

A Novel TDC Scheme: Combinatorial Gray Code Oscillator Based TDC for Low Power and Low Resource Usage Applications

Jinyuan Wu and Jingjing Xu

Abstract— In common digital electronics design practice, combinatorial loop is less preferable, especially in the multi-bit cases. An exception of multi-bit feedback system that is allowed to be implemented in combinatorial loop is the Gray code sequence generator. The Gray code sequence generator runs by itself without being driven by an external clock. It steps through Gray code sequence significantly faster than the Gray code counter under the same silicon technology. In high energy physics experiments, it is necessary to find a low resource usage and low power TDC (Time-to-Digital-Converter) scheme for high channel count detectors. A Gray code oscillator based TDC is a good candidate for such a scheme. In this paper, we describe our work of design and implementation of a TDC based on combinatorial Gray code oscillator in a Xilinx Kintex-7 FPGA (Field-Programmable Gate Array). The implementation and test results are presented. The test result shows that the TDC using only 8 logic elements is able to reach a RMS precision of 160 ps for time difference measurements of a single pulse. The single pulse measurement precision can be further improved to 51 ps with the weighted average scheme using data from the same TDC channels captured by 4 consecutive system clock edges.

Index Terms— Time to Digital Converter, Gray Code Oscillator, FPGA Applications

I. INTRODUCTION

In common digital electronics design practice, combinatorial loop is less preferable, especially in the multi-bit cases. Combinatorial loops typically are only used in limited single bit situations. The examples of single bit loops include latches, static memories or flip-flops in which the feedbacks are positive, and ring oscillators in which the feedback is negative.

If a multi-bit feedback is required, such as in counters or infinite impulse response filters, the feedback loops are usually built as a combinatorial-register structure and a clock signal is used to drive the registers. In multi-bit feedback systems, the propagation delay for various feedback paths are typically different. Therefore, at one state in a sequence, before the logic levels of all output bits arrive back to the inputs, the combinatorial logic may not generate the next state cleanly, but rather a set of complex intermediate states. So in most real

application, it is necessary to use a set of register to intersect the feedback loop, waiting logic levels from all bits all arrive to the input so that the combinatorial logic outputs become stable. The stable state is then presented at the outputs of the registers so that the system steps through a deterministic sequence.

An exception of multi-bit feedback system that is allowed to be implemented in combinatorial loop is the Gray code sequence generator. Indeed, most Gray code sequence generators are Gray code counters which do have registers and the sequence is stepped through controlled by a clock signal as shown in Fig. 1(a). But to generate Gray code sequence, it is unnecessary to use a register to intersect the feedback loop and it can be built purely with combinatorial loops as shown in Fig. 1(b).

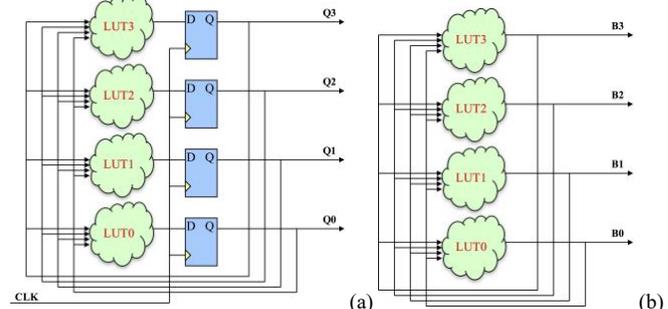


Fig. 1. (a) Gray Code Counter and (b) Combinatorial Gray code oscillator.

It is known that in Gray code sequence, there is only a single bit transition between two consecutive states and all other bits remain unchanged in the two states. At one state, the transition of a single bit output propagates through one feedback path back to the input, causing another bit to flip in the next state. Unlike typical multi-bit feedback system, the difference of the propagation delays in various feedback paths is not harmful since there is only one path that is relevant at a time for Gray code. So the combinatorial feedback loops will not cause glitches in the system as in plain binary sequence. We use term “Gray code oscillator” to distinguish this kind of purely combinatorial structure from typical Gray code counter. Gray code oscillator was used in Reference [1] for temperature

measurement in the FPGA (Field-Programmable Gate Array) devices.

We will discuss the detailed features and conditions of stable oscillation of the gray code oscillator in this paper.

The Gray code counter was used in early history of TDC (Time-to-Digital-Converter) designs [2-3] and today some designers may still use it in low resolution TDC, coarse time counter, single slope ADC or Wilkinson ADC. It can be shown that in most cases, it is unnecessary to use Gray code. As long as the counter is driven by a periodic clock signal, a regular binary counter works as good as a Gray code counter for time measurement.

However, if the sequence generator is not driven by a periodic clock, it becomes necessary to use Gray code in the TDC. On the other hand, the oscillation of combinatorial feedback logic structure using typical binary sequence is not stable and is not suitable for TDC implementation.

A recent effort of implementing Gray code TDC is found in Reference [4], but its scheme still uses large number of delay cells and only save resources in encoding logic. The implementation of the delay cells in this reference are actually flip-flops, rather than self-oscillating structures.

In high energy physics experiments, it is necessary to find a low resource usage and low power TDC scheme for high channel count detectors. A Gray code oscillator based TDC is a good candidate for such a scheme. In this paper, we describe our work of design and implementation of a TDC based on combinatorial Gray code oscillator in a Xilinx Kintex-7 FPGA [5] (XC7K325T-2FFG900C). The TDC was implemented in a Xilinx KC705 evaluation board [6] and the test result will be presented in later sections.

II. THE STRUCTURE OF THE GRAY CODE OSCILLATOR TDC

The TDC consists of two primary parts: combinatorial Gray code oscillator and D flip-flop registers for sampling, as shown in Fig. 2.

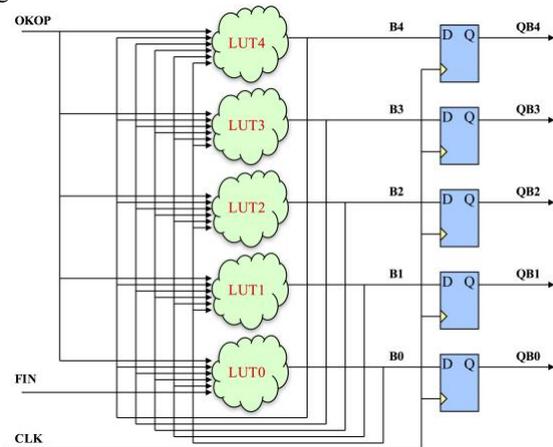


Fig. 2. The Gray code oscillator TDC

In the Gray code oscillator, a signal OKOP is connected to all bits as a master control signal. When OKOP = 0, all bits are held = 0, ready to measure a new event. When an input arrives to the TDC, OKOP is turned on and held = 1 (usually using a latch or a DFF), which allows the oscillator to step through the

Gray code sequence, causing the code B[] change states rapidly. The oscillating speed is determined by the propagation delay of the combinatorial feedback loops, which is typically 250 ps in our FPGA implementation. The timing diagram of the TDC operation is shown in Fig. 3.

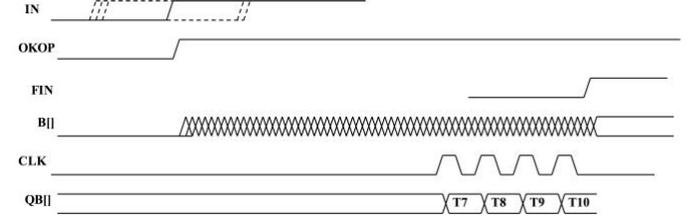


Fig. 3. Timing diagram of the TDC operation

The sampling clock signal for the registers, CLK, provides a known timing reference. In high energy physics experiments, the clock is usually derived from the accelerator beam crossing signal, which is 40 MHz in LHC (Large Hadron Collider), for example. At the CLK leading edge, the values of B0 to B4 are captured by the registers, temporarily held as QB0 to QB4 for further readout. This captured Gray code represents number of states stepped through from OKOP becomes 1 to the leading edge of the CLK, and therefore, the arrival time of the TDC input is measured.

In actual applications, one may choose a burst of pulses for the CLK signal. For example, in our TDC test discussed later, the DFF are clocked using burst of pulses and the repeating period of the burst is 40 ns which is equivalent to a 25 MHz beam crossing. In this 40 ns period, there are 16 pulses with intervals of 2.5 ns. In our test, the Gray code values T7, T8, T9 and T10 are sampled by the pulse edges 7, 8, 9 and 10, respectively and are recorded for further analysis.

The TDC structure shown above may look like the Gray code counter but there are a few important differences. It can be seen that the feedback in the TDC is in the combinatorial portion, before the registers, which is different from the regular Gray code counter. Also, in counters or other pipeline structures, the logic levels at the D input ports of the flip-flops become stable after sufficient long time, before the clock rising edge. While in the TDC structure shown above, the outputs of the combinatorial portion never become stable. The Gray code oscillator steps through the states in the Gray code sequence and the registers take a “snap shot” of the changing states for time measurement.

In FPGA, the combinatorial oscillator is implemented with lookup tables (LUT), and in Xilinx Kintex-7 family, each LUT accepts 6 inputs. In our test, one of the 6 inputs is used for master control signal OKOP, and the remaining 5 inputs are used for feedback from 5 bits of the oscillator outputs.

There are several variations of the Gray code and we choose the original reflected binary code (RBC) scheme. In the original Gray code sequence, the lowest bit B0 has no dependency on itself and therefore it is unnecessary to feedback B0 to LUT0. The saved input for LUT0 is used for finish command input signal FIN. When FIN = 0, the oscillator cycles through all 32 Gray code states repeatedly. If FIN = 1, the oscillator runs to the last code and is held unchanged. The FIN

command is designed to prevent the oscillator from running indefinitely which causes extra power consumption unnecessarily. The FIN signal is generated in the extending higher bit block which we will discuss next.

III. THE HIGHER EXTENSION BITS OF THE TDC

In practical applications, longer time measurement range may be needed, which needs more code bits especially when the Gray code oscillator is fast. Adding bits to the Gray code oscillator is possible in theory, but it would need lookup tables with more inputs. A more practical approach is to extend the higher bits separately from the Gray code oscillator. In our test, we have extended two bits beyond the 5-bit Gray code oscillator to indicate 4 cycles of 32-step oscillation, making the oscillation range up to 128 steps. In practical design, we implemented three bits: B5a, B5b and B6a. The logic generating higher extending bits that the authors implemented is shown in Fig. 4.

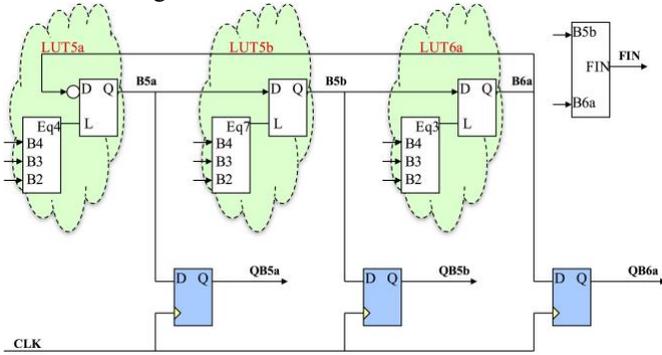


Fig. 4. The higher extending bits of the Gray code oscillator TDC

The combinatorial portions for the higher extending bits are essentially transparent latches that open and latch data at different times. The operating timing is controlled by the Gray code oscillator sequence through bits B4, B3 and B2. The logic levels of B5a, B5b and B6a are sampled at the leading edge of CLK and stored as QB5a, QB5b and QB6a, respectively.

Initially, B5a, B5b and B6a are all reset and latched to 0. When the Gray code oscillator steps to the state that $(B4, B3, B2) = 100$, corresponding to steps 28-31 in the Gray code sequence, the latch for B5a opens, B5a becomes 1. The sequence of the other bit changes is shown in Table I.

Sequence Number of Action	B4	B3	B2	B5a	B5b	B6a
0				0	0	0
28-31	1	0	0	1	0	0
52-55	1	1	1	1	1	0
72-75	0	1	1	1	1	1
92-95	1	0	0	0	1	1
116-119	1	1	1	0	0	1

Table I. The truth table of higher bits

Since the extending higher bits are not in the Gray code oscillator, their bits may flip at the same time as a lower bit flips, and once a changing higher bit aligns CLK leading edge at the same time, it may cause a complicate measurement error. To avoid this type of errors, we use multiple bits to represent a

higher bit. So the two higher bits, denoted as QB5 and QB6, are actually chosen from QB5a, QB5b and QB6a for different regions of QB4 down to QB0 as shown in Table II.

QB4 down to QB0 (Sequence Number)	QB5	QB6
0-15	QB5a	QB5b
16-23	QB5a	QB6a
24-31	QB5b	QB6a

Table II. The truth table of the higher bit selection

The bits selected in the table above will not flip in the corresponding regions. Their transitions only happen in other regions.

Now the higher two bits QB5 and QB6 indicate cycles 0, 1, 2 and 3 of 32-step oscillation with $(QB6, QB5) = 00, 01, 11, 10$. They look like a 2-bit Gray code sequence but they are not generated by a Gray code oscillator.

The signal FIN is set = 1, when $B6a = 1$ and $B5b = 0$, which indicates that the Gray code oscillator sequence is reaching final steps. The FIN signal is fed to LUT0 so that the oscillation stops precisely when it reaches step 126 and held at this condition until OKOP is reset to 0, preparing for the next event.

The configurable logic block (CLB) inside the Kintex-7 FPGA containing one channel of the Gray code oscillator TDC is shown in Fig. 5.

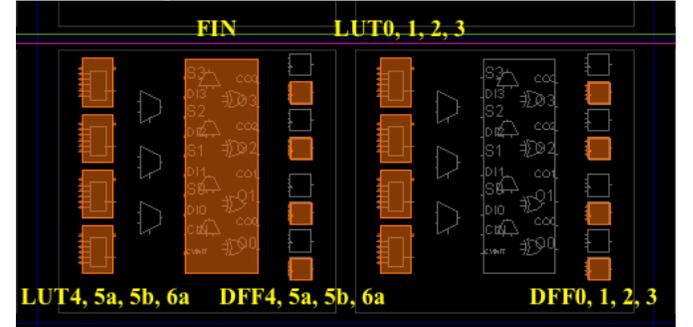


Fig. 5. The configurable logic block (CLB) inside FPGA containing one channel of the Gray code oscillator TDC

Each CLB inside the Kintex-7 FPGA contains two slices, each with 4 six-input LUT and 8 flip-flop registers (4 usable for our purpose). The 5 combinatorial blocks for the Gray code oscillator, 3 for the higher extending bits are placed in 8 lookup tables. The 8 sampling registers are also placed directly next to the corresponding LUT. The FIN logic is generated using the CARRY4 primitive inside the left logic slice to ensure compactness and short propagation delay. A CARRY4 primitive contains 4 multiplexers and two of them are used to generate FIN signal with the following multiplexing logic: $FIN = (B6a=0) ? 0 : ((B5b=0) ? 1 : 0)$.

With this arrangement, all critical timing paths, including the combinatorial feedback paths and LUT to register paths, are contained in a CLB.

IV. THE TEST RESULTS OF THE GRAY CODE OSCILLATOR

To verify that the oscillator steps through the Gray code sequence, we use an input signal with scanning delay relative

to the CLK signal leading edge to make repeating measurements. The Gray code oscillator runs at a speed of approximately 250 ps per step and it is difficult to monitor the oscillation steps directly. We use the register array to monitor the operation of the oscillator indirectly. The CLK drives the register array with a 40 ns repeating rate. The input signal which turns the OKOP signal to 1 shifts relative to the CLK about 40 ps per period. When OKOP turns to 1, the oscillator starts to step through the Gray code sequence and at the leading edge of CLK, a Gray code value will be captured in the register array. Once the relative delay between OKOP and CLK scans through a time range, we expect to see a sequence of Gray code in the recorded data, as shown in Fig. 6. In this plot, B0 to B4 raw values (0 or 1) are multiplied by 1 to 5 so that they are clearly separated.

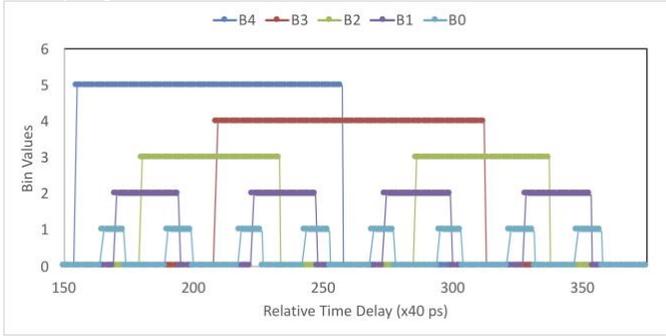


Fig. 6. The Gray code in the recorded data

Since the relative delay between OKOP and CLK decreases, downward counting Gray codes is seen. It can be seen that the codes in the plot do not have the same width due to differences of the propagation delays of combinatorial feedback paths. However, all 32 codes are clearly presented with no missing codes. In each event, after OKOP turns to 1, such a 32-code sequence is repeated 4 times until reaching step 126, when FIN signal causes the oscillator to halt.

To test the stability of the oscillation, we have disabled the FIN signal so that the Gray code oscillator runs without stopping. The test result is shown in Fig. 7.

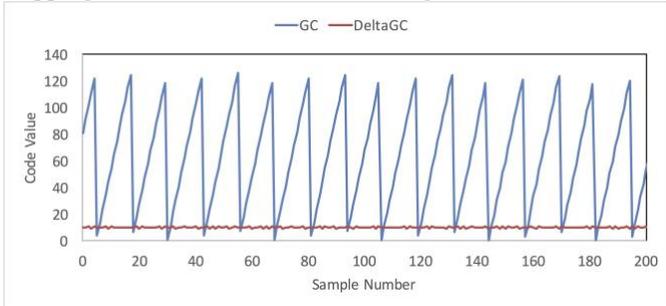


Fig. 7. The long term oscillation of the Gray code oscillator

The register array is clocked with 400 MHz system clock while the Gray code oscillator is operating. The state of the oscillator is sampled every 2.5 ns and within 2.5 ns, the oscillator runs through about 10 steps. In the plot above, “GC” is the binary count of the sampled Gray code and “DeltaGC” is the difference between 2 adjacent samples, which is about 10.

The data is recorded about 15 min after reset and it can be seen that the oscillator remains normal oscillating after the Gray

code sequence has been repeated many billion rounds.

In the multi-bit feedback system, the feedback propagation delay for paths between various bits are not identical, especially when it is implemented in FPGA. The consequence of different delays of feedback paths is that different codes may have different time lengths, which represented as different bin width in the TDC output values, as shown in Fig. 8.

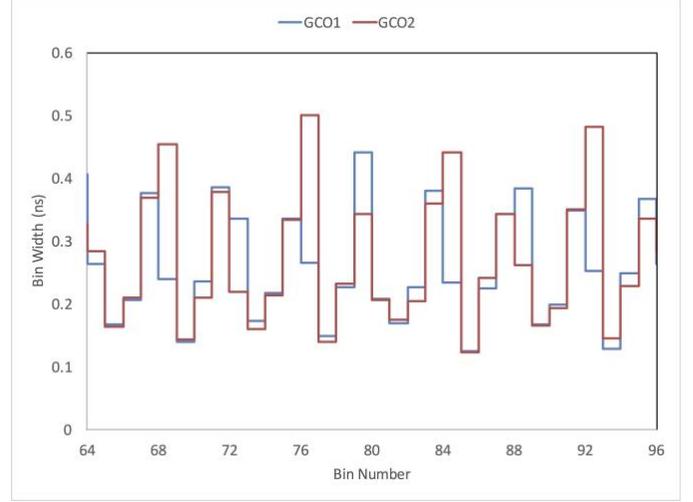


Fig. 8. The bin widths of different TDC output values

We have implemented two Gray code oscillators and due to non-uniformity of the FPGA device, the same code in two oscillators may not be the same.

Note that the bin width plot shown above uses natural time unit (ns) for the vertical axis. If the vertical axis is rescaled with the unit the least significant bin (LSB) of the TDC, the plot becomes the differential non-linearity (DNL) plot.

In our Gray code oscillator, the logic functions that generate lower 5 bits are independent on the higher bits. Therefore, it is reasonable to expect the bin width pattern shown in Fig. 8 can be extended to other ranges of 32 counts, i.e., 0-31, 32-63 and 96-126.

Since the TDC bin widths are not uniform, it is necessary to create a calibration table to convert the raw TDC code to the arrival time of the input pulse. If the k -th bin in a TDC has a width w_k , the arrival time when TDC outputs the code n can be written:

$$t_n = \frac{w_n}{2} + \sum_{k=0}^{n-1} w_k \quad (1)$$

It should be emphasized that it is crucial to calibrate to the centers of the bins, i.e., the first term representing the half width must not be omitted. It is not impossible for one to implement the sum term only and omit the half width term when the calibration algorithm is buried in complicate codes. It can be shown that the RMS measurement errors are the minimum when the times are calibrated to the centers of the bins.

The calibration table for the two Gray code oscillators that we implemented in our test can be shown in Fig 9.

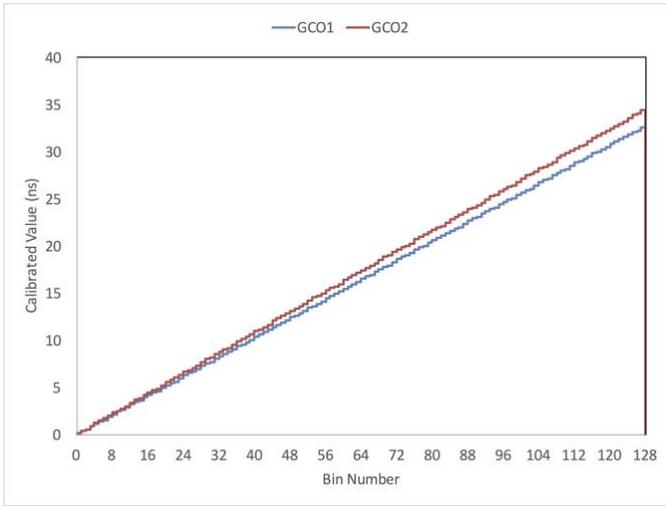


Fig. 9. The calibration table for the two Gray code oscillators

With the calibration table, one can find the input pulse arrival time from the raw code produced by the TDC. From the plot above, we can see that the two Gray code oscillators do not have the same oscillation speed. The oscillator GCO1 uses less time to run from code 0 to 127 than GCO2 and therefore GCO1 is faster than GCO2. The average bin width of GCO1 is 256 ps and the one of GCO2 is 271 ps.

V. THE TEST RESULTS OF THE TDC

We implemented two channels of TDC based on the combinatorial Gray code oscillator and tested their performance. The test emulates typical application in an accelerator based high energy experiment, in which what to be digitized is a time interval between an input pulse and the beam crossing clock. The digitized times from two TDC channels are subtracted from one another to find the time differences and measurement precision is calculated using measurements of multiple events.

It is crucial to use an appropriate clocking and test pulse generating scheme so that the test results are not biased by the unwanted correlation between signals and clocks. Our TDC channels are clocked using burst of pulses with period of 40 ns which is equivalent to a 25 MHz beam crossing.

Test pulses with a constant delay between them are fed to the two TDC channels. The test pulses are generated with cascaded phase lock loops (PLL) from the same crystal oscillator used to generate the TDC clock. The test pulses drift about 40 ps relative to the TDC clock in every 40 ns period. The reason of using cascaded PLL rather than using unrelated crystal or random signal source is to ensure an even timing coverage relative to the TDC clock and to eliminate statistical fluctuation.

In the 40 ns period, there are 16 pulses with intervals of 2.5 ns, which are actually 400 MHz clock pulses. For test purpose, all 16 pulses are kept in our design, while in real applications, it is obvious that the users can keep only used pulses to reduce power consumption. The Gray code values sampled by the pulse edges 7, 8, 9 and 10 are recorded in our test. The recorded raw codes are first multiplied with the average bin width of the corresponding TDC channel as a rough conversion of the arrival times without calibration and their differences are booked into

the histogram.

The histograms of the time differences digitized by the two TDC channels are shown in Fig. 10.

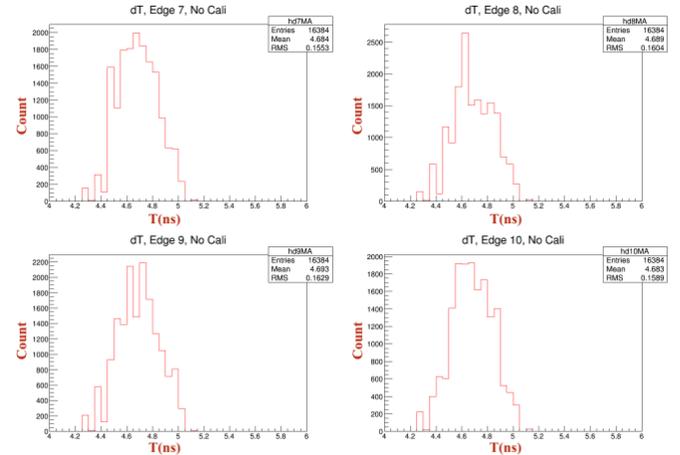


Fig. 10. The time differences measured by two TDC channels

It can be seen that the time difference measurement precision for single digitization is about 160 ps, that yields an equivalent TDC bin width [7] of 392 ps ($160 \times \sqrt{12}/\sqrt{2}$). (In other words, if we use two TDC channels with uniform bin widths of 392 ps, we will achieve the time difference measurement precision about 160 ps.)

Recall that the average bin widths of our Gray code oscillators are about 256 and 271 ps, but the widths of bins are not the same. Wider bins make very significant contributions to the measurement errors causing large standard deviation.

Clearly, temperature variations will cause changes of the propagation delays in semiconductor devices and the TDC bin widths will change for different temperatures. However, if the calibration process discussed above is performed just before the data taking, it can be reasonably expected that the effect of bin width variations can be accounted for. In fact, the measurement data themselves sometimes can be used for calibration.

In our tests discussed here, data were taken at various environments for different times after powerup. The results for these conditions remain the same after calibrations.

VI. IMPROVING TDC WITH MULTIPLE MEASUREMENTS

Some practical applications may need better TDC measurement precision that can be fulfilled by combining multiple measurements.

The simplest method of using multiple measurements is making an average of measured values. In our test described above, a single input pulse is measured 4 times at clock edge 7 to 10. By simply calculating the plain arithmetic average, one can expect that the measurement error will reduce accordingly. The differences of the averaged time values are plotted in histograms as shown in Fig. 11.

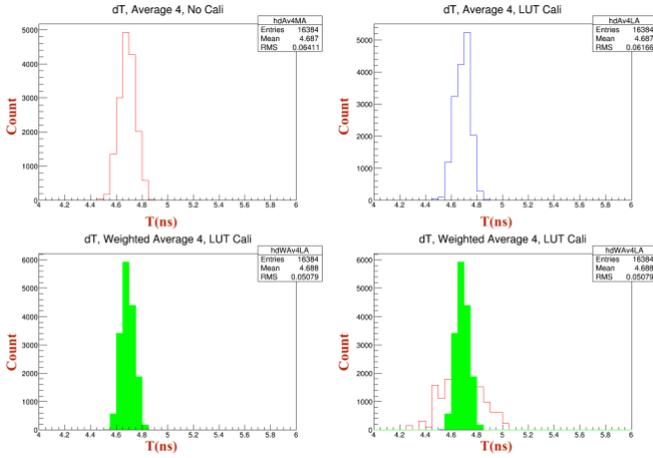


Fig. 11. Results of several average schemes

It can be seen that the difference of the averaged measurements indeed has a much narrower peak than the single measurement. The first histogram corresponding to the rough conversion without calibration as described earlier. Even without bin-by-bin calibration, the measurement errors can still be reduced significantly (from 160 ps down to 64 ps).

In the second histogram, the raw TDC codes in each measurement are converted using the calibration table. With the bin-by-bin calibration, it can be seen that the measurement precision is further improved (down to 62 ps).

Since the TDC raw codes in the 4 measurements may have different bin widths, it is clear that they have different measurement precisions. Obviously, a narrower bin has a finer measurement precision. Therefore, a weighted average will have even narrower peak in the histogram, as shown in the third plot above. The RMS error in our tests have reduced to 51 ps. The fourth plot compares the peak widths of the weighted average and the original histogram for single measurements.

To further understand the schemes of plain average and weighted average, an example extracted from the experiment data is shown in Fig. 12.

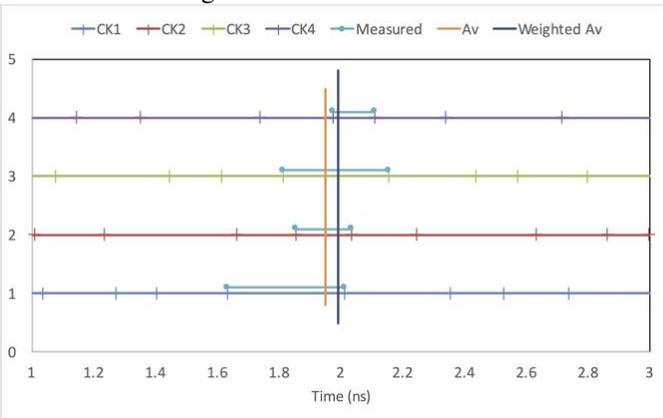


Fig. 12. Relative timing positions of TDC bins for measurements at the 4 clock edges

The measurements by the different clock edges can be visualized as shifted bins as plotted above. In a given event, the raw TDC codes may have different bin widths. Narrower bins can be trusted more and therefore we assign higher weights to narrower bins when we calculate the weighted averages. A

simple choice of the weight is the inverse of squared bin width, as in the following equation:

$$T_{WA} = \frac{\sum_{m=7}^{10} t_m / w_m^2}{\sum_{m=7}^{10} 1 / w_m^2} \quad (2)$$

In the equation above, 4 measurements for edges 7 to 10 are used to calculate the weighted average and t_m and w_m are calibrated time and width of the bin, respectively.

With the weight of inverse squared bin width, the influence of wider bins to the final average is reduced while narrower bins contribute to the measurement more. As a comparison, regular arithmetic average treats all measurements with the same weights which may not provide good results if the bin width variation is large. In the plot shown above, the line marked “Av” is the arithmetic average and one can see that it is not inside to range of the bin of the measurement 4. This is because measurement 1 has a very wide bin which should not make very big contribution in the average. On the other hand, the weighted average in the plot above falls into all measurement bin range, which is more meaningful than the arithmetic average.

It should be emphasized here that improved measurement precisions discussed above are still “single shot measurement precisions”, since the Gray code oscillator is started by a single shot input and the multiple measurements is done by capturing the oscillator values at different clock cycles. Obviously, the multiple measurements are taken by a single TDC channel, rather than multiple channels. However, users may choose to gang several TDC channels together to make multiple measurements, or may also choose to combine the approaches of both ganging channels and multiple clock cycles for further improvements of the measurement precisions. Again, these schemes are still belong to single shot measurements.

In some applications, the input signals are repeating periodic pulses. The TDC naturally makes multiple measurements and once the results are averaged, more dramatic improvement of measurement precision can be achieved, but this case belongs to periodic signal measurements.

VII. DISCUSSIONS

Comparing TDC schemes based on thermometer code and Gray (or binary) code, the later uses less resources since each bit carries more information than that in thermometer code, which makes the Gray (or binary) code based TDC a good candidate for low resource scheme. Resource usage is directly connected with the power consumption. For example, with less sampling DFF registers, less clock input ports are toggled with the CLK signal.

In the early history of TDC, a system clock is used to drive a Gray code counter and the input pulse drives the DFF register array. It should be pointed out that when a system clock is available to drive the counter, it is unnecessary for the counter to use Gray code if the timing uncertainty is appropriately confined [8]. However, if time sequence is not generated by a counter driven by the system clock, such as in the combinatorial oscillator based TDC discussed in this paper, Gray code becomes necessary. In fact, if the multi-bit combinatorial

feedback oscillator is not running in Gray code, it will not oscillate stably.

The reason of using combinatorial oscillator rather than using counters is that the oscillators have finer bin width for a given silicon technology. When a counter is implemented, the clock period must be larger than the sum of the worst case propagation delays of the combinatorial logic, feedback paths plus D flip-flop which limits the counter speed. While in a combinatorial Gray code oscillator, the propagation delays due to the flip-flops are eliminated and there is no need to wait for the slowest feedback paths. Therefore, the combinatorial Gray code oscillators run faster than counters under the same silicon technology. For example, in our oscillators discussed earlier, the typical time interval of running each Gray code step is about 250 ps, which would require a 4 GHz system clock if it is implemented with a Gray code counter that is not possible for the FPGA chosen.

Of course the advantage of the counters is the even bin width of the TDC output codes since the bin width is determined by the clock period, which can be viewed as a drawback of the combinatorial Gray code oscillators. Given the availability of bin width calibration, however, either online or offline, uneven bin width is not an issue anymore. It is less important to pursue smaller differential non-linearity than in early TDC history, especially when either extra silicon resource, power consumption or design efforts are needed.

In many applications such as drift chamber or similar detectors, a single measurement with a single TDC channel at 150 ps as we achieved in this work may already be satisfactory. If a better measurement precision is needed, one may combine several measurements, either time values captured with the burst pulses in the CLK signal as we discussed earlier, or ones from several parallel TDC channels. Since the Gray code oscillator TDC scheme uses less silicon resource, it is more likely to be affordable to fulfill the system requirement within the silicon area and power consumption budget than other TDC schemes. In our design described in this paper, a channel of TDC uses only 8 logic elements (a logic element is counted as a combinatorial lookup table and a DFF register) for its timing critical part including the Gray code oscillator and the sampling registers and therefore it is possible to fit more than 1000 TDC channels even in a low-cost FPGA.

The uneven bin width for the Gray code oscillator TDC sometimes is a good feature when we need to combine multiple measurements into a finer measurement. For example, when using several samples as discussed earlier, it is crucial for the bin edges of various samples not to align together, otherwise the arithmetic or weighted average will not improve the measurement precision. The TDC codes with uneven bin width have less chance to align together in such case.

The Gray code oscillator TDC is not only suitable for FPGA implementation, but also for ASIC implementation. Note that the original Gray code scheme, the reflected binary code has a very unified data dependency: all up transitions of B0 cause up transitions of B1; all down transitions of B0 are caused by B1 down transitions; all transitions of bits other than B1 and B0 are caused by down transitions of B0, which further cause the up

transitions of B0. With an appropriate design of the logic generating Gray code that takes the advantage of unified data dependency, it is possible to further reduce the feedback loop delay so that a higher oscillating speed or finer bin widths can be achieved.

REFERENCES

- [1] S. Nolting, L. Liu and G. Payá-Vayá, "Two-LUT-based synthesizable temperature sensor for Virtex-6 FPGA devices," *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, Ghent, 2017, pp. 1-4.
- [2] R. D. Cousins, C. Friedman and P. L. Melese, "32-channel digital 6-bit TDC with 2.5 ns least count," in *IEEE Transactions on Nuclear Science*, vol. 36, no. 1, pp. 646-649, Feb. 1989.
- [3] LeCroy Co., "Model 2277 32 Channel CAMAC TDC", (1991) available via: { <https://teledynelecroy.com/lrs/dsheets/dslib.htm> }
- [4] C. Li and H. Kobayashi, "A gray code based time-to-digital converter architecture and its FPGA implementation," *2015 IEEE International Symposium on Radio-Frequency Integration Technology (RFIT)*, Sendai, 2015, pp. 178-180.
- [5] Xilinx Corporation, "*Kintex-7 FPGAs Data Sheet (DS182)*", (2018) available via: {<http://www.xilinx.com/>}
- [6] Xilinx Corporation, "*KC705 Evaluation Board for the Kintex-7 FPGA User Guide (UG810)*", (2018) available via: {<http://www.xilinx.com/>}
- [7] J. Wu, "Uneven bin width digitization and a timing calibration method using cascaded PLL," 2014 19th IEEE-NPSS Real Time Conference, Nara, 2014, pp. 1-4.
- [8] J. Wu, J. Odeghe, S. Stackley and C. Zha, "Improving single slope ADC and an example implemented in FPGA with 16.7 GHz equivalent counter clock frequency," *2011 IEEE Nuclear Science Symposium Conference Record*, Valencia, 2011, pp. 2183-2187.