

# **A High-Resolution Flash Time-to-Digital Converter Taking Into Account Process Variability**

**Nikolaos Minas**  
**David Kinniment**  
**Keith Heron**  
**Gordon Russell**

# Outline of Presentation

- **Introduction**
- **Background in Time-to-Digital Converters**
- **Theory**
- **FPGA implementation**
- **Calibration**
- **Results**
- **Conclusions**

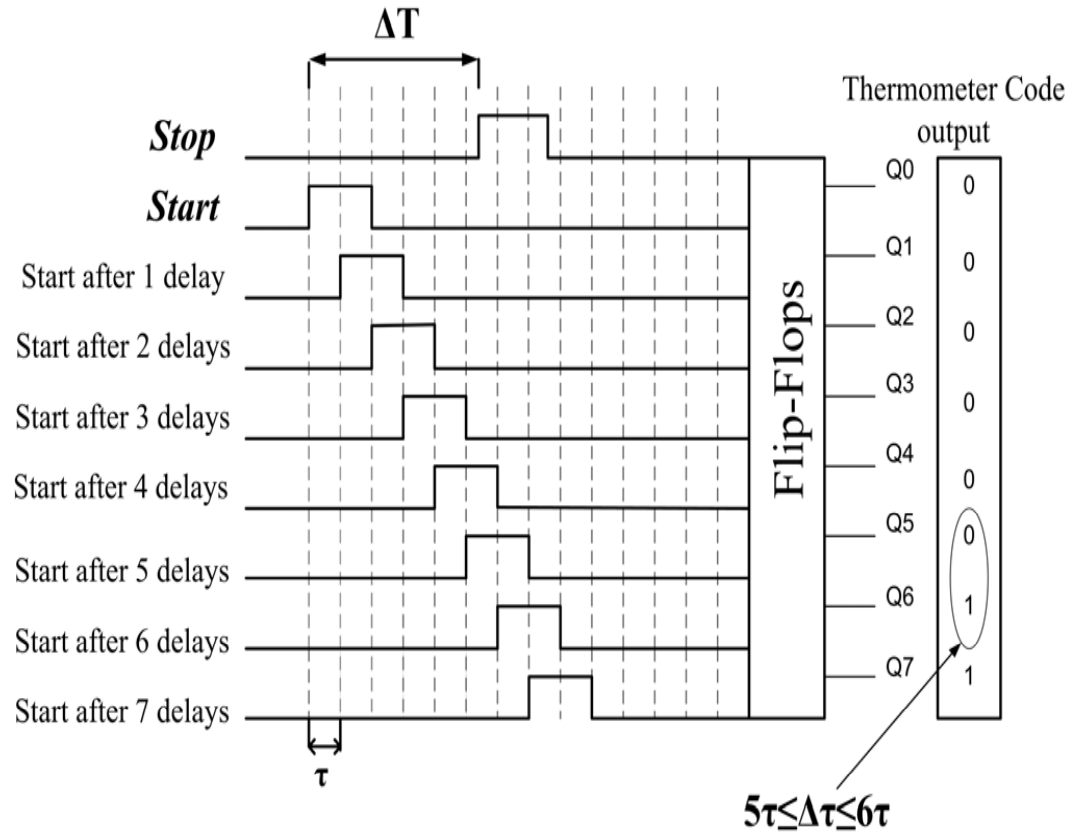
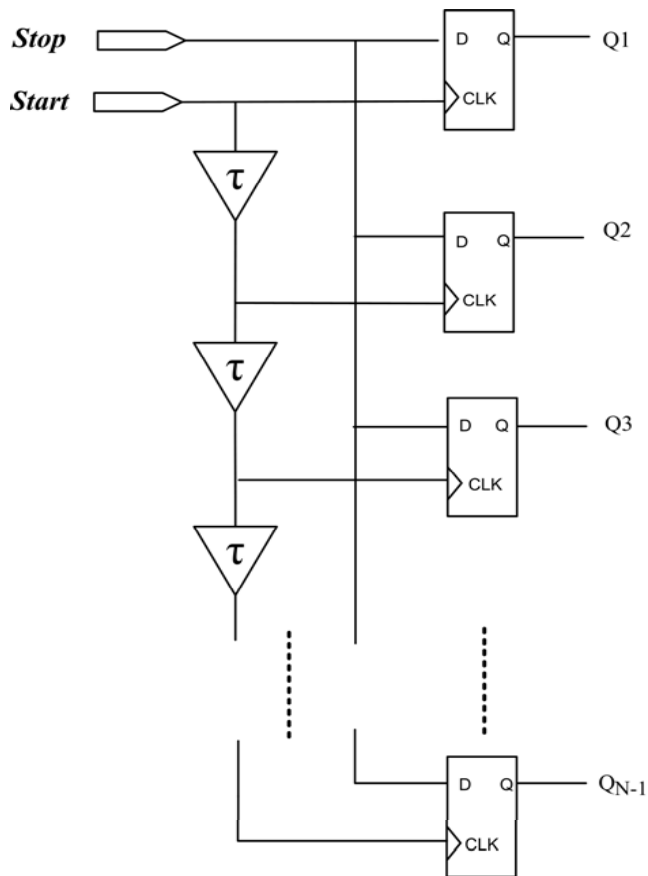
# Introduction

- **Timing issues are a major concern in the design of high performance circuit**
- **System operation is often based on Timing Assumptions. To ensure correct operation these assumptions have to be verified**
- **Investigation into the cause of timing issues cannot accurately undertaken using external test equipment**
- **On-Chip timing measurements offer a more accurate, faster and cost effective alternative**

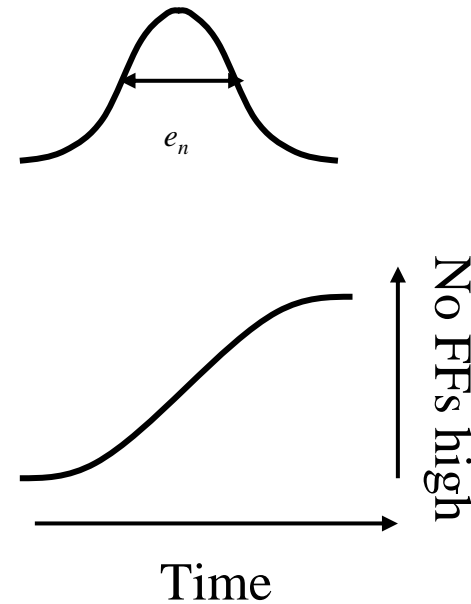
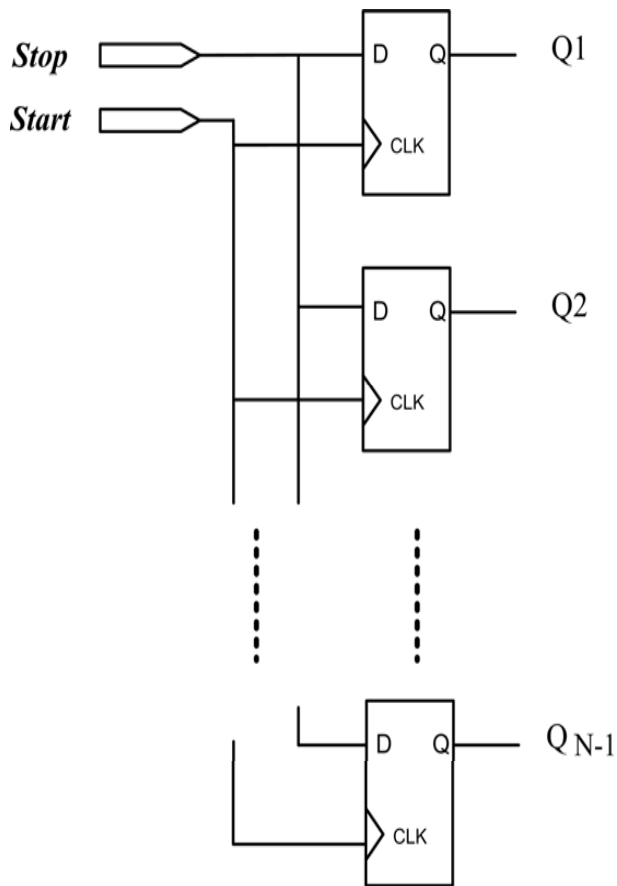
# Time-to-Digital Converters (TDC)

- **Time-to-Digital converters operate by comparing an input signal to various reference edges**
- **Use of Flip-Flop or MUTEXes to compare two edges**
- **Many different configuration of the TDC depending on the resolution required**

# Single Delay Chain TDC

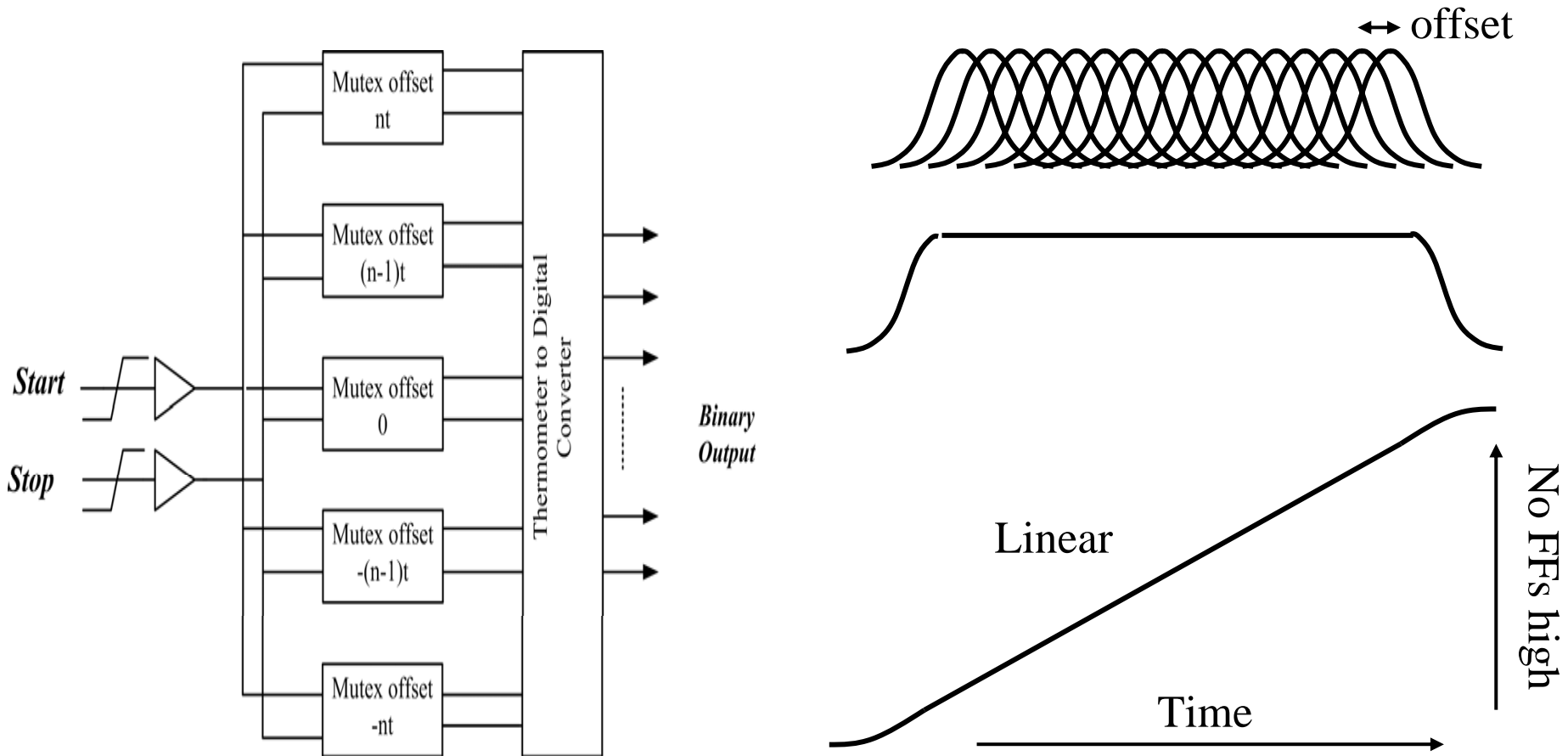


# Process variation based TDC



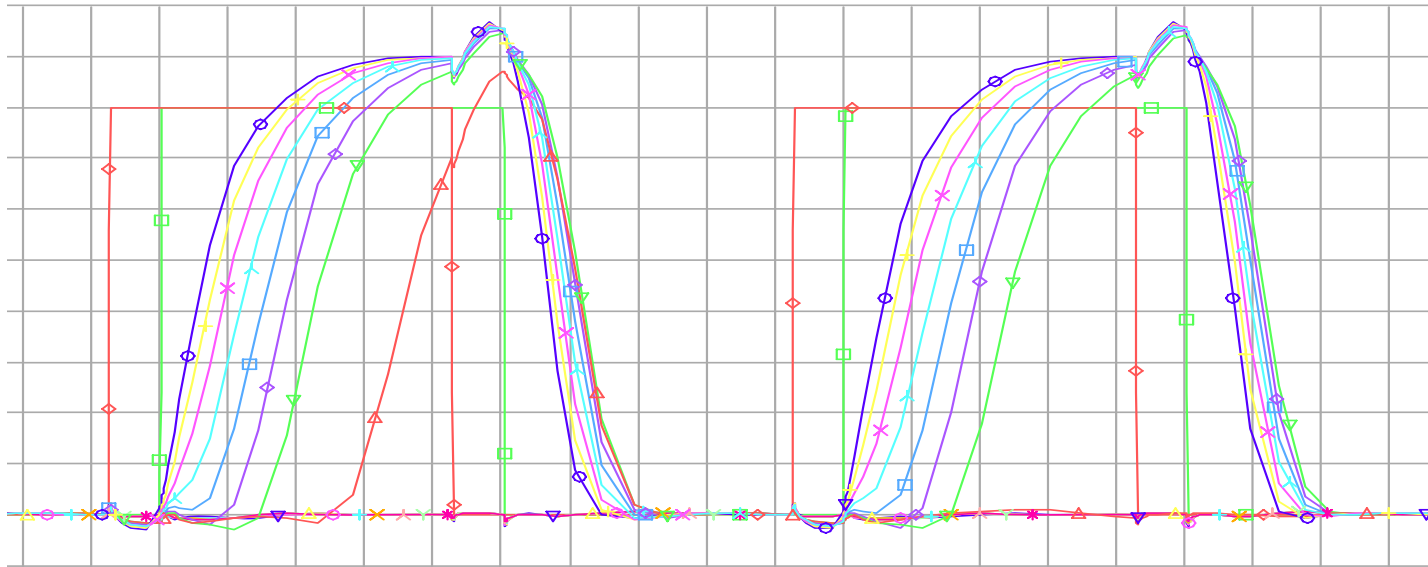
- Exploits the random offsets of Flip-Flops or arbiters to perform time quantization
- Each stage has to be individually calibrated

# Asymmetrical MUTEX-based TDC

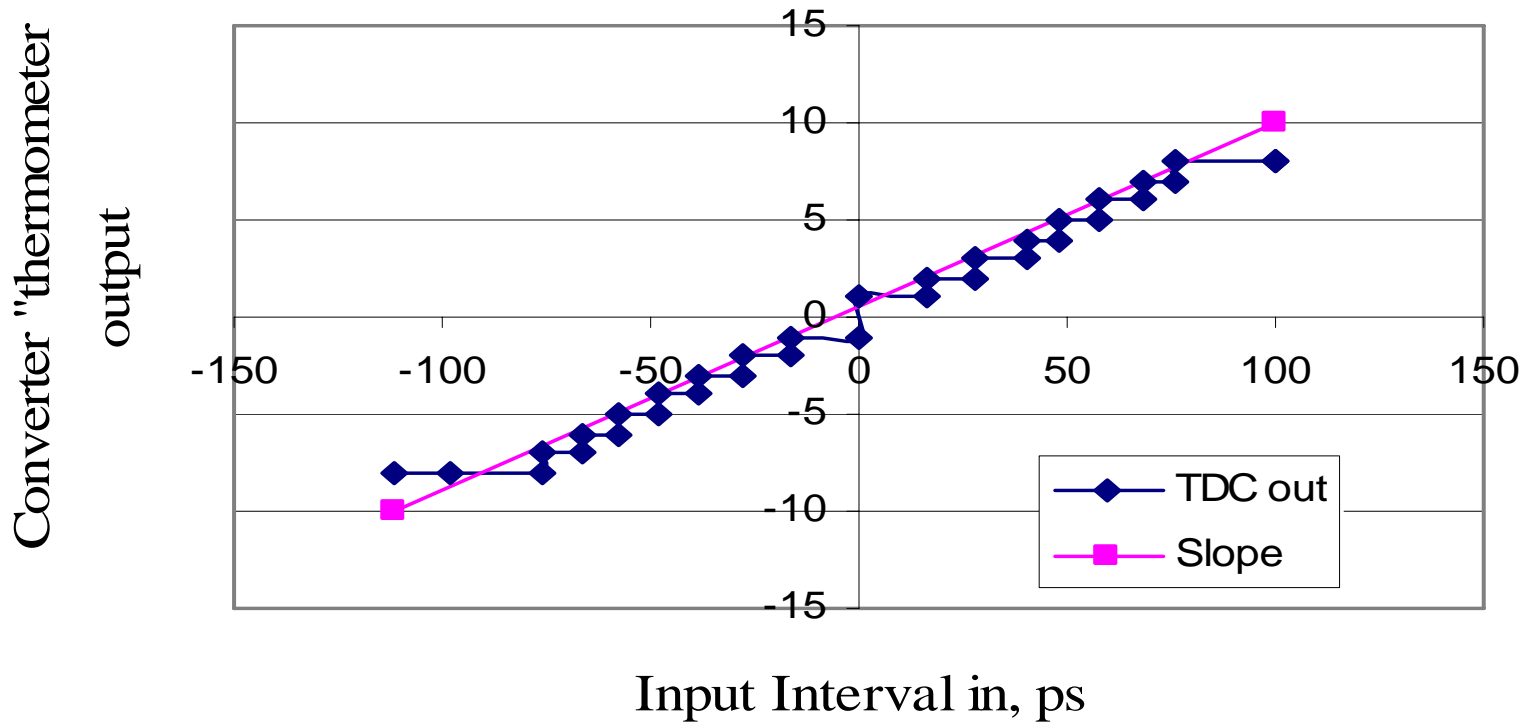


# Simulation results

- Simulation has been done using ORCAD 10 with 0.18 $\mu$ m process models
- Initially, array consists of 16 MUTEXes
- Two groups of 8 with the inputs reversed on the second group

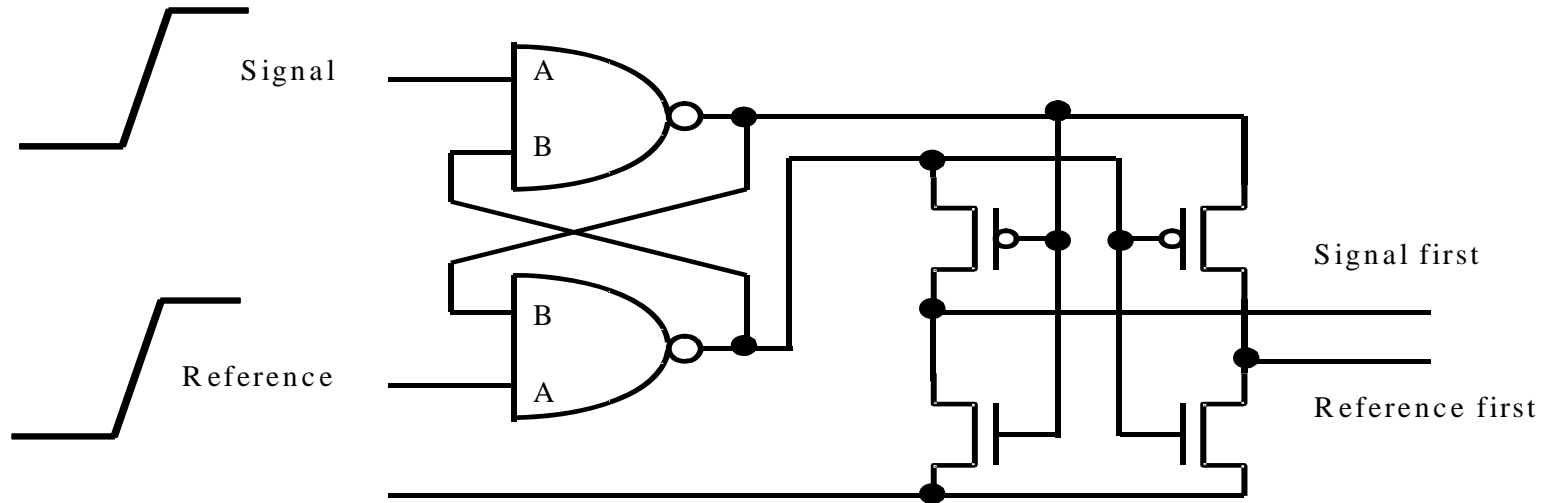


# Calibration curve



- The resolution of the TDC is approximately 10ps

# Effects of process variability



- The width and length parameters of each transistor were varied by a random amount, with standard deviation of 10%
- The distribution of the offset is normal with a standard deviation of 2.028ps. Due to the random variation of the offset the error in time measurement is around 2ps

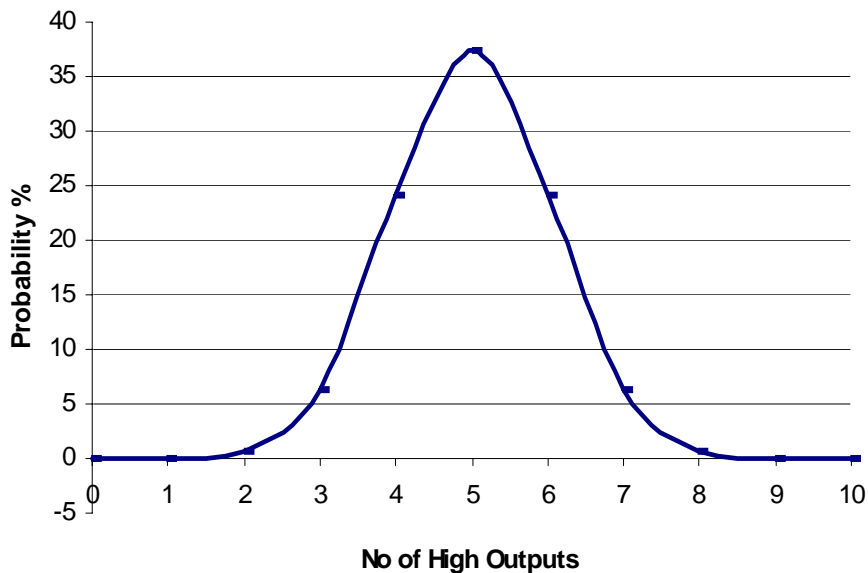
# Probability of a high output

MUTEX	0	1	2	3	4	5	6	7	8	9
Offset, ps	-4.5	-3.5	-2.5	-1.5	-0.5	+0.5	1.5	2.5	3.5	4.5
Probability of a high output, %	98.8	96.0	89.4	77.3	59.9	40.1	22.7	10.6	4.0	1.2

- **The probability of a MUTEX output changing at any particular time can be calculated from the cumulative error function with a deviation of 2ps**
- **Here the 10 MUTEXes are set to change state at 1ps interval**
- **With a distribution of 2ps, a MUTEX with an input of 0ps is 50% likely to set high, and one with -1.5ps is still only 77.3% likely to be high**

# Probability for a given number of high outputs

Number	0	1	2	3	4	5	6	7	8	9	10
Probability of number of highs, %	0.00	0.02	0.65	6.41	24.20	37.43	24.20	6.41	0.65	0.02	0.00



- $2^{10}$  output patterns
- 86% of TDCs will give a count of 4, 5, 6
- The standard deviation of the error is 1.1ps
- Improvements in accuracy of a factor of 2

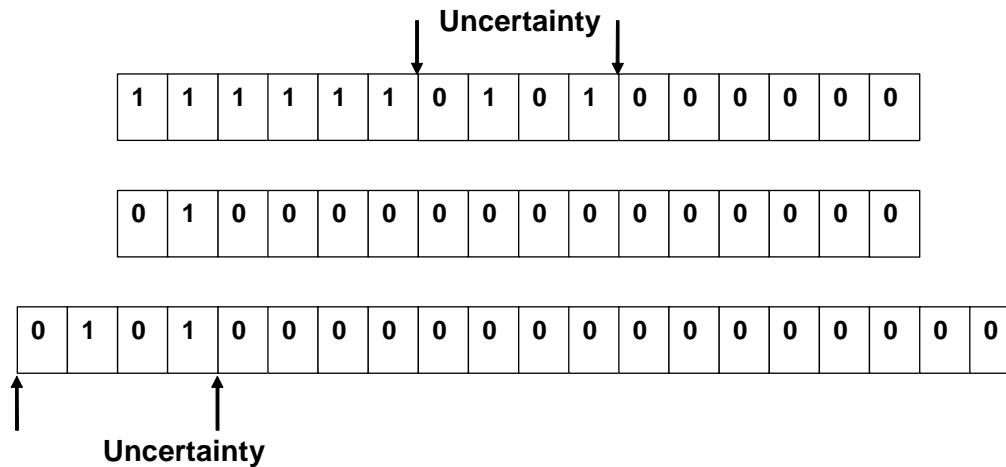
# Calculating the standard deviation of the measurement error

- With a spacing of 1ps there are 4 MUTEXes contributing to the measurement because they are within the standard deviation of the set points  $(2ps + 2ps)/1ps = 4$ , and the effective accuracy is improved by  $\sqrt{4}$ , from 2ps to 1ps
- If there is a random variation in the offset, then the standard deviation due to this variation will be approximately

$$\sigma / \sqrt{\frac{2\sigma}{s}} \quad \text{or} \quad \sqrt{0.5 \times s \times \sigma}$$

- s is the time step between successive MUTEXes

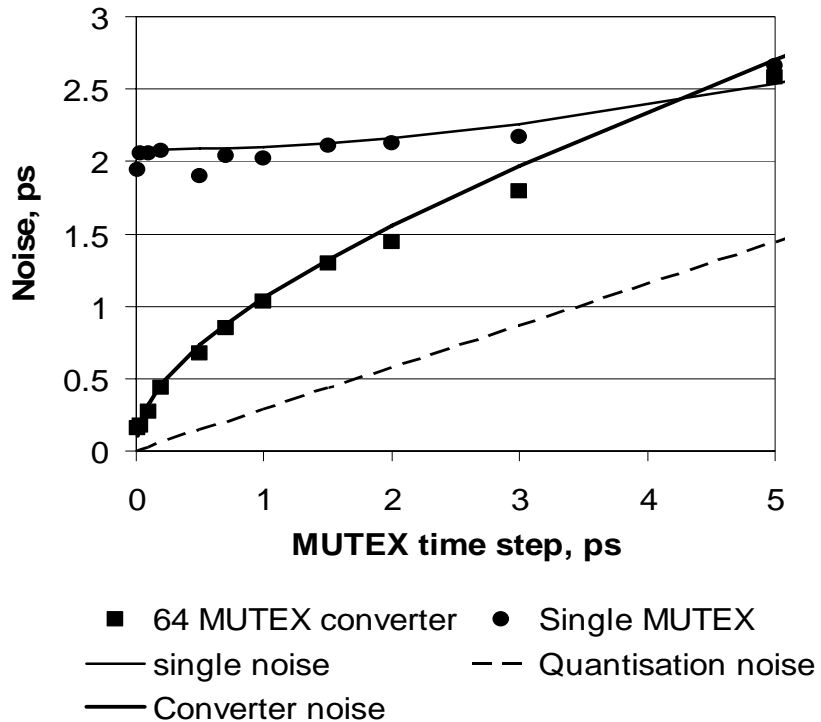
# Effects of uncertainty at the extreme ends of the scale



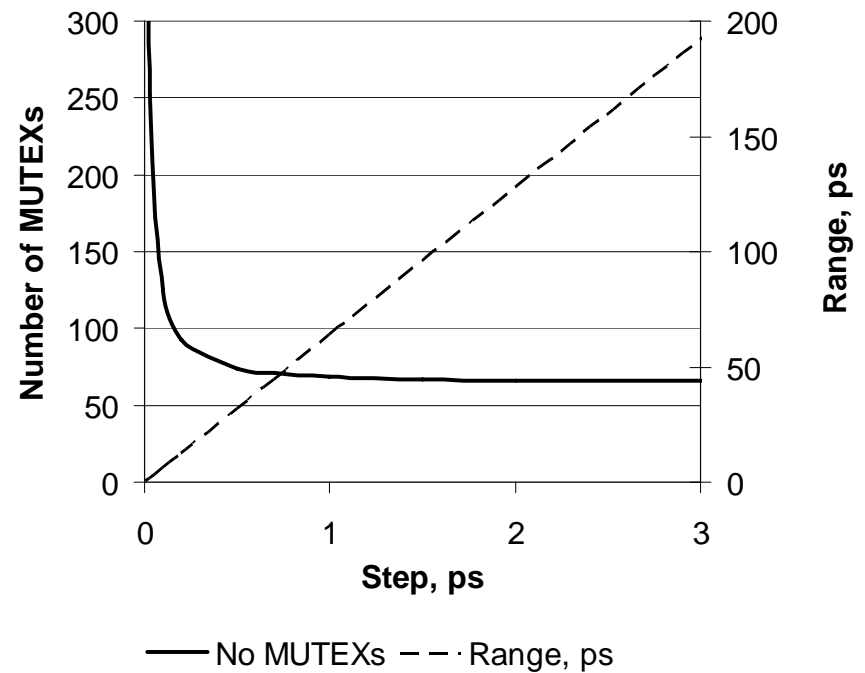
- The problem of uncertainty can be overcome by adding guard stages at either end of the TDC
- That way the number of high outputs varies linearly between 2 to 18 rather than 0 to 16
- The number of extra bits needed is given by  $\sigma/s$

# Noise and cost of a 64 MUTEX TDC

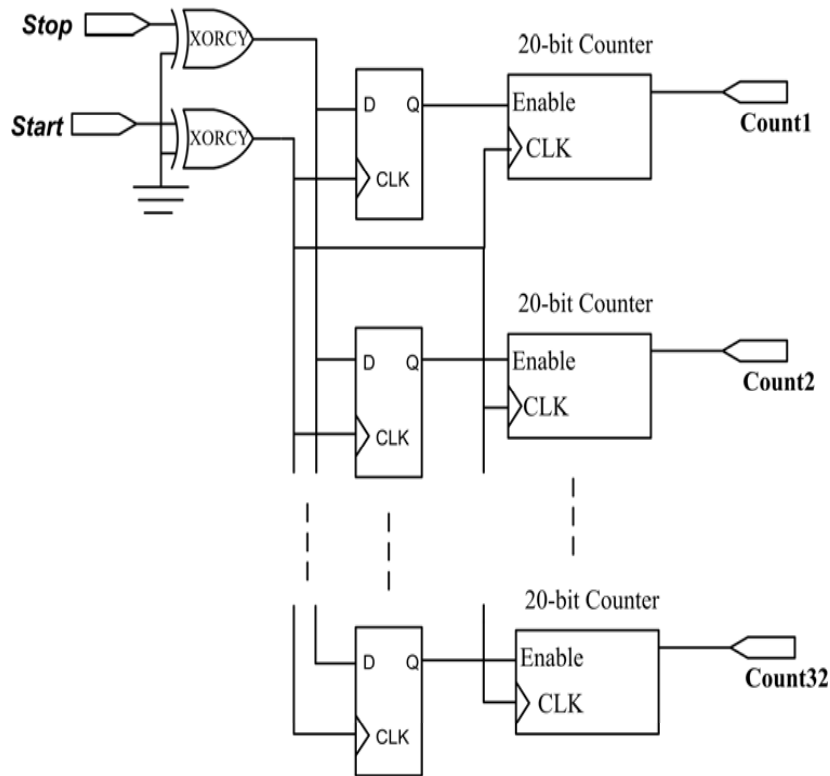
(A)



(B)

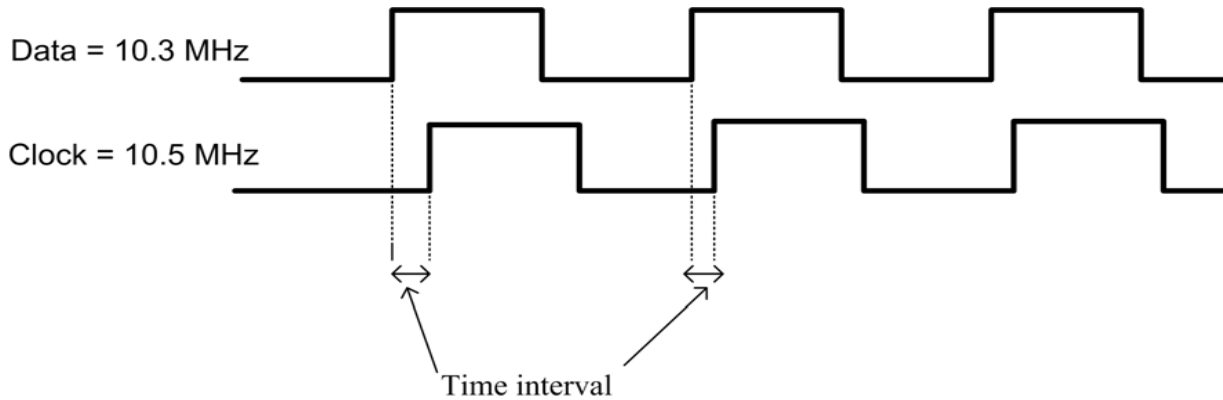
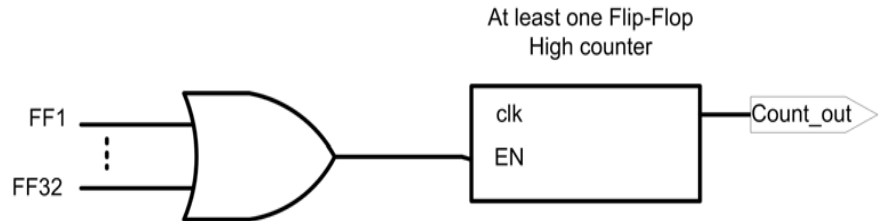
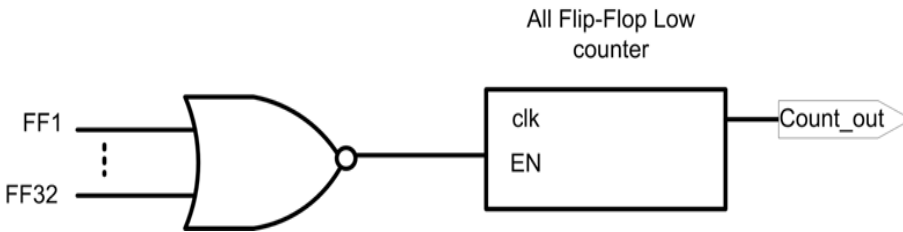
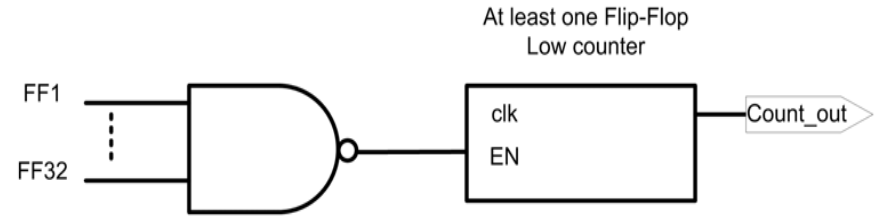
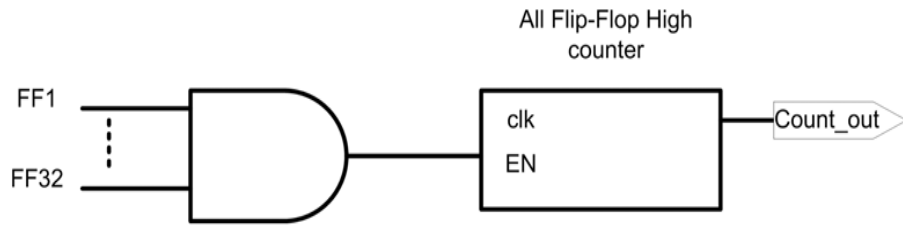


# FPGA-based TDC implementation

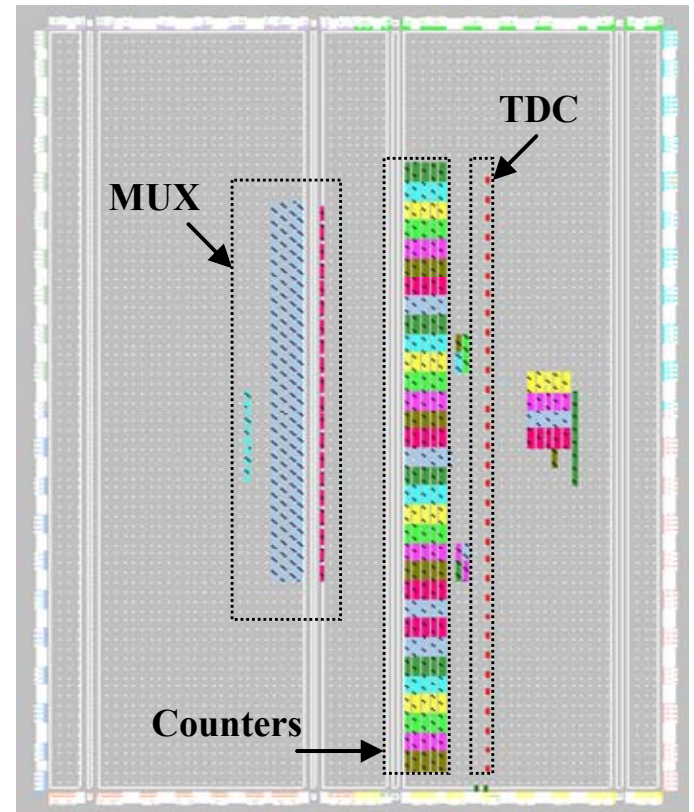
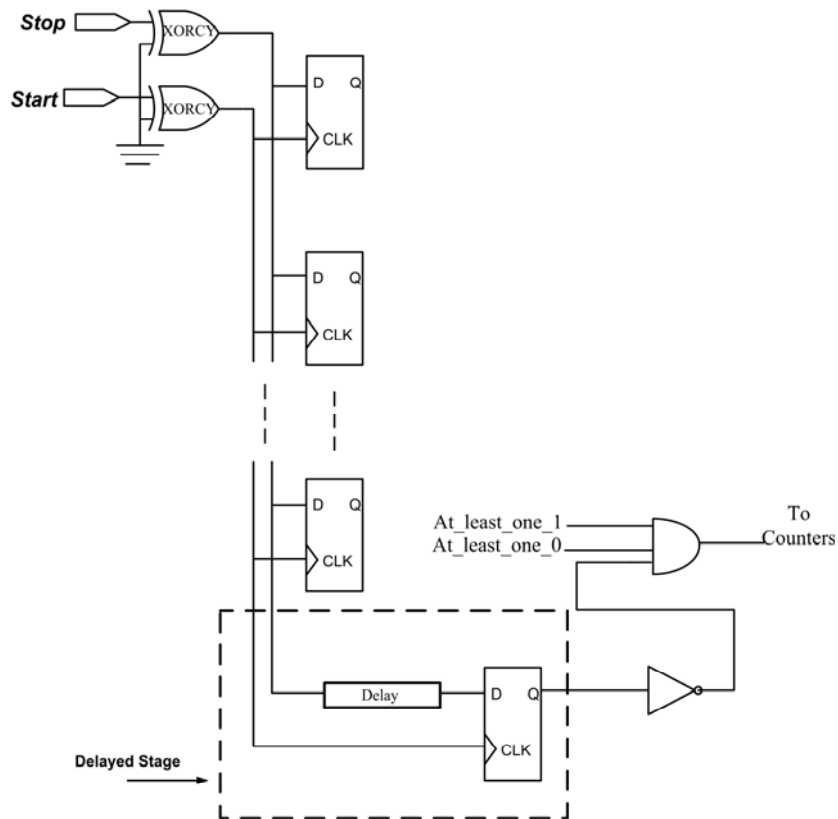


- Using XOR gates it was possible to achieve uniform routing for both data and clock
- Both data and clock paths are laid out progressively
- The difference in delay is between the general purpose and the clock interconnects
- A reference counter is used to count the number of clock cycles that the TDC is operational

# Calibration



# Delaying the data signal



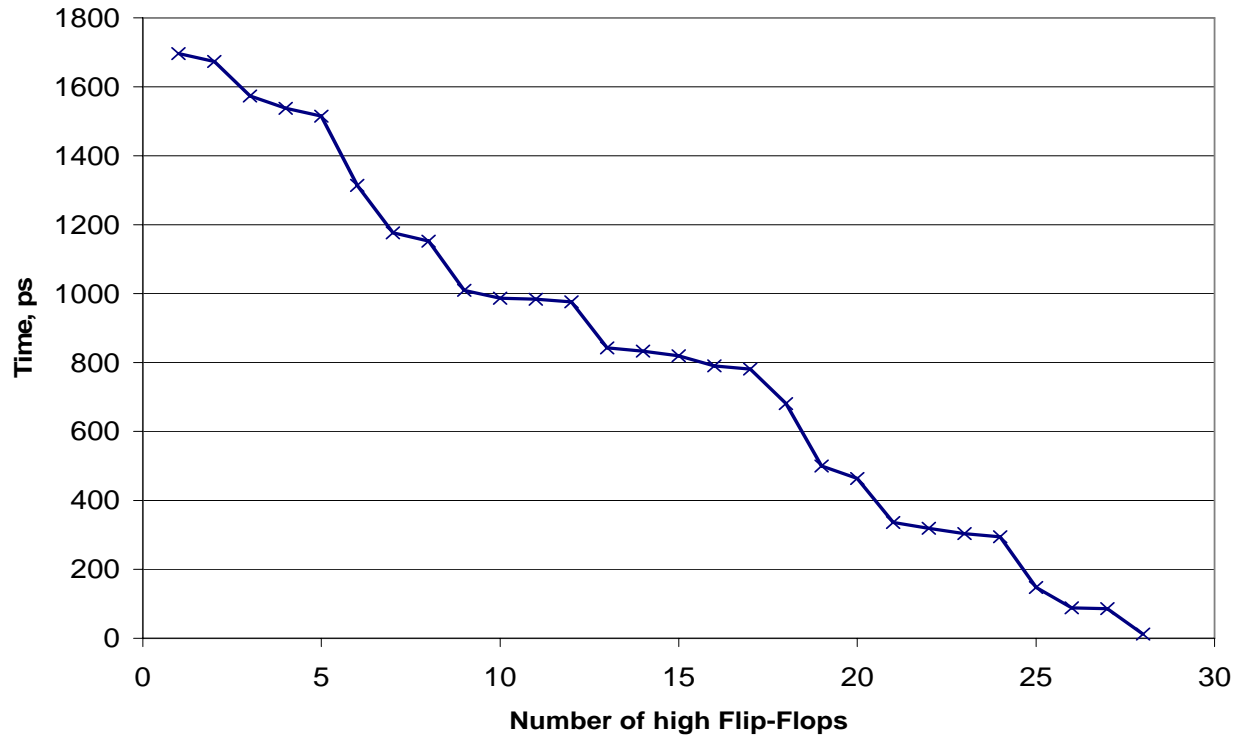
# 1. Results

- **The TDC was run for  $2^{20}$  clock cycles**
- **The recorded events with at least one flip-flop high can be found by:**  
*At\_least\_one\_FF\_1 – All\_FF\_1*
- **The recorded events with at least one flip-flop low can be found by:**  
*At\_least\_one\_FF\_0 – All\_FF\_0*
- **The full range of the TDC can be found by :**

$$Time\_range = \frac{(highest\_value\_Counter) \times 97.08ns}{2^{20}}$$

- **Because of the end effect only 27 stages are used, the range is 1.69ns with a resolution of 62.8ps**
- **To compensate for any errors in the measurements the value of each stage was put in an ascending order of magnitude**

## 2. Results



- The accuracy for after ordering the values has improved to **69.3ps** from **109.1**

# Conclusions

- **The method offers the possibilities of measuring very small time intervals**
- **The simulation results using asymmetric MUTEXes suggest resolutions down to few picoseconds**
- **The accuracy near the limits of the TDC is improved by taking into account the end effect**
- **The proposed design was implemented as a proof of concept in an FPGA with resolution of 62.8ps**
- **The problem of errors in the measurements was also overcome and showed improvements of a factor of 1.57**
- **The calibration technique is straight forward, easy to implement and it does not require any external equipment**