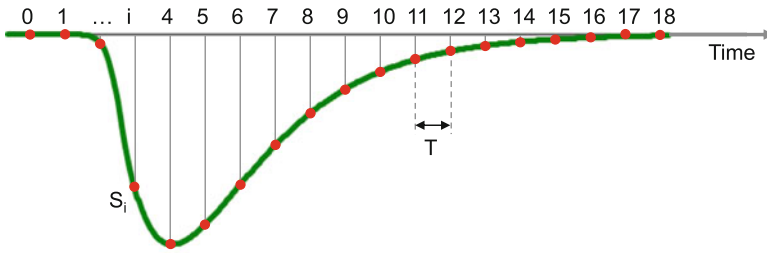
The detector systems that are commonly used for AI will typically be made up of large area detector arrays of which each element will require its own data acquisition channel. Digital DAQ systems are particularly suited to these multi-channel implementations and significantly outperform their analogue counterparts. Another significant advantage of digital processing is that the detected radiation events can be analyzed in far greater detail opening up the possibility of sophisticated detection algorithms thereby increasing overall detection performance.

### 7.3.1 Advantages of Digital DAQ

Digital DAQ offers a number of significant advantages over analog DAQ and can now be considered the DAQ of choice when designing a new radiation detection system. A summary of some of the advantages of digital DAQ is given below and the sections of this chapter that follow demonstrate how these advantages have been realized in modern digital DAQ systems:

- A significant reduction in hardware and compact solutions. A single digital hardware module is capable of performing multiple functions such as energy calculations, event timing, counting and pulse shape analysis
- More cost-effective and reliable compared with analog
- Good linearity and stability leads to good reproducibility
- Wider dynamic range and uniformity
- Digital techniques allow better correction of pile-up and noise effects due to baseline fluctuations and ballistic deficit [1],
- Easy synchronization and correlation over several DAQ channels
- Low dead time in the acquisition leading to high count rates
- Flexibility—algorithms can be tailored (changed and adapted) to better fit the application
- Tuning and calibration: software programming instead of manual control; faster and automatic

Despite the huge range of benefits that a digital DAQ system can offer there are also some disadvantages that one should always be aware of when considering whether analog or digital DAQ is most appropriate for a given application. Properly setting up a digital DAQ system requires good knowledge of digital signal processing algorithms and the relevant control parameters; more so than with an equivalent analog DAQ system. It can take more time for beginners to understand and configure



**Fig. 7.19** Illustration of the sampling process

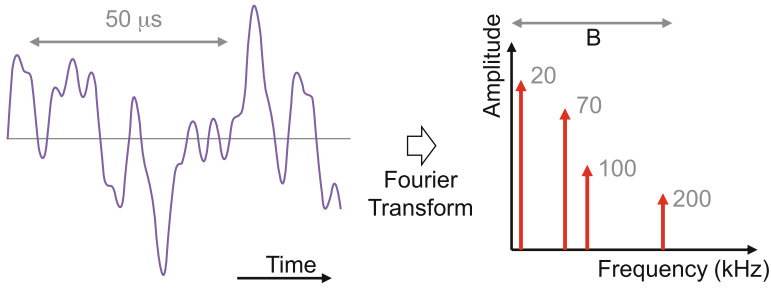
a digital DAQ system because the user interface generally has more parameters to vary, the layout is not necessarily standardized and the configuration process is not always obvious.

### 7.3.2 Digital Sampling

Digitization is the process by which a continuous time signal is converted to a discrete time signal. Sampling is performed by measuring the value of the continuous signal every  $T$  units of time, where  $T$  is referred to as the sampling interval. The digitization process results in a sequence of samples (that can be stored in memory on a digital device) that represent the original signal. The sampling of a detector current pulse is illustrated in Fig. 7.19, where  $S_i$  is the sampled value of the pulse amplitude at sample number  $i$ .

For the sampled signal to be an adequate representation of the original analog signal, there are certain rules that must be obeyed. These rules are fundamental in ensuring the preservation of information when moving from the analog domain to the digital domain and result from a subject are known as *information theory*. Linked to information theory is the *sampling theorem* that dictates the minimum sampling period required to faithfully represent a continuous time signal in the digital domain.

Any continuous time signal can be represented by a summation of sine waves at different frequencies, each of which represents a frequency component. The continuous time signal can be represented as a *spectrum* of frequency components in what is known as the frequency domain. The continuous time signal is said to reside in the time domain and it can be converted to the frequency domain via the Fourier transform [14]. An example of this transform is illustrated in Fig. 7.20, which shows a time domain signal on the left and its frequency domain representation on the right, which in this case is composed of four sinusoidal components of varying amplitude and frequency. These frequency components span a bandwidth,  $B$ .



**Fig. 7.20** Time domain and frequency domain representations of a continuous time signal

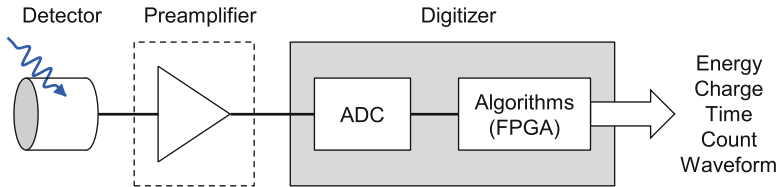
A signal is said to be band-limited if it contains no energy at frequencies higher than some bandwidth  $B$ . A signal that is band-limited is constrained in how rapidly it changes in time, and therefore how much detail it can convey in a certain time interval.

Referring to Fig. 7.19, the Sampling Theorem states that the uniformly spaced discrete samples are a complete representation of the signal if its bandwidth is less than half the sampling rate. Put another way, for an exact digital representation of the continuous time signal, the sampling rate must be at least twice the highest frequency in that signal; known as the Nyquist rate [14]. So in determining the required sampling rate for a detector output signal, the main consideration is the highest frequency component of that signal. The consequence of sampling below the Nyquist rate is an effect known as aliasing, where different frequency components of the signal become indistinguishable when sampled. To avoid this problem, a digital sampling system (such as that described in Sect. 7.3.3) will have an anti-aliasing filter as its first stage to remove any frequency components that violate the Nyquist criterion. When choosing or designing a digital sampling system, one should take care that the anti-aliasing filter is not removing wanted high frequency components.

In practice the sampling rate is chosen to be somewhat higher than the Nyquist rate to allow for imperfections in the design of the anti-aliasing filter.

### 7.3.3 The Digitizer

Recent developments in high-speed digitizers have paved the way for a more elegant hardware solution to the analog DAQ chains presented in Sect. 7.2 particularly so for DAQ systems with a high number of channels. Digitizers offer a one-box solution that can be dynamically configured to perform many different DAQ functions. The analog DAQ chain is replaced by a digital DAQ chain. The essential difference between the analog and digital systems is that the latter digitizes the detector output



**Fig. 7.21** Digital data acquisition chain

signals very early in the chain (usually at the detector output or just after the charge integrating preamplifier if one is used in the DAQ chain) and then performs the pulse processing in the digital domain.

The digitizer is today the core component of most digital DAQ systems. The analog DAQ chains made up of a series of hardware modules is replaced by a single hardware digitizer module that performs analog-to-digital conversion as well as applying digital algorithms. These algorithms mimic the functions of the modules one would have in an analog DAQ chain such as shaping amplifiers, discriminators, peak sensing, scalers and charge-to digital converters. A typical block diagram for a digitizer-based DAQ chain is shown in Fig. 7.21.

The key components of the digitizer are the analog-to-digital (ADC) conversion stage and a processing stage where different pulse processing algorithms can be applied to the digitized detector pulses.

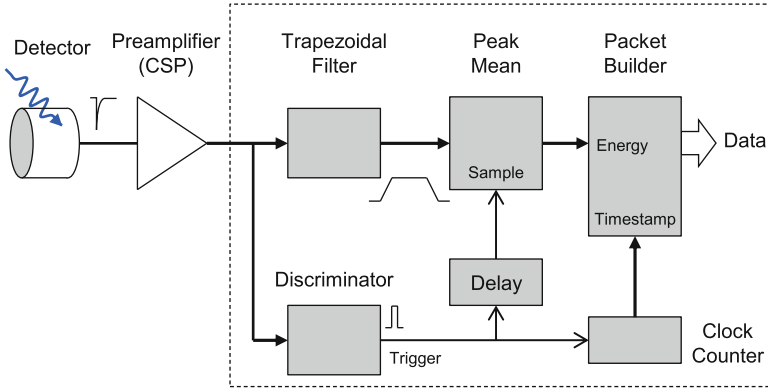
The specification of a digitizer is largely defined by the specification of the ADC and the amount of processing power available to run algorithms. The ADC is largely defined by the rate at which it samples the input signal from the detector and the precision to which it produces those samples. The required specification of ADC is dictated by the frequency content of the input pulse and the energy resolution of the detector. The faster the time development of the detector pulse, the higher the sampling rate needed. The required sampling rate can be chosen in accordance with the sampling theorem described in Sect. 7.3.2. As a guideline, charge integrated pulses from detectors such as sodium iodide and high-purity germanium have a time development in the  $\mu\text{s}$  range and 100 MSa/s (million samples per second) is sufficient to digitally capture these signals. The current pulse from a liquid scintillator or PVT (polyvinyl toluene) plastic has a time development in the range of 10s to 100s of ns; a sampling rate in the region of 250 MSa/s might be needed for energy measurements and perhaps 500 MSa/s for Pulse shape discrimination measurements. The higher the energy resolution of the detector, the more digital bits needed to faithfully represent the measured energy values. Examples of the number of ADC bits needed for different detector types can be found in Sect. 7.2.4. The output of the ADC is a sequence of digital samples (representing the analog pulse presented at its input) which is fed to the digital pulse processing stage.

In its simplest form, there is no pulse processing carried out on the digitizer hardware and the digitized detector pulses are passed directly to a computer, where the pulse processing can be performed in software. This mode of operation is commonly referred to as *waveform* or *raw* capture mode. The obvious disadvantage of raw mode is in the sheer amount of data that must be passed between the hardware and the host computer and this can often preclude real-time operation of the DAQ system; this issue is discussed in detail in Sect. 7.4. The ability to carry out pulse processing on the digital hardware itself opens up the possibility of real-time operation of complex algorithms. Pulse processing is performed on a hardware chip (integrated circuit), such as a field-programmable gate array (FPGA), and just the results of those pulse processing algorithm operations are passed to the host computer rather than the whole digital pulse waveform. Not only can the algorithms be performed faster in hardware but the amount of data passed to the host can be reduced substantially, and in the process easing the data transfer specification on the host interface. The hardware pulse processing device of choice is the FPGA (but there are alternatives such as microcontrollers, Digital Signal Processors or Complex Programmable Logic Devices) which is an array of reconfigurable gates (groups of transistors) that can be programmed (and re-programmed) to form processing blocks such as logic functions, comparators, latches and memory [15]. These elementary processing blocks are used to build up more complex functions such as a shaping network, constant fraction discriminator or charge-to-digital converter.

### 7.3.4 Triggering

The concepts of triggering are explained in Sect. 7.3.4 and apply equally in the digital domain. Both the leading edge discriminator and constant fraction discriminator can be implemented digitally. But digital DAQ systems have the advantage of being able to store data and “look backwards in time”, and so the trigger schemes can be more sophisticated if desired. For example, a system can be set to produce a trigger only if the input signal remains above a fixed threshold for a certain period of time; or only trigger if a certain amount of time has elapsed since the previous trigger. With digital systems, it is also relatively easy to generate a trigger based on the time coincidence of two or more events either on the same DAQ channel or across different channels of a multi-channel DAQ system. This can be extended further to generate triggers based on a sequence of events that fit a certain pattern, which could be a pattern based on, say, the energy of the event or the particle type. Smart triggering of this type and the benefits that it can bring are discussed further in Sect. 7.4.





**Fig. 7.22** Digital pulse height analysis data acquisition chain. The processing blocks within the dotted outline are performed in the digital domain

### 7.3.5 Pulse Height Analysis

Pulse height analysis (PHA) is the digital equivalent of the analog peak sensing ADC data acquisition chain described in Sect. 7.2.4. The PHA digital DAQ chain takes the charge integrated preamplifier signal as its input. The preamplifier signal is appropriately sampled and the remainder of the processing is performed in the digital domain. One possible DAQ chain implementation for digital PHA is illustrated in Fig. 7.22. The processing blocks within the dotted outline are performed in the digital domain; for real-time operation these blocks would typically reside in programmable hardware (e.g. a field-programmable gate array) on the digitizer module.

The trapezoidal filter creates a trapezoid or *flat-top* output signal, whose maximum amplitude is proportional to the peak height of the preamplifier pulse. The trapezoidal filter is the digital equivalent of the analog shaping amplifier shown in Fig. 7.5. The trapezoidal filter is typically composed of the digital equivalent of a delay line or similar filter network [15], to produce the flat-top output signal. The width of the flat-top can be controlled, in the same way that the shaping time can be varied on an analog shaping amplifier. The flat-top should be sufficiently wide so that its peak amplitude can be sampled accurately; in practice, the peak amplitude will be taken as the average over a pre-defined number of samples. The discriminator (which can be either a leading edge triggered or a constant-fraction discriminator; see Sect. 7.3.4) detects the start of the preamplifier signal. The discriminator provides a trigger to the clock counter, which counts the number of events and registers the event time (the time stamp). The trigger is also used as the timing signal to determine when the Peak Mean stage is to sample the flat-top

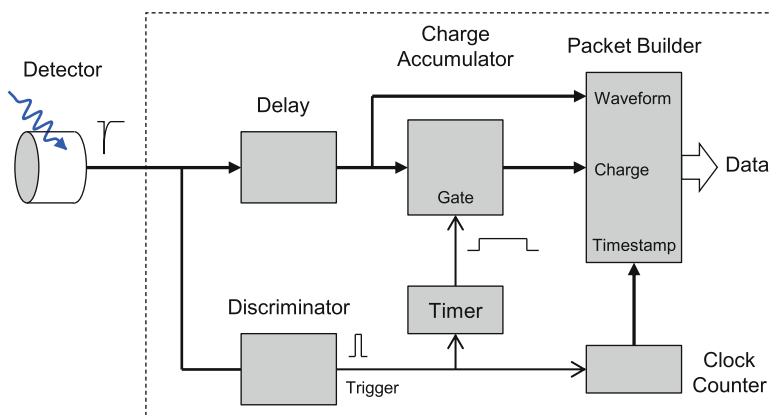
amplitude; the delay can be tuned to make the sample time line up with the centre of the flat-top. The sampled value is proportional to the deposited charge and hence the energy of the radiation event.

Depending on the specific implementation, the time stamp and energy value (and any other relevant information such as the DAQ channel number, event number or DAQ parameter settings) can be built into a data packet to be sent to the host computer. The relevant information from these data packets can be taken to form a list-mode data file (see Sect. 7.3.8), which can be further processed to generate an energy or time spectrum (histogram).

### 7.3.6 Charge Integration

Charge integration (CI) is the digital equivalent of the Charge-to-Digital analogue data acquisition chain described in Sect. 7.2.5. The CI digital DAQ chain takes the detector current pulse as its input signal. The current pulse is appropriately sampled and the remainder of the processing is performed in the digital domain. A typical data acquisition chain for digital charge integration is shown in Fig. 7.23. The processing blocks within the dotted outline are performed in the digital domain; for real-time operation these blocks would typically reside in programmable hardware (e.g. a field-programmable gate array) on the digitizer module.

The *charge accumulator* essentially sums the digital sample values of the detector current pulse, which is equivalent to integrating the signal. The Charge Accumulator requires a charge integration window (a time period over which to integrate the detector current pulse), which is provided by the timer. The start signal for the timer is provided by the discriminator (which can be either a leading edge



**Fig. 7.23** Digital charge integration data acquisition chain. The processing blocks within the dotted outline are performed in the digital domain

triggered or a constant-fraction discriminator; see Sect. 7.3.4), which triggers when an input pulse is detected. The delay unit (prior to the Charge Accumulator) is necessary to align the current pulse with the charge integration window, as there is an inherent delay introduced by the discriminator relative to the current pulse. The discriminator output also triggers the clock counter, which counts the number of events and registers the event time (the time stamp). The “raw” digitized current pulse (waveform) is usually available as an output on such a digital DAQ system and may be recorded to file for later processing.

Depending on the specific implementation, the time stamp, charge value and waveform samples (and any other relevant information such as the DAQ channel number, event number or DAQ parameter settings) can be built into a data packet to be sent to the host computer. The relevant information from these data packets can be taken to form a list-mode data file (see Sect. 7.3.8), which can be further processed to generate an energy or time spectrum (histogram).

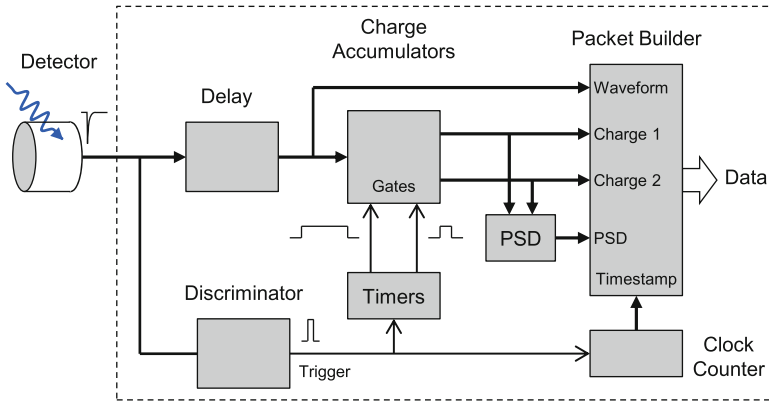
### 7.3.7 Digital Pulse Shape Discrimination

The use of PSD to separate neutrons from  $\gamma$  rays was described in some detail in Sect. 7.2.11 and a number of analog PSD methods were presented, namely rise time discrimination, zero cross-over, charge comparison and constant time discrimination. Any one of these methods may be implemented in the digital domain because all of the building blocks that make up the analog DAQ chains for each of these methods can be implemented digitally. Charge integration is the method that has been most widely adopted for digital PSD and is the method that will be described in detail here.

The DAQ chain for the charge integration PSD method takes the detector current pulse as its input signal. The current pulse is appropriately sampled and the remainder of the processing is performed in the digital domain. A typical data acquisition chain for charge integration PSD is shown in Fig. 7.24, which is essentially a modified version of the CI DAQ chain shown in Fig. 7.23 to include a second charge gate. One gate is used to measure the short duration (or prompt) charge of the detector current pulse and a second gate used to measure the long duration (or total) charge; as is required for the charge integration PSD method (see Fig. 7.18).

The processing blocks within the dotted outline are performed in the digital domain; for real-time operation these blocks would typically reside in programmable hardware (e.g. a field-programmable gate array) on the digitizer module.

The charge accumulator sums the digital sample values of the detector current pulse (which is equivalent to integrating the signal) over the short and long time windows (charge gates). The two charge integration intervals are provided by two timers. The start signal for both timers is provided by the discriminator (which can be either a leading edge triggered or a constant-fraction discriminator; see Sect. 7.3.4), which triggers when an input pulse is detected. The delay unit (prior



**Fig. 7.24** Digital PSD data acquisition chain based on charge integration. The processing blocks within the dotted outline are performed in the digital domain

to the Charge Accumulator) is necessary to align the current pulse with the charge integration window, as there is an inherent delay introduced by the discriminator relative to the current pulse. The discriminator output also triggers the clock counter, which counts the number of events and registers the event time (the time stamp). The PSD stage in Fig. 7.24 calculates a PSD value (a measure of neutron and  $\gamma$ -ray separation) from the two charge integration values, which is typically calculated as the ratio of the delayed charge component (see Fig. 7.18) to the overall charge (the long gate charge).

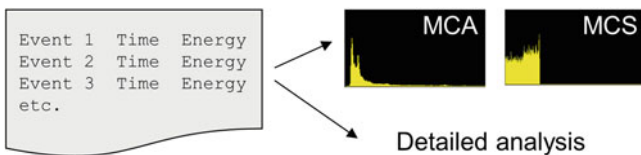
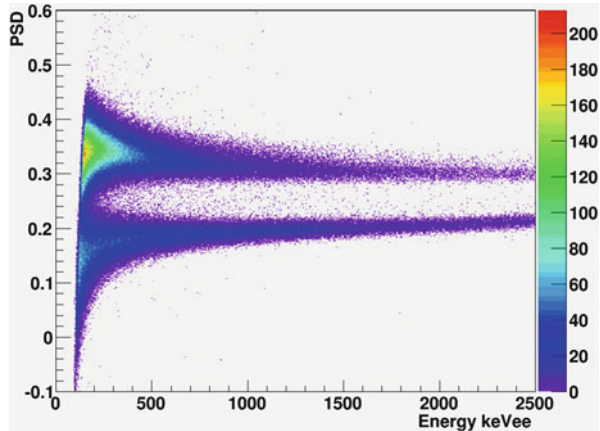
The “raw” digitized current pulse (waveform) is usually available as an output on such a digital DAQ system and may be recorded to file for later processing.

Depending on the specific implementation, the time stamp, two charge values, a PSD value and waveform samples (and any other relevant information such as the DAQ channel number, event number or DAQ parameter settings) can be built into a data packet to be sent to the host computer. The relevant information from these data packets can be taken to form a list-mode data file (see Sect. 7.3.8), which can be further processed to generate an energy, time or PSD histogram. Further, one can combine any of these histograms to produce two dimensional (2D) or even three dimensional (3D) histograms to illustrate the radiation behavior. A useful and commonly used 2D histogram is energy versus PSD value, which shows neutron and  $\gamma$ -ray separation as a function of energy. An example of such a 2D histogram is shown in Fig. 7.25, which is for a hydrocarbon-based liquid scintillator.

### 7.3.8 Time Stamp List Mode

Time stamp list mode (or more commonly just list mode) is the capability to record and retain the time and energy of each radiation event (as opposed to traditional

**Fig. 7.25** Two-dimensional histogram of PSD value versus energy for a hydrocarbon-based liquid scintillator. The histogram was created using a digital PSD DAQ chain based on charge integration



**Fig. 7.26** Time stamp list mode data acquisition

multichannel analyzers that throw away timing information and multichannel scalers that throw away energy information). List mode is most certainly the future direction of DAQ systems; each radiation event is tagged (as illustrated in Fig. 7.26) and maintained in a list, which can just be a simple text file on the host computer. In its simplest form, the list might contain just a time stamp and an energy value for each radiation event. From the list mode data it is then possible, either in real-time or offline, to create either the energy spectrum of an MCA (see Sect. 7.2.7) or the time spectrum of an MCS (see Sect. 7.2.8).

For PSD analysis the list would include additional parameters per event that describe the shape of the pulse (in the case of pulse shape discrimination based on the charge integration method the list might include two additional parameters with values for the prompt and delayed charge; see Sect. 7.3.7).

The main advantage of list mode (compared with DAQ systems that just produce histograms as their output) is that a much more sophisticated analysis can be performed on the measurement or experiment data. With list mode it is possible to produce energy histograms that span a certain time interval or produce time histograms that cover a given energy range. List mode provides a means of replaying the entire acquisition and displaying the same information in different ways. For example, it is possible to post-process the list mode file to identify coincident events of, say, events of a given energy within a certain time window; the same list mode file can be parsed multiple times to look at different energy events or different time

windows. One can also search for patterns such as a neutron event followed by  $\gamma$  rays of a certain energy within a certain time period. In AI, the time development of certain radiation events is often of importance and appropriate algorithms can be run on list mode files to extract such information; this is discussed further in Sect. 7.5.4.

### 7.3.9 *Multichannel DAQ Systems*

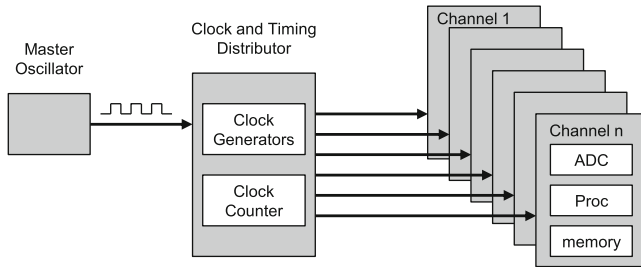
Any large detector system will be made up of multiple detector elements each of which will require its own data acquisition channel. There are many reasons why digital DAQ systems are particularly suited to multi-channel implementations when compared with their analog counterparts. One of the biggest advantages is the physical size of the electronics modules needed. A simple analog DAQ chain (for example Peak Sensing ADC; Sect. 7.2.4) requires at least three different modules and a more complex analog DAQ chain (for example rise time discrimination PSD; Sect. 7.2.11) may require six modules or more. With just a few DAQ channels the total number of modules becomes excessive and for a system with tens or hundreds of channels, the implementation becomes impractical. Of course there are always analog modules that combine, say, multiple shaping amplifiers, which can help to reduce the module count but this only goes part-way to solving the issue. In comparison, it is already possible to get 8 or 16 channels of digital DAQ on a single module roughly the same dimensions as a single analog DAQ module. One can envisage that in future, with advancements in integrated circuit technology, it will be possible to get more digital channels per module.

Another advantage that digital DAQ has is in the flexibility of the hardware. The pulse processing functions of a real-time digital DAQ chain will be held in a programmable device such as a field-programmable gate array (FPGA). The pulse processing functions can be updated and reconfigured in a matter of seconds by downloading new functions to the FPGA. This introduces the notion of a multipurpose multichannel DAQ system, where the DAQ chain implementation for individual DAQ channels can be tailored to different detector hardware or experimental configurations.

Other advantages of digital DAQ for multichannel implementations include the relative ease in which cross-channel processing can be done (for example coincidence triggering) and being able to update DAQ parameters on a channel-by-channel basis or on a group of channel simultaneously.

#### 7.3.9.1 **Channel-to-Channel Synchronization**

A single digitizer hardware module will have a number of identical acquisition channels (typically 4, 8, 16 or 32 channels on modern digitisers for radiation detection). The sampling clock and the clock signal required by internal logic and memory units will typically be generated by a clock tree, which is itself driven by



**Fig. 7.27** Typical clock generation and distribution in a hardware digitizer module

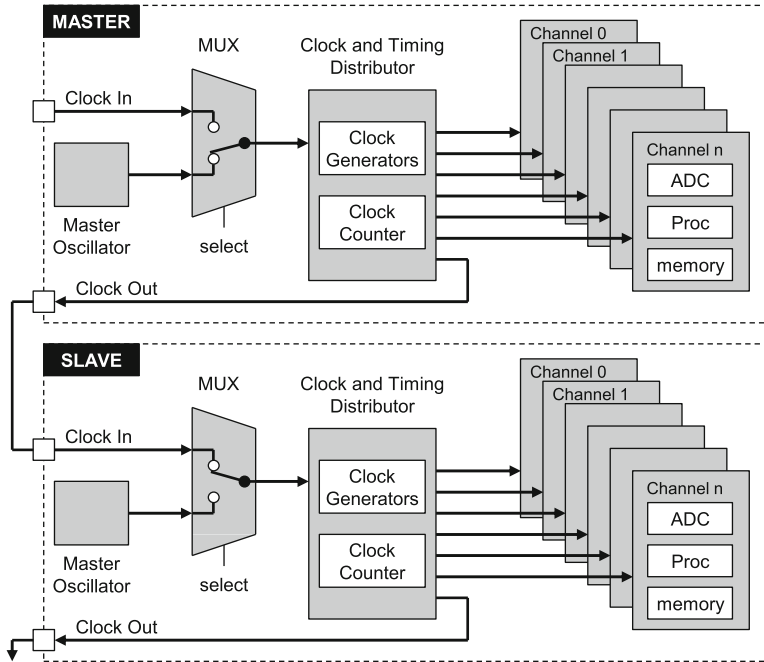
a master oscillator. An example of this is shown in Fig. 7.27. The clock generators derive the various clock signals needed by the system (typically through cascaded multiplication and division stages) and importantly these different clock signals are phase and time aligned. Each channel also has access to the same system clock counter (which might be centrally located or reside as a separate synchronized unit on each channel). Each channel will have its own analog to digital converter (ADC), pulse processor (Proc) and working memory.

Such a clock generation scheme means that signal sampling and data processing can be handled synchronously. By the very nature of the design of the digitizer, individual channels will be synchronized to each other and so a time stamp on one channel can be compared relative to the timestamp on another channel. Channel-to-channel synchronization is particularly important where, say, signals from a multi-element detector are being summed or if channel-to-channel event timing is required (such as for coincidence or time-of-flight measurements).

An issue only arises when channel-to-channel synchronization is required between two channels on two separate digitizer modules. Each digitizer will have its own master oscillator so in general both the frequency and phase of the clock signals on the two digitizers will be different. Furthermore, the notion of “time zero” on one digitizer will be different for the two digitizers. In summary, the timestamp of a given channel on one digitizer has no relationship to the timestamp of another channel on a different digitizer. If more DAQ channels are required than are available on a digitizer module then some means of synchronizing digitizer modules is needed. Fortunately there are digitizer solutions available from certain manufacturers that offer synchronization functionality.

### 7.3.9.2 Synchronization of Two or More Digitizer Boards

Figure 7.28 details the clock distribution topology between two digitizer modules in a Master-Slave configuration. A clock input and clock output is provided on each digitizer module so that the clock signal can be distributed across multiple digitizers. A multiplexer (MUX) is used to select whether or not a digitizer uses its own internal oscillator or takes its clock from a master digitizer. The Clock Distributor on the



**Fig. 7.28** Clock distribution topology between two digitizer modules in a Master-Slave configuration

Master board generates the sampling clocks for each of its DAQ channels, which are inherently synchronized. In addition, the Clock Distributor on the Master provides an output to feed the Slave board. The Slave uses this input clock rather than its own internal clock. Further slave digitizers can be added to the external clock chain.

The Master-Slave topology provides a means of synchronising the time-base of multiple digitizers but the local clock signal on each digitizer will not necessarily be phase aligned (the rising and falling edges of the clocks will not necessarily line up); in general, clocks on different digitizers will be out of phase. Depending on the timing accuracy required, the phase misalignment may or may not be an issue. If clock phase alignment is required then a phase adjustment mechanism is required; this can normally be achieved by applying a small delay to the output clock on the master digitizer. In practice, phase alignment can be a delicate operation and the exact procedure will be specific to the design of the digitizer.

It has been described how the clocks on two different digitizers can be frequency and phase aligned but there is the remaining issue of aligning “time zero” across all digitizers in a Master-Slave configuration. Alignment of “time zero” is necessary so that the time stamp registers on different digitizers are synchronized. It is usually insufficient to use a software signal to start the acquisition across multiple digitizers due to latencies in software and on the digitizer-host interface. The usual method



for aligning “time zero” is though a cable connection at a hardware level so that all digitizers see the start signal at the same time.

Clock and timing synchronization across digitizers is very complex and great care must be taken in matching cable lengths and setting phase delays. At the high clock rates of 10s and 100s of MHz, signal propagation times along cables become comparable with clock periods and compensating for cable delays can be difficult. For these reasons, the external clocks that go from digitizer to digitizer are often made to run at a lower frequency than the sampling clock of the digitizer. Phase-locked loops are then used to multiply the external clock frequency to the higher sampling frequency required by the ADC and signal processing logic.

## 7.4 Data Processing and Storage

The way in which data is processed, stored and transferred through different stages is an important consideration in the design of a data acquisition system. This is particularly so for digital DAQ systems, where vast amounts of data can be moving through the DAQ chain.

### 7.4.1 Analog Systems

Figure 7.29 illustrates the flow of data in a generic analog data acquisition system. Analog processing (such as charge integration and pulse shaping; as described in Sect. 7.2.3) is performed on the current pulse from the radiation detector producing an analog value representing a quantity of interest. That quantity could be, for example, the energy of the radiation event or pulse rise-time for a pulse shape discrimination measurement. That analog value is converted to a digital value using

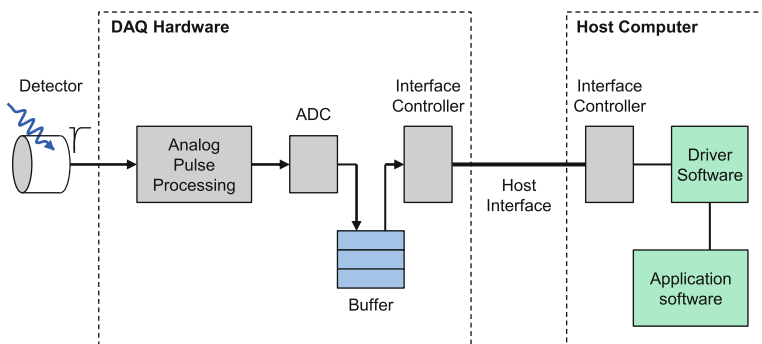


Fig. 7.29 Data flow in a generic analog data acquisition system

an analog-to-digital converter (ADC) and stored in a local buffer on the DAQ hardware. This buffer will have a certain depth, meaning that it can hold a certain number of events (ADC values) ready for transfer to the host computer. Event data from this local buffer is transferred to the host computer via the host interface (managed by the interface controllers on the DAQ hardware and host computer).

There is a notion here of data being produced by the DAQ hardware and being consumed by the host computer. The maximum rate (count rate) at which the DAQ hardware can produce data is limited by the dead time of the DAQ hardware. The dead time is made up of a settling time for the analog signal, ADC conversion time and time to clear the circuits ready for the next event. The DAQ hardware dead time is typically of the order of a few microseconds. The ultimate count rate that can be achieved is limited by the total dead time of the detection system with contributions from the detector itself (for example the decay constants of scintillator light pulses), the front end electronics (for example the time constant of the shaping amplifier) and the aforementioned digital conversion dead time. Depending on the value of these dead time contributions, the theoretical count rate could be many tens of kHz or even hundreds of kHz (however, in practice, the random nature of radioactivity and the resulting effects of pulse pile-up, will be a further limitation on the achievable event rate; also, other system design considerations such as maximizing energy resolution performance will be at the expense of count rate).

The maximum rate at which the digitized event data can be transferred to the host depends on the maximum data transfer rate of the host interface. The most favored interface for modern systems is USB (Universal Serial Bus) running in high speed mode, which has a theoretical data transfer rate of 480 Mb/s (megabits per second) or 60 MB/s (megabytes per second); the transfer rate that can be achieved in practice (useable data rate) is about half this value. If it is assumed that each event (count) produces a digital value of no more than 16 bits (2 bytes) then a USB interface operating at high speed could sustain an event rate of around 15 ME/s (million events per second), which is significantly higher than the rate that the DAQ hardware is able to produce. In practice the data transferred to the host may contain supplementary information or metadata but even taking this into account, the USB host interface is more than good enough to support a single channel of analog DAQ hardware. Indeed, many analog DAQ channels can be supported over a single USB connection.

As mentioned earlier, the DAQ hardware will have a data buffer, which should be sufficiently deep to allow for the fact that the host computer is invariably not a real-time system. The buffer is a “holding bay” for the digital event data for those periods of time where the host interface controller at the host end is either preparing for a data transfer or busy due to another host system operation.

### 7.4.2 Digital Systems

Figure 7.30 illustrates the flow of data in a generic digital data acquisition system. Unlike analog DAQ systems, the current pulse from the radiation detector (or the charge integrated pulse following a charge sensitive preamplifier) is converted to a digital waveform and the remainder of the DAQ processing is carried out in the digital domain. It is possible that those digital waveforms could be passed directly to the pulse processor stage but it is more usual that the digital pulse waveforms will be held in a local waveform buffer. This buffer will have a certain depth, meaning that it can hold a certain number of pulse waveforms ready for the pulse processor stage. This buffer will also have a certain width, which is equal to the number of samples in the pulse waveform. Each sample will have a number of bits (binary digits). This buffer arrangement is illustrated in Fig. 7.31, where waveform  $S_0$  has samples  $S_{00}$  through  $S_{0n}$  (where  $n$  is the number of samples per waveform) and each sample is a 16-bit (2-byte) number. The buffer can hold a maximum of  $m$  waveforms.

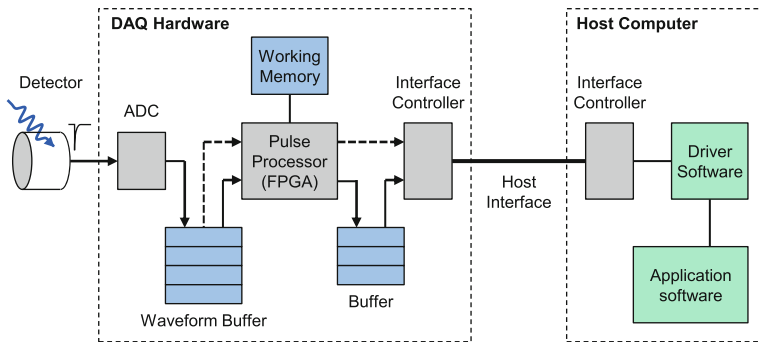


Fig. 7.30 Data flow in a generic digital data acquisition system

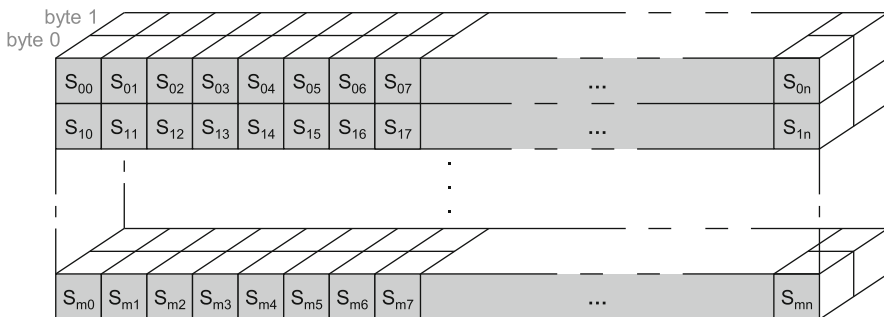


Fig. 7.31 Example of the arrangement of sample data in the local waveform buffer

The width of the buffer is equal to the pulse duration divided by the ADC sampling period. The number of bits in each sample depends on the bit resolution of the ADC. As an example, consider a detector current pulse of  $1\ \mu\text{s}$  being sampled at  $500\ \text{MSa/s}$  by a 14-bit ADC. The sampling period is  $2\ \text{ns}$  (reciprocal of the sampling rate) giving a buffer width of 500 samples, which will be taken as 512 samples since memory arrangements work better in powers of 2. The depth of each sample is traditionally stored in units of bytes (8 bits) so each 14-bit sample will occupy 2 bytes of memory. If we wanted to store, say, up to 256 waveforms in the local buffer then this would require a memory storage block of 256 kB.

The local waveform buffer is made sufficiently deep to allow the pulse processor stage time to perform whatever functions are necessary on the raw waveforms. If the pulse processor can operate in real-time or near real-time then the buffer can be quite shallow because the waveforms can be processed at the same rate (or similar rate) at which they arrive from the ADC. However, there is a mode of operation where the raw waveforms are passed directly to the host without any (or very little) processing on the DAQ hardware and in this case the buffer needs to be sufficiently deep to allow for the host interface data transfer speed and latency; and to allow for the fact that the host computer is invariably not a real-time system. It is worth exploring the raw waveform mode a little further to understand the data transfer rates involved. Expanding on the example given earlier of a  $1\ \mu\text{s}$  being sampled at  $500\ \text{MSa/s}$  by a 14-bit ADC, the theoretical pulse waveform storage rate is 1 million pulses per second (reciprocal of  $1\ \mu\text{s}$ ). At 1 kB per waveform, the theoretical incoming data rate from the ADC is approximately 1 GB/s (gigabytes per second), which is approximately 30 times greater than the 30 MB/s or so that can be sustained over a USB interface running in high speed mode. Indeed, 1 GB/s would even be challenging for USB running in super speed mode (theoretical transfer rate of 640 MB/s; commonly referred to as USB 3.0). Furthermore, the DAQ system is likely to have more than one channel. So it can be seen that sustained throughput of waveform data is challenging at best. Fortunately in AI, advantage can be taken of the burst nature of radiation (e.g. pulsed interrogation from a linear accelerator and the resulting bursts of induced radiation) to avoid overflow of the waveform local buffer. The buffer needs to have sufficient depth to acquire the detector pulses resulting from a single interrogating pulse; the waveform data can be processed and/or sent to the host in the intervening time before the next interrogating pulse arrives. Pulsed interrogation is discussed further in Sect. 7.5.5.

Referring once again to Fig. 7.30, the pulse processor (which is typically a field-programmable gate array) operates on the data in the local waveform buffer to perform functions such as PHA, CI and PSD; as described in Sects. 7.3.5–7.3.7. The pulse processor will have some working memory to hold parameters and calculation results from the signal processing operations performed. The working memory would most likely be internal to the FPGA or might be a combination of internal and external memory. The result of the pulse processor will be a handful of values for each processed waveform; for example, the energy and time of arrival of the event

or the PSD value. These values are stored in another local buffer, similar to that at the final stage of the analog data acquisition data flow diagram of Fig. 7.29. This buffer will have a certain depth, ready for transfer to the host computer. Event data from this buffer is transferred to the host computer via the host interface (managed by the interface controllers on the DAQ hardware and host computer). Similar to the description in Sect. 7.4.1, there is a notion here of data being produced by the DAQ hardware and being consumed by the host computer. The buffer is a “holding bay” for the post processing results data (timestamps, energy values etc.) and should be sufficiently deep to allow for the fact that the host computer is invariably not a real-time system.

Returning once again to the example given earlier, the pulse processing results for, say, a charge integration digital DAQ chain is a timestamp and a charge value. The timestamp might be a 32-bit (4 byte) number and the charge value a 16-bit number. Comparing these 6 bytes with the 1 kB needed to represent the whole pulse waveform, that’s a ratio of more than two orders of magnitude. Looking at it a different way, the data transfer rate to the host for our theoretical event rate of 1 million pulses per second, reduces from 1 GB/s (for the raw pulse waveform) to 6 MB/s (for the timestamp and charge value). In practice the data transferred to the host may contain supplementary information or metadata, which will modify these data rate numbers to a certain extent but the basic argument still holds true. The relative merits of waveform output versus pulse processed values are discussed further in Sect. 7.4.4.

It can be seen that data flow management is a major consideration in the design or choice of a digital DAQ system. The size of the various buffers and data transfer rates of the busses and interfaces that run between them must be carefully specified or chosen to achieve a balanced system. Invariably there will be a limiting factor (or bottle-neck) that will affect the overall performance of the DAQ system; depending on the particular measurement or experimental configuration, it could perhaps be a buffer size, a bus speed or the processing speed of the pulse processor that is this limiting factor. The data processing chain is only as strong as its weakest link and the result of any link not performing to the required level is loss of data (usually the overflow of a data buffer).

Figure 7.30 is a simplified representation of a digital data processing chain but in practice there are various techniques that can be employed to alleviate potential bottlenecks in the chain. For example, depending on the pulse processor architecture, it may be more efficient to process multiple samples or waveforms in parallel or use pipe-lining techniques to make best use of processor clock cycles. There may be DAQ hardware user configurable parameters that can be set to make the most efficient use of available memory or control data transfer modes (e.g. data block size and frequency of transfer). Individual DAQ systems vary greatly in their implementation and so a detailed description is not possible here, but one should be aware that the default settings of such configuration parameters may not be optimal for a particular measurement or experiment and should be adjusted accordingly.

### ***7.4.3 Data Formats and Data Storage***

Choices can be made on the format of digital data and the manner in which that data is stored and manipulated. There are advantages and disadvantages of different data formats. It was described in Sect. 7.4.2 that pulse waveform samples are a digital value, typically ranging from 0 to 65,536 (a 16-bit value). When stored in DAQ hardware memory (for example in the local waveform buffer) this 16-bit value will occupy 2 bytes. But if that sample value is transferred to the host computer then most likely we will want that value to be stored in a file on the computer data storage drive. The two most common formats for storing that data is ASCII or binary. ASCII stands for American Standard Code for Information Interchange and is the most common format for storing text files on a computer. Each sample value is converted to a decimal number and stored in a text file typically with a space character or carriage return after each sample; that file is human-readable and can be opened with any standard text reader or editor. In binary format each sample is stored as a native binary number and stored in a file as a contiguous sequence of binary values. The binary file is not human readable and one needs to know the specific format for that file in order to read or process it.

The advantage of the ASCII format is that it can be read like a normal text file and the sample values just read as a sequence of numbers. Of course, one needs to know such information as how many samples there are in each waveform but this information is often included as ASCII metadata at the start of the file. The disadvantage is that ASCII characters on average take up more file space compared with a binary file. The ASCII file is also of variable length whereas the binary file is of fixed size for a given number of samples. As a simple example, if the sample values were to be stored as 16-bit values then in binary format each possible sample value from 0 to 65,536 will occupy just two bytes each; the equivalent ASCII file would have sample values of either one, two, three, four or five characters depending on the sample value. The size of an ASCII file can vary depending on the data being stored but in practice ASCII files tend to be two or three times larger than a binary file containing the same information. If waveform data files are several megabytes or gigabytes in size then there are clear advantages for using a binary file format. But, as mentioned earlier, the disadvantage of using binary files is that one must keep a separate record of the file format because it will not ordinarily be possible to read or process the file without knowledge of the file format. The same principles of ASCII versus binary apply to the storage of other data such as charge values or timestamps. The binary format can be taken several steps further to minimize file sizes and this is described in Sect. 7.4.5.

### ***7.4.4 Real-Time Versus Offline Processing***

Analog systems of the type described in Sect. 7.2 are, by their nature, real-time systems. Real-time means that detector pulses are being captured, processed and

stored on the host computer at the same rate that the pulses are occurring. It was explained in Sect. 7.4.1 that there is at least one local buffer in the hardware of an analog DAQ system to help with the flow of data to a non-real-time host computer. But the DAQ hardware itself runs in real time and on average, the whole DAQ chain from the arrival of pulses from the detector to writing processed data (e.g. energy value, event time or pulse rise time) to file can be taken as running in real-time. Digital DAQ systems based on digitizer hardware (as described in Sect. 7.3.3) produce vast amounts of data (in particular the sampled waveform data) and real-time operation is more of a challenge. Where it is neither possible nor necessary to process the data in real-time, one can transfer the raw waveforms to the host computer and the pulse processing algorithms can be run there. Offline processing gives the option of running more sophisticated algorithms (perhaps taking much longer to run but producing more precise results) in non-real-time and it is also possible to “replay” the same data using different algorithms.

Historically (less than a decade ago), the normal operational mode for a digital DAQ system would be to collect sampled waveform data using digitizer hardware and passing this data directly to the host computer to perform the pulse processing, which might be PHA, CI or PSD. It was explained in Sect. 7.4.3 how there are challenges associated with the interface bandwidth required to transfer raw waveform data to the host. As DAQ hardware waveform sampling rates and sample resolution continue to increase (as digitizer hardware technology advances), greater demands are placed on the host interface bandwidth. So the host interface is one limitation in the ability to process data in real-time when running in raw waveform (offline) mode. Another limitation is the speed at which the host computer can run the required pulse-processing algorithms, meaning that the host computer must be well specified and configured to maintain sensible processing times.

The way to avoid these limitations associated with the host interface bandwidth and host computer processing power is to perform the pulse processing on the DAQ hardware itself; this is described in Sect. 7.4.2. It is only in the last 5 years or so that DAQ hardware pulse processing has become widely available in everyday off-the-shelf digital DAQ systems. In particular, digitizer hardware technology has evolved to include programmable hardware integrated circuits (typically field-programmable gate arrays) that are specialized in digital pulse processing and can run in real-time. With an FPGA-based digitizer, the pulse processing is off-loaded from the host computer to the DAQ hardware with only basic control data (very low bandwidth) flowing between the host computer and DAQ hardware. There are however challenges in developing efficient pulse processing algorithms that are capable of running in real-time. There are also challenges in creating firmware (software programmed into read-only memory) code that can fit in the limited space and use the limited resources available on a typical cost-effective FPGA. But, as always, there are advantages and disadvantages of operating a digital DAQ system in real-time versus offline mode. The advantages have already been discussed in Sect. 7.4.2, but the main disadvantages are a loss of data fidelity and an accompanying loss of flexibility in how data is processed, stored and (potentially) post-processed.

To explain the relative merits in terms of data storage and processing for real-time versus offline operation, we have to look at how and where data is stored through a DAQ processing chain. We start by looking at how data might be stored with an analog DAQ system compared to a digital one. With an analog system the data values transferred to the host (e.g. energy, event time or pulse rise time) are typically built into a histogram and then that data value is thrown away. The histogram file that is eventually written to file will be of a (preconfigured) fixed number of histogram channels (or “bins”). With a 14-bit analog to digital converter this would be 16,384 bins ( $2^{14}$ ) and in an ASCII file format the histogram would just be a list of 16,384 numbers representing the number of counts (radiation events) in each bin. This histogram file would perhaps be just a few tens of kilobytes in size, which by modern standards is a tiny file. Moving on to digital DAQ systems, the detector data could be stored as raw waveform samples (which takes up a significant amount of file space; see Sect. 7.4.2) or as list mode data (which takes up perhaps two or more orders of magnitude less file space than raw waveforms); or the same data can be reduced to a histogram file that is many more orders of magnitude smaller again. But the reduction in file space in moving from waveforms to list-mode to histograms is accompanied by a commensurate loss of information; if one is only interested in, say, a time spectrum or an energy spectrum then the histogram format would be the best solution for long-term storage of measurement data. But if there is a need to post-process the data at a later date (for example to generate an energy spectrum for just a particular time interval) then one would want to keep data in list mode format but at the expense of using more file space. In the case of histogram mode or list-mode, operation in real-time, with the pulse processing performed in hardware would be the preferred solution because it is not a problem that the waveform data gets thrown away.

However, if there is a need to post process the data to, say, apply an alternative pulse shape discrimination algorithm then one must transfer the raw pulse waveforms to the host computer to store to file (which invariably means that real-time operation is challenging or not possible at all).

### 7.4.5 *Data Reduction*

Section 7.4.2 highlighted the challenges that can be encountered storing and transferring the large amounts of data often associated with digital DAQ systems, particularly those systems with high sampling rate and high sample resolution.

It can be advantageous to the data throughput and data management of such systems to reduce the amount of data to be processed, transferred or stored. Data reduction can be achieved in many ways ranging from more efficient methods of arranging a given set of data through to eliminating superfluous data altogether.



There are several techniques that can be used to achieve data reduction in digital DAQ systems, the most noteworthy are listed below but many other techniques exist:

- Smart triggering
- Data cuts
- Data packing
- Data compression
- Partial waveform capture
- Waveform parameters

Smart triggering is a general term applied to triggering schemes that are more sophisticated than the standard trigger mechanism where each pulse that meets the trigger criteria (either leading edge or constant fraction discrimination; see Sect. 7.3.4) is duly processed by the rest of the data acquisition chain. One example of smart triggering is hardware coincidence logic where a pulse is only processed if it occurs within a given time window of another pulse (either on the same DAQ channel or a different DAQ channel). For analog DAQ systems, additional hardware modules are required to implement this coincidence logic, which ensures that data is only recorded for those pulses that meet the coincidence criteria and thereby reduces the amount of data being passed to the host computer. In a digital DAQ system the coincidence logic can be implemented in re-configurable logic (for example, a field-programmable gate array) at an early stage of the digital DAQ chain; pulses that do not meet the coincidence criteria are discarded as early as possible to reduce both the load on the hardware pulse processing and the transfer of data to the host computer. Hardware coincidence logic can be particularly effective in a detector system where the radiation rate is very high but the wanted events are just a small fraction of those radiation events that initially trigger the DAQ system.

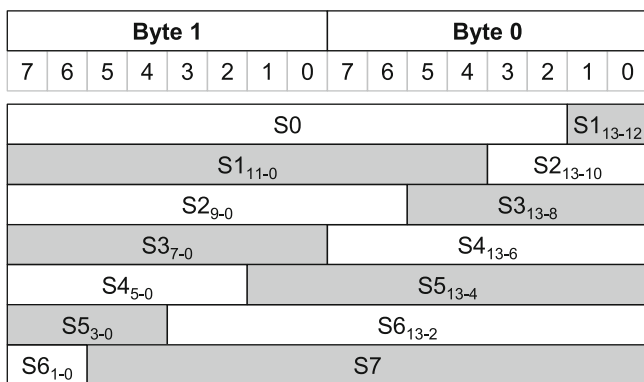
Other examples of smart triggering are where a trigger is produced only if the input signal remains above a fixed threshold for a certain period of time, or only trigger if a certain amount of time has elapsed since the previous trigger. Another example is that the DAQ system be configured to only trigger if it detects a sequence of events that fit a certain pattern, which could be a pattern based on, say, the energy of the event or the particle type (for example a neutron followed by a gamma ray above a certain energy). These types of smart triggering schemes are generally reserved for digital DAQ systems, which have great flexibility in implementing the logic and memory allocations required. But these sophisticated schemes require careful implementation in the DAQ hardware if they are to be effective in reducing the data at the appropriate stages in the DAQ chain and thereby easing the processing load on the DAQ system overall.

Data cuts generally operate much further along the DAQ chain than smart triggering and the main aim with data cuts is to reduce the load on the host interface. As described in Sect. 7.4.2, host interface bandwidth limitations are more of an issue with digital DAQ rather than analog DAQ. Referring to Fig. 7.30, with data cuts,

pulse-processed data (e.g. charge value, timestamp or PSD value) is only transferred to the host if it meets certain pre-specified criteria. For example, the system can be configured to only transfer events that are classified as neutrons based on the PSD value. In a radiation environment that is heavily polluted with (potentially unwanted) gamma rays, this can bring about a significant reduction in data transfers. As another example, the system can be configured to only transfer events that are above a certain charge threshold; this can be particularly beneficial in a pulse shape discrimination system where a standard voltage threshold (as used for leading edge and constant fraction triggering schemes) is not suitable for masking out unwanted low energy events.

More effective than cuts to the pulse-processed data are cuts to the waveform data associated with that pulse-processed data. That is, a waveform is not transferred to the host unless the pulse-processed data resulting from that waveform meets certain pre-specified criteria. Since waveform data can be many orders of magnitude greater than pulse-processed data (as described in Sects. 7.4.2 and 7.4.4), the load on the host interface can be reduced significantly.

Data packing is a technique that can be employed to make best use of the space available for data storage. This space could be memory blocks on the DAQ hardware, random access memory (RAM) on the host computer or file storage space. Commercially available storage memory is almost exclusively configured as storage elements that are multiples of a byte (8 bits); it was mentioned in Sect. 7.4.2 that a 14-bit value from an analog-to-digital (ADC) conversion would most likely be stored in 2 bytes, which means that 2 bits (12.5% of the available memory) are effectively wasted for each 14-bit value that is stored. A way to avoid this wastage is to pack data to fill all of the available bits; this is depicted in Fig. 7.32, where eight 14-bit samples (S0 through S7) are packed to fit the space that would normally hold seven 14-bit samples.



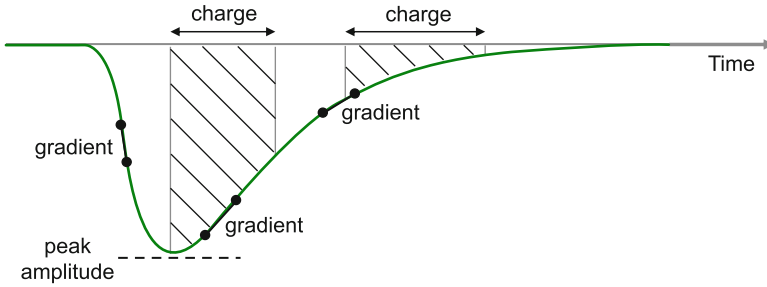
**Fig. 7.32** An example of data packing for 14-bit data values. Eight 14-bit samples (S0 through S7) are packed into the space of seven (2-byte) words. The subscript notation indicates the range of bits for a particular sample

Data packing is most effective for data values that just exceed a byte boundary. For example, in a DAQ system with a 10-bit ADC, 6-bits (37.5% of the available memory) are wasted for every stored event, which is a significant amount of storage space; but that wasted space can be clawed back utilizing data packing. Data packing not only makes more efficient use of data storage space but also reduces the data transfer bandwidth requirements by the same factor as the data space savings. Data packing can yield big savings in data storage and data transfer bandwidth requirements but the disadvantage is that the data value boundaries have to be carefully managed and there can be circumstances where the management overhead can out-weigh the savings.

Data compression is a general term used to describe techniques that can be applied to data to reduce the amount of space it occupies in storage memory and reduce data transfer bandwidth required. Examples of data compression that might be suitable for DAQ systems include run-length encoding (RLE), Huffman coding and zero suppression, which are all lossless compression techniques (meaning that there is no loss of information in the compression process) [16]. Data compression offers savings in data storage and data transfer bandwidth requirements but the disadvantage is that compression and decompression algorithms take up processing time themselves and there can be circumstances where the compression overhead out-weighs the savings.

Partial waveform capture is a compromise to full waveform mode in digital DAQ systems (described in Sects. 7.4.2 and 7.4.4). Instead of transferring all samples of the complete pulse waveform to the host computer, only the relevant parts of the waveform are transferred. For example, if the rising part of the detector pulse is the only part of interest then the host interface bandwidth requirements can be alleviated somewhat by not having to transfer all of the samples that make up the complete pulse. As the rising part of the waveform is generally much shorter than the falling part (the pulse decay), the data transfer bandwidth requirement can be reduced by a substantial factor. In a more sophisticated DAQ implementation, it is also possible to capture more than one section of the detector pulse waveform (for example, to capture the baseline prior to the pulse rise, or the region around the pulse peak, or maybe sections in the pulse tail for the identification of pulse pile-up). Another partial waveform capture technique is to decimate (only retain every  $n$ th sample, where  $n$  is the decimation factor) some parts of the waveform (typically those parts that are slow moving and so do not require a high sampling rate). In summary, with partial waveform capture it is possible to capture enough of the pulse waveform to allow some offline post-processing (see Sect. 7.4.4) but without suffering the problem of excessive data transfer to the host computer.

Lastly, the detector pulse can be represented by a number of values that describe the pulse waveform. In comparison with partial waveform capture, a value is calculated to represent a section of the pulse waveform rather than capturing all of the samples that make up that section. A range of waveform samples is replaced with a single value giving a reduction in data. As an example, consider the pulse waveform shown in Fig. 7.33.



**Fig. 7.33** An example of pulse waveform representation as a series of values that capture features such as pulse gradient, charge integration and peak amplitude

This example shows multiple values based on the pulse gradient, charge integrated periods and also the peak amplitude (but there are many other features that could be represented by a value). The idea here is that the calculations for these values are performed in real-time on the DAQ hardware resulting in a reduced set of data to transfer to the host (compared with transferring all of the samples of the pulse waveform). Calculating values in this way is just an extension to the time stamp list mode (TSLM) representation described in Sect. 7.3.8; the difference here is that the list of values are more extensive and can be customized to suit the application. With the use of multiple values to represent certain features of a pulse waveform, it is possible to capture enough of the waveform features to allow some reasonably detailed offline post-processing (see Sect. 7.4.4) whilst at the same time maintaining a very low data transfer rate to the host computer.

Some of the data reduction techniques described in this section (namely, basic smart triggering, data cuts, basic partial waveform capture, some compression algorithms) are readily catered for on some commercial digital DAQ systems. But other techniques require modification to the hardware pulse processor firmware (typically code running on a field-programmable gate array) of commercial DAQ systems or would have to be realized through a custom digital DAQ implementation.

## 7.5 AI Data Acquisition Challenges

This section covers some aspects of DAQ system design and configuration that are of particular importance and relevance to AI.

### 7.5.1 Event Rate

The single most challenging issue with DAQ systems for AI is the event rate both from the interrogating source and the radiation induced in the material under

inspection. The radiation particle rates that might be encountered in an AI system (such as the nuclear car wash [17, 18]) very much depends on the particular configuration and position of the source and detectors. But it is not unusual to see both neutron and photon rates of the order of  $10^3\text{--}10^5\text{ s}^{-1}\text{cm}^{-2}$  at several meters stand-off, particularly in the immediate microseconds and milliseconds following an interrogating pulse. Such high count rates induce unwanted effects in any DAQ system, analog or digital.

The obvious effect of a high count rate is pulse pile-up, which is discussed in Sect. 7.5.2. Another less obvious effect is in how the baseline estimation calculations for the incoming detector pulse can be upset; this is covered in Sect. 7.5.2.1.

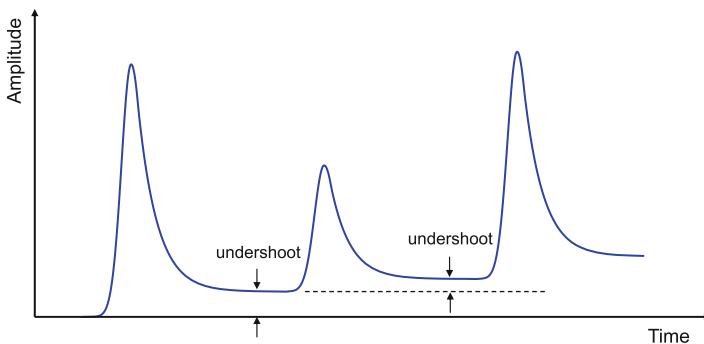
Another obvious adverse effect of a high event rate is in the high resulting data rate that must be processed, transferred and stored throughout the DAQ chain (see Sect. 7.4).

## 7.5.2 Pulse Pile-Up

The interfering effect between successive pulses from a detector is commonly referred to as pulse pile-up. Since radiation events are random, pile-up can in theory occur even at low count rates, but in practice pile-up only becomes significant at higher counting rates.

### 7.5.2.1 Pile-Up in Analog DAQ Systems

In analog DAQ systems, pile-up phenomena are of two general types: The first type is known as tail pile-up and is a result of (charge integrated and shaped) detector pulses being superimposed on a voltage or current offset from a preceding pulse. This superposition results in a change in the pulse height, which manifests as a shift in its measured energy. Figure 7.34 illustrates the case of pulse pile-up due to



**Fig. 7.34** Tail pile-up resulting from undershoot of a preceding pulse in an analog DAQ system

undershoot of a preceding pulse. In the figure, the second pulse is superimposed on the tail of the first pulse and so the peak amplitude (as measured relative to zero) is higher than it should be, giving a false energy value. The pile-up effect can be compounded for successive pulses.

One remedy for tail pile-up is pole-zero compensation. As described in Sect. 7.2.3, the output signal from the charge sensitive preamplifier is shaped with a multistage CR-RC<sup>n</sup> network into a near-Gaussian pulse waveform. The assumption in the design of the shaping network is that the input signal is a step voltage, but in reality the preamplifier has an appreciable decay (typically a 50  $\mu$ s decay constant) that has a measurable effect on the response of the shaping network [1]. The result is that the tail of the shaped pulse will show an undershoot (zero crossover) before returning to the baseline. The way to remedy this undershoot is to introduce a pole-zero compensation resistor across the capacitor of the first stage (differentiator) of the shaping network (see Fig. 7.4), which modifies the transfer function accordingly and compensates for the undershoot. Overcompensation however, will result in overshoot (the shaped pulse not returning to the baseline quickly enough; as depicted in Fig. 7.34). Both overshoot and undershoot can lead to the undesirable effect of tail pile-up.

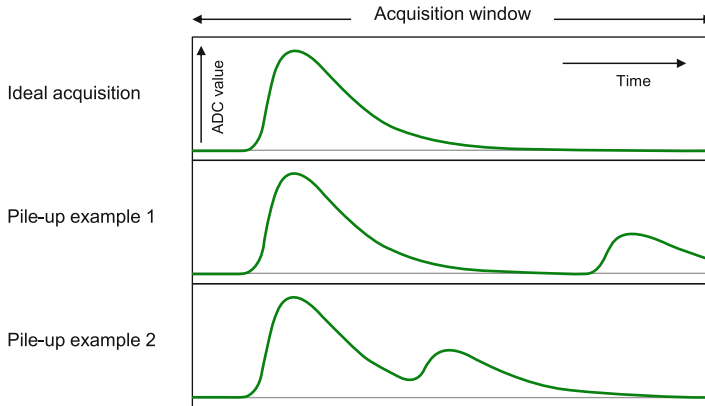
The second type of pulse pile-up is known as peak pile-up and occurs when two pulses are sufficiently close together to be counted as a single pulse. The apparent pulse amplitude will be equivalent to the sum of the two individual amplitudes giving an energy value that is the sum of the energy of the two individual events. In an analog DAQ system there is little that can be done to detect and/or remedy peak pile-up, but one can limit the effects of peak pile-up by suitable design and configuration of the setup to keep pulse durations short and event rates low at the DAQ input.

### 7.5.2.2 Pile-Up in Digital DAQ Systems

In a digital DAQ system, the current pulse (or charge integrated pulse) from the detector is digitized and the pulse processing is carried out on that digitized waveform. Following a trigger, an acquisition window is opened and the analog waveform is sampled for a predetermined number of samples (known as the record length). Pile-up is said to have occurred if another pulse arrives within the acquisition of the preceding pulse. Some examples are illustrated in Fig. 7.35.

For both PHA and CI digital DAQ topologies, the presence of an unwanted pulse within the acquisition window can result in an erroneous calculation of the charge value or energy value. For the PSD digital DAQ topology, the erroneous calculation of a charge value directly affects the calculated PSD value and results in particle discrimination performance degradation.

The solution for pulse-pile up in digital DAQ systems is to employ algorithms that operate on the contents of the acquisition window prior to the pulse data being passed to the pulse processing stage. These algorithms typically fall into two categories: Pile-up rejection and pile-up correction. A pile-up rejection algorithm



**Fig. 7.35** Pulse pile-up in the acquisition window of a digital DAQ system

will search for the presence of additional pulses within the acquisition window and rejects the entire record (data packet) if it fails to meet the rejection criteria (the rejection criteria is set according to the level of DAQ performance degradation due to pile-up that can be tolerated for a particular measurement). A pile-up correction algorithm is more sophisticated and will attempt to not only identify the presence of additional pulses within the acquisition window but also attempt to either remove the errant pulse(s) or separate the superimposed pulses into individual pulses that can be passed to the pulse processor in turn. There are many well established techniques and algorithms for combatting pulse pile-up, with new algorithms being developed all the time [19–21].

### 7.5.3 Baseline Estimation

In any DAQ system, the measurements on the detector current pulse (or the charge-integrated pulse) are made relative to a baseline (that is taken as the zero reference for any amplitude measurements). In an analog DAQ system, this baseline is generally fixed with perhaps some rudimentary baseline compensation control. In a digital system, the baseline can be fixed, but more usually the DAQ attempts to make baseline estimation measurements when there are no incoming pulses (i.e. the baseline is continually estimated and adjusted between the pulses). In a digital system the baseline will be estimated by taking the average value of a set number of samples during a period where there are no pulses. The longer the time duration (or the more samples used in the calculation) the more accurate the baseline estimation will be.

At a low event rate there is plenty of opportunity for the DAQ system to make estimations of the baseline and there is the luxury of being able to configure the

baseline estimation period to be relatively long. But as the event rate increases, the time period between pulses decreases and there is less time available to make baseline estimation measurements. The net effect of an increasing event rate is that the baseline will be estimated less often and the most recent baseline estimation may not necessarily be appropriate for the pulse that is about to be processed; and so the amplitude measurements made on that pulse are relative to a false baseline.

The false baseline problem can be minimized to some extent by reducing the baseline estimation period (albeit at the expense of a less accurate baseline estimation measurement, which may or may not be an issue). But as the event rate increases even further, there will come a point where baseline estimation measurements become too infrequent or cannot be made at all because the event rate is too high. Some digital DAQ systems have various baseline estimation modes and settings that can be optimized to suit the application and this can help to improve high event rate performance and accuracy but there will come a point where normal DAQ functionality breaks down.

### ***7.5.4 Time Development of Active Signatures***

AI involves stimulating some material of interest with a radiation source and then making measurements on the induced radiation. The induced radiation will have certain characteristics (commonly referred to as a signature) that help to identify the type and composition of that material. These signatures are typically distinct by radiation type (neutron, gamma ray etc.), energy and time development (for example, the radiation decay profile of induced fission). So, in AI, it is desirable to have a DAQ output that shows an energy spectrum as a function of time, which in practice would be a two dimensional histogram of energy and time. In the analog domain this amounts to combining the functionality of a MCA with that of a MCS to produce a two-dimensional spectrum of energy versus time; see Sects. 7.2.7 and 7.2.8. With AI, the MCS dwell time is likely to be quite short (in the order of microseconds or milliseconds). These short timescales mean that in practice, this 2D histogram functionality is difficult to achieve using standard commercially available analog MCA and MCS units and a custom DAQ solution will most likely be needed. In the digital domain however, this 2D histogram functionality can be achieved relatively easily by creating a sequence of energy histograms from the list mode data (see Sect. 7.3.8) at set time intervals.

Figure 7.36 is an illustration of a 2D histogram of data acquired from a <sup>3</sup>He proportional counter. In this case, the histogram is showing the decay profile of the energy spectrum of the radiation being measured. Calculations can be made on the decay constant(s) to give information on the material under inspection in an AI detection system. The example of Fig. 7.36 is composed of 100 histogram bins each for energy and time but these bin sizes would be adjusted to fit the required energy resolution and dwell time.



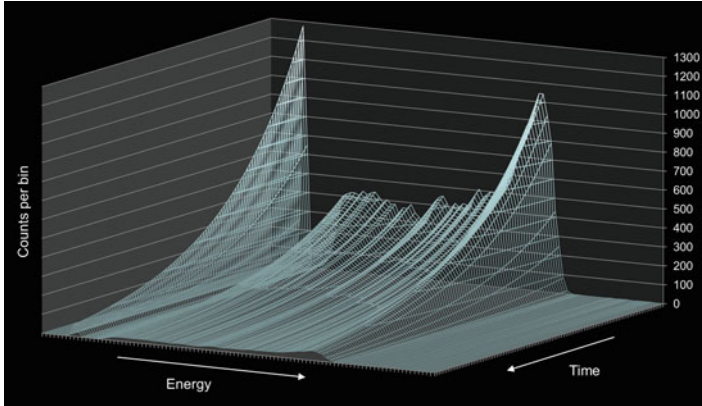


Fig. 7.36 Two dimensional histogram showing the energy spectrum as a function of time

Depending on the particular AI system, a measurement of the type shown in Fig. 7.36 would most likely be made following a pulse of interrogating radiation. This might be a one-shot pulse from, say, a flash x-ray machine or a sequence of pulses from a linear accelerator or neutron generator. In the latter cases the DAQ measurements would be synchronized with the interrogating pulse stream.

### 7.5.5 Pulsed Interrogation

In an AI system (such as the Nuclear Carwash [17, 18]), the interrogation source is typically a high intensity beam of radiation that is pulsed for a given time duration at a given repetition rate. AI sources of this type include linear accelerators, continuous wave x-ray generators and neutron DD or DT generators. For clarity, the pulse of radiation from these accelerators and generators is hereafter referred to as a radiation “burst” to be distinct from detector pulses in a DAQ chain.

The radiation bursts from these accelerators and generators will be short in duration (typically tens or hundreds of microseconds) but very high intensity (for example,  $10^7$ – $10^8$  neutrons/s into  $4\pi$  for a DT generator); the intensity is sufficiently high to overwhelm the DAQ system. So it is usual to inhibit the DAQ system (and often the detector systems too) during the interrogating radiation burst and just enable the DAQ during the periods between the bursts. The inhibit control to the DAQ is derived from a logic level synchronization (sync) signal provided as an output on the accelerator or generator. In an analog DAQ system, the sync signal can be used as a gate to inhibit or enable the acquisition of data. In a digital system, the sync signal can also be recorded a separate channel and can be used (either in real-time or offline) to identify the time regions of interest relative to the interrogating pulse.

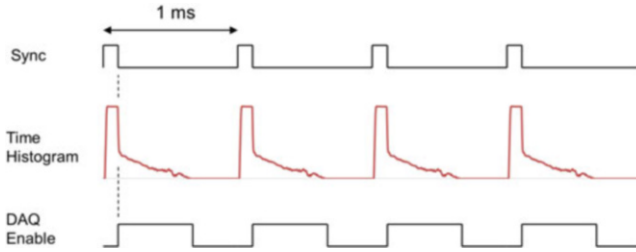


Fig. 7.37 Synchronization of data acquisition to the pulsed interrogating source

An example of how the DAQ might be inhibited and enabled in a pulsed AI system is shown in Fig. 7.37, which is for a particular AI technique known as Differential Die-Away Analysis (DDAA) [22, 23].

In this example, the source emits high intensity radiation in 100  $\mu$ s bursts at a burst repetition rate of 1 kHz (period of 1 ms). Referring to Fig. 7.37, the sync pulse from the source (in this case a DT neutron generator) is at a logic level “high” when the radiation is being emitted and logic level “low” when the radiation beam is off. Figure 7.37 shows the die-away signal (the AI signature of interest) of the induced radiation following every pulse of source radiation. In this example, the die-away signal is represented as a time histogram of neutron events (in this case neutron events detected by a hydro-carbon based liquid scintillator). Simple logic and a timer can be used to create the DAQ enable signal so that the DAQ is only acquiring data during the radiation periods of interest and, importantly, is not overloaded by the intense source radiation.

It is also interesting to note that the pulsed nature of the interrogating source (and the resulting induced radiation signatures) can help with data flow management throughout the DAQ chain. Data buffers on the DAQ hardware can be used to manage the flow of data to the host computer (see Figs. 7.29 and 7.30). With knowledge of the interrogating pulse structure, it is possible to configure the size of those data buffers to be deep enough to hold all of the data resulting from a single interrogating pulse. There is then a short period of time (depending on the burst repetition rate) before the next burst, which can be used to transfer the data to the host computer, thereby avoiding excessive data bandwidth demands on the host interface.

## References

1. G. Knoll, *Radiation Detection and Measurement* (Wiley, New York, 2010). <https://books.google.com/books?id=4vTJ7UDel5IC>
2. K. Krane, *Introductory Nuclear Physics* (Wiley, New York, 1987). <https://books.google.com/books?id=ConwAAAAMAAJ>

3. R. Evans, *The Atomic Nucleus* (Krieger Publishing Company, Malabar, 2003). <https://books.google.com/books?id=mWTFPQAACAAJ>
4. D. Gedcke, W. McDonald, Nucl. Inst. Methods **55**, 377 (1967). [http://dx.doi.org/10.1016/0029-554X\(67\)90145-0](http://dx.doi.org/10.1016/0029-554X(67)90145-0). <http://www.sciencedirect.com/science/article/pii/0029554X67901450>
5. J. Gál, G. Bibok, Nucl. Inst. Methods **163**(2), 535 (1979). [http://dx.doi.org/10.1016/0029-554X\(79\)90145-9](http://dx.doi.org/10.1016/0029-554X(79)90145-9). <http://www.sciencedirect.com/science/article/pii/0029554X79901459>
6. W. Leo, *Techniques for Nuclear and Particle Physics Experiments: A How-to Approach* (Springer, Berlin, 2012). <https://books.google.com/books?id=yc4qBAAQBAJ>
7. L.M. Bollinger, G.E. Thomas, Rev. Sci. Instrum. **32**(9), 1044 (1961). <http://dx.doi.org/10.1063/1.1717610>
8. S. Wu, Q. Yue, W. Lai, H. Li, J. Li, S. Lin, Y. Liu, V. Singh, M. Wang, H. Wong, B. Xin, Z. Zhou, Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip. **523**(1), 116 (2004). <http://dx.doi.org/10.1016/j.nima.2003.12.024>. <http://www.sciencedirect.com/science/article/pii/S0168900204000336>
9. C. Cialella, J. Devanney, Nucl. Inst. Methods **60**(3), 269 (1968). [http://dx.doi.org/10.1016/0029-554X\(68\)90131-6](http://dx.doi.org/10.1016/0029-554X(68)90131-6). <http://www.sciencedirect.com/science/article/pii/0029554X68901316>
10. G. McBeth, J. Lutkin, R. Winyard, Nucl. Inst. Methods **93**(1), 99 (1971). [http://dx.doi.org/10.1016/0029-554X\(71\)90144-3](http://dx.doi.org/10.1016/0029-554X(71)90144-3). <http://www.sciencedirect.com/science/article/pii/0029554X71901443>
11. T. Miller, Nucl. Inst. Methods **63**(1), 121 (1968). [http://dx.doi.org/10.1016/0029-554X\(68\)90314-5](http://dx.doi.org/10.1016/0029-554X(68)90314-5). <http://www.sciencedirect.com/science/article/pii/0029554X68903145>
12. D.W. Glasgow, D.E. Velkley, J.D. Brandenberger, M.T. McEllistrem, Nucl. Inst. Methods **114**(3), 535 (1974). [http://dx.doi.org/10.1016/0029-554X\(74\)90179-7](http://dx.doi.org/10.1016/0029-554X(74)90179-7). <http://www.sciencedirect.com/science/article/pii/0029554X74901797>
13. T. Alexander, F. Goulding, Nucl. Inst. Methods **13**, 244 (1961). [http://dx.doi.org/10.1016/0029-554X\(61\)90198-7](http://dx.doi.org/10.1016/0029-554X(61)90198-7). <http://www.sciencedirect.com/science/article/pii/0029554X61901987>
14. E. Ifeachor, B. Jervis, *Digital Signal Processing: A Practical Approach*. Electronic Engineering Series (Addison-Wesley, Wokingham, 1999). <https://books.google.com/books?id=ZxPwygAACAAJ>
15. P. Horowitz, W. Hill, *The Art of Electronics* (Cambridge University Press, Cambridge, 2015). <https://books.google.com/books?id=LaiWPwAACAAJ>
16. S. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing* (California Technical Publishing, San Diego, 1997). <https://books.google.com/books?id=rp2VQgAACAAJ>
17. J. Hall, S. Asztalos, P. Bilotft, J. Church, M.A. Descalle, T. Luu, D. Manatt, G. Mauger, E. Norman, D. Petersen, J. Pruet, S. Prussin, D. Slaughter, Nucl. Instrum. Methods Phys. Res. Sect. B Beam Interact. Mater. Atoms **261**(1), 337 (2007). <http://dx.doi.org/10.1016/j.nimb.2007.04.263>. <http://www.sciencedirect.com/science/article/pii/S0168583X07007355>. The Application of Accelerators in Research and Industry
18. D. Slaughter, M. Accatino, A. Bernstein, P. Bilotft, J. Church, M. Descalle, J. Hall, D. Manatt, G. Mauger, T. Moore, E. Norman, D. Petersen, J. Pruet, S. Prussin, Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip. **579**(1), 349 (2007). <http://dx.doi.org/10.1016/j.nima.2007.04.058>. <http://www.sciencedirect.com/science/article/pii/S0168900207006456>. Proceedings of the 11th Symposium on Radiation Measurements and Applications
19. M. Nakhostin, Z. Podolyak, P.H. Regan, P.M. Walker, Rev. Sci. Instrum. **81**(10), 103507 (2010). <http://dx.doi.org/10.1063/1.3499241>
20. J. Liu, H. Li, Y. Wang, S. Kim, Y. Zhang, S. Liu, H. Baghaei, R. Ramirez, W.H. Wong, in *2007 IEEE Nuclear Science Symposium Conference Record*, vol. 6 (2007), pp. 4230–4232. <https://doi.org/10.1109/NSSMIC.2007.4437051>

21. M.D. Haselman, J. Pasko, S. Hauck, T.K. Lewellen, R.S. Miyaoka, *IEEE Trans. Nucl. Sci.* **59**(5), 1823 (2012). <https://doi.org/10.1109/TNS.2012.2207403>
22. K.A. Jordan, T. Gozani, *Nucl. Inst. Methods Phys. Res. B* **261**, 365 (2007). <https://doi.org/10.1016/j.nimb.2007.04.294>
23. T.J. Shaw, D.A. Strellis, J. Stevenson, D. Keeley, T. Gozani, *AIP Conf. Proc.* **1099**(1), 633 (2009). <http://aip.scitation.org/doi/abs/10.1063/1.3120116>