공학석사 학위논문

Development of Multichannel Digital Pulse Processing Platform for recording time of arrival

입자도달시간 검출을 위한 다채널 디지털 펄스 처리기 개발

2017 년 **2** 월

The Graduate School Seoul National University

Dept. of Electrical and Computer Engineering

SHANG HYUB KIM

입자도달시간 검출을 위한 다채널 디지털 펄스 처리기 개발

Development of Multichannel Digital Pulse Processing platform for recording time of arrival

지도 교수 서종모

이 논문을 공학석사 학위논문으로 제출함

2017 년 2 월

서울대학교 대학원

전기컴퓨터공학부

김상협

김상협의 공학석사 학위논문을 인준함

2017 년 2 월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

Abstract

Applications to measure accurate time intervals of sub-nano second resolution become popular as sensor and detector technology develops rapidly in the field such as nuclear physics, bio-medical nuclear medicine diagnosis (PET, gamma cameras), time-of-flight measurement and range finding area. As a result, a demand is increasing for digital pulse-processing platform that simultaneously records time of arrival, or time-to-digital converter (TDC), with single-shot accuracy from multiple channels. In this thesis, a multichannel, high timing resolution time-to-digital converter with single-shot accuracy is presented. For low-cost system and flexible reconfiguration, Field Programmable Gate Array (FPGA) is used, and adjustable phase shifted clock scheme is proposed to ensure single-shot accuracy while holding ability to implement multiple channels, which is one of benefits of the phase shifted clock scheme. A TDC system is built based on the proposed architecture. Testing and characterization was done on the system to evaluate the performance of timing accuracy, single-shot capability, timing jitter, Differential Non-Linearity (DNL), Allan deviation (Time base stability) and system throughput.

Key Words: Time-to-digital Converter, TDC, time of arrival, time-of-flight, phase shifted clock
Student Number: 2012-20749

Contents

List of Tables

List of Figures

Introduction

1.1 Time-to-Digital Converter

It is getting more important to measure accurate time intervals (sub-nano second resolution) in many applications such as nuclear physics, bio-medical nuclear medicine diagnosis system (PET, gamma cameras), time-of-flight spectrometers and range finding applications as sensor and detector technology develops rapidly.

The time-to-digital converter (TDC) provides a high resolution, accurate, and high throughput time measuring by converting time intervals or time of arrival into a digital code just such as ADC translates an analog voltage level into a digital code.

1.1.1 Time interval

Simply a timing resolution shows capability of measuring an interval of time of arrival. In many applications such as high-energy and particle physics, sub-nano second precise time-interval measurement has been required. Due to rapid developing of CMOS technology, TDC becomes useful in a wider filed with a very fine timing resolution [4]. Recent ASIC-based TDCs demonstrate less than 10 pico-second timing resolution with single-shot measurement [5]. Some FPGA-based TDC technology shows less than 100 pico-second timing resolution with multi sampling [2][3][5].

1.1.2 Dead time

Dead time is a minimum time between two measurements or frames. The dead time needs to be minimized for capturing streaming pulses continuously, and is usually required to be less than several microseconds [3]. Pipelining in the processing module [12], alternating the inputs of TDC logic [13], or switching operation modes between measuring and calibrating [14] is proposed to decrease the dead time. The Dead time determines the throughput of the system, and becomes more critical when implementing many channels (>100 channels) is required

1.1.3 Dynamic range

Dynamic range is a maximum time interval that can be measured. Some applications require very large dynamic range up to several hours. Mostly dynamic range is determined by a bit width of a counter running at operating frequency, which is usually ranging from 100MHz to the maximum of 500MHz or even more. Sometimes the dynamic range becomes a factor to affect the system throughput because it takes a certain portion of bus width when a digital code is being generated. Also, very large dynamic range requires a very good clock quality of high stability to ensure long term accuracy of pulse pair resolution.

1.1.4 Differential non-linearity (DNL)

Differential non-linearity is a variation of a single bin width and can be expressed as the measurement number of counts divided by the expected number of counts [2]. A positive value means the bin width is bigger than the expected or calculated resolution, and a negative values means the bin width is smaller than the expected resolution. A zero DNL shows an ideal bin width and each bin has no variation. A TDC with a large DNL is lacking single-shot capability, since it is more likely that each measurement falls into adjacent or even farther bin.

Usually FPGA-based TDC suffers from its large DNL than analog-based or ASIC-based TDC. FPGA has predefined logic cell and routing architecture, and clocking resources. The logic cells are placed in a predefine location inside a block, called logic array block or configurable logic block, and give very little control to adjust propagation or routing delays. And the routing path from the clock source to logic cells differs and is predefined, so the clock skew is induced leading to non-linearity [2]. In ASIC-based TDC, it is possible to minimize or eliminate these delay variation or clock skew by manually controlling logic cells and clock resources.

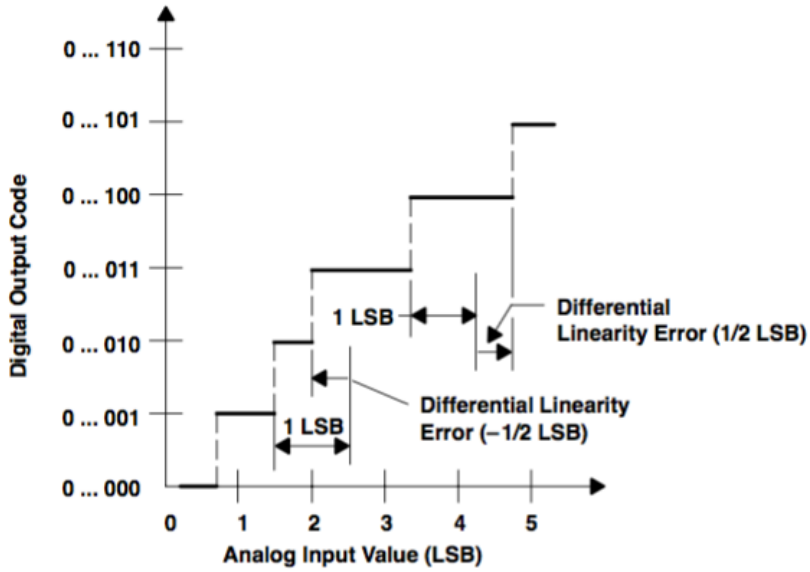


Fig. 1.1 DNL represented in transfer function [7]

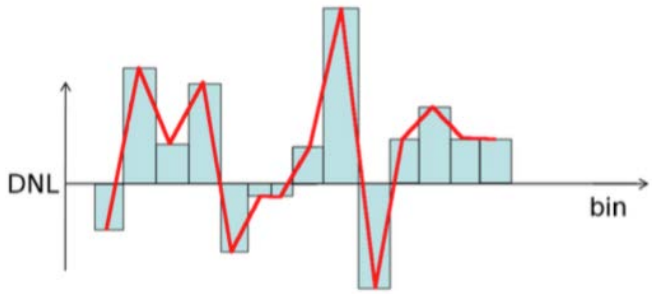


Fig. 1.2 DNL represented in bin histogram [2]

1.1.5 Dynamic Calibration

Besides a static non-linearity caused by different propagation and routing delays, the bin width can change during the operation due to temperature and power supply voltage. It is required to compensate these variations dynamically to minimize DNL. For example, ASIC-based TDC adjusts a control voltage to change the phase of clock source, or to tune the internal cell delays [2]. In an FPGA-based TDC, since it is impossible to compensate using an analog method, digital calibration such as post-processing is preferable.

2. State of the Art

Generally TDC is implemented in two ways: analog and digital. Analog-based TDC uses current source to charge a capacitor, and samples the voltage level when it gets a trigger. Digital-based TDC usually consists of coarse and fine counter. A coarse counter is a simple counter running at high speed, and fine counter converts a sub-period value into a digital code using various methods such as delay chain or shifted clock sampling.

2.1. Analog-based TDC

The traditional approach to time-to-digital conversion using analog method is to convert the time interval into a voltage level, and digitize into a digital code using analog-to-digital converter (ADC) [4]. Start signal is used to trigger a pulse generator to charge an integrator, and stop signal to stop charging. [Fig. 2.1] The pulse width of V_p, a time interval between start and stop, is then converted to a voltage level V_{eq}, which is then digitized by ADC later. The dynamic range (DR) is determined by the maximum number of bits N the ADC offers, and minimum time resolution T_{LSB}.

$$DR = 2^N \bullet T_{LSB} \text{ [7]}$$

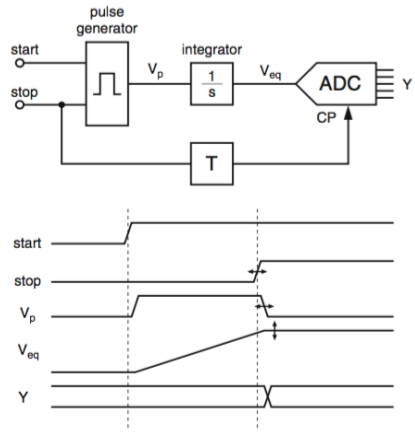


Fig. 2.1 Block (above) and signal diagram (below) of basic analog time-to-digital converter [4]

There are several issues that may degrade the TDC performance unless considered carefully. The building blocks, i.e. the pulse generator, the integrator, and the ADC have to meet the full linearity of demands of the overall TDC [7]. One of the elaborate approaches to resolve this is to use dual-slope conversion. Similarly, start signal is used to trigger a pulse generator, and stop signal stops charging. In this case, the stop signal triggers a second integrator that has reduced integration constant $\frac{1}{p}$. A comparator then generates a signal while it is detecting when the output of the first and second integrator becomes equal. The interval of the signal generated by the comparator is stretched by the factor $(1+p)$. When p is large enough, a fast digital counter can be used for the conversion.

But, due to the stretched time interval, this limits the measurement throughput, thus the system has increased dead time.

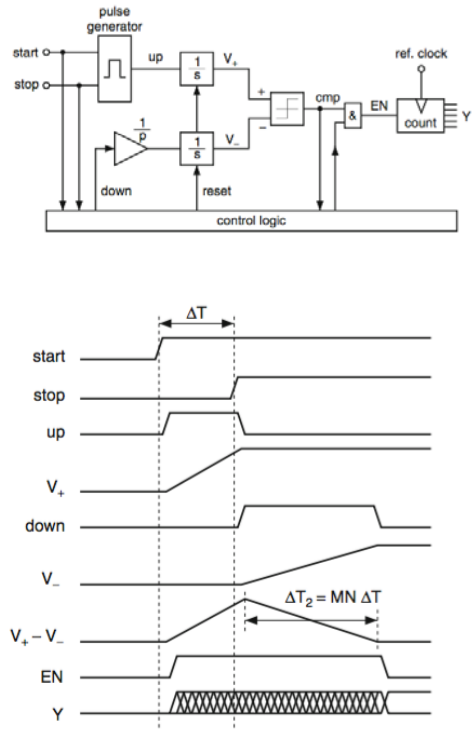


Fig. 2.2 Block (above) and signal diagram (below) of dual-slope analog-to-time-interpolation [4]

2.2. Digital-based TDC

Mostly the digital-based TDC consists of a coarse and fine counter. The coarse counter runs at high speed operating clock. It can be implemented in a simple binary counter. But in designs that use a stop signal directly from a HIT or TDC input to stop the operating of the flip-flop in the counter, an unstable state can happen. Because of asynchronous relationship between the operating clock and input signal, the flip-flop in can go into meta-stable state. To prevent this from happening, using two counters in parallel with two different phases is proposed [6]. Another way is to use a gray code

counter, so the error would be only one bit, since the gray code counter flips only one bit at a time [8].

The fine counter, which also runs at the same operating clock for the need of synchronization with coarse counter, is designed to provide the timing resolution beyond the operating clock period. To implement the fine counter, methods described below have been developed.

2.2.1. Delay Line TDC

Delay line TDC achieves fine timing resolution by delaying input signal through a chain of digital buffers. The timing resolution is then determined by the delay of the buffers. A start signal travels along the delay chain, and as a stop signal arrives, it records the states of the delay chains at the moment. Then the output sampled by the stop signal becomes a thermometer code, and this indicates how far the start signal propagated after the rising edge of the flip-flop, which is running at the same clock that the coarse counter runs at.

So the time interval ΔT ($T_{stop} - T_{start}$) is,

$$\Delta T = N \bullet T_{LSB} + T_{coarse} \text{ (in ideal case)}$$

where T_{LSB} is the delay of a buffer in the delay line, N is the number of buffers that the start signal propagated through, and T_{coarse} is a count value from the coarse counter.

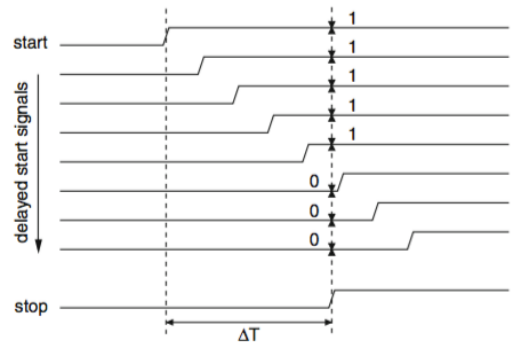


Fig. 2.3 Operating principles of Delay line TDC [4]

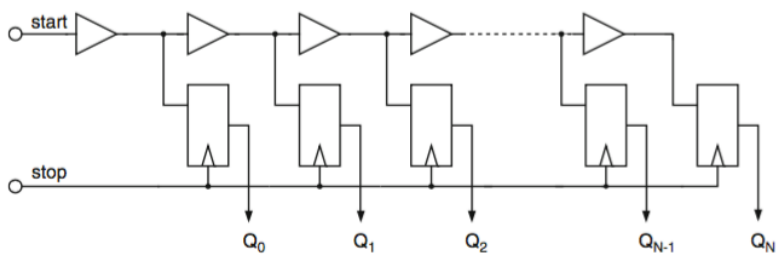


Fig. 2.4 Basic delay-line based time-to-digital converter [4]

Any deviation in the delay line or skew from the clock distribution tree affects the non-linearity of the TDC performance. Non-uniform delay of the chains reduces single-shot

2.2.2. Inverter Delay Line TDC

The timing resolution of the delay line TDC can be improved by using CMOS inverters instead of buffers [1][4]. In this case, instead of detecting HIGH-LOW transition to measure the propagation when buffer chain is used, the alternation of HIGH-LOW sequence is used to detect the propagation of the start signal.

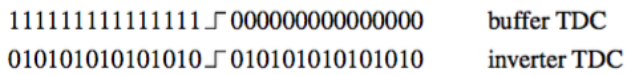


Fig. 2.5 Detecting the propagation for buffer and inverter TDC [4]

The rise and fall delay of the inverter need to match to be equal to minimize the non-linearity of the TDC performance [1][4].

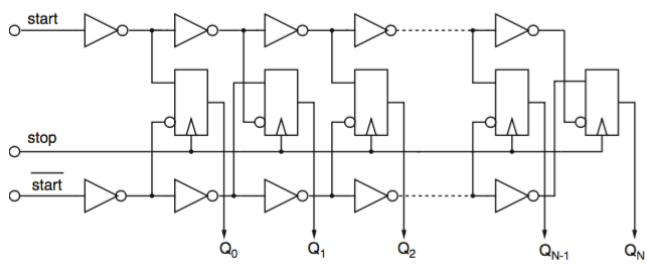


Fig. 2.6 Delay-line based time-to-digital converter using inverter chains [4]

2.2.3. Carry-chain Delay Line TDC

In FPGA-based TDC, a fine counter is implemented using carry chain structures that is specific to vendors or the device architecture [9]. It is also called tapped delay line TDC. The carry chains have a set of predefined cells and routings as shown Fig. 2.7. The timing resolution is then determined by the propagation delay of the multiplexers. In FPGA architecture, the delay of the multiplexer is short enough to have a good timing resolution. The routings of the multiple carry chains are straightforward. This makes carry chains a good alternative of buffers in the FPGA-based TDC design.

As shown in Fig. 2.8, the output of the delay components, the carry chains, is sampled by the followed flip-flop. As the start signal travels along the delay line, the outputs of the flip-flops indicate the thermometer code of how far it propagates after the most recent clock. The clock driving the flip-flops is the same fast clock that operates the coarse counter. In order to achieve a desired fine timing resolution, the clock frequency usually goes up to 500 to 600MHz.

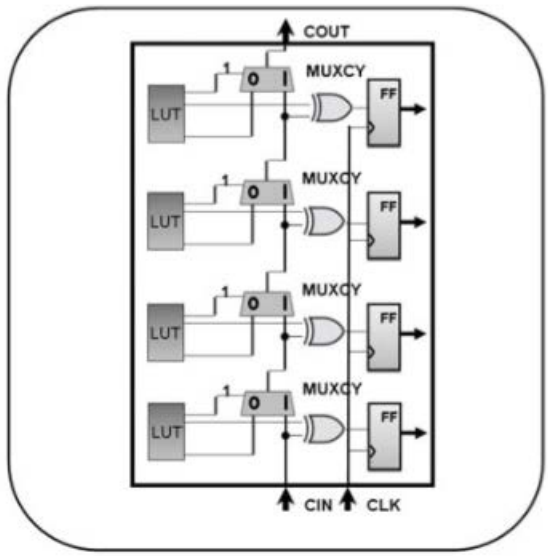


Fig. 2.7 Carry chain block in the Virtex-5 device [9]

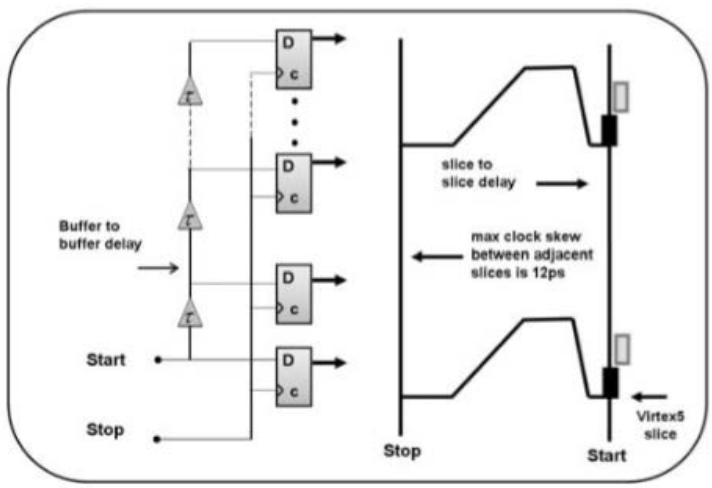


Fig. 2.8 Delay line TDC using carry chains inside the FPGA [9]

But, the FPGA-based tapped delay line TDC suffers from the non-linearity of the performance. Clock skew can be one of the reasons due to the predefined routing of the clock distribution tree. Another would be discrepancy of routing delays of the carry chains. The routing delays inside the carry chain block are kept identical, but the

ones between the blocks have longer delays. There are approaches to compensate the non-linearity due to this uneven delay in the carry chain using post-processing [10].

2.2.4. Vernier Delay Line TDC

The Vernier Delay line TDC makes use of the difference between two propagation delay values to significantly better timing resolution out of gate delay. The Vernier TDC has a pair of delay lines that has slight different propagation delays. A start signal travels through one of the delay line, while a stop signal samples as it propagates through the other line. The design implemented on the commercial FPGA has a timing resolution of 200ps [11]. Since CMOS process is very temperature sensitive so frequent calibration is required [3].

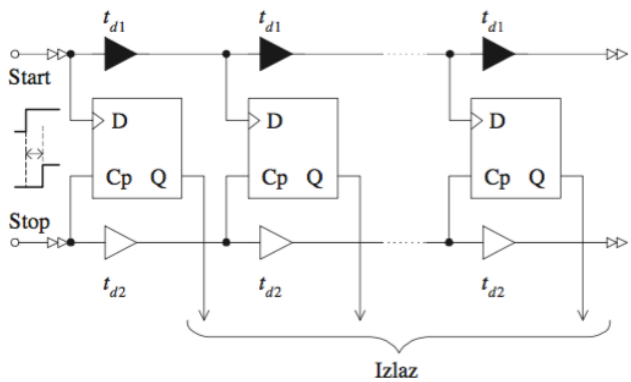


Fig. 2.9 The Vernier Delay line TDC [3]

2.2.5. Shifted Clock Sampling TDC

The Shifted Clock Sampling (SCS) TDC uses multiple phases in the sampling clocks to achieve sub-period timing resolution. A TDC input is routed to the flip-flops that have the same clock frequency with equally distanced phases. Then the outputs of the flip-flops are synchronized to the main operating clock.

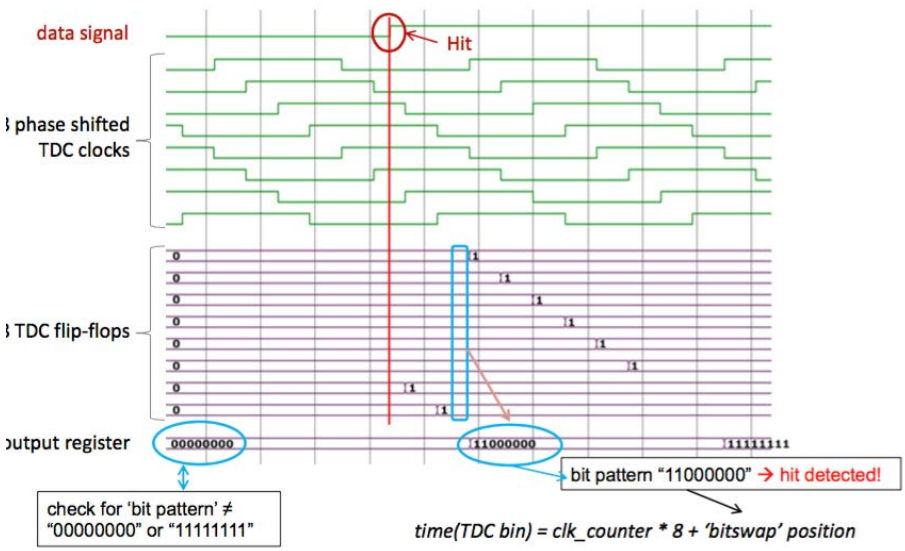


Fig. 2.10 Signal timing diagram of SCS TDC (8 phase shifted clocks) [5]

As a LOW-HIGH transition happens, the values of outputs of the flip-flops changes from "00000000" to "11000000" (Fig. 2.10). The number of 1's from the leftmost indicates how far the signal propagates from the most recent clock (main operation clock). The main operating clock is used to run a coarse counter. So the time stamp value T_{TS} of

the transition is,

$$T_{TS} = T_{coarse} - N \bullet T_{LSB}$$

Where T_{LSB} is a bin size in second ($= \frac{\text{Clock Period}}{\text{\# of Phase Shifted Clocks}}$), N is the number of 1's from the leftmost, and T_{coarse} is a count value from the coarse counter. At every clock period, if the outputs of the flip-flops are not "00000000" or "11111111", the hit is recognized and the data can be transferred to the decoding logics.

In order to avoid setup and hold requirements due to its multiple clock domains, a concept of 'partition' is introduced. To synchronize the outputs of the flip-flops, the clock domains are partitioned in two groups to maximize the setup time before reading out (Fig. 2.11). The first partition reads only outputs of the first 4 flip-flops, and the second the last 4 flip-flops.

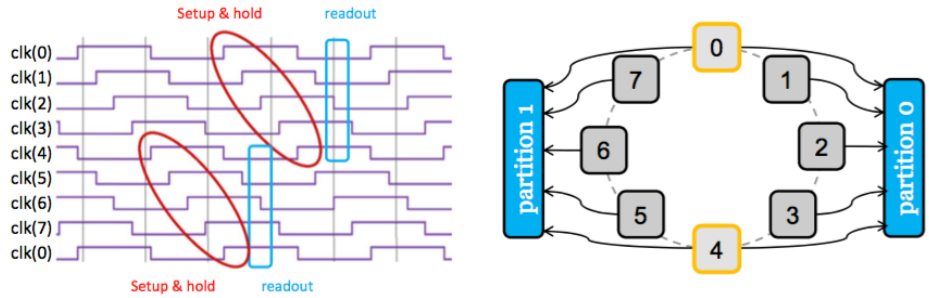


Fig. 2.11 Two partitions used for readout to meet timing requirements [5]

The SCS TDC is advantageous when implementing many channels (>100 channels) within one FPGA. For example, an image sensor array with time-of-flight (TOF) feature

can easily need more than 100 independent channels to process its TOF pulses simultaneously. The FPGA-based SCS TDC is a good candidate for such applications.

Achieving good linearity in the SCS TDC is a challenging task. The main contributions to the non-linearity are the clock phase error/clock skew and the routing skew of the input signals to the different flip-flops [5]. The clock phase is well controlled by the clock resources available in the FPGA. It is inevitable to have clock skew due to its predefine clock distribution trees, but this mainly affects the deviations among the channels and controlled to be within acceptable range by the tool. The routing skew is more difficult to control, because the FPGA implementation tools don't provide a way to choose certain routes [5]. To minimize the variations due to the routing, the user-defined placement was used so that the auto-router finds appropriate connections. Though the routing deviation is controlled to some extent, it has inherent non-linearity characteristics, which would affect the performance of the TDC especially for single-shot application.

3. Implementation

A multi-channel TDC that is capable of measuring a high resolution with high throughput is needed in many applications such as high-energy physics, positron emission tomography (PET), and time-of-flight 3D cameras. Those applications are based on detector arrays that produce pulses as particles such as photons hit the cells of the arrays. For example, a time-of-flight 3D camera detects the reflected photons after it bounced back from the object. The time interval gives information about the distance from the object to the camera. For the 3D camera application, at least 64 or 128 channels with high throughput capability are required to display images with 3D depth information simultaneously. Also, time measuring with single-shot accuracy is needed to measure accurate photon arrivals.

In order to achieve a high performance TDC with single-shot accuracy, Application-Specific Integrated Circuit (ASIC) based TDC would be the best fit in general. The manually controlled delay chains with a fine propagation delay achieve a resolution of a few picoseconds with low DNL [2-9]. It is worth the time and money to develop ASIC-based TDC for a specific application if used in large amounts. But, for applications that need custom requirements and don't expect massive production, FPGA-based TDC would be an alternative.

Some FPGA-based TDC achieves a high resolution with high channel counts. But, all of them lack single-shot accuracy. In

this paper, a multi-channel, high resolution FPGA-based TDC with single-shot accuracy is developed.

3.1. Implementation Objectives

From the requirements of the possible applications, the implementation objectives are extracted (Table 3.1). To implement high-channel counts in FPGA device while achieving high resolution, SCS architecture is selected. The flexible implementation of the number of channels is desirable between 64 and 256 channels. The DNL performance less than 1.5 LSB is capable of providing sufficient single-shot accuracy. Maximum throughput of 3.5MSPS per channel is equivalent to 448MSPS of the system throughput when 128 channels are implemented. For applications such as optical communication system using pulse position modulation (PPM), maximum 30 seconds of a large dynamic range is required, which determines a word size of one time stamp to be 64-bit word. If one sample or time stamp consists of 64-bit word, the total system throughput will be 3.5GB/sec.

Architecture	FPGA-based Shifted Clock Sampling
Channel count	Min. 64 channels Max. 256 channels
Timing Resolution	< 200ps
DNL	< 1.5 LSB
Throughput	Max. 3.5MSPS/channel (when 128 channels implemented) Max. 450MSPS system throughput
Dead Time	< 100ns
Dynamic Range	Max. 30 seconds

Table. 3.1 Implementation objectives

3.2. Architecture

In order to achieve high-channel counts in FPGA device, the SCS architecture is chosen. The SCS architecture needs only a few flip-flops that are equivalent to the number of phase-shifted clocks to implement a single channel. But the main drawback is its non-linearity due to the routing skew. FPGA has its predefined routing patterns, and it is a high challenging task to make the routing path from the input equivalent to each flip-flop (Fig. 3.1) The deviation of the routing paths directly contributes to the DNL. Moreover, each channel should have the same routing characteristics, so be able to work as an independent channel (Fig. 3.2).

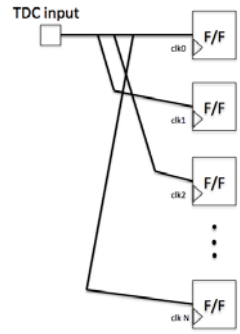


Fig. 3.1 The deviation of the routing paths affecting the DNL

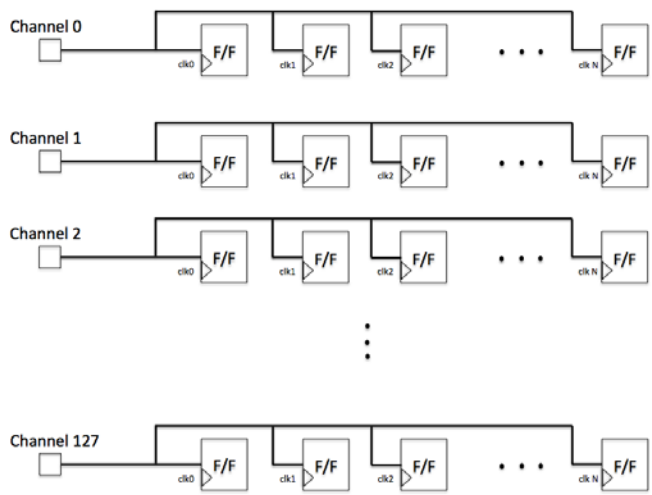


Fig. 3.2 Channels should have identical routing characteristics

3.2.1. Compensator

To compensate the deviation of the routing path from the input to each flip-flop, a compensator is proposed to adjust the clock phases instead of changing the routing paths. A compensator moves the clock phase value to nullify the effect of the routing variations. Through a calibration process, it detects the static deviation due to the routing skew as well as the dynamic deviation due to temperature and power supply

change. After calculating the amount to compensate, it changes the clock phase value so that all flip-flops have “effective” identical routing path from the TDC input.

3.2.2. Routing for multiple channels

It is important to keep identical the routing characteristics of all channels in order to make all channels working as an independent channel. Some FPGA vendors provide advanced routing options to some extent. In the proposed design, directed manual routing is used to duplicate the routing paths over channels (Fig. 3.3).

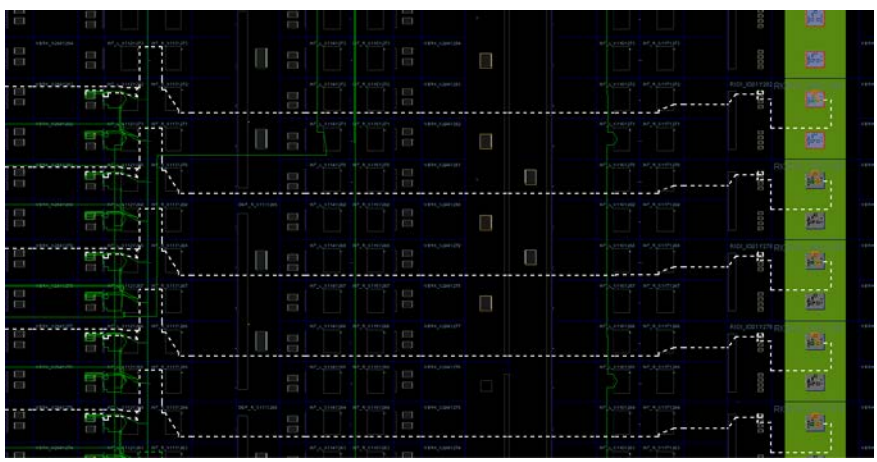


Fig. 3.3 The TDC front-end routing pattern identical across channels

3.2.3. FPGA selection

A Virtex-7 FPGA is selected to implement the proposed SCS architecture. The Virtex-7 series FPGA has clock management modules called Clock Management Tiles (CMT). Each CMT has one mixed-mode clock manager (MMCM) and one phase-locked loop (PLL). The Virtex-7 series FPGA has up to 24 CMTs depending on the device size, which provides sufficient clock resources to generate phase-shifted clocks. The clocks are controlled within a block called clock region, and each clock region can have up to 12 clocks that drive flip-flops within the region. The number of CMTs available in a specific device and efficient allocation of the flip-flops within a clock region determine the maximum number of phase-shifted clocks and timing resolution.

The Xilinx development platform provides constraints format based on the standard Synopsys Design Constraints that has been in use for more than 20 years. It supports various constraints to allow advanced place and routing options. The manual routing direction provides capability to make identical routing characteristics across the channels.

3.3. The System Design

The designed system consists of two main modules – the FPGA TDC and Digital Processing Unit (DPU) (Fig. 3.4). The FPGA TDC contains the modules to capture, process and store time stamp values. The FPGA TDC has one XC7VX485T-FFG1761 FPGA device and DDR3 memory chips for time stamp storage. All the internal modules are connected using the standard ARM® Advanced Microcontroller Bust Architecture (AMBA®) Advanced eXtensible Interface (AXI) protocol to the AXI interconnect then to the AXI master, which communicates with the DPU via the LogiCORE™ AXI Chip2Chip Interface.

As pulses come to the TDC inputs via LVDS, the TDC Front-End module detects rising edge transitions (can be programmed to detect falling edge) of pulses and delivers to Time-stamping Module. The Time-stamping Module converts the coarse counter value and the thermometer code of the fine counter into the time stamp value. Inside the Time-stamping Module, a simple Run Length Encoder (RLE) is used to reduce the fine counter code to fit the bus width of the data path. The time stamp value is then stored temporarily in the FIFO-type buffer. The FIFO AXI Direct Memory Access (DMA) Master moves the time stamp data into the off-chip DDR3 memory chips. The memory space of the DDR3 memory is directly accessible via the AXI by the DPU.

The DPU is responsible for functions of controlling, post-processing and data transfer. The DPU has an ARM®-based

Zynq XC7Z020 system-on-chip (SoC), which functions as an AXI master in the whole system. The DPU communicates with PC software via TCP/IP over Gigabit Ethernet, receiving commands and transferring time stamp data. Especially the Zynq SoC in the DPU can access the time stamp data via the AXI, so post-processing can be applied while the data still stay in the memory space. The embedded Linux OS (Petalinux 2014.4) is ported on the DPU, so the standard GNU C library can be applied for easy functional expansion in the future. This provides usefulness when there is a need of on-board processing to apply existing C-based algorithm code into the system.

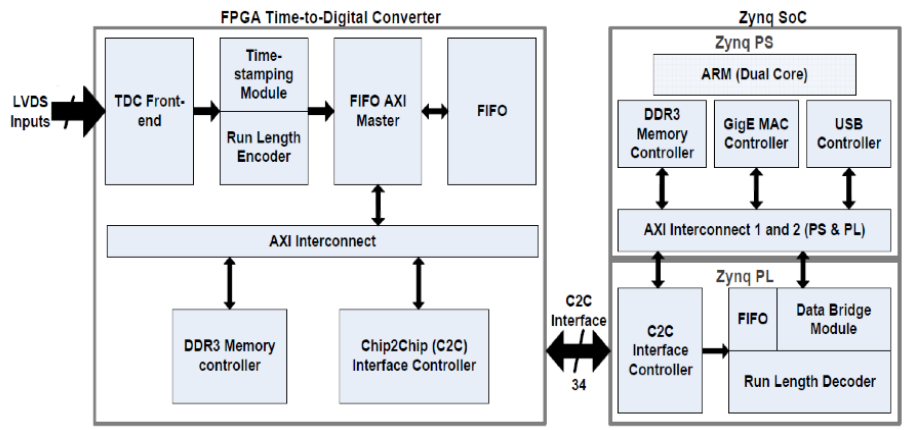


Fig. 3.4 Block diagram of the SCS TDC with Digital Processing Unit

3.3.1. FPGA TDC

Inside the FPGA TDC, all the peripherals are connected to the AXI master via the AXI interconnect. This maximizes reusability, flexibility and modularity in using design IPs generated in the system. For example, any module with AMBA interface can directly be connected to the FPGA TDC, which allows adding in-system processing features such as autocorrelation function in the future easily.

3.3.1.1. TDC Front-End Module

The TDC Front-End Module (TFEM) is a cluster of flip-flops for phase shifting sampling. The TFEM for one channel consists of 24 flip-flops (12 for sampling block, and 12 for synchronizer). Twelve 500MHz clocks with equally distanced phases drive the flip-flops. This provides 166.66ps ($=\frac{2ns}{12}$) of resolution. The TFEM detects rising edge transition by monitoring changes of pulse trains, and hands over 12 bit data per channel to the Time-stamping Module (TSPM). In the TFEM, a careful consideration is required when synchronizing data from the flip-flops with different phases in order to avoid setup & hold time violations.

3.3.1.2. Time-stamping Module

The Time-stamping Module (TM) calculates time stamp value from the coarse counter value and thermometer code from TFEM. The coarse counter runs at 500MHz. A RLE is used to reduce the bit width of a packet to fit the bus width of the 64-bit data path. This allows more room to the integer value of the time stamp in a packet that it has a large dynamic range of capturing time. The time stamp data is then transferred to the internal buffer FIFO.

3.3.1.3. FIFO AXI DMA Master

The FIFO AXI DMA Master is used to transfer time stamp data to the DDR3 memory. The DDR3 memory controller has an AXI-based interface that any AXI master can access the memory space. To maximize the data transfer rate or throughput, it uses 4 times widened bus width, 256-bit, of a time stamp packet. A clock for the AXI bus architecture runs at 200MHz and the AXI data bus width is 256-bit, so the system throughput is estimated to be minimum 200MHz x 4 samples (per clock cycle) x memory controller efficiency (as low as 60%) = 480MSPS, which would suffice the system throughput requirement.

3.3.1.4. Chip2Chip Interface Controller

The Chip2Chip Interface Controller is a vendor-specific IP from Xilinx LogiCORE™. It functions such as a bridge to seamlessly connect two devices over an AXI interface [16]. One side of the Chip2Chip interface becomes master device, the other side slave device. In the current design, the DPU is a master device, and FPGA TDC a slave device.

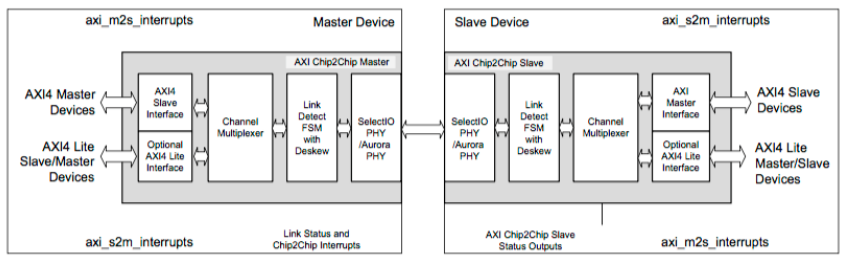


Fig. 3.5 AXI Chip2Chip Block Diagram

3.3.2. Digital Processing Unit

The Digital Processing Unit (DPU) has a Zynq SoC chip that has two ARM processor cores. Besides the ARM cores, it has hard-wired Gigabit Ethernet controller, DDR3 memory controller and AXI interconnect. With the embedded Linux loaded, the standard TCP/IP API functions can be used that the control and data communication made easy. With the hard-wired DDR3 memory controller, up to 1GB for program and data is provided, which will allow enough memory space for additional software features in the future. The Zynq SoC has a channel that the processor can access the programmable logic (PL) region via the AXI so that the user-

defined AXI peripherals can be seen by the processor in the memory map.

3.3.2.1. Data Bridge Module

The Data Bridge Module (DBM) is used when time stamp data is handed over to the PL peripherals. Additional post-processing can be done in a module in the PL region, and then the results are transferred to the DBM. The DBM has an access to the processor via the AXI interconnect and moves data to the processor region. The Run Length Decoder is needed in this case prior to the DBM.

3.3.3. Scalability

One of the key features of the proposed TDC design is modularity and scalability. Since the FPGA device itself is reconfigurable, the flexible configuration of channel counts and device selection depending on the types of applications is highly desirable.

Each TDC channel is designed as a modular component that allows scaling with minimal design changes such as modifying parameter values. Each channel works independently, and has a complete chain of pulse capturing and processing blocks (Fig. 3.7). Using this scheme, this design can be scaled to any number of channels, only limited by available FPGA resources. Fig. 3.8 shows how channels are combined to build multiple

TDC channels. The output bus of each channel FIGO goes to the Arbiter logic, then the system FIFO. The Arbiter can be configured by changing one parameter (number of channels), hence increasing the cannel number doesn't require design changes or modification except configuring parameter.

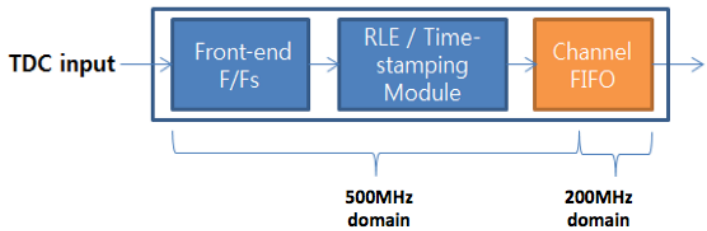


Fig. 3.6 Scalable, modular single channel SCS design

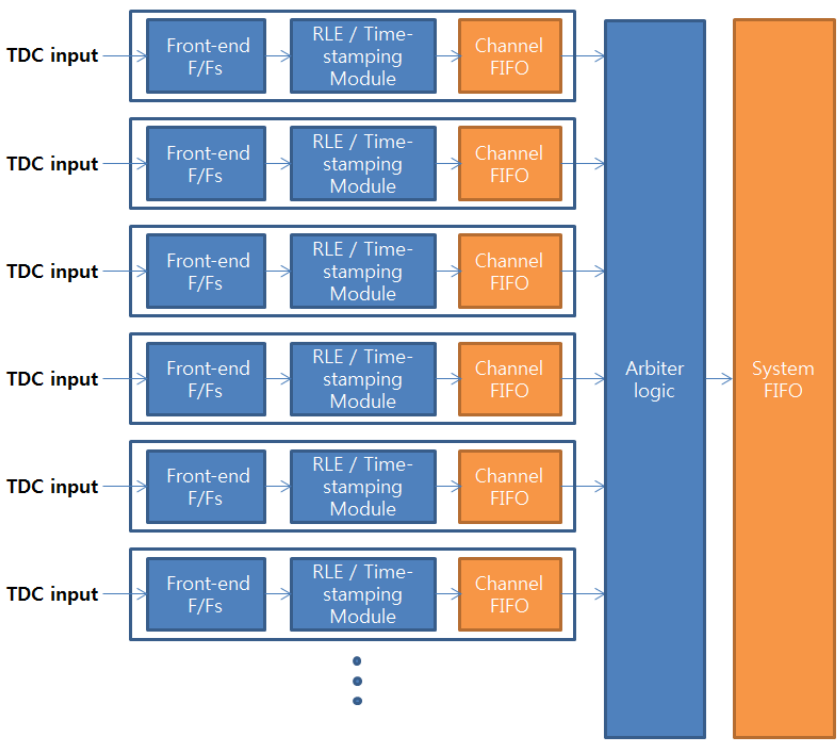


Fig. 3.7 Scaled SCS TDC Front-end

One of the challenging parts of this design is the requirement of distributed resource allocations due to the fact that each channel should have its own independent blocks, i.e. flip-flops, Time-stamping Module and FIFO. It means that some hardware specific resources such as FIFO need to be fit into the allocated area per channel, and routing wires should be done within each block. Fig. 3.9 shows the implemented 199-channel design using “block” to delimit the area of channels in order to keep the design modularized.

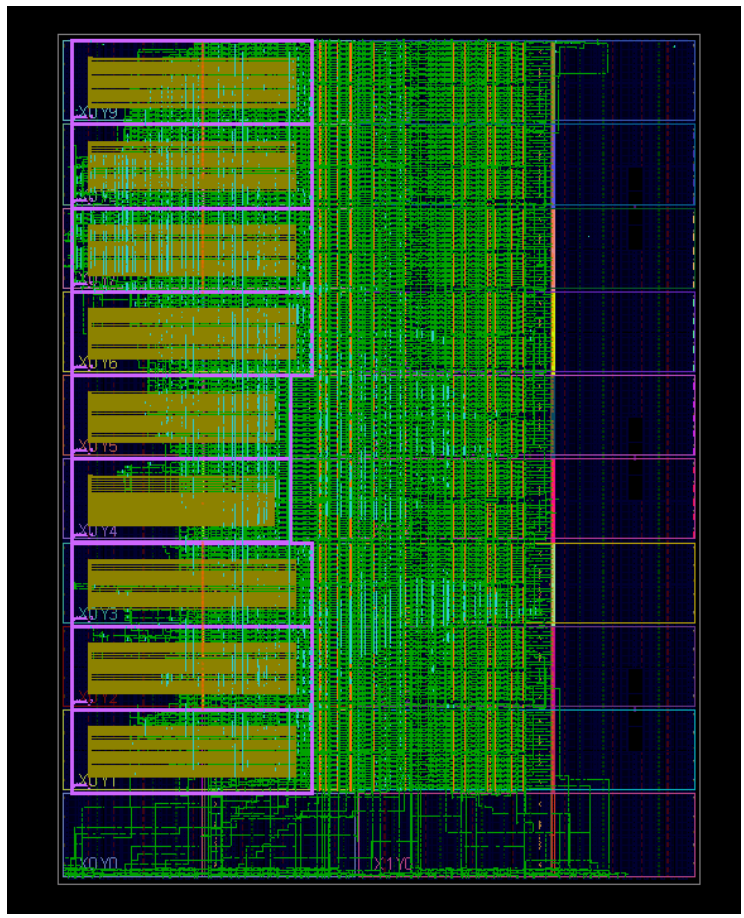


Fig. 3.8 Implemented 199-channel design

3.3.4. System Build

The FPGA TDC board and DPU board from the design above are built and tested (Fig 3.8). Evaluation boards are used to demonstrate the performance of the architecture.

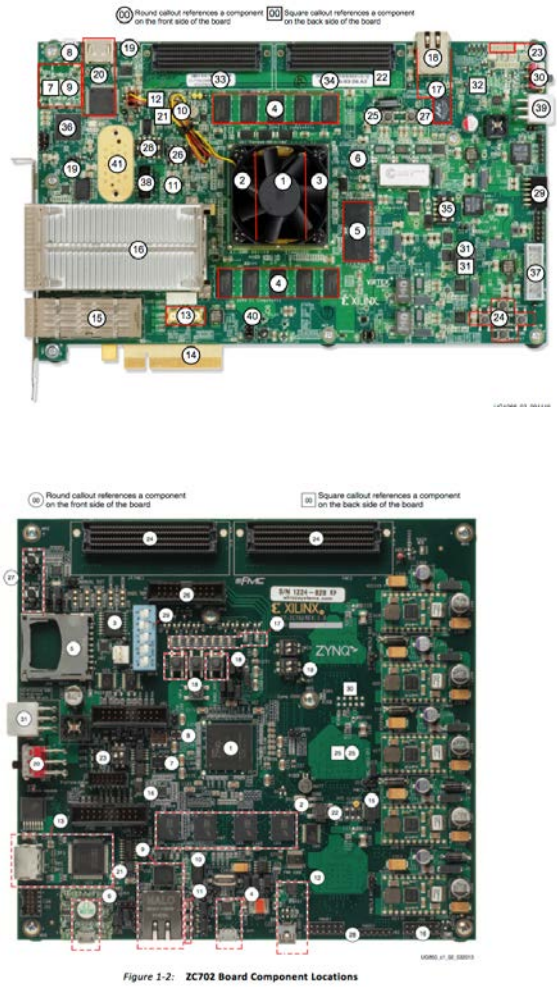


Fig. 3.9 Photos of FPGA TDC board (top) and DPU board (bottom)

3.3.5. PC Software

PC software is created to communicate with and receive time stamp data from the TDC system. The software saves time stamp data into CSV format so that various characterization and analysis can be done easily. The operating parameters can be configured through the software such as acquisition time or the number of TDC-enable pulses to capture.

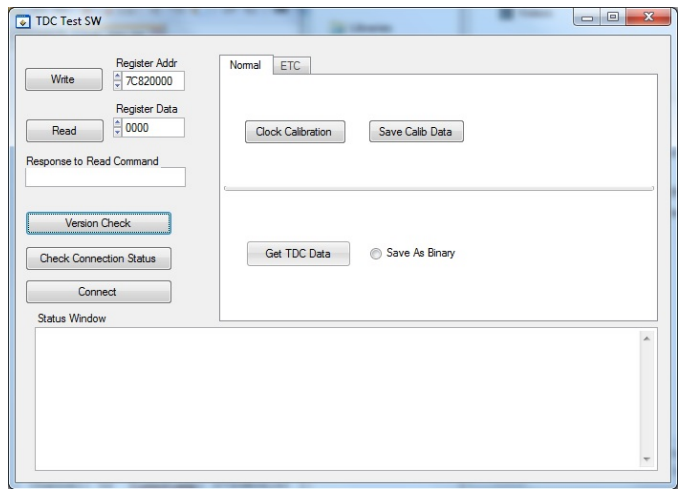


Fig. 3.10 Snapshot of the software GUI

4. Test and Characterization

Test and characterization was done to evaluate the performance of the TDC system. Set-up for testing and characterization is shown Fig. 4.1. A Virtex-7 evaluation board (VC707) is used to generate test patterns to the TDC system. The evaluation board can be configured to output test patterns with various pulse periods. It is able to receive external clock source so high quality, long-term stable clock is fed to the system from the pulse generator.

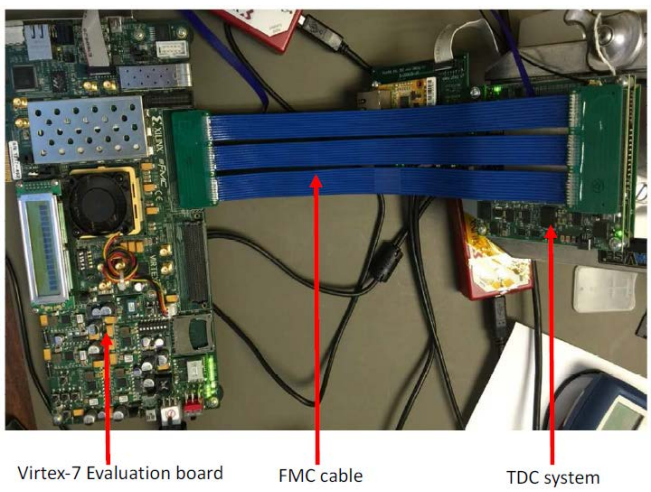


Fig. 4.1 Set-up for testing and characterization

4.1. Timing accuracy

Test patterns with varying pulse periods were injected to measure timing accuracy. Table 4.1 shows the mean and standard deviation values when test pulse trains are applied with multiple periods. The timing measurement of the

designed TDC is accurate, keeping the uncertainty in each capture within 1 LSB.

Test Pulse Period (ns)	Mean (ns)	Std. (ps)	Single-shot Timing uncertainty
125.34	125.34	24	Max. 1 LSB
149.76	149.76	16	Max. 1 LSB
1000.0	999.99	52	Max. 1 LSB
320010.5	320010.49	20.1	Max. 1 LSB
999,996.20	999,996.19	27.5	Max. 1 LSB

Table 4.1 Test result for timing accuracy

Fig. 4.2 is a profile of time stamp when test pulse trains with a period of 320010.5 ns are injected. In the plot, the time stamp values stay between 320010.34 ns and 320010.5 ns, which is within a range of 1 LSB (166ps).

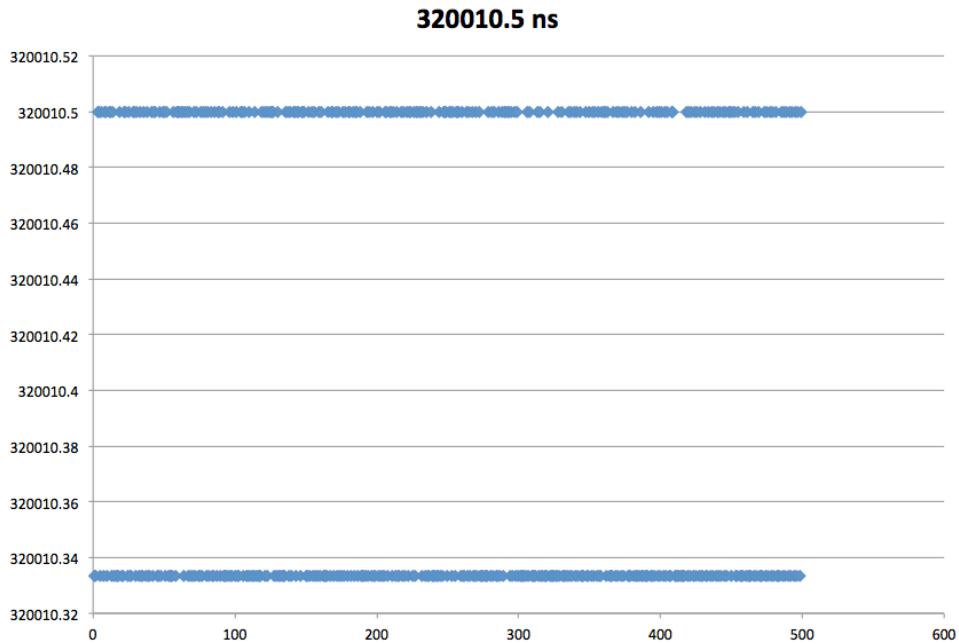


Fig. 4.2 Single-shot timing uncertainty within 1 LSB

Fig. 4.3 shows the timing jitter performance when two adjacent pulses with 1us apart are applied. It has 151.6ps of the full width at half maximum (FWHM), which is less than 1 LBS.

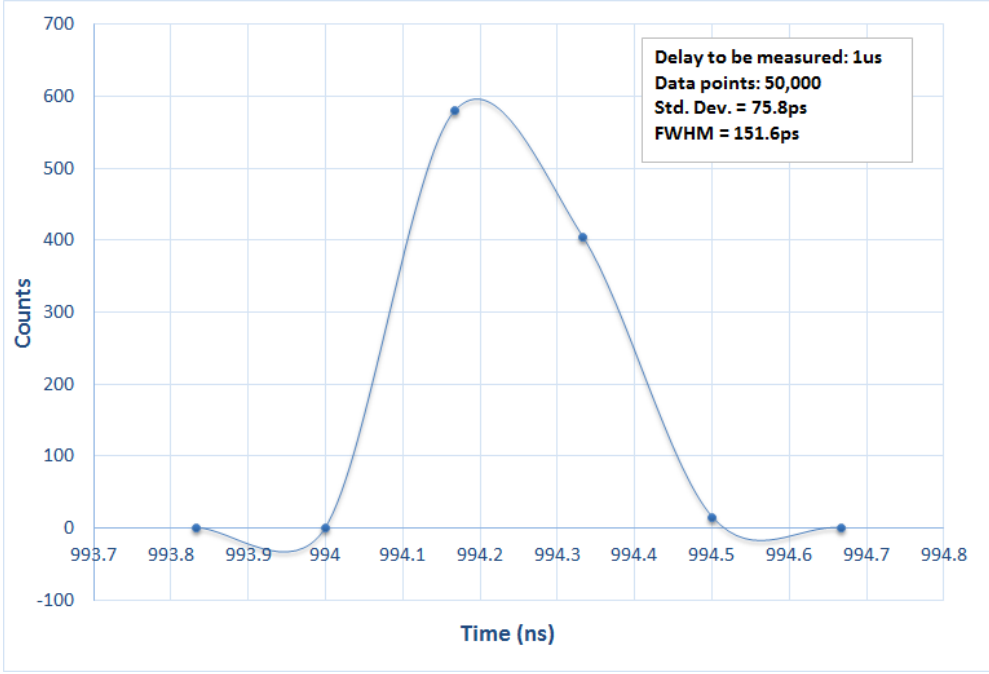


Fig. 4.3 Timing Jitter

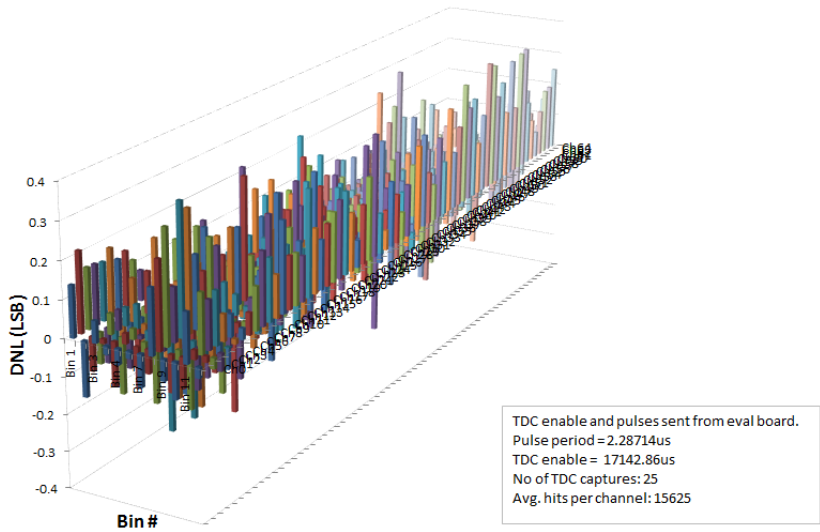
Table 4.2 shows the timing precision performance when two adjacent pulses with a time interval of 100ns and 1us are applied. Those two measurements have similar standard deviation values, and delay errors, which requires long-term stability oscillator for next revision.

Delay Inputs (ns)	Data points	Delay Measured (ns)	Standard Deviation (ps)
100	50,000	94.39941	83.37
1000	50,000	994.14046	75.81
1000000	50,000	1000031.28	85.4

Table 4.2 Timing precisions

4.2. DNL

Fig. 4.4 shows the DNL performance of the system of all 64 channels. For an ideal TDC, in which the DNL = 0 LBS, each bin equals 1 LSB (166.7ps in this case) and the transition values are spaced exactly 1 LSB apart. A DNL error specification of less than or equal to 1 LSB guarantees no missing codes, which ensure single-shot accuracy.



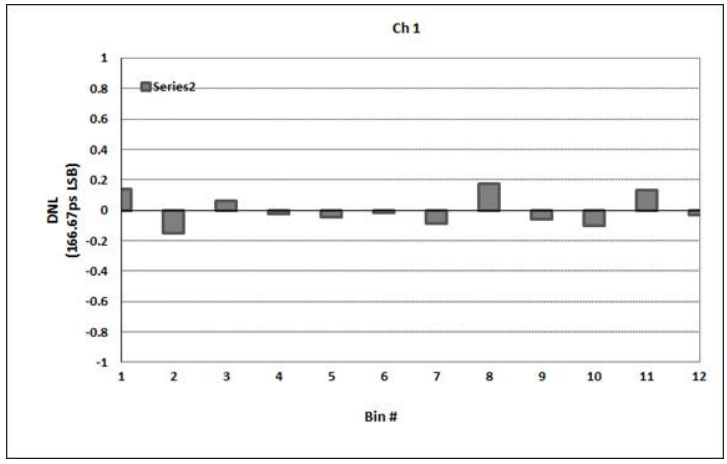


Fig. 4.4 DNL of the TDC system

Factors causing channel-to-channel DNL come from internal routing paths of the clock. Even though, a routing from an input of each channel to each flip-flop is controlled (Fig. 3.3 and Fig. 4.5), the clock paths to each channel are not identical (Fig. 4.6), so this affects channel-to-channel

deviation.

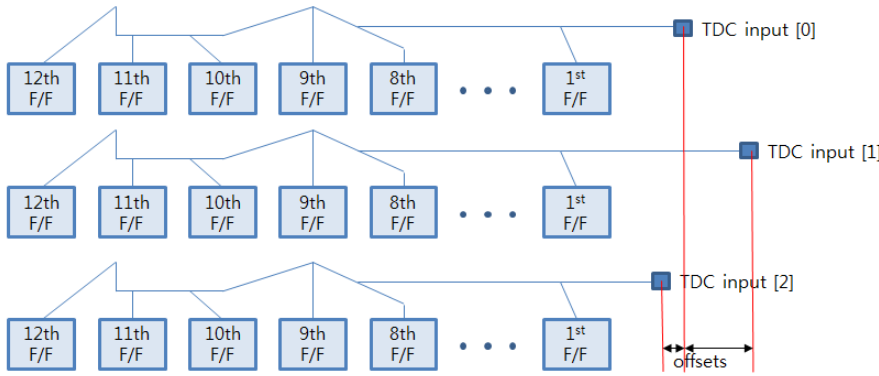


Fig. 4.5 Relationships of TDC input channels

From Fig. 4.6, clocks are distributed from a clock tree and share a route until they reach the terminal where they go to each flip-flop. The shared routes have constant delays so can be compensated later, but the branch to each flip-flop is hard to calibrate. The implementation tool of the FPGA vendor gives an estimate of maximum deviations among those routes, and these values stay within certain ranges, which are 20 ps in this design. The fact that the DNL among the channels are within 0.5 LBS means all other factors, except the clock skew that is controllable to the limited extent, are kept identical or minimized to ensure channel-channel independent operation.

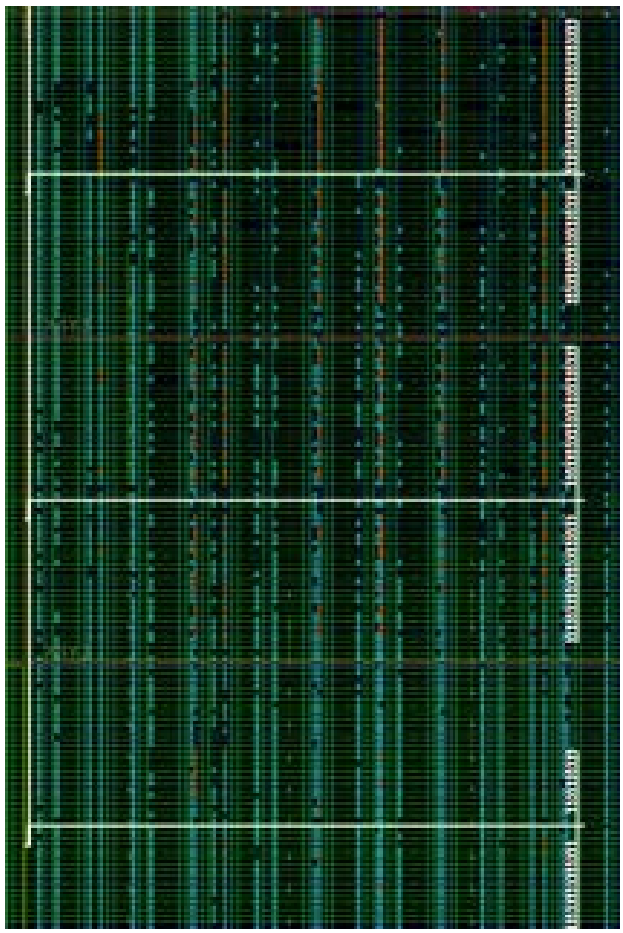


Fig. 4.6 Routing of clocks to each flip-flop

4.3. Time base stability (Allan deviation)

Allan deviation is used to measure frequency stability of the system. From the sampling time 0.1 second, the Allan deviation value increases again, shows the clock of the system starts drifting gradually. This explains the delay errors in Table 4.2. For example, in the case of measuring adjacent pulses with a time interval of 1,000,000 ns 50,000 times, the observation or sampling time spans up to 50 seconds. As the data point accumulates, the deviations of the delay due to clock frequency instability are added up.

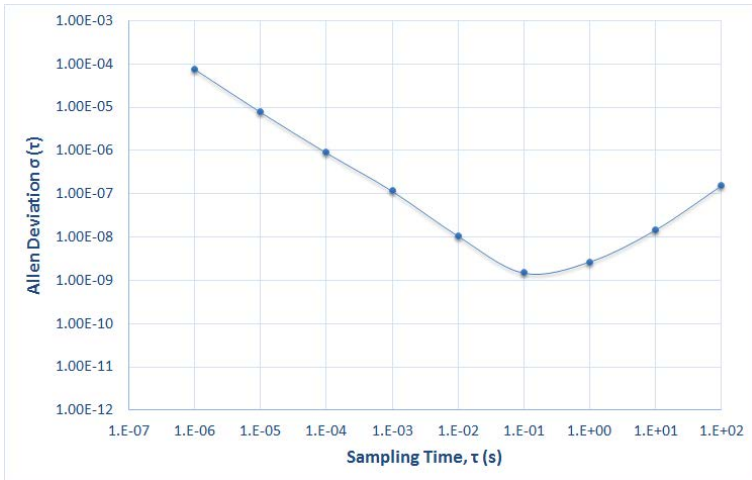


Fig. 4.7 Allan deviation of the TDC system

4.4. Throughput

Maximum system throughput is measured in all 64-channel (Table 4.2). Until 512 MSPS, there is no data loss in various test conditions; pulse period and acquisition time. But, from 520 MSPS, data loss starts happening when capturing for a long acquisition time. Above 550 MSPS, it loses accounts on all acquisitions. The throughput performance measured surpasses the design requirements.

System Throughput (MSPS)	Data loss
376	No
426	No
512	No
520	Yes

Table 4.2 System throughput and data loss

Dead Time requirement is evaluated. Based on the testing, the minimum Dead Time is 20ns (10 clock cycles of the system clock). Maximum dynamic range is 68 seconds, which is above the design requirements.

5. Discussion

The project presented a Time-to-digital converter system that records time of arrival in high resolution from multiple channels simultaneously with single-shot accuracy. In order to keep the system cost low and have flexible reconfiguration depending on various applications, FPGA device is used, and phase shifted clock scheme is applied for implementing many channels. To achieve single-shot accuracy, a design with the adjustable clock phase is proposed.

A TDC system, with the Digital Processing Unit to provide versatile functions such as in-system processing, is build and characterized. From the characterization result, the system shows high timing accuracy of 166ps timing resolution, DNL of less than 0.5 LSB in all channels, 151.6 pico seconds of FWHM, and 4GB/s of the system throughput.

The TDC system proposed in this thesis can be useful in applications that need high channel counts, high timing resolution performance with single-shot accuracy, and versatile in-system processing such as High Energy Physics experiments, 3D time-of-flight camera system, pulse position modulation-based optical communication system and nuclear medical diagnosis system.

Bibliography

[1] Chen Yao, “Time to Digital Converter used in ALL digital PLL”

[2] Harmen Menninga, “Implementation, Characterization, and Optimization of an FPGA-based Time-to-Digital Converter”

[3] G. S. Jovanovic, M. K. Stojcev, “Verner’s Delay Line Time-to-Digital Converter”, *APPL. MATH. INFORM. AND MECH*., vol. 1, pp. 11-20 (2009)

[4] S. Henzler, “Time-to-Digital Converters”, *Springer Series in Advanced Microelectronics 29*

[5] M. Buchele, H. Fischer, F. Herrmann, K. Konigsmann, C. Schill, S. Schopferer, “The GANDALF 128-Channel Time-to-Digital Converter”, *Technology and Instrumentation for Particle Physics*, pp. 1-8, 2011

[6] C. Veerappan, “Data acquisition system design for a 160x128 single-photon image sensor with on-pixel 55ps time-to-digital converter”, December 2010

[7] Texas Instruments, “Understanding Data Converters”, Application Report, 1995

[8] J. Wu, “Several key issues on implementing delay line based TDCs using FPGAs”, *IEEE Transactions on Nuclear Science*, vol. 57, pp. 1543-1548, June 2010.

[9] A. Aloisio, P. Branchini, R. Cicalese, R. Giordano, V. Izzo, S. Loffredo, R. Lomoro, ”High-Resolution Time-to-Digital Converter in Field Programmable Gate Array

[10] Jinyuan Wu and Zonghan Shi, “The 10-ps Wave Union TDC: Improving FPGA TDC Resolution beyond Its Cell Delay”

[11] Kalisz J., et al., “Single chip interpolating time counter with 200ps resolution and 43-s range”, *IEEE Transaction on Instrumentation and Measurements*, vol. 46, No. 4, pp. 851-856, 1997

[12] J. Wang, S. Liu, Q. Shen, H. Li, Q. An, “Fully fledged TDC implemented in field-programmable gates arrays”, *IEEE Transaction on Nuclear Science*, vol. 57, pp. 446-450, April 2010

[13] C. Favi and E. Charbon, “A 17ps resolution, temperature compensated time-to-digital converter in FPGA technology”, *FPGA09*, vol. 1, pp. 1-8, February 2009

[14] S. Sakunia, “Ping-pong-pang instrumentation amplifier”, *Faculty of Electrical*

Engineering, Mathematics and Computer Science Electronic and Instrumentation, pp. 1-71, September 2010

[15] J. Wu, Z. Shi, and I.Y. Wang, “Firmware-only implementation of time-to-digital converter (TDC) in Field-Programmable Gate Array (FPGA)”, *IEEE Embedded Systems Letter*, no. 177-181, pp. 56-59, 2004

[16] Qi Shen, Shubin Liu, Binxiang Qi, Qi An, Shengkai Liao, Chengzhi Peng, Weiyue Liu, “A Multi-chain Measurements Averaging TDC Implemented in a 40nm FPGA”

[17] Foudil Dadouche, Timothe Turko, Wilfried Uhring, Imane Malass, Jeremy Bartringer, Jean-Pierre Le Normand, “Design Methodology of TDC on Low Cost FPGA Targets”, International Conference on Sensor Technologies and Applications 9th, pp. 29-34, 2015

[18] Juha Kostamovaara, Sami Kurtti, Jussi-Pekka Jansson, “A receiver – TDC chip set for accurate pulsed time-of-flight laser ranging”, *EMEA*, 2012

[19] Jozef Kalisz, “Review of methods for time interval measurements with picosecond resolution”, *Metrologia*, vol. 41, pp. 17-32, 2004

[20] Xi Qin, et al., “Development of a High Resolution TDC for Implementation in Flash-Based and Anti-Fuse FPGAs for Aerospace Application”, IEEE Transactions on Nuclear Science, vol. 60, no. 5, pp. 3550-3556, October 2013

초록

입자도달시간 검출을 위한 다채널 디지털 펄스 처리기 개발

김 상 협
전기컴퓨터공학부
서울대학교 대학원

핵의학, 의공학 진단 분야, time-of-flight 및 정밀 거리 측정과 같은 고해상도 시간 간격을 측정하는 응용 분야가 점점 많이 확산되고 있다. 특별히 다채널 동시 검출이 가능하며 single-shot 측정이 가능한 정확도를 가진 디지털 펄스 처리기의 요구가 늘어나고 있다. 본 논문에서는, 이와 같은 필요성을 바탕으로, 적은 비용으로 개발이 가능하고, 다양한 응용 분야에 맞춰 수정이 가능하도록 Field Programmable Gate Array (FPGA)를 이용하여 도달 시간 검출을 위한 디지털 펄스 처리기인 Time-to-digital Converter를 개발하였다. 다채널 구성에 유리한 클럭 위상 변화 구조를 바탕으로 single-shot 측정이 가능한 정확도가 나오도록 위상 조정이 가능한 설계를 제안하였다. 제안된 설계를 바탕으로 디지털 펄스 처리기를 개발하여 시간 측정 정확도 및 DNL, Time base stability 및 데이터 전송 속도에 대한 성능 분석을 시행하였다.

주요어: Time-to-digital Converter, TDC, time of arrival, time-of-flight, phase shifted clock

학번: 2012-20749