

FPGA-Based High Area Efficient Time-To-Digital IP Design

Min-Chuan Lin, Guo-Ruey Tsai, Chun-Yi Liu, Shi-Shien Chu
Department of Electronics Engineering, Kun-Shan University, Taiwan
mclin123@mail.ksu.edu.tw

Abstract—This paper proposes a novel design for a highly area efficient FPGA-based TDC (Time to Digital Converter) IP (Intelligent Property) with resolution less than 30ps. To avoid the unpredictable internal place and route (P&R) delay, a modified ring oscillator is presented. By integrating the gates delay and P&R delay, a design by combining Schematic and VHDL codes, can generate a predictable and stable TDC module built in a Xilinx FPGA.

I. INTRODUCTION

Precise measurements of the time intervals between two or more physical events are frequently needed for many applications in science and industry. We have adopted Software-Definition-Instrumentation concept to implement re-configurable and multi-functional instruments, and proposed a precise phase measurement with broad frequency range [1]. For example, to get a phase resolution with one degree, the system clock for the FPGA should be 360 times of the frequency of the test signal. For a system clock with frequency of 100M Hz, the frequency of the test signal will be limited below 278 KHz. We need a TDC technique to overcome the high system clock frequency requirement and upgrade the resolution of the phase measurement.

Many TDC techniques have been proposed like Tapped Delay Lines (TDL) [2,3], Delay Locked Loop (DLL) [4], Vernier Delay Line (VDL) [5], Multi-level TDC [6], and Triggered Ring Oscillator [7]. Most implementation tools of these TDC techniques are carried out either by full custom ASIC design or standard cell ASIC design. These two approaches all have the advantages of high speed and preciseness, but still face against the problems of both high development cost and slow time-to-market. Moreover, those design modules can not be integrated into a FPGA which is usually used as the hardware platform for a re-configurable SoRC (System on Re-configurable Chip). In other words, all those designs can not be directly called by the HDL (Hardware Description Language) program. To develop measurement system with characteristics including both re-configuration and multi-functions, in general, we prefer a FPGA-based SoRC which is more suitable for the application system design with the requirements of fast time-to-market and low development cost.

Kalisz first used a FPGA to implement TDC module in 1997 [2]. The adopted Tapped-Delay-Lines technique is designed into a QuickLogic pASIC, and a 10 ns time interval measurement with 200 ps resolution is announced. The Xilinx Spartan-3 and VirtexIIPro series FPGA are all equipped with DCM (Digital Clock Manager) modules

which can precisely control the system clock. We have proposed and implemented a phase shifting TDC module with 70 ps resolution by DCM modules [8]. Using Tapped delay lines or Vernier Delay Line concept, most FPGA-based TDC design usually suffer from large area consumption and unpredictable P&R delay. The time delay from gates themselves together with the unpredictable internal P&R delay, are the main limitations to a stable ps-resolution TDC IP design. This is the inevitable FPGA hardware restriction.

Section two will introduce the design of a novel high resolution TDC module using two oscillators with frequency difference near tens of pico-second. Section three carries out the circuit design and its simulation analysis when the oscillators are implemented in a Xilinx FPGA. Section 4 illustrates the implementation results. Finally, we have the conclusions.

II. Design of a novel high resolution TDC module

Fig. 1 shows the common digital method of time interval measurement. The system clock can count out the coarse value T_{12} which is represented by nT_{clk} , where n is the number of counted clocks and T_{clk} is the system clock period. When using the counter method, we assume that the measured interval T_m is asynchronous with regard to the system clock. It means that neither the start edge nor the end edge of the measured time interval is correlated in time with the system clock pulses. In other words, there is a uniform probability distribution of the time interval between the active edge of the system clock pulse and the start edge of the measured time interval. The situation is same with the end edge of the measured time interval. Depending on the true value of the interval T_m and its time location with regard to the system clock, the maximum quantization error of a single measurement may reach about $\pm T_{clk}$, i.e., the coarse measurement resolution (LSB: least significant bit) is limited to T_{clk} . So, it needs other fine measurement methods for T_1 and T_2 such that we have

$$T_m = T_1 + T_{12} - T_2 = nT_{clk} + T_1 - T_2 \quad (1)$$

Where, T_1 is the time interval between the active edge of the system clock and the rising edge of the measured time interval, and T_2 is the time interval between the first inactive edge of the system clock and the falling edge of the measured time interval. Both T_1 and T_2 are shorter than T_{clk} .

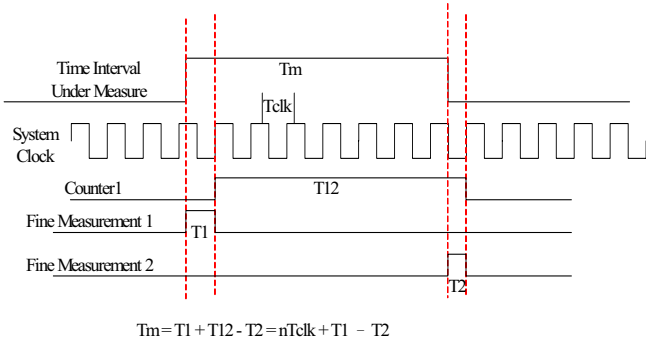


Fig. 1 illustrates the measurement of the time interval T_m .

To measure T_1 and T_2 , we generate two controllable oscillators with slightly different frequencies f_1 and f_2 , and then we can have an incremental time resolution

$$T_{rs} = T_{\text{slow clock}} - T_{\text{fast clock}}. \quad (2)$$

where, $T_{\text{slow clock}} = 1/f_1$ and $T_{\text{fast clock}} = 1/f_2$. To measure the T_1 , we first need to generate the slow clock with rising edge coinciding with the start edge of T_1 . Second, the fast clock is generated with rising edge synchronous with the end edge of T_1 . Fig. 2 illustrates the timing.

The circuit in Fig.3 is used to decide the coincidence between slow and fast clocks. The respective counters-counter 1 and counter 2, store the numbers n_1 and n_2 . When the quantization error is ignored, the measurement result is

$$T_1 = (n_1 - 1)T_{\text{slow clock}} - (n_2 - 1)T_{\text{fast clock}} \quad (3)$$

When $T < T_{\text{slow clock}}$, then $n_1 = n_2$, and

$$T_1 = (n_2 - 1)T_{rs} \quad (4)$$

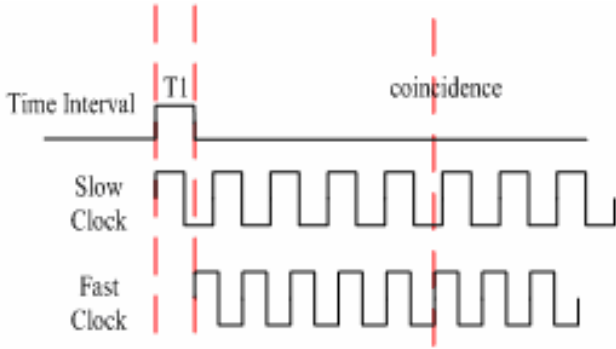


Fig. 2 shows the timing for the slow clock, the fast clock and T_1 .

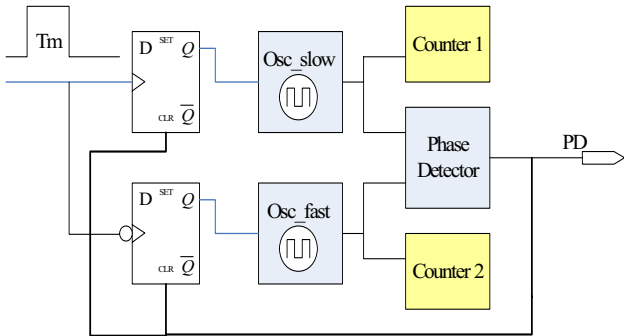


Fig. 3 shows phase detector for pulse coincidence detection between slow clock and fast clock.

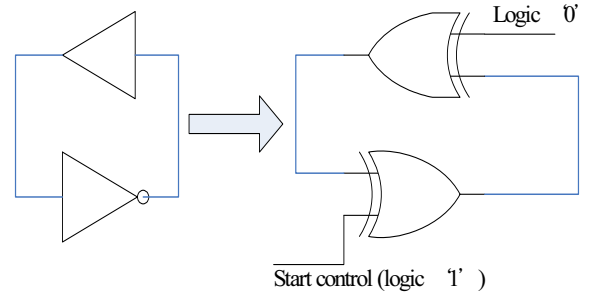


Fig. 4 illustrates the modification from basic ring oscillator to modified ring oscillator which is designed with XOR gates

III. Circuit design and function simulation of the two oscillators

Fig. 4 shows the modified ring oscillator which uses the XOR gates to replace the inverter and buffer in the basic ring oscillator. Without synthesizing LUT (Look up Table), we can utilize the independent XOR gates in the configurable logic block (CLB). The main task is to implement the specified XOR gates with slight routing delay difference. By virtue of the recent hardware technique, we can generate two controllable oscillators with frequency difference down to several ten's pico-second.

Fig. 5 shows the implementation circuit of the controllable ring oscillator. Both AND and XOR gates (working together as delay#1) are synthesized by LUT. Working as the delay#2, the buffer is implemented by the XORCY component inside the CLB, where the normal output of the XORCY component is used for fast oscillator, and the local output of the XORCY is for slow oscillator.

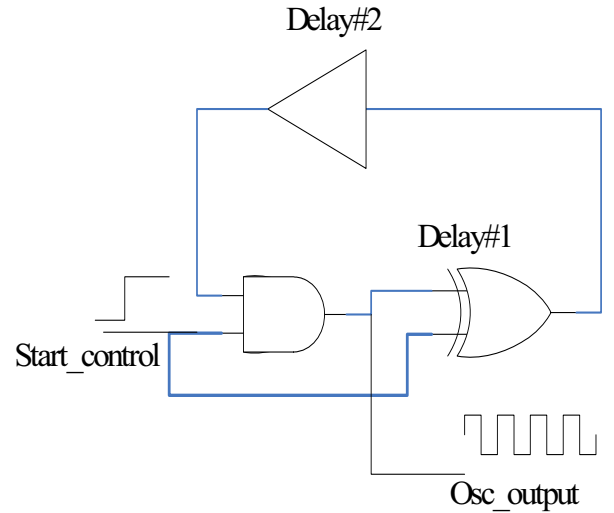


Fig. 5 shows the implementation circuit of the controllable ring oscillator.

The phase detector circuit in Fig. 3 is used to detect the history of the phase difference between the two oscillators, thus providing information about the phase change. Because the rising edge of the slow oscillator signal is always set to lead the fast oscillator clock signal at the start of the measurement process, the measurement process stops when the slow oscillator clock signal begins to lag the fast oscillator clock signal. To accomplish this, the phase detector is implemented by using VHDL coding shown as table 2 below.

Table 2 VHDL codes of phase detector

```

Process (osc_s, osc_f)
begin
    if rising_edge(osc_f) then
        q1 <= osc_s;
        q2 <= q1;
    end if;
    ccout <= not(q1) and q2;
end process;

```

The output of AND gate will switch from ‘0’ to ‘1’ when an input sequence of “10” is detected. We use this signal to control the counter. Fig. 6 shows the behavior simulation waveform for the phase detector output.

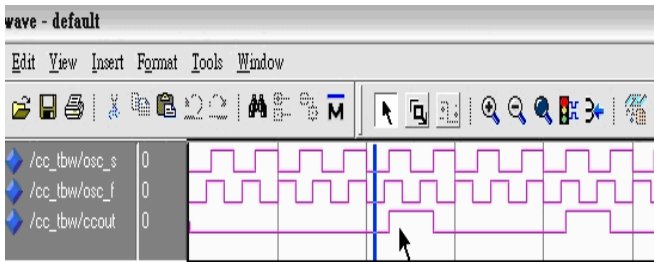


Fig. 6 displays the behavior simulation waveform for the phase detector output.

I. FPGA REALIZATION AND ITS PERFORMANCE

We use two kinds of FPGA types, Spartan3 and Virtex4 for controllable oscillator's realization and performance analysis. Because the XST synthesis tool of ISE8.1 will simplify the entry design schematics, so the oscillator output must feed through an independent buffer device (BUFG). The two controllable oscillators truly occupy two CLBs separately. When using normal output of the XORCY component for fast clock, shown as Fig. 7, we get the period of 3.8764ns. When using local output of the XORCY component, shown as Fig. 8, an output waveform measured by Agilent Infiniti oscilloscope, shown as Fig. 9, tells the period of 3.9068ns. The period difference between the two oscillators will make TDC resolution down to 30.4ps.

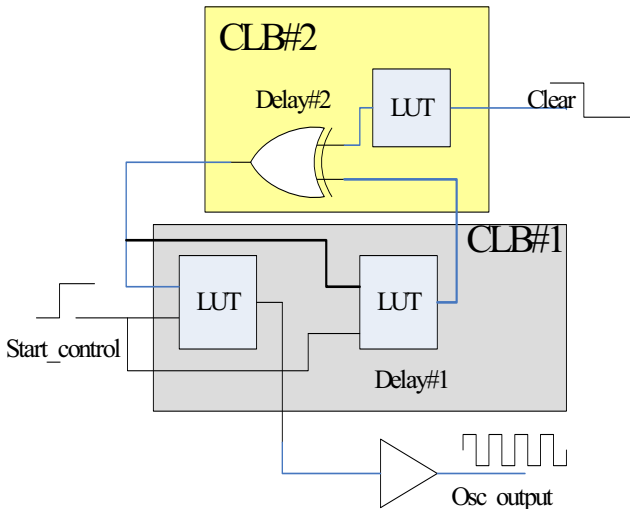


Fig. 7 illustrates the synthesis circuit using normal XORCY output in XC3S200FT256 chip.

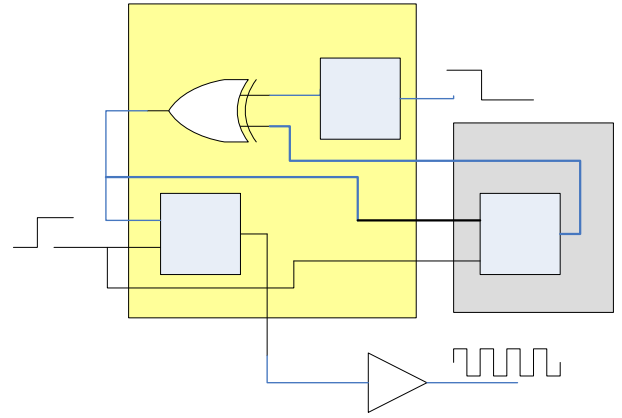


Fig. 8 shows the synthesis circuit using local XORCY output in XC3S200FT256 chip.

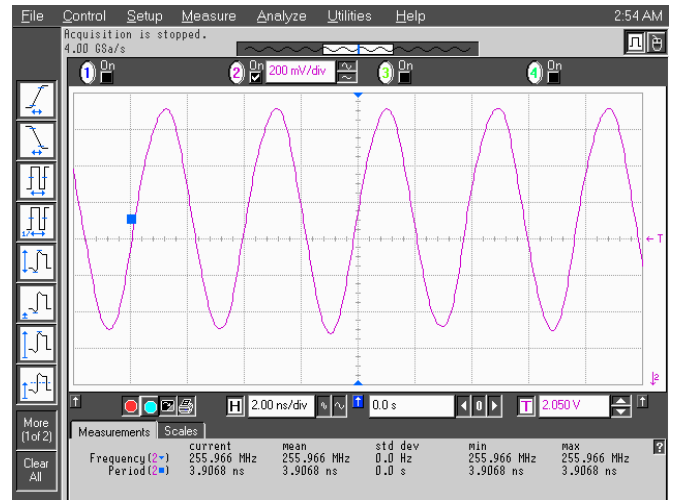


Fig. 9 shows the output waveform of this slow oscillator is measured by Agilent InfiniTi oscilloscope. The period is 3.9068ns.

When using XC4VLX25-FF668-10 of ML401 as testing device, we have 3.7580ns period for fast oscillator and 3.7853ns period for slow oscillator, then we can have the TDC resolution down to 27ps.

IV. CONCLUSIONS

In order to manage pico-second resolution TDC, we use the XORCY components in each CLB, utilize the P&R delay difference, and then get two ring oscillators. Because the modified ring oscillators only consume two or three CLB's, they are highly area efficient. By setting P&R area constraints, we can generate two controllable oscillators with predictable, precise and slightly different frequencies. The period difference of these two ring oscillators can be set down to 30 pico-seconds, and suitable for the TDC module in the software definition instrumentation system design.

ACKNOWLEDGE

This research has been sponsored by National Science Council (NSC94-2215-E-168-004), Taiwan, R.O.C.

REFERENCES

- [1]. Guo-Ruey Tsai, Min-Chuan Lin (2006), "FPGA-based reconfigurable measurement instruments with functionality defined by user", *Eurasip Journal on Applied Signal Processing*, V 2006, 2006, pp 1-14.
- [2]. J. Kalisz, R. Szplet, J. Pasierbinski, A. Poniecki (1997), "Field Programmable Gate Array Based Time-to-Digital Converter with 200-ps Resolution", *IEEE Trans. Instrumentation and Measurement*, Vol. 46, No. 1, pp51-55, Feb. 1997.
- [3]. Jian Song, QiAn, Shubin Liu (2006), "A High-Resolution Time-to Digital Converter Implemented in Field-Programmable-Gate-Arrays", *IEEE Trans. of nuclear Science*, Vol. 53-1, pp.236-241, 2006.
- [4]. Santos, D. M. (1996), "A CMOS Delay Locked Loop and Sub-Nanosecond Time-to-Digital Converter Chip", *IEEE Trans. of nuclear Science*, Vol. 43-3, pp.1717-1719, 1996.
- [5]. P. Dudek, S. Szezepanski, J. Hatfield (2000), "A High Resolution CMOS Time-to-Digital Converter Utilizing a Vernier Delay Loop", *IEEE Trans. Solid State Circuits*, pp240-247, 2000.
- [6]. Chong-Sii Hwang, Poki Chen, Hen-Wai Tsao (2004), "A High-Precision Time-to-Digital Converter Using a Two-Level Conversion Scheme", *IEEE TRANSACTIONS ON NUCLEAR SCIENCE*, Vol. 51, No. 4, P1349-1352, AUGUST 2004.
- [7]. A. H. Chan, G. W. Roberts (2004), "A Jitter Characterization System Using a Component-Invariant Vernier Delay Line", *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, Vol. 12, No. 1, pp79-95, JANUARY, 2004.
- [8]. Min-Chuan Lin and Guo-Ruey Tsai, (2005), NSC93-2215-E-168-004 Project Report, National Science Council, Taiwan, ROC.