Helicity Decoder for E08-027

Chao Gu

University of Virginia

cg2ja@virginia.edu

August 9, 2014

The E08-027 (g2p) experiment uses inclusive electron scattering cross-section differences to determine the g_2 structure function. The cross-section differences are defined between two different helicity states of the incoming electrons. Thus the helicity scheme of the beam needs to be recorded and decoded correctly to obtain the true helicity states of the incoming electrons. This technical note will summarize the helicity scheme during the experiment and the helicity decoder for the offline analysis.

1 Introduction to the Helicity Scheme

At Jefferson Lab, the polarized electron beam is produced by illuminating a GaAs photocathode with circularly polarized photons [1]. The beam helicity needs to be reversed to measure helicity-dependent observables like the cross-section differences in the g2p experiment. The spin of the photo-emitted electron is correlated to the circular polarization state of the photon. It can be either aligned parallel (1 or +) or anti-parallel (0 or -) to the electron momentum direction, which are the two helicity states of the electron.

A programmable logic generator known as the Helicity Control Board is installed at the injector to control the helicity of the electron beam [2]. It generates a logic signal known as the Helicity Flip signal to control the polarity of the high voltage of the Pockels Cell on the Laser Table in the injector. The Pockels Cell is a crystal that acts as a quarter-wave retardation plate when a high voltage is applied on it. Flipping the polarity of the high voltage of the Pockels Cell changes the circular polarization state of the laser and hence

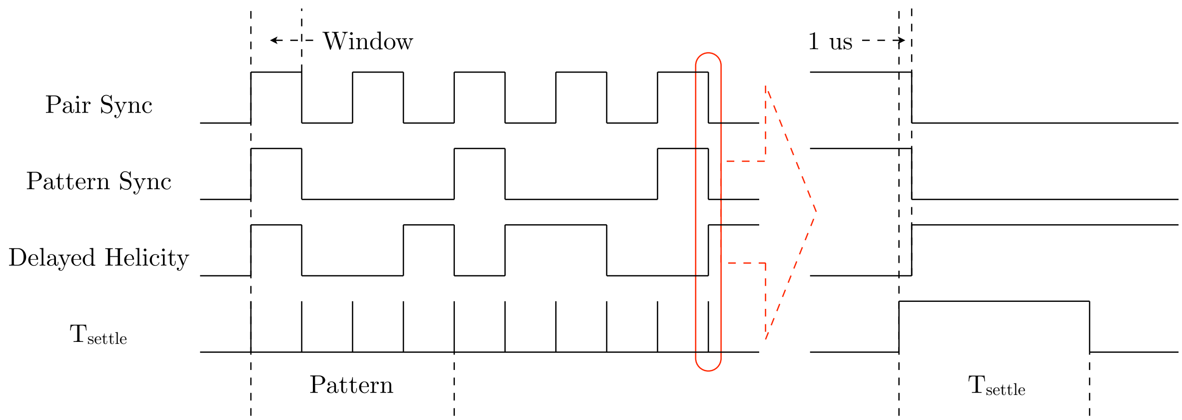


Figure 1: Helicity signals received by experiment DAQ. The right side shows the time sequence of these signals, notice that the T_{settle} signal is 1 μs prior to the other three to avoid misalignment.

changes the helicity of the electron beam [3]. Since there is no mechanical movement, the Pockels Cell can be used to provide relatively fast reversal of the beam helicity. Normally, the beam helicity is flipped at 30 Hz. However during the g2p experiment, the helicity was flipped at 960.02 Hz to be compatible with other experimental halls.

The actual sequence of the beam helicity is a series of identical length helicity windows in which helicity is stable. See Figure 1. To minimize the low frequency systematic uncertainty, the helicity signal always shows up in some symmetric multi-window patterns, like a double-window Pair or a four-window Quartet. For example, the helicity sequence in a Quartet pattern can either be $(+--+)$ or $(-++-)$ so any linear background is cancelled out.

The helicity of the first window of each pattern is determined by a pseudo-random generator in the Helicity Control Board. The generator is a 30-bit shift register, the algorithm of which is shown in Figure Figure 2. Any correlation between the helicity of the beam and other data acquisition (DAQ) components is removed by using this pseudo-random generator. To minimize any other possible systematic effects, the helicity signal received by experiment DAQ is delayed by 8 helicity windows. However, the actual helicity of the incident electrons can still be extracted since the pseudo-random algorithm is fully known.

Due to the non-zero response time of the Pockels Cell to the HV change, the helicity during the transition time between two helicity windows is not stable. A T_{settle} signal is generated to deal with this problem. This signal is composed by a T_{settle} part and a T_{stable} part. $T_{settle} + T_{stable}$ equals to the time length of a helicity window and the T_{settle} is chosen to be slightly longer than the transition time of the Pockels Cell. Any data

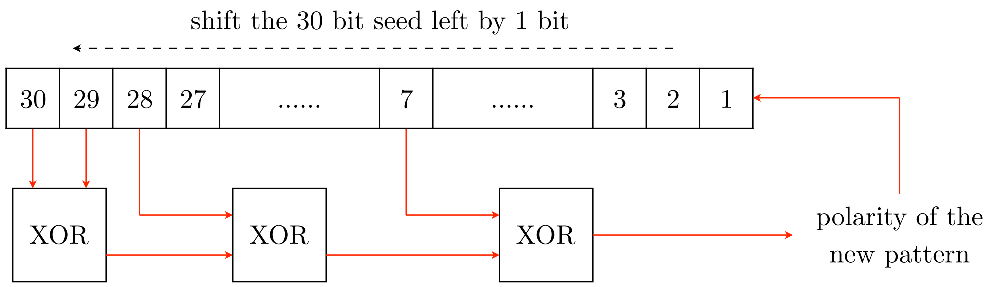


Figure 2: The 30-bit shift register in the Helicity Control Board. The polarity of a new quartet is calculated by applying an XOR (exclusive disjunction) operation to the bit 30, bit 29, bit 28 and bit 7 of a 30-bit register. Then the register is left-shifted by one bit and the new bit 1 is set by the XOR result. The repeat length of this generator is $2^{30} - 1 = 1,073,741,823$ bits.

taken during the T_{settle} part is excluded from the helicity related analysis.

Aside from the Delayed Helicity and the T_{settle} signal, the experiment DAQ also receives two more signals from the Helicity Control Board. The Pattern Sync signal indicates the start of a helicity pattern with a logic 1 and remains 0 in other helicity windows. The Pair Sync signal begins with a logic 1 at the first window of a helicity pattern and then toggles between 0 and 1. These two signals are useful to help predicting the actual helicity. Figure 1 shows the relations of these signals and their time sequence.

The helicity scheme generated by the Helicity Control Board can be varied. E08-027 shares the same helicity scheme with Hall C QWEAK experiment. During the experiment, the helicity pattern is set to be Quartet. The T_{settle} and T_{stable} was set to 70 μs and 971.65 μs respectively so the helicity reversal rate is 960.02 Hz. However the typical DAQ rate of g2p expriment was $5 \sim 6$ kHz. The existing helicity decoder (THaQWEADHelicity in the Hall A analyzer) to extract the actual helicity from the Delayed Helicity signal did not work at this DAQ rate. A new helicity decoder was designed for the g2p experiment.

An insertable half-wave plate (IHWP) located upstream of the Pockels cell could also be used to reverse the beam helicity manually. Insertion of the half-wave plate was performed several times per day to check and to help cancel the helicity dependent systematic effects.

2 Data Acquisition Setup

To calculate the asymmetry, each recorded event needs to be sorted by the helicity of the electron beam. The number of accepted events in each helicity state is normalized by the total charge from BCM and the DAQ live time with the same helicity. Thus, the BCM signal, the triggers (T1∼T8) and L1A signal of Hall A DAQ need to be sorted by the beam helicity. In the g2p experiment, these signals and the detected physical events are addressed as two different issues.

The helicity signals described in the previous section are copied to three different electronics during the g2p experiment. The helicity of the physical event is recorded by the trigger interface (TI). The TI has 12 state registers (TIR). Four of these registers are used to record all of the four helicity signals, Pattern Sync, Pair Sync, Delayed Helicity and T_{settle}, respectively (See Figure 1). The electronics setup is based on Ref. [4] and the recorded helicity is referred as TIR helicity in the rest of this section. Besides, decoding the TIR helicity requires timing information. The standard 103.7 kHz fast clock signal of Hall A was used to set a time-stamp for each physical event.

The BCM signals, triggers and L1A signals are all pulse signals. Helicity-gated SIS3801 scalers are used in the experiment to count these signals. Figure 3 shows the workflow of a SIS3801 scaler. The scaler contains 32 data registers to count, 8 control registers and a FIFO (First-In-First-Out) data buffer. The data signals are sent to the data registers and the T_{settle} signal is sent to one of the control registers to make a veto gate. The data registers only count during the T_{stable} part of a helicity window. Once the counting of one helicity window is finished, the FIFO reads the counting results and save them temporarily. Two additional control registers are used to record the Delayed Helicity and Pattern Sync signals. The FIFO also reads these and

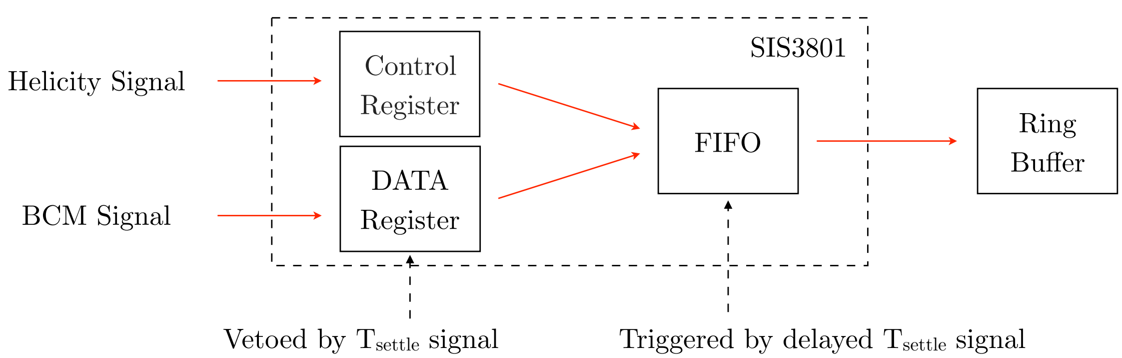


Figure 3: Workflow of a SIS3801 scaler.

records the helicity state of each helicity window.

However, the FIFO is not capable to store a large amount of data. A ring buffer, which is able to store the counting results of 1000 helicity windows, is set in the memory of the VME crate to keep the counting results. The ring buffer is read by the DAQ system every 50 physical events to reduce DAQ dead time. After each readout, the ring buffer is cleared for new counting results. With SIS3801 scalers, the helicity-gated counting results are saved to the raw data file marked with their helicity. The helicity recorded by the ring buffer is referred as ring buffer helicity hereafter.

3 Helicity Decoder

As mentioned in Section 1, the helicity recorded by the DAQ of experimental halls is delayed by 8 helicity windows compare to the actual helicity of the beam. In the accelerator injector, the beam helicity is determined by a pseudo-random generator which is shown in Figure 2. Since the algorithm of this pseudo-random generator is well defined, the actual helicity can be extracted via the same pseudo-random algorithm. The idea is to read 30 continuous helicity quartets to retrieve the random seed, and use this seed to predict the reported helicity, i.e. the delayed helicity as well as the actual helicity. The prediction can be compared to the reported helicity of the first window of each pattern in the helicity sequence to make sure it is correct.

3.1 Predict Actual Ring Buffer Helicity

The ring buffer saves a full sequence of the helicity as mentioned in the previous section. The sequence breaks only if no event was written during the time period of 1000 helicity windows that in our case is about a second. This is because the capacity of the ring buffer is 1000 helicity windows, and the newly coming data flush the old one out if the DAQ system does not read the ring buffer to clear it.

The method of decoding the ring buffer helicity is shown in Figure 4. If the random seed is not set, the program reads the ring buffer helicity in sequence and selects out the windows with Pattern Sync 1. The helicities of these windows are appended to the random seed. The seed is used to predict the reported helicity and the actual helicity of the next helicity pattern once all 30 bits are collected. For a Quartet pattern ("$+--+$" or "$-++-$"), only the helicity of the first window needs to be predicted by the seed,

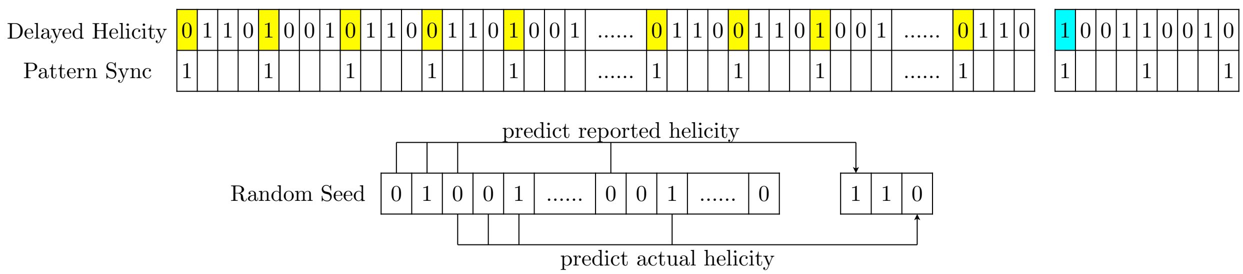


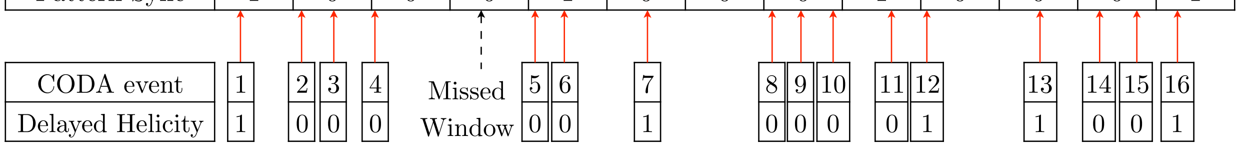
Figure 4: Predict actual ring buffer helicity. The windows marked as yellow are used to generate the random seed. The seed is used to predict the reported helicity and the actual helicity of the newly coming window (marked as cyan), using the 30-bit shift register algorithm shown in Figure 2. The actual helicity is behind the reported one by 2 quartets or 8 helicity windows. In this example, the reported helicity of the cyan window is 1 but the actual helicity is 0.

because the helicities of the second and the third windows of the quartet are always opposite to the first window and the helicity of the forth window is always equal to the first one. After prediction of all four windows in one quartet, the random seed is left-shifted by one bit and the new bit 1 is set with the reported helicity of the first window of the predicted pattern. The prediction is verified with the reported helicity sequence. If the prediction does not agree with the reported value for any reason, all windows in this quartet are marked as bad and the 30-bit random seed is reset immediately and generate again.

3.2 Predict Actual TIR Helicity

For TIR helicity, the decoding algorithm is still based on the prediction method. However, it is possible that several raw events are saved in one helicity window, or no physical event is saved during several helicity windows. Figure 5 shows an example of TIR helicity. It does not show any obvious pattern, which makes the decoding more difficult. It is critical to find a method to locate each event in the helicity sequence before any prediction can be proceed. Since the helicity windows all have identical time length, it is possible to identify each raw event in the helicity sequence if they are labeled by some kind of time-stamp. The standard 103.9 kHz fast clock signal of Hall A was used to set the time-stamp during the experiment. The clock signal is counted by an ungated scaler (which means it is not helicity gated), and read by the DAQ system for each event. The helicity reversal frequency is 960.02 Hz in the experiment, so the time length of each window is about $T_{\mathrm{w}} = 103900 \div 960.02 \approx 108.2$ scaler counts. Some of the TIR event may be saved during the T_{settle} part of a helicity window. These events were excluded from the decoding process and marked as “unstable” by the decoder.

Delayed Helicity	1	0	0	1	0	1	1	0	0	1	1	0	1
Pattern Sync	1	0	0	0	1	0	0	0	1	0	0	0	1



CODA event	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Delayed Helicity	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1
Pattern Sync	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1

Figure 5: An example of the TIR helicity. The sequence on the top is the normal helicity sequence. The physical event stream at the bottom shows how the TIR helicity breaks the normal helicity sequence.

The first step to decode the TIR helicity is still to generate the random seed. For convenience, the helicity windows with Pattern Sync 1 are referred as Pattern Sync windows hereafter. Any events in these Pattern Sync windows are also referred as Pattern Sync events. The helicity of Pattern Sync events is used to generate the random seed, however, there are 3 different situations for the TIR helicity:

1. The event is the first event of a Pattern Sync window, and no Pattern Sync window is missed before this event. In this case, the helicity of this event should be appended to the random seed.
2. The event is the second (or third, ...) event of a Pattern Sync window. In this case it is ignored because the first event of this window has been appended to the random seed.
3. The event is the first event of a Pattern Sync window, but one or more Pattern Sync windows are missed before this event. In this case, all existed 30 bits of the random seed are reset.

In the decoder, a time interval $\Delta T_{\text{pattern}}$ is calculated to determine these 3 situations. Assuming the time-stamp of the current event is T and the previous Pattern Sync event is $T^{\text{last}}_{\text{pattern}}$, $\Delta T_{\text{pattern}}$ can be expressed as $\Delta T_{\text{pattern}} = T - T^{\text{last}}_{\text{pattern}}$. Figure 6 shows the restrictions on $\Delta T_{\text{pattern}}$ for these 3 situations. If $\Delta T_{\text{pattern}} < 2\,T_{\text{w}}$, the previous Pattern Sync event and the new one are in the same window, and it is case 1 described above. If $\Delta T_{\text{pattern}} > 6\,T_{\text{w}}$, at least one Pattern Sync window is missed, and this is case 3. Notice that the possible value of $\Delta T_{\text{pattern}}$ are $0 \sim 1\,T_{\text{w}}$, $3 \sim 5\,T_{\text{w}}$, $7 \sim 9\,T_{\text{w}}$ or etc. The restrictions are chosen to be just inbetween two ranges to avoid any possible error due to the scaler fluctuation.

Once the random seed is generated, the second step is to predict the reported and the actual helicity for each event with the seed. Since the random seed needs to be left-shifted by 1 bit whenever a helicity quartet

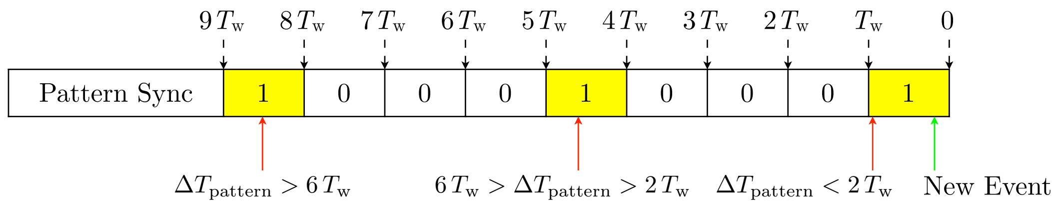


Figure 6: Thresholds on $\Delta T_{\rm pattern}$ to determine whether a Pattern Sync window is missed or not if the new event is a Pattern Sync event. Here $T_{\rm w}$ is the time length of a helicity window in unit of scaler counts. The red arrows are different possibilities of the previous event. Yellow backgrounds indicate the possible range of $\Delta T_{\rm pattern}$, and the thresholds are set in between these ranges to avoid any possible error due to scaler fluctuation.

is finished, it is critical to determine n, the number of missed Pattern Sync windows. Besides $\Delta T_{\rm pattern}$, another time interval ΔT is used to determine n. Assuming the time-stamp of the previous event is $T^{\rm last}$, ΔT can be expressed as $\Delta T = T - T^{\rm last}$. The random seed must be left-shifted by n bits before making the helicity prediction as described below. When shifting, the new bits are always calculated with the algorithm in Figure 2. Due to the particularity of the Pattern Sync event, three different situations need to be considered separately:

1. Both the new event and its previous event are Pattern Sync events. In this case, the restrictions on $\Delta T_{\rm pattern}$ shown in Figure 6 still works. If $\Delta T_{\rm pattern} < 2\,T_{\rm w}$, the new event is in the same window of the previous one. The same random seed is used to predict the actual helicity. If $(4 \times n + 2)\,T_{\rm w} < \Delta T_{\rm pattern} < (4 \times n + 6)\,T_{\rm w}$, n Pattern Sync windows are missed (n can be 0). The random seed is left-shifted by n bits, then the prediction for the new event is made. After the prediction, the seed is left-shifted by 1 bit to prepare for the next prediction.

2. The new event is a Pattern Sync event but its previous event is not. In this case, the time interval ΔT is used to determine n, the number of missed Pattern Sync windows, as $n = \mathrm{int}[\Delta T/(4\,T_{\rm w})]$. The random seed is left-shifted by n bits, then the prediction for the new event is made. After the prediction, the seed is left-shifted by 1 bit to prepare for the next prediction.

3. The new event is not a Pattern Sync event. In this case, the time interval $\Delta T_{\rm pattern}$ is used to determine n as $n = \mathrm{int}[\Delta T_{\rm pattern}/(4\,T_{\rm w})]$. The random seed is left-shifted by n bits, then the prediction for the new event is made. But in this case, the prediction only tells the actual helicity of the first window in the Quartet pattern to which the new event belongs to. The actual helicity of this particular event can be determined according to its Pattern Sync, Pair Sync and the reported helicity value, as shown in Table 1. Unlike cases 1 and 2, the seed does not need to be left-shifted after the prediction, but the

Prediction of the First Window is +				Prediction of the First Window is −			
Reported Helicity	Pattern Sync	Pair Sync	Actual Helicity	Reported Helicity	Pattern Sync	Pair Sync	Actual Helicity
1	1	1	+	1	1	1	−
0	0	0	−	0	0	0	+
0	0	1	−	0	0	1	+
1	0	0	+	1	0	0	−
0	1	1	+	0	1	1	−
1	0	0	−	1	0	0	+
1	0	1	−	1	0	1	+
0	0	0	+	0	0	0	−

Table 1: Find the actual helicity of a event according to its reported helicity and Pair Sync value.

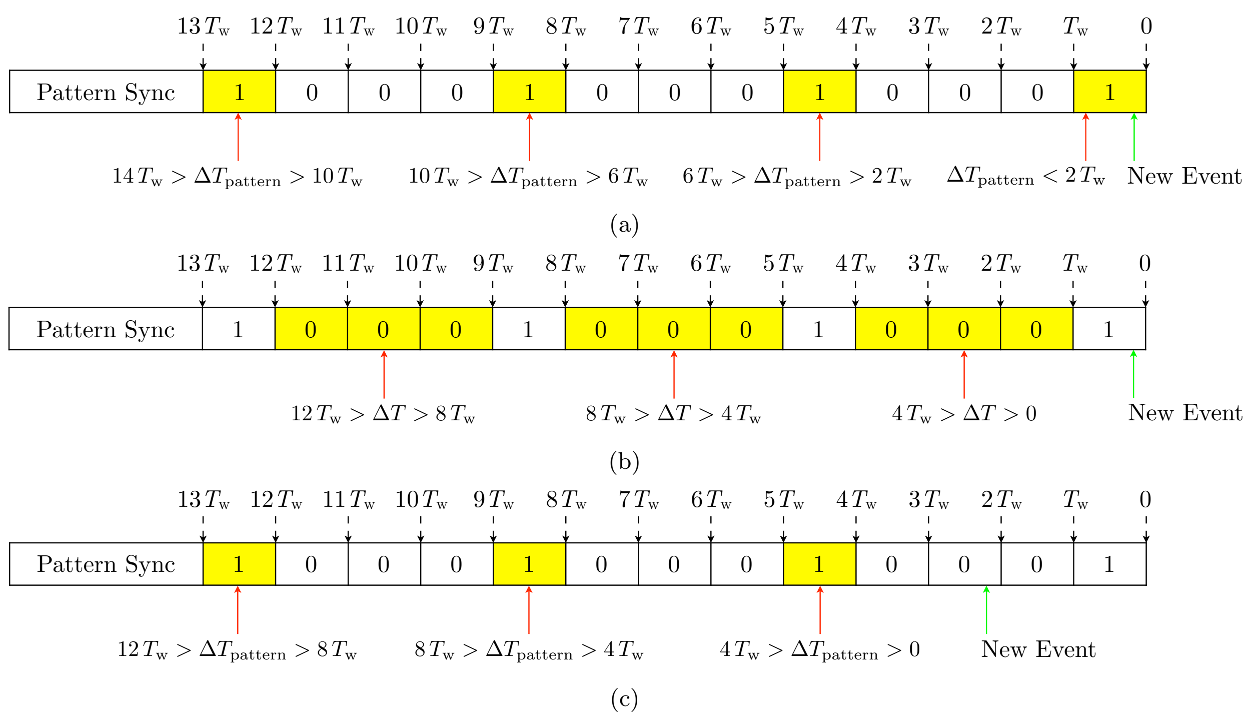


Figure 7: Thresholds to determine the number of missed Pattern Sync windows: (a) Both the new event and its previous event are Pattern Sync events; (b) The new event is a Pattern Sync event but its previous event is not one; (c) The new event is not a Pattern Sync event. Yellow backgrounds indicate all possible time ranges for the previous event. And the thresholds are chosen in between the possible ranges of ΔT and $\Delta T_{pattern}$.

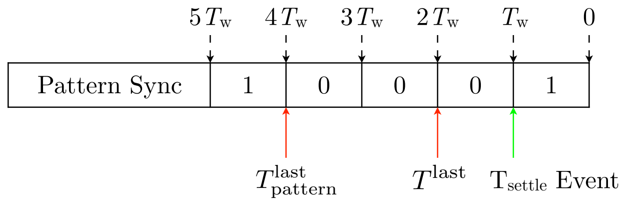


Figure 8: Calibrate T^{last} and $T^{\mathrm{last}}_{\mathrm{pattern}}$ with $\mathrm{T}_{\mathrm{settle}}$ events.

$T^{\mathrm{last}}_{\mathrm{pattern}}$ need to be increased by $n \times 4\,T_{\mathrm{w}}$ in case the next event is still not a Pattern Sync event.

Figure 7 illustrates the thresholds on ΔT and $\Delta T_{\mathrm{pattern}}$ to determine n for these 3 situations. The prediction of the reported helicity of each event is checked with the reported helicity written in the raw data file. If the prediction fails, all events in this quartet are marked as bad. The 30-bit random seed is reset and regenerated.

Due to the fluctuation of the scaler, the time interval ΔT and $\Delta T_{\mathrm{pattern}}$ may not always satisfy the thresholds described and in Figure 7. The excluded $\mathrm{T}_{\mathrm{settle}}$ events are used to calibrate T^{last} and $T^{\mathrm{last}}_{\mathrm{pattern}}$ since the time length of helicity windows is fixed. As shown in Figure 8, a $\mathrm{T}_{\mathrm{settle}}$ event which is right before a Pattern Sync window is selected for calibration. Assuming the time-stamp of this $\mathrm{T}_{\mathrm{settle}}$ event is T, the T^{last} is set to $T - T_{\mathrm{w}}$ and the $T^{\mathrm{last}}_{\mathrm{pattern}}$ is set to $T - 3\,T_{\mathrm{w}}$. Once calibrated, T^{last} and $T^{\mathrm{last}}_{\mathrm{pattern}}$ are not used to store the time stamp of previous events any more. The values of T^{last} and $T^{\mathrm{last}}_{\mathrm{pattern}}$ are increased by $n \times 4\,T_{\mathrm{w}}$ if n patterns are finished during the prediction. And any qualified $\mathrm{T}_{\mathrm{settle}}$ events are used to recalibrate their values. The fluctuation of ΔT and $\Delta T_{\mathrm{pattern}}$ is reduced by at least half if calculated with calibrated T^{last} and $T^{\mathrm{last}}_{\mathrm{pattern}}$, the number of missed Pattern Sync windows can be determined more accurately, and the fail rate of the helicity prediction is reduced.

3.3 Align TIR Helicity with Ring Buffer Helicity

The purpose to align TIR helicity with ring buffer helicity is to insert the helicity-gated informations into the physical data stream. As mentioned in Figure 2, the helicity random seed repeats every 2^{30}-1 bits, thus it never repeats repeats during one particular run, which is usually hour-long. Therefore the random seed can be used as the "fingerprint" to do this alignment.

Before alignment, the quality of the prediction result is checked to avoid false asymmetry. If the actual

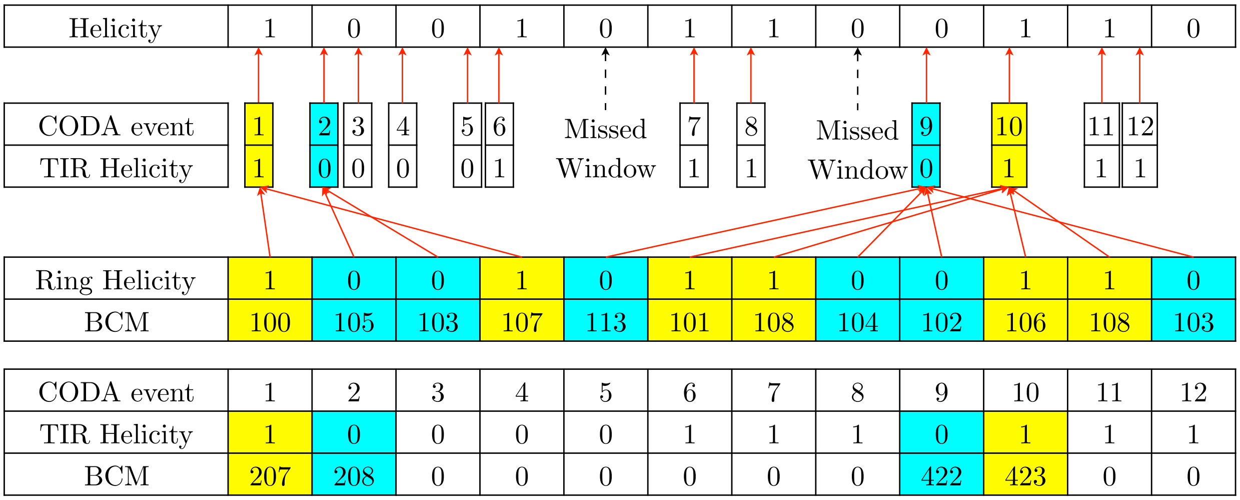


Figure 9: Align TIR helicity with ring buffer helicity. Take BCM as an example of the helicity-gated data. The second quartet in the helicity sequence missed two 0 helicity windows, so the BCM values of this pattern is added to the values of the next quartet to be saved in the physical event stream.

helicity of a event is not predictable due to some error, events in the same helicity quartet are also marked as bad during the checking. For the ring buffer helicity, the BCM information is used to determine beam trips. The data taken during the beam trip and within 30 helicity quartets before and after the beam trip is excluded from the data analysis to prevent any systematic error.

Figure 9 shows an example of the alignment. Here BCM is selected as an example of the helicity-gated data. For each helicity quartet in the ring buffer helicity, two BCM values C_+ and C_- are calculated for $+$ and $-$ helicity. The random seed saved for this pattern is compared with all the random seeds saved in the TIR helicity. If a matching pattern is found and the pattern contains at least one event with $+$ helicity and one event with $-$ helicity, C_+ is saved to the first event with $+$ helicity and C_- is saved to the first event with $-$ helicity. If no pattern matches, C_+ and C_- is added to the BCM values of the next helicity quartet in the ring buffer helicity. This method preserves the most helicity-gated data in the physical event stream so they can be used in helicity-related calculation.

4 Test with Charge Asymmetry

The helicity decoder is tested with beam charge asymmetry during the experiment. The beam charge asymmetry A_Q can be expressed as:

$$A_Q = \frac{Q_+ - Q_-}{Q_+ + Q_-}. \tag{1}$$

Here $Q_\pm$ are the beam charge with helicity ± 1. The beam charge asymmetry can be adjusted in the injector. For the test, beam with large charge asymmetry is required from the injector and is measured with HRS DAQ (SIS3801 scaler), Hall A Møller DAQ and Hall C DAQ simultaneously. The Møller DAQ and Hall C DAQ are used as reference of this test. The results of the test are listed in Table 2. The calculation result of the new helicity decoder agrees well with the Møller DAQ and Hall C DAQ, indicating our new decoder can be used for the analysis.

ID	Left HRS	Right HRS	Moller	Hall C
1	-0.91%	-0.91%	-0.92%	-0.91%
2	-0.56%	-0.56%	-0.56%	-0.56%
3	-0.092%	-0.095%	-0.090%	-0.094%

Table 2: Beam charge asymmetry test with different DAQs.

References

[1] F. Gross. Making the Case for Jefferson Lab. *J. Phys. Conf. Ser.*, 299:012001, 2011.

[2] R. Flood, S. Higgins, and R. Suleiman. *Helicity Control Board User's Guide.* Jefferson Lab, 2010.

[3] J. Hansknecht. *Injector Laser Table Element Definitions.* Jefferson Lab, 2007.

[4] R. Michaels. *Schematic of Electronics for Helicity Info in Hall A HRS during QWEAK.* Jefferson Lab, 2010.