Design of High Speed Front End Circuitry for Neutrino Detectors

A Project Report

submitted by

HARSHIT S. VAISHNAV

in partial fulfilment of the requirements
for the award of the dual degrees of

BACHELOR OF TECHNOLOGY

and

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2011

THESIS CERTIFICATE

This is to certify that the thesis titled **Design of High Speed Front End Circuitry for Neutrino Detectors**, submitted by **Harshit S. Vaishnav** to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology** and **Master of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Nagendra Krishnapura
Project Advisor
Assistant Professor
Dept. of Electrical Engineering
IIT Madras, 600 036

Place: Chennai
Date: 26th May 2011

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Nagendra Krishnapura for his constant guidance and support throughout the project and also during the rest of the academic curriculum. It was his classes on Analog Circuits, with their remarkable insight and intuition which initially pulled me into analog circuit design. His unique way of approaching problems has never ceased to amaze me. Interaction with him over the last three years has helped me learn a lot of new things. I would also like to thank Dr. Y. Shanthi Pavan, whose classes and assignments showed me some of the beautiful math involved in circuit design.

I would also like to thank my lab mates in the TI lab for being patient enough to sort out the problems I had with the simulator.

I would like to take this opportunity to thank my VLSI batch mates Foil, Nimit, Mayank, Pi, Srinidhi and Chouksey for being such a good company during classes and through the final project. Those tea sessions and the unending discussions thereafter will never be forgotten. Thanks are also due to Major, Darbha, Chandy, Venky, Karthik and all my wing mates for the fun times in the hostel. College life would not have been the same without you guys. Thanks are also due to Abhimanyu and Amit for the fun I had with them, both inside the campus and outside.

Finally, and most importantly, I dedicate this thesis to my parents and my sister Dhara for their love, support and encouragement without which this would not have been possible.

ABSTRACT

KEYWORDS: Time-to-digital converter, TDC.

In this project, design of a front end circuitry for neutrino detectors is presented. The system consists of a high speed front end amplifier and latching circuitry. A time to digital converter (TDC) is used to measure the time of arrival of the input with respect to a reference start. A delay locked loop is used to stabilize the delays against PVT variations. A phase locked loop is used to generate a high frequency clock from a low frequency input clock. A digital back-end is designed to process the data digitally and to output a serial data stream.

The design is implemented in 0.13 μm CMOS process. The TDC has a resolution of 125 ps and a range of 131 μs. It occupies 0.24 mm^2 area and consumes negligible static power. The DLL occupies 0.12 mm^2 area and consumes a power of 2 mW. The amplifier has a DC gain of 41.5 dB and a bandwidth of 513 MHz and consumes 0.2 mW power. The design will be sent for fabrication in the UMC 0.13 μm CMOS process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

TDC	Time-to-Digital Converter
INO	Indian-based Neutrino Observatory
ADC	Analog-to-Digital Converter
PVT	Process-Voltage-Temperature
DLL	Delay Locked Loop
PLL	Phase Locked Loop
VCDU	Voltage Controlled Delay Unit
PFD	Phase Frequency Detector
DNL	Differential Non Linearity
INL	Integral Non Linearity
VCO	Voltage-Controlled Oscillator
MOSFET	Metal-Oxide-Semiconductor Field Effect Transistor
NMOS	N-type Metal-Oxide-Semiconductor
PMOS	P-type Metal-Oxide-Semiconductor
CMOS	Complementary Metal-Oxide-Semiconductor
FPGA	Field Programmable Gate Array
UMC	United Microelectronics Corporation

CHAPTER 1

Introduction

Neutrino physics experiments in the last several decades have provided many new and significant results. Many highly multi-disciplinary research groups are working on neutrino detection and related research. The India-based Neutrino Observatory (INO) is one such particle physics research project which primarily aims to study atmospheric neutrinos. The neutrino detector of INO consists of a massive magnetized iron calorimeter (INO, 2008). The primary detection mechanism is via detection of muons produced in charged neutrino interactions. The detector comprises of layers of iron sheets interleaved with planar active detector elements. Each metal sheet contains a mesh of 32 by 32 readout channels. This helps us to determine the (x, y) of the neutrino hit on a given metal sheet. The index of the metal sheet will give us the z coordinate of the neutrino hit. We also need to determine the timing of the neutrino hit with respect to a reference start. Upon determining all these, we can know the (x, y, z, t) profile of the neutrino trajectory. For this, we need to be able to detect the hit on a given plate and process it accurately.

Avalanche mechanisms in the detector array give rise to a voltage spike which needs to be processed using high speed circuits. Fast and high gain amplifier along with latching circuitry is required to generate a digital signal which goes high when a neutrino hit is detected. The time of the neutrino hit also has to be measured with respect to a reference start. The system must be robust and should be low power because high temperatures in the detector render cooling mechanisms less efficient. The system proposed in this thesis is targeted to be of use in these detectors. With this as the motivation for the thesis, we now proceed to describe briefly the design of the proposed system.

1.1 Designed system

The system proposed in this thesis is shown in Fig. 1.1. The front-end amplifier amplifies the input signal. The TDC measures the time interval between the latched version of the amplifier output and a reference start signal. A delay locked loop (DLL) is used to stabilize the delays in the TDC. A phase locked loop (PLL) at the front end increases the frequency of the input clock. The output of the TDC is processed in the digital back-end and output as a serial data stream. A digital offset cancellation circuit is used to remove the effect of the input referred offset of the amplifier. Since the event rate of neutrino hits is low, we can use a low frequency clock for for the digital back-end since the data changes quite infrequently. The clock frequency of the offset cancellation circuitry can also be quite low since the reasons of offset are either process mismatch or temperature, neither of which require a high speed correction technique.

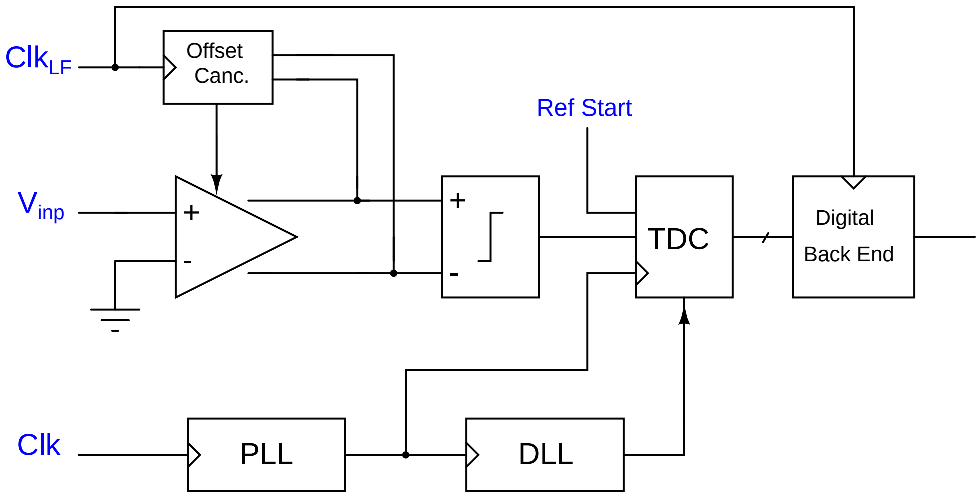


Figure 1.1: The complete system

1.2 Overview of the thesis

The rest of the thesis is organized as follows.

Chapter 2 describes the basics of time to digital converters and the design of our TDC. It also explains the design of the DLL which was used to stabilize the delays.

Chapter 3 explains the design of the analog front end which consists of the amplifier and a PLL.

Chapter 4 compiles all the simulation results and tabulates important characteristics of the TDC and the DLL.

Chapter 5 concludes the thesis and discusses some possible future extensions of this work.

CHAPTER 2

Time to Digital Converters: Fundamentals and Design

2.1 Introduction

A Time to Digital Converter (TDC) is a type of data converter which converts the time interval between two inputs into a digital code. This kind of conversion is useful in time measurement as well as for encoding data on the time axis, a recent area of research called time mode signal processing (TMSP). High resolution TDCs thus have applications in a number of measurement systems, e.g. time-of-flight particle detectors, laser range-finders, logic analyzers and clock jitter and skew measurement. In many applications, the time of arrival of a signal is measured with respect to a reference start and hence measuring arrival time of a signal is also the same as measuring the time interval between two signals. The focus of our design is on using a TDC to record the timing of neutrino hits. A neutrino hit generates a small voltage (few mV amplitude) which can be processed to obtain a digital signal. Our interest lies in measuring the time interval between this signal and a reference start. The parameters to be considered while evaluating a particular architecture are its resolution, range, power consumed, area and susceptibility to PVT variations.

The simplest implementation of a TDC is just a counter which measures the number of clock cycles between the two inputs. While this offers simplicity, its time measurement resolution is limited to one clock period of the counter's clock. Increasing the counter clock frequency to improve the resolution can be infeasible either due to limitation of speed in the existing technology or because of increased power dissipation. There are many alternative techniques to achieve higher timing resolutions. We will discuss three common techniques and advantages and issues with each of the techniques. The design of the TDC with the chosen architecture will be explained in the next section.

2.2 TDC architectures

2.2.1 Flash or single delay line

The delay line architecture (Abas *et al.*, 2007) is in principle similar to the working of a flash ADC. In a flash ADC the input voltage is compared to a set of voltages uniformly distributed between the maximum and the minimum voltage. The output thermometer code encodes the input voltage in a digital form. A delay line based TDC works on a similar concept. The input time period is in essence compared to a set of time periods and a thermometer code is generated. Let the time interval between the two signals be T, the time to be digitized. The start signal i.e. the signal which arrives before the other is passed through a delay chain of n delay elements each having a delay T_d as shown in Fig. 2.1. So, the output of the i^{th} delay element ($0 \leq i \leq n$) is delayed by an amount iT_d w.r.t the start signal. The output of each delay element is compared with the stop signal to determine when the delayed outputs cross the stop signal. If the output of the j^{th} element crosses the stop signal for the first time, then

$$(j-1)T_d < T < jT_d$$

Flip flops are used to determine if the outputs of the delay elements have crossed the stop signal. The range of measurement is nT_d.

The advantage with the delay line based TDC is the simplicity of design that it offers. The design just requires a chain of inverters and flip flops. The disadvantages are that it has a timing resolution T_d which cannot be less than twice the minimum inverter delay in the given technology. In addition to that, the inverter delays are highly dependent on process and to a lesser extent on supply voltage and temperature variations which tend to make the TDC's resolution PVT dependent. This can be overcome by using a DLL to stabilize the delays, which increases power dissipation and total area.

2.2.2 Vernier delay line

Vernier line TDC (Abas *et al.*, 2007),(Li and Chou, 2007) works on a principle similar to that of Vernier calipers. The idea is to delay the start and stop signals by different amounts and to let the start signal "catch up" with the stop signal. The start signal is passed through a delay chain having n delay elements each having a delay T_1. The stop signal is also passed through a delay chain with each element

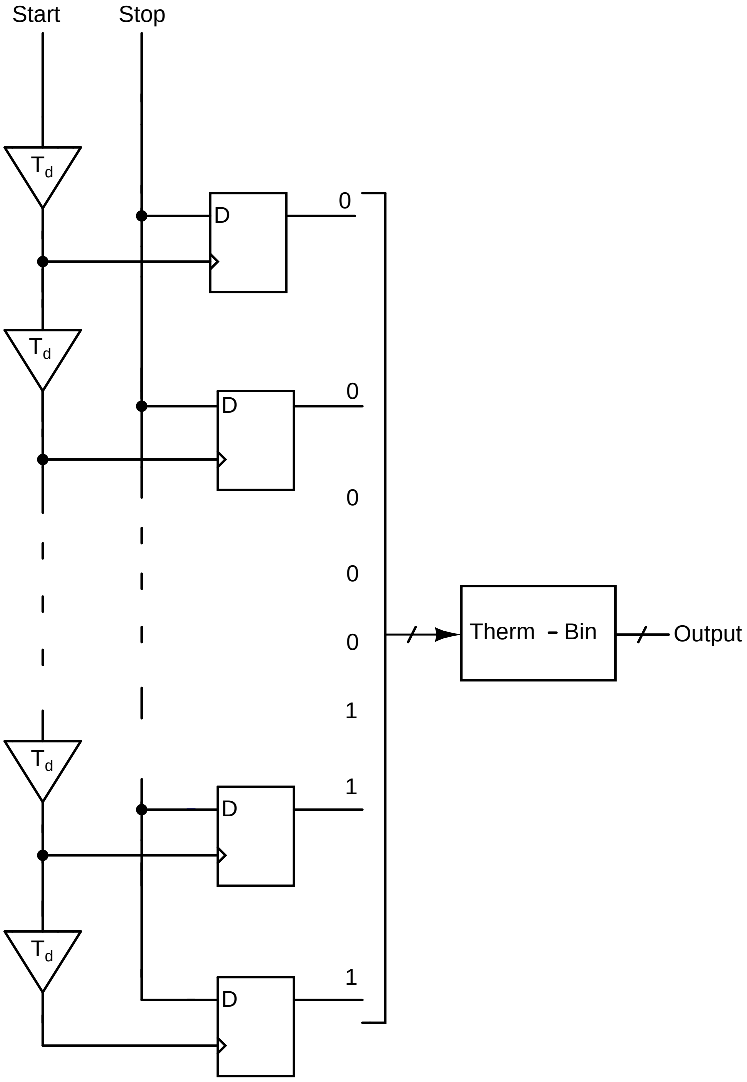


Figure 2.1: Flash or single delay line TDC.

having a delay of $T_2 (< T_1)$ as shown in Fig. 2.2. So, after each delay element, the time interval between the delayed start and delayed stop signals reduces by an amount $\Delta T = T_1 - T_2$. Eventually, the delayed start signal will cross the delayed stop signal. Like in the single delay line architecture, flip flops determine when the start signal crossed the stop signal. If the crossing occurs after the j^{th} element, then

$$(j-1)\Delta T < T < j\Delta T$$

Thus, the Vernier delay line is able to achieve a timing resolution of $T_1 - T_2$ which can be many times smaller than T_1 or T_2, which is what is achieved by a flash TDC. The disadvantage is increased area and the need for two DLLs to lock the two delays T_1 and T_2. A single DLL can also be used to lock the two delays but the circuit complexity and area goes up.

2.2.3 Time to amplitude converter

This architecture (Abas *et al.*, 2007) is different from the others in that it is a purely analog architecture. A current source is used to charge a capacitor during the interval between the two signals as shown in Fig. 2.3. The voltage across the capacitor ramps up and the final voltage is directly proportional to the time interval between the two signals. An ADC is used to generate the digital output from this voltage. Parasitics and input capacitance of the ADC at the capacitor terminal change the charging rate at that node. To reduce the effect of these added capacitances, a higher capacitance has to be used. The advantage of this scheme is that it does not require a DLL. The disadvantages are high power dissipation, large area due to the capacitor and need for a high resolution ADC, which might nullify the area advantage of not having a DLL.

2.2.4 Other architectures

Other techniques exist for measurement of time intervals which were not simulated. One of them is the pulse shrinking technique where the time interval to be measured is shrunk by a constant factor α every cycle till the time interval cannot be measured. Number of cycles taken can be used to determine the input time interval. This method requires a very stable feedback loop and a pulse shrinking cell, both of which are not very easy to design. Another approach is to use two oscillators running at slightly different frequencies, one enabled by the start signal and other enabled by the stop signal. After some cycles, one oscillator output

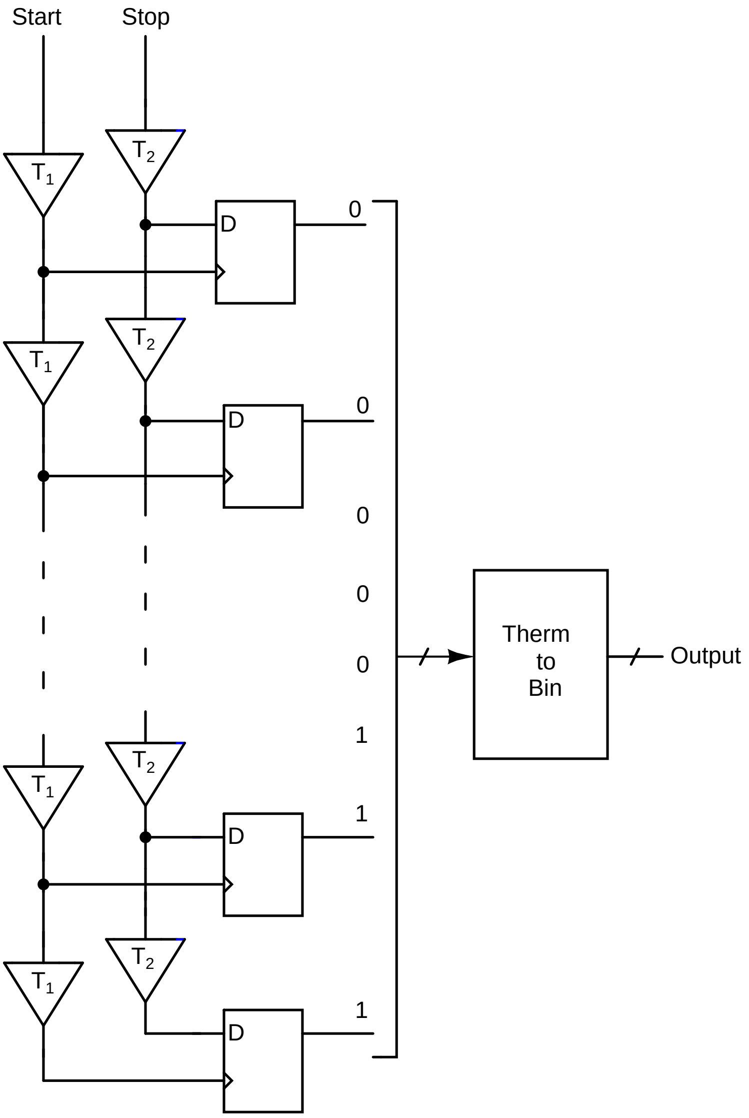


Figure 2.2: Vernier TDC.

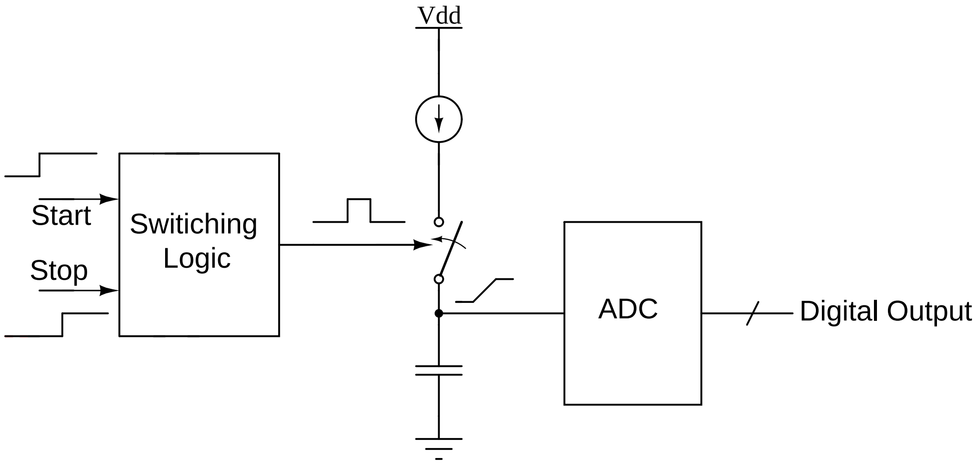


Figure 2.3: Time to amplitude converter based TDC.

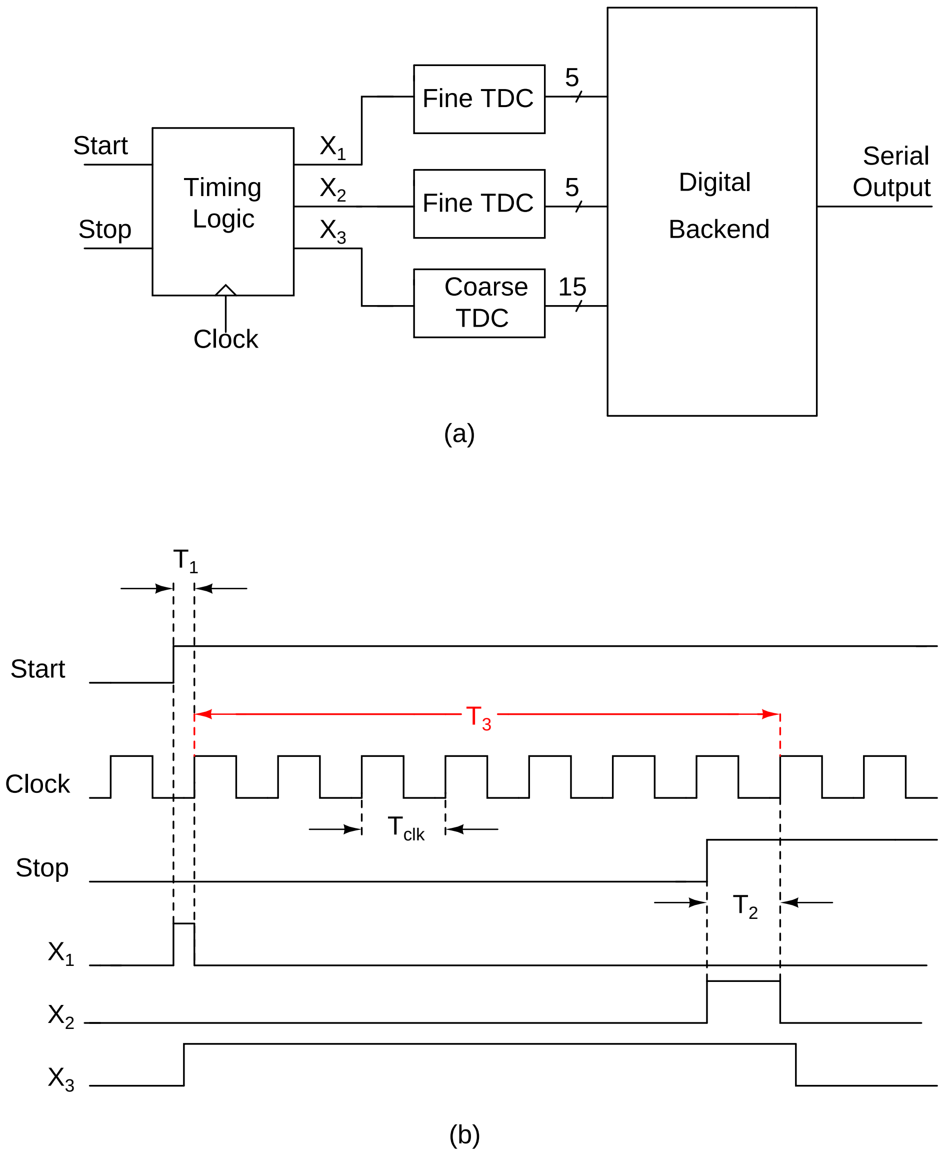
will cross the other and the number of cycles elapsed gives an indication of the input time interval. The big advantage with this method is that mismatch in the individual delay elements does not affect the TDC non linearity as long as the total periods of the two oscillators are locked. The disadvantage is that we need two oscillators which are locked to slightly different frequencies.

2.2.5 Comparison

Flash TDC offers a very simple and robust design. Vernier TDC is similar to flash TDC, but offers higher resolution at the expense of more area and higher complexity. Other architectures also have a potential of providing higher resolution at the expense of area, power and complexity. The resolution requirements of the INO project however can be met by a flash TDC. A high range, less area and low power solution is desired. So for the present work, a flash TDC architecture was chosen. A DLL was used to stabilize the delays against PVT variations. The exact architecture and design details are discussed in the next section.

2.3 TDC: Designed system

The architecture chosen for designing the TDC is a composite coarse-fine architecture. The main idea is to measure the time as sum of two parts: one measured by a coarse counter which gives a very high range but low resolution and the remaining part measured using a high resolution method which has a lower range. Consider the two inputs as shown in Fig. 2.4(b). Let the time interval between the two inputs be ΔT, the time to be digitized. The time interval ΔT can be split into three parts T_1, T_2 and T_3 w.r.t a system clock as shown in the figure. T_1 is the time between the start signal and the next rising edge of the clock. Similarly T_2 is the time between the stop signal and the rising edge immediately after it. T_3 is time between the two rising edges of the clock mentioned above. Clearly, $T_1, T_2 < T_{clk}$ and hence have to be measured by a fine TDC whereas $T_3 > T_{clk}$ and can be arbitrarily large within the range of the TDC. Hence it is measured using a coarse TDC. Also we have, $\Delta T = T_3 + T_1 - T_2$.



(a)

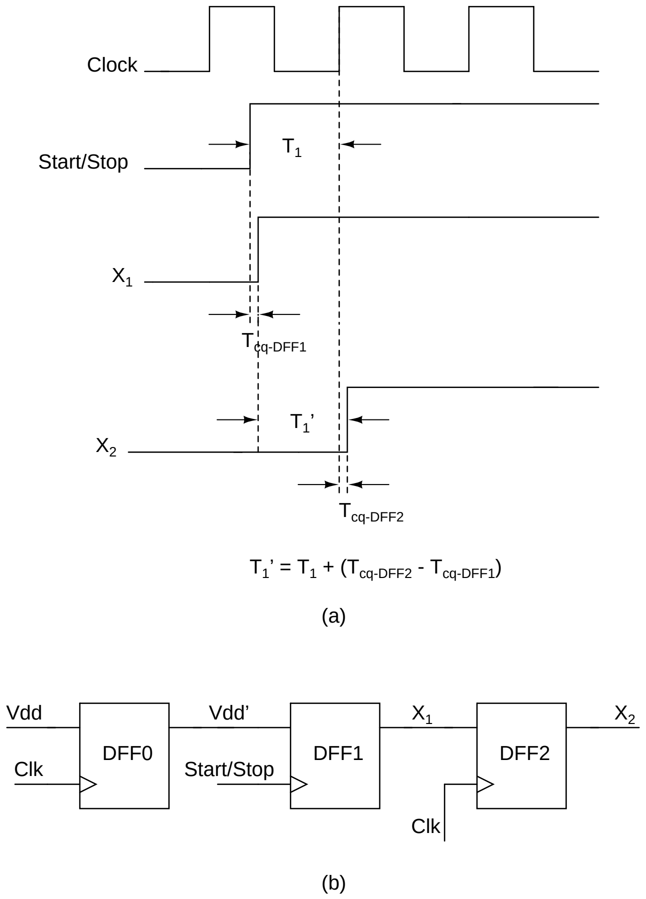
(b)

Figure 2.4: (a) The TDC system (b) Input signals and clock.

The block diagram of the complete design is shown in Fig. 2.4(a). A timing logic block extracts the signals corresponding to the intervals T_1 and T_2 and feeds them to the fine TDCs. It needs to ensure that the signals it outputs have the same time interval as the inputs it receives. This means ensuring identical design, similar loading and symmetric layout of devices on the signal path. It feeds the coarse TDC with a signal corresponding to the interval T_3. The fine resolution measurement of T_1 and T_2 is performed by the fine TDC and T_3 is measured by the coarse TDC. The thermometer coded output of the fine TDC is corrected for 'bubbles' and is converted to binary in the digital back end and ΔT is calculated from T_1, T_2 and T_3. T_1 and T_2 are measured with 5 bit resolution and T_3 is measured with 15 bit resolution. A serializer circuit in the digital back end converts the 16 bit parallel TDC output into a serial output data stream at a lower clock rate. Each of the blocks in Fig. 2.4(a) are explained below.

2.3.1 Timing logic

The timing logic block extracts the signals which need to be sent to the fine and coarse TDC. For the coarse TDC, whose implementation essentially involves just a counter, the timing logic block generates an enable signal. This enable signal goes high with start signal and goes low with arrival of the stop signal. The coarse TDC should count the number of clock cycles when this enable is high. For extracting signals to be sent to the fine TDC, a simple implementation using only flip flops was chosen as shown in Fig 2.5



(a)

(b)

Figure 2.5: Timing logic block.

As shown in the figure, the interval T_1 between the start/stop signal and the clock has to be extracted into two step signals spaced by the same amount. When the start/stop signal goes high, X_1 goes high. X_2 goes high the next time clock goes high. Hence the signals X_1 and X_2 are the signals that have the same interval as the interval between the start/stop signal and the next clock edge. It can be argued that we can directly use the start/stop signal instead of X_1 but then the outputs will be spaced by $T_{cq-DFF2}$ instead of $T_{cq-DFF2}-T_{cqDFF1}$, as in the current design. Since the two DFFs are assumed to be matched, we can expect the output interval to be almost the same as the input interval. Simulations however showed that the delay also depends on the amount of current drawn from the D input of the flip flops. If the D input of DFF1 is connected to the voltage source V_{dd}, it can draw more current than DFF2 which is connected to the output of DFF1. To rectify this, we connect DFF1 input also to a flip flop output, assuming that the start/stop signals arrive at least one cycle after the reference clock has started. The outputs X_1 and X_2 are given to the fine TDC. Since X_1 is loaded by DFF2 as well as the fine TDC whereas X_2 is loaded only by the fine TDC, the delay of flip flops can be different. This can easily be rectified by loading X_1 with a dummy load identical to the input capacitance of the fine TDC.

2.3.2 Coarse TDC

The coarse TDC is basically a 15 bit digital counter with a clock period of 4 ns. It is enabled with the start signal and disabled with the stop signal. It uses the reference clock running at 250 MHz for its counting. It was implemented in Verilog and synthesized and routed using automated CAD tools *Design Vision* and *Encounter*. It can measure time periods till 2^{15} clock cycles or $131\mu s$. The range of the complete TDC is same as the range of the coarse TDC. This range is highly scalable since the number of bits of the counter can be increased almost arbitrarily. The serializer at the output ensures that the number of pins does not become a limiting factor in scaling the range. The area consumed by the coarse TDC is 80μm × 20μm.

2.3.3 Fine TDC

The fine TDCs measure the remaining two time intervals with a fine resolution of 125 ps and a range of 4 ns. As mentioned before, a single delay line or flash architecture shown in Fig. 2.1 was chosen. The start signal is passed through a delay chain of 32 delay elements each having a delay $T_d = 125$ ps. The number of delay elements is chosen to be a power of 2 so as to completely utilize the bits of the output binary code. The output of the i^{th} delay element $(0 \leq i \leq n)$ is delayed by an amount iT_d w.r.t the start signal. The output of each delay element is compared with the stop signal to determine when the delayed outputs cross the stop signal. If the output of the j^{th} element crosses the stop signal for the first time, then

$$(j-1)T_d < T < jT_d$$

. Flip flops are used to determine if the outputs of the delay elements have crossed the stop signal. The output is a thermometer code where the number of 1's in the code represents the number of LSBs in the input.

The delay elements in the simplest implementation can be just a pair of inverters in series. But if the delay elements are chosen to be buffers made from inverters, then the delay of each element varies significantly with variations in process, supply voltage and temperature. Simulations show that between the *ss* (slow) and the *ff* (fast) process corners, the inverter delay varies by 60% of its value at the *tt* (typical) corner in the 130nm CMOS process. This would result in the LSB resolution being strongly dependent on process variations. Also, in a composite coarse-fine architecture such as this, it is necessary that the range of the fine counter be the same as the resolution of the coarse counter. The range of a simple inverter chain can vary by 60% across the corners, making it infeasible for the coarse-fine architecture to work correctly. So we need delay elements whose delay does not change with process variations.

One approach is to use a voltage controlled delay unit (VCDU) where the control voltages are adjusted so as to give the same delay in spite of process variations. The voltages are tuned by a DLL (Section 2.4) which is locked to a fixed delay. The VCDU is basically similar to an inverter with additional transistors to control the current used to charge or discharge the output. The topology shown in Fig 2.6 is called current starved topology since the top and bottom transistors 'starve' the transistors in the signal path for current.

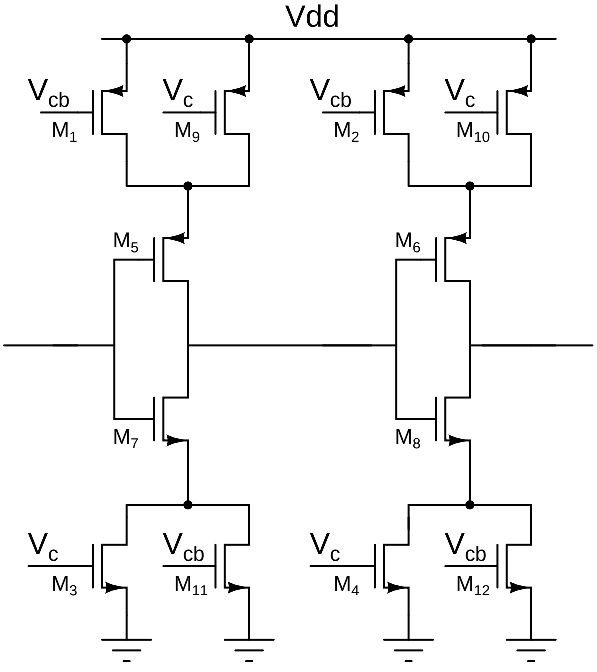


Figure 2.6: Voltage controlled delay unit (VCDU)

The transistors $M_{1,2,3,4}$ provide a current to the transistors $M_{5,6,7,8}$ which depends on the control voltages V_c and V_{cb}. This controls the delay from the input to the output of the VCDU. The transistors $M_{9,10,11,12}$ are used to ensure that there is a finite delay between the input and output in case the control voltage V_c falls below the threshold voltage of $M_{3,4}$. This can happen during the initial cycles when the DLL which generates the voltages V_c and V_{cb} is yet to lock. An alternative possibility exists for making a VCDU where the supply pin of a standard inverter is replaced by the control voltage. While this does ensure dependence of delay on control voltage, the drawback is that current is drawn from the control voltage node which may result in the control voltage change. The VCDU implemented in this design has only gate capacitance loads at V_c and V_{cb} and hence current drawn is much lesser.

The VCDU delay characteristics are shown below for various process corners along with the voltages corresponding to a delay of $125ps$. As we can see, all three voltages are well within the swing limits of the charge pump which drives these voltages and also above the threshold voltage of the transistors in the VCDU.

The delayed start signal is compared with the stop signal by the D flip flops to determine which signal arrived first. The stop signal is fed to the D input and the delayed start signal is fed to the clock input. If the stop signal has arrived more than one setup time before the delayed start signal, then the output will be a logic

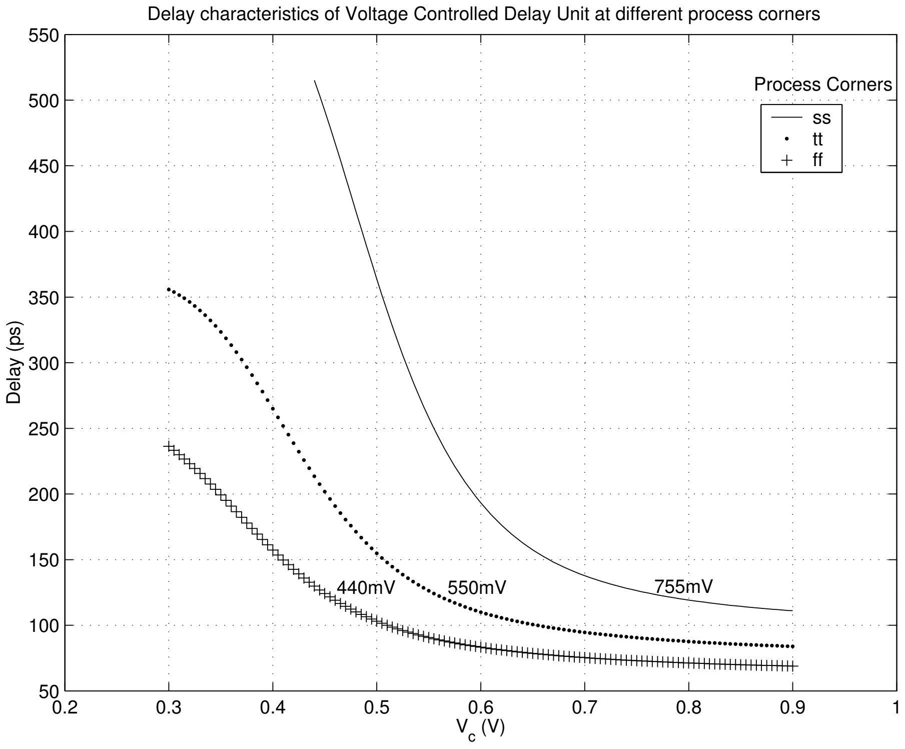


Figure 2.7: VCDU Characteristics. The voltages shown correspond to 125 ps delay.

1 otherwise a 0. In case the start and stop signal arrive within one setup time of each other, the D flip flop will go into a metastable state where the output can be unpredictable. This also causes the TDC characteristics to be unpredictable for a duration of one setup time near each code transition. So, there is a need to reduce the setup time of the flip flops. We can do so by modifying the standard master slave topology a little (Zhou *et al.*, 2001). Consider a master slave topology of a D flip flop as shown in Fig 2.8(a)

When the clock ϕ is low, the data D is loaded to node A and held at the output of the master stage. When the clock goes high, this data is propagated to the output terminal. Now assume that the data changes before the clock changes from 0 to 1. This change in data has to propagate to the node B. If the clock arrives before the change of data has propagated to the node B, the previous value of data will be propagated to the output. There's a "race" condition where the node A is being driven by two different sources and there will be some delay before one of the values is latched on. So, we can say that in the worst case, the setup time is given by

$$t_{su} = t_{G1} + t_{INV1} + t_{INV2} + t_{G2} + t_{race}$$

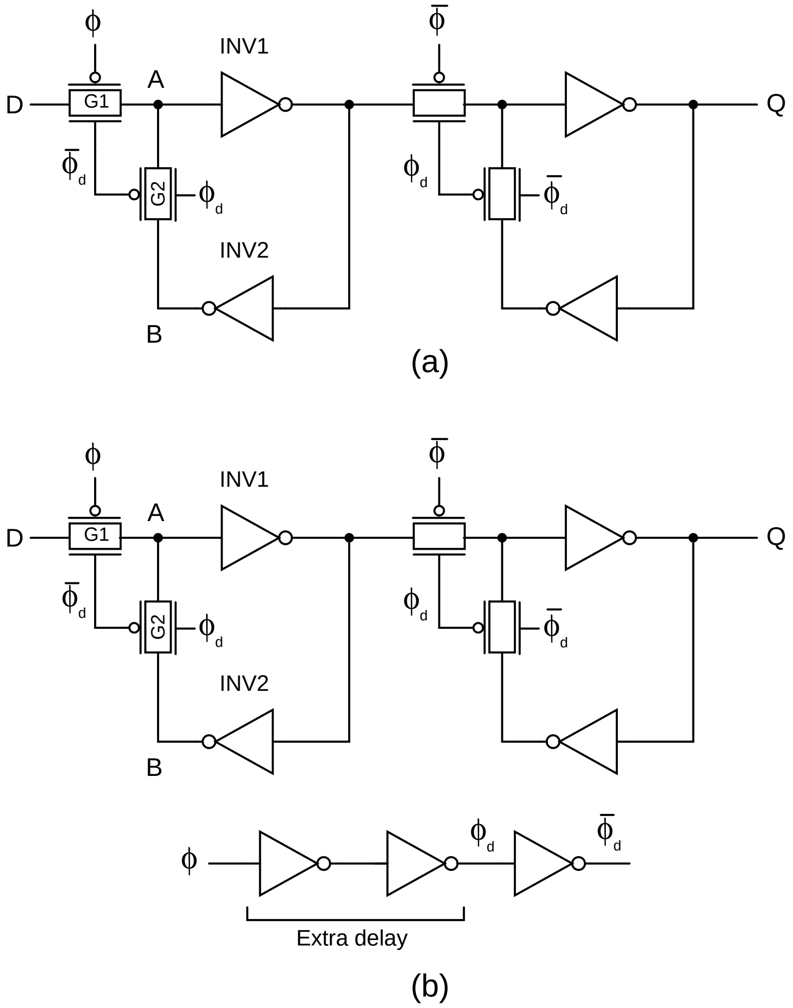


Figure 2.8: (a) Standard master slave topology of a D flip flop (b) Reduced setup time D flip flop.

The modified design uses two clock phases, one for gate $G1$ and other for gate $G2$. Gate $G2$ is given the delayed clock phase ϕ_d and $G1$ is given the clock ϕ. When ϕ goes high, ϕ_d is still low and hence if some new data was written on the D input, it can propagate through $G1-INV1$ path without the $INV2-G2$ path affecting the data value. The setup time is hence greatly reduced and the setup time in this case is given by

$$t_{su-new} = t_{G1} + t_{INV1}$$

Simulations show that the modified design has a setup time of 14 ps while the original design had 65 ps setup time.

The other possibility i.e. with delayed start at the D input of flip flop also works but the capacitance looking into the D input depends on the value of the clock and hence results in delay mismatch between different delay elements.

2.3.4 Digital back end

The digital back end consists of digital circuitry which takes input from the output of the main TDC circuitry. The digital back end can run at very low frequencies since the expected event rate in neutrino experiments is very low. The backend consists of an adder circuit which adds the number of ones in the output thermometer code. Apart from this, we also have adder/subtracter circuitry for adding and subtracting TDC outputs so as to obtain $\Delta T = T_3 + T_1 - T_2$. The output digitized time interval is 16 bits in size. To avoid dedicating 16 pins for this output, we use a serializer which sends output serial data. The serializer circuit is simply a counter which counts till 16 and assigns one TDC output bit to the serial output in each cycle. As mentioned above, the neutrino hits rates are quite low and somewhat arbitrarily, a serializer clock rate of 40 kHz was chosen, which implies a serial clock rate of 640 kHz. This means that the 16 bit measurement data is transmitted from the output approximately every 1.56 μs.

The digital back end, not being too stringent in its speed and accuracy requirements was synthesized, placed and routed used automated CAD tools *Design Vision* and *Encounter*. The area consumed is 130μm × 80μm.

2.4 Delay locked loop

As mentioned before, we need to ensure that the delay of the VCDU is constant in spite of process variations. For this purpose, we use a DLL which locks the delays of its elements to a fixed value. It does so by altering the control voltages using a feedback loop. If the delay elements in the DLL and the TDC are designed and laid out in an exactly identical manner, the delays in the TDC will also be constant in spite of process variations. Block diagram of a DLL is shown in Fig. 2.9. The working is similar to that of a PLL except that the VCO is replaced by a delay line and no frequency division takes place. The phase frequency detector (PFD) detects the phase difference between the input clock and the feedback signal. Depending on which signal is lagging, the PFD gives UP and DN output signals so as to correct the phase difference. The charge pump changes the control voltage of the delay line appropriately so as to reduce the magnitude of the phase difference at the input of the PFD. The DLL designed as part of this project has 32 delay elements and an input clock of period 4 ns. So, each delay element is locked to 125 ps. This delay is replicated in the delay line of the TDC by using the same control voltage and nominally identical design of delay line. Assuming PVT variations affect the delay line in the DLL and that in the TDC identically, the delays in the TDC will also be 125 ps, independent of PVT variations. We now describe the design of each of the components of the DLL.

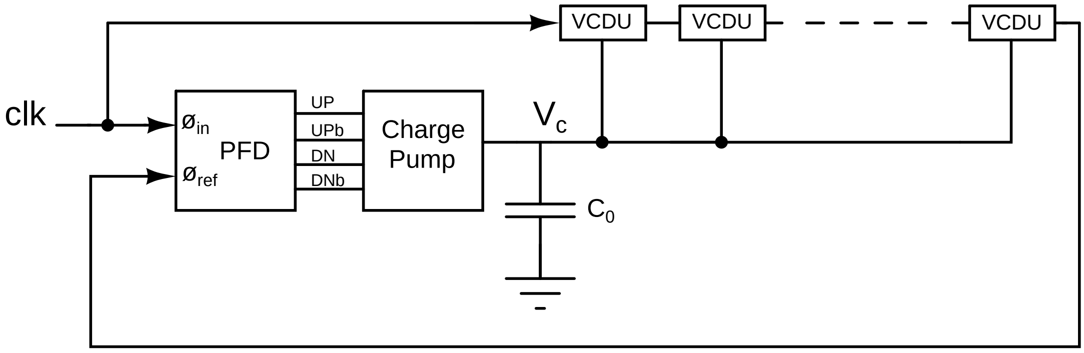


Figure 2.9: Delay Locked Loop

2.4.1 Phase frequency detector

The PFD is used to detect the phase difference between its input signals. A standard architecture is used with slight modifications. The basic architecture is shown in Fig 2.10(a). Let us assume that the feedback signal is lagging the input clock by $\Delta t > 0$. In this case, the outputs of the PFD will be as shown in Fig 2.10(b)

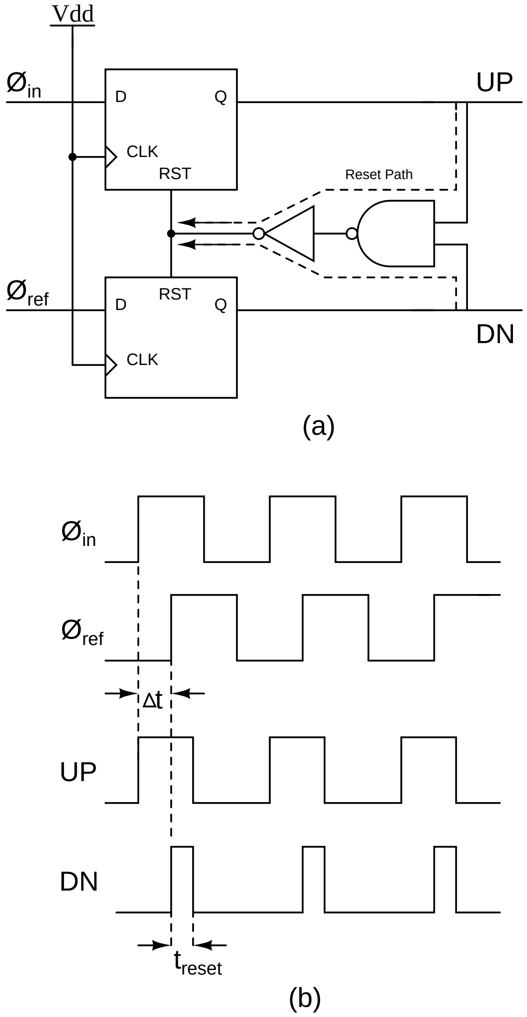


Figure 2.10: (a) Archituecture of PFD (b) Input output waveforms

Here t_{reset} is the reset delay which is the sum of delay through the reset path shown in the figure and the clock-to-Q delay of the FF. UPb and DNb are just complements of UP and DN. Now, in a standard implementation, the four signals, UP, DN, UPb and DNb are directly fed to the charge pump. Since $\Delta t > 0$, the upper branch of the charge pump charges the output during the interval Δt and both the branches are ON for the time interval t_{reset}. If the current being pumped out of the charge pump is same as the current being pulled in, the net charge deposited on the capacitor C_0 during the interval t_{reset} is zero and the output voltage of the charge pump doesn't change. However due to random process and systematic design mismatches, the two currents are not the same. In this case, the output voltage of the charge pump changes even during the reset period. This results in a non-zero locking offset in the DLL. The locking offset is given by

$$t_{off} = \frac{I_{mis}}{I_0} t_{reset}$$

where I_{mis} is the mismatch between the currents in the charge pump and I_0 is the nominal current. To avoid this problem, the following method was used.

The signals UP and DN have a common period of duration t_{reset} when both of them are ON. Instead of giving the signals as they are to the charge pump, we can remove the common period by simple digital logic and feed the charge pump with just one (say UP) of the signals which is high for the duration Δt. The DN signal is identically zero. Similarly, for $\Delta t < 0$, DN is high for the duration Δt and the UP signal is identically zero. The common interval is removed by taking the AND of UP and DNb and that of UPb and DN and using them as UP and DN respectively in the charge pump. This removes the constraint of the charge pump currents to be identically matched. The signals are shown in Fig 2.11 for clarity.

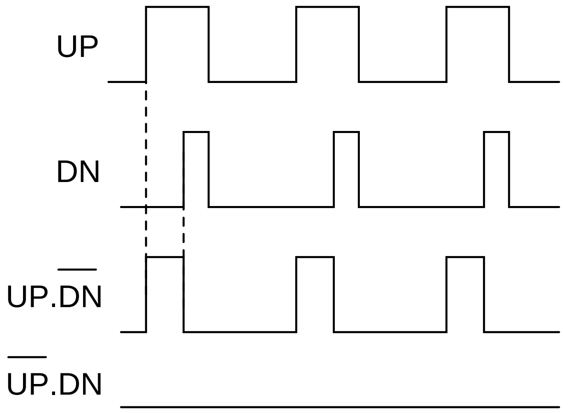
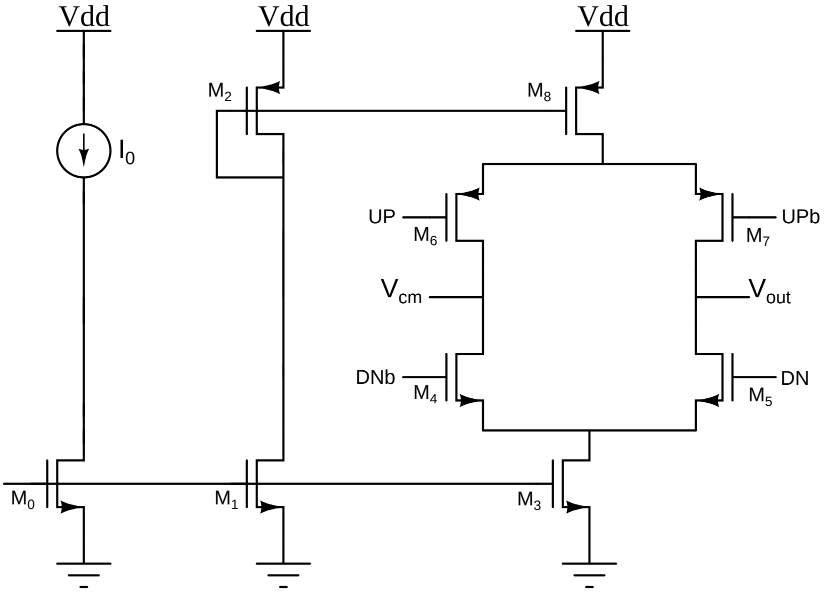


Figure 2.11: Removal of reset delay from UP and DN signals

Simulations results showed the original design as having a locking offset of 198 ps while the modified design has a locking offset of 4 ps.

2.4.2 Charge pump

The charge pump is used to adjust the control voltage of the delay elements depending on the outputs of the PFD. The charge pump should pump a current I_0 into the output when the UP signal is high and should pull out the same current when the DN signal is high. A simple architecture with four switches is used as shown in Fig 2.12. When UP is high, i.e UPb is down, DN must be low because of the chosen topology of the PFD. Hence the current from the top current source is pumped into the output node and the output voltage increases. Similarly when DN is high and UP is low, the bottom current source pulls out current from the output node, causing the output voltage to reduce. The drains of the transistors in the other branch of switches are connected to $V_{cm} = 600mV$ so as to have an output DC operating point near V_{cm}. A current of $10\mu A$ and $50\mu A$ is chosen for the mirror circuitry and the switches respectively. The circuit diagram with the sizing of transistors is shown in Fig 2.12.



M_0, M_1	2.12um/1.5um	M_4, M_5	1.2um/0.12um
M_2	3.6um/1.5um	M_6, M_7	4.02um/0.12um
M_3	10.6um/1.5um	M_8	18um/1.5um
I_0, V_{cm}	10uA, 600mV		

Figure 2.12: Charge pump circuit

The transistors in the mirror circuitry are chosen to be long so as to reduce the channel length modulation effect and hence to increase the accuracy of the mirroring. The transistors which are used as switches have minimum length so as the increase the switching speed.

2.4.3 Delay line

The delay line is replicated from its implementation in the TDC so as to ensure that the delays in the TDC are the same as the delay to which each delay element in the DLL locks. The loading and physical layout of the delay line is hence kept exactly the same as in the TDC.

Since the VCDUs use two control voltages having opposite incremental polarity, we need to generate them both in the DLL. V_{cb} is generated from V_c using the circuit shown in Fig. 2.13. The sizing ensures that the incremental gain from V_c to V_{cb} is -1. The sizes are chosen large since V_{cb} has to drive a large number of VCDUs.

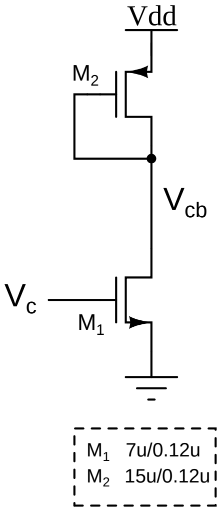


Figure 2.13: Control voltages

2.5 TDC non linearities

In the analysis so far, it was assumed that all the elements in the delay line have identical delays. This assumption is in fact not correct because of the mismatch between devices on an IC. The main cause of mismatch in delays is the mismatch in threshold voltages of transistors. We develop here a model for the delay of the delay elements in terms of the transistor parameters β and V_{th}. We will then quantify the non linearity of the TDC output code due to process variations in these parameters.

Consider a falling input transition as shown in Fig 2.14. We assume that the rise and fall times of all the delay elements are same. This is because we have designed the delay unit to have equal rise and fall times. Also, the rise/fall time of each element is same as that of the other element because in case of long inverter chains, the rise and fall times become constant as the signal travels through the chain. Let the rising/falling times be $t_f = t_r$. As the input begins to fall, the output starts rising after the transistor M_1 comes out of cut-off region. This happens when the input falls one V_t below V_{dd}. Hence this delay, t_d is proportional to the falling time of the input. Simulations show $t_d = \frac{t_f}{2}$ for a wide variety of loads and input slews. If the output rise time is t_r, the delay between $V_{dd}/2$ transitions of input and the output is $t_d + (t_r - t_f)/2$. As discussed above, we can assume $t_f = t_r$ and hence the delay of the delay element is $t_d = \frac{t_f}{2}$. So, we basically need a model to calculate t_f from the transistor parameters.

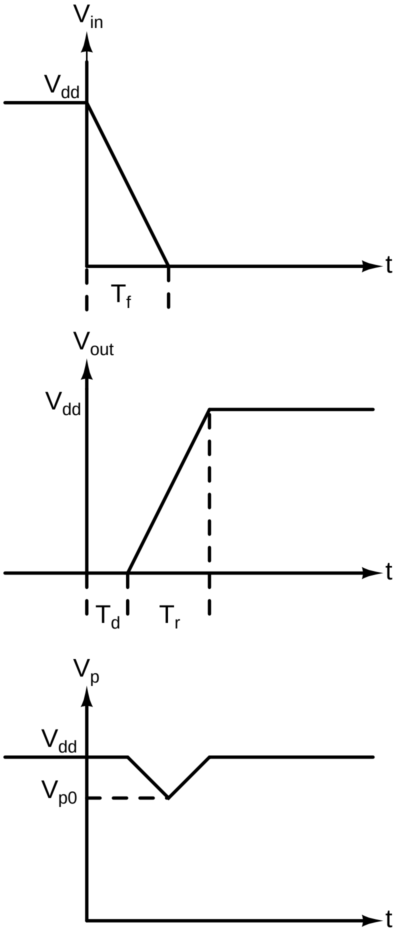


Figure 2.14: VCDU - Input, output and intermediate node waveforms

For this purpose, we consider the working of the delay element in more detail. Consider the voltage controlled delay unit as shown in Fig. 2.15. We neglect the additional two transistors which were used to ensure delay propagation in case of $V_c = 0$. The transistors M_p and M_n control the amount of current used to

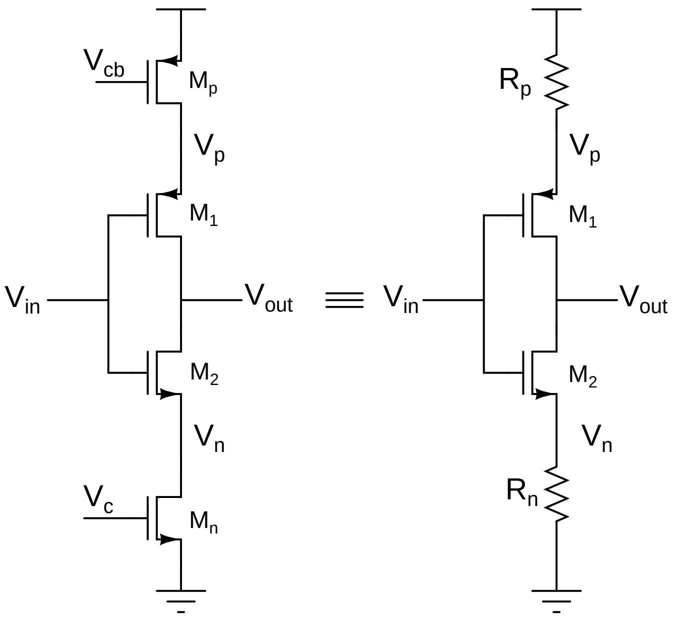


Figure 2.15: VCDU and its approximate equivalent

charge or discharge the output capacitor. The nodes V_p and V_n vary a little while charging or discharging the output. The variation however is small enough for the transistors M_p and M_n to remain in linear region. Hence we can assume them to be resistors. We consider only the falling transition here and we can replace the transistor M_p by a resistance $R_p = \frac{1}{\beta_p(V_{sg}-V_{cb}-V_{thp})}$. When the input goes low, V_p starts reducing since the current pulled by M_1 is more than the current being pulled from V_{dd}. At some point, the voltage starts rising back and goes to its initial value of V_{dd}. This happens when the current being pulled by M_1 out of V_p is same as the current being pulled from V_{dd}. Let the minimum voltage at node V_p be V_{p0}. We get,

$$\frac{V_{dd}-V_{p0}}{R_p} = \beta_1 V_{dsM_1}(V_{p0}-V_{th1})$$

Assuming that V_p has equal falling and rising times, we can assume $V_{out} \simeq \frac{V_{dd}}{2}$ for calculating V_{p0}. Expanding, we get

$$V_{p0}^2 - V_{p0}\left(V_{th1} + \frac{V_{dd}}{2} - \frac{1}{\beta_1 R_p}\right) + V_{dd}\left(\frac{V_{th1}}{2} - \frac{1}{\beta_1 R_p}\right) = 0 \qquad (2.1)$$

Using the sizes and operating points from our design, we get $V_{p0} = 0.91V$ which is close to the simulation result of $0.96V$. We are however more interested in using this to determine an expression for the delay which will help us determine the impact of threshold voltage mismatch on the TDC linearity. Since the final voltage at V_p is same as the initial voltage, no net charge is deposited at this node. Hence, all the current that was pulled from V_{dd} through the resistor was used to charge the output capacitor. The current waveform through the resistor shown in Fig. 2.15 is also triangular since $I_p = \frac{V_{dd}-V_p}{R_p}$ and we have modeled the waveform of V_p as triangular. Hence, if the maximum current is I_{p0}, then the average current will be $\frac{I_{p0}}{2}$. So, if the rise time of the output is t_f, then the average charge deposited at the output is $\frac{I_{p0}t_f}{2}$. Assuming the capacitance at the output to be C, and equating the charge deposited to CV_{dd}, we get

$$t_f = \frac{2CV_{dd}}{I_{p0}}$$

Expressing in terms of V_{p0} and using $t_d = \frac{t_f}{2}$, we get

$$t_d = \frac{CV_{dd}R_p}{V_{dd} - V_{p0}}$$

where V_{p0} is given by (2.1). To quantify the variation in delay due to variation in threshold voltages, we consider the partial derivatives $\frac{\partial t_d}{\partial V_{th1}}$ and $\frac{\partial t_d}{\partial V_{thp}}$. Differentiating (2.1) and solving, we get

$$\frac{\partial V_{p0}}{\partial V_{th1}} = \frac{V_{th1} - V_{dd}/2}{2V_{p0} - V_{dd}/2 - V_{th1} + \frac{1}{\beta R}}$$

From $t_d = \frac{CV_{dd}R_p}{V_{dd}-V_{p0}}$ we get

$$\frac{\partial t_d}{t_d} = \frac{\partial V_{th1}}{(V_{dd} - V_{p0})} \frac{V_{p0} - V_{dd}/2}{2V_{p0} - V_{dd}/2 - V_{th1} + \frac{1}{\beta R}}$$

Similarly, for small changes in V_{thp}, we get

$$\frac{\partial t_d}{\partial V_{thp}} = \frac{CV_{dd}R_p}{(V_{dd} - V_{p0})^2} \frac{\partial V_{p0}}{\partial V_{thp}} + \frac{CV_{dd}}{(V_{dd} - V_{p0})} \frac{\partial R_p}{\partial V_{thp}}$$

$$= \frac{CV_{dd}R_p}{(V_{dd} - V_{p0})^2} \left(\frac{(V_{th1} - V_{dd})\frac{\beta_p}{\beta_1}}{2V_{p0} - V_{dd}/2 - V_{th1} + \frac{1}{\beta R}} \right) + \frac{CV_{dd}}{(V_{dd} - V_{p0})} \left(\frac{1}{\beta_p(V_{dd} - V_c - V_{thp})^2} \right)$$

$$\frac{\partial t_d}{t_d} = \left(\frac{-\frac{\beta_p}{\beta_1}\partial V_{thp}}{2V_{p0} - V_{dd}/2 - V_{th1} + \frac{1}{\beta R}} \right) + \frac{\partial V_{thp}}{\beta_p R_p (V_{dd} - V_c - V_{thp})^2}$$

Using $C = 10fF$ (from transistor sizes and parasitic capacitance density of 10fF/μm^2), $V_{th1} = V_{thp} = 0.3V$ and $\beta = 75\mu A/V^2$, we get

$$\frac{\delta t_d}{t_d} = 0.67\delta V_{th1} + 2.38\delta V_{thp}$$

Table 2.1: Mismatch in delays for nominal delay of $t_d = 68ps$

	δt_d: Simulation	δt_d: Model
$\Delta V_{th1} = 5mV$	$290fs$	$245fs$
$\Delta V_{thp} = 5mV$	$845fs$	$810fs$

The delay block was simulated by slightly varying V_{th1} and V_{thp} and results are as shown in Table 2.1. We can see that the model is reasonably accurate in modeling the delay.

As a final step, we calculate the variance of the mismatch from the known value of variance of threshold voltages. We know that $\sigma(V_{th}) = \frac{A_{VT}}{\sqrt{WL}}$. For 130nm UMC process, we have $A_{VT} = 4mV\mu m$. So, we get, for our sizes, $\sigma\left(\frac{\delta t_d}{t_d}\right) = 0.011$. Hence, we have $\sigma\left(\delta t_d\right) \simeq 0.01LSB$. Hence the differential non linearity (DNL) of the TDC output code has a variance of 0.01 LSB. The integral non linearity (INL) is zero for the the codes at the ends because the total delay is locked by the DLL. The INL peaks in the middle and attains a maximum of $\frac{\sqrt{32}}{2}\sigma\left(\frac{\delta t_d}{t_d}\right) \simeq$ 0.03 LSB. Hence we see from our analysis that the DNL and the INL are both well below 0.1 LSB.

CHAPTER 3

Analog Front-End

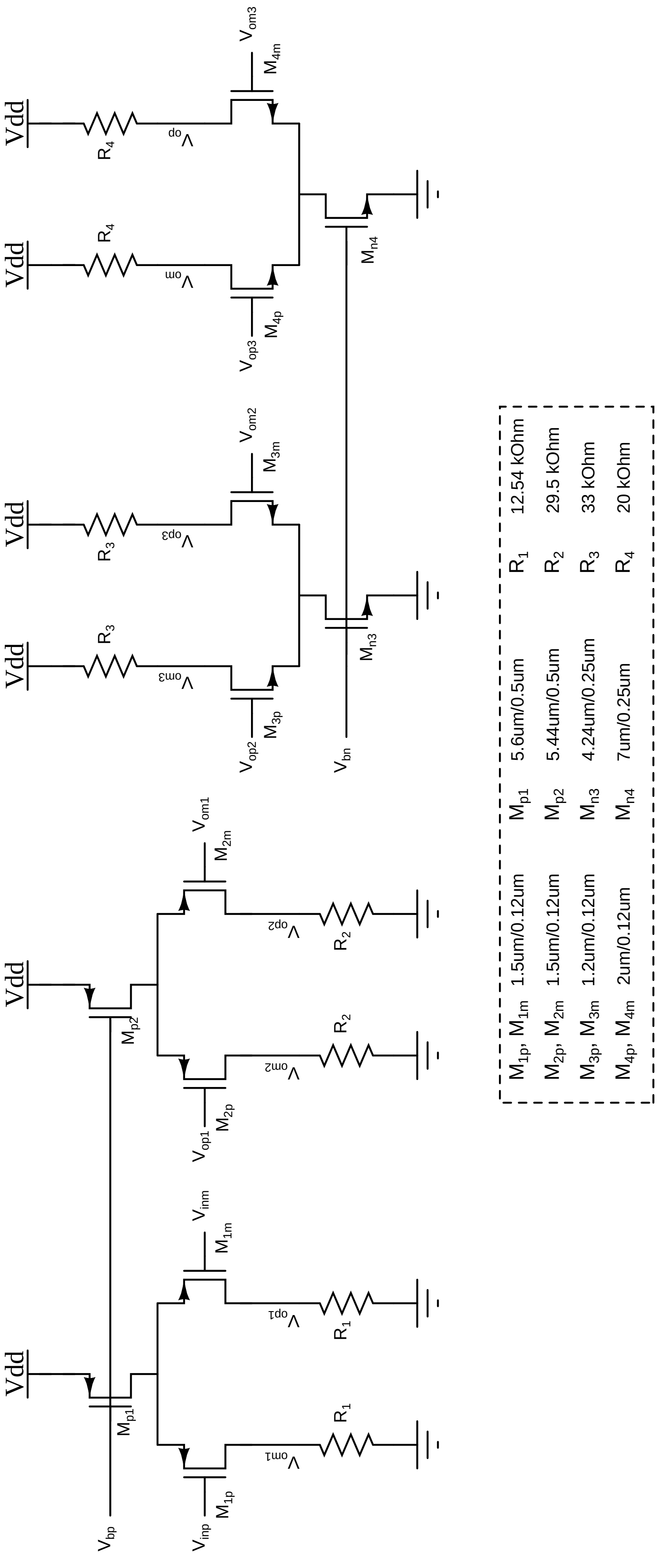
As mentioned in introduction, this design is intended to be used in a neutrino observatory apparatus. The apparatus aims at tracking the trajectory of neutrinos as they pass through the region of interest. The apparatus consists of a vertical stack of iron plates. Each plate has a mesh of 32 by 32 readout channels. This helps us to determine the (x, y) of the neutrino hit on a given metal plate. The index of the metal plate will give us the z coordinate of the neutrino hit. We also need to determine the timing of the neutrino hit with respect to a reference start. Upon determining all these, we can know the (x, y, z, t) profile of the neutrino trajectory. For this, we need to be able to detect the hit on a given plate and process it accurately. At the location of the hit, a small voltage is generated due to avalanche mechanisms of the materials used between the plates. The voltage spike is about 1 mV in amplitude and is about 10-20 ns wide. We want to take this as the input to our system and determine all the required parameters like amplitude, width of pulse etc. The associated front end circuitry is discussed in this chapter.

3.1 Front end amplifier

The voltage spike at the input of our system has to be first amplified to be able to process it further. As mentioned above, the amplitude of the spike is around 1 mV. So,we choose the gain of the amplifier to be 100 so that the output will have an amplitude of around 100 mV which can be used by an ADC to digitize the spike. The rise time of the spike is about 1 ns. So, the bandwidth of our system should be such that around 3 time constants are around 1 ns. From this, we get a bandwidth of 478 MHz. Hence, we choose the specifications of our amplifier as: a gain of 100 and a bandwidth of 500 MHz.

Note that the amplifier can be used in an open loop configuration to get high bandwidth since the gain of 100 is not a stringent requirement. So the stability of the amplifier is not an issue for designing the amplifier. So, we can have multiple poles before the unity gain frequency as long as the -3dB bandwidth meets the requirement of 500 MHz at all process corners. Since short channel transistors are used for high bandwidth, gain per stage is small and we have to use multiple stages to achieve the gain. The standard topology of a fully differential amplifier with active loads can be used. However, we then need to use a separate common mode feedback circuitry to stabilize the common mode. Also, the active loads add significant parasitic capacitances at the output of each stage, making the target bandwidth of 500 MHz difficult to attain. So, rather than going for a topology with active loads, which is generally used for achieving very high gains, we can go for a differential pair with resistive loads. Here the output common mode is automatically set as the DC operating point and the parasitic capacitances of the resistors are much less than that of transistors. Four stages were used for this amplifier.

An additional constraint we haven't discussed yet is that the input from the metal plates has a zero DC voltage. If we use the first stage as an NMOS one, we need to AC couple the input to provide for a DC bias for the NMOS differential pair. This will increase the size of the design, as coupling capacitors can be bulky. Instead of using coupling capacitors, we can use an input PMOS stage to which the input can be directly applied. This also helps for noise purposes, since the flicker noise of the a PMOS transistor is lower than that of an NMOS. So, we choose the first two stages as PMOS stages and the next two as NMOS stages. The designed amplifier with the transistor sizing is shown in Fig 3.1.



M_{1p}, M_{1m}	1.5um/0.12um	M_{p1}	5.6um/0.5um	R_1	12.54 kOhm
M_{2p}, M_{2m}	1.5um/0.12um	M_{p2}	5.44um/0.5um	R_2	29.5 kOhm
M_{3p}, M_{3m}	1.2um/0.12um	M_{n3}	4.24um/0.25um	R_3	33 kOhm
M_{4p}, M_{4m}	2um/0.12um	M_{n4}	7um/0.25um	R_4	20 kOhm

Figure 3.1: Amplifier circuit diagram

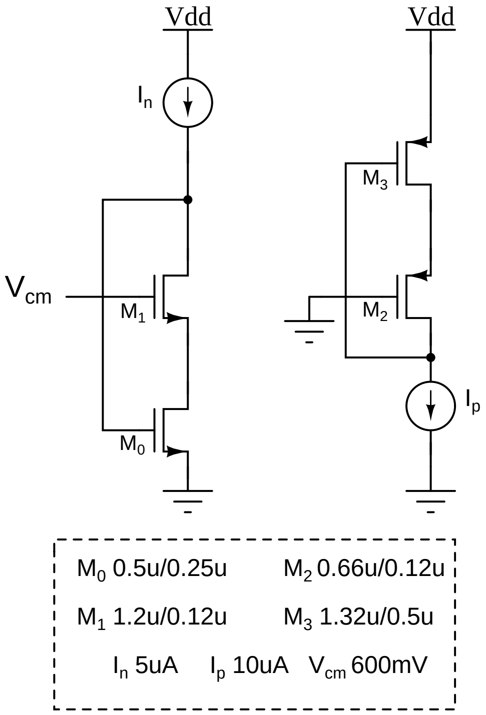


Figure 3.2: Cascode current mirroring scheme

A cascode current mirroring scheme is used as shown in Fig 3.2. This helps in more accurate mirroring since it uses identical V_{ds} in addition to identical V_{gs}. The stages are not AC coupled and the output of one stage is directly fed to the next stage. This helps in saving the area of coupling capacitors and also allows the offset cancellation schemes to work.

The AC magnitude response is shown in Fig 3.3 below. The transient response for a 1 mV spike is shown in Fig 3.4 with the output scaled by a factor of gain so as to plot both of them on same scale. We can see that the gain is more that 100 and the shape of the output is almost same as the input shape.

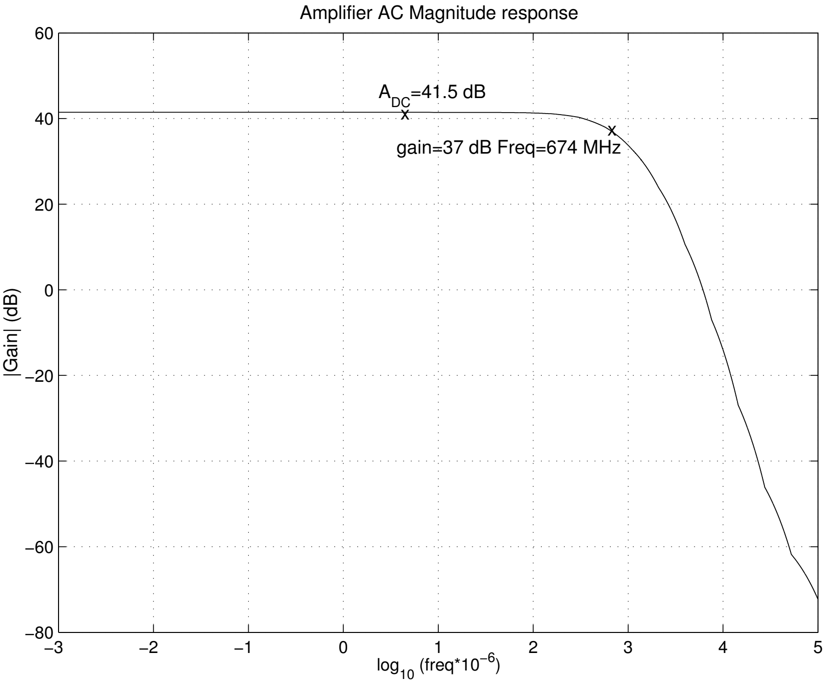


Figure 3.3: AC magnitude response of amplifier

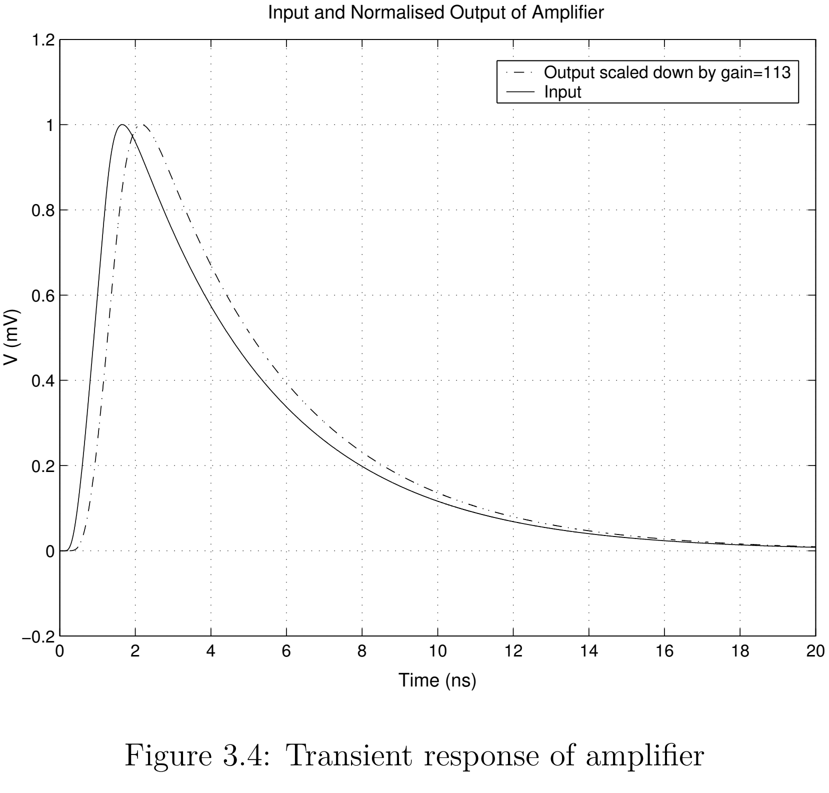


Figure 3.4: Transient response of amplifier

3.1.1 Offset cancellation

While the amplifier design as mentioned above works at all process corners, it does not consider random mismatch between the elements and assumes them to be exactly identical. We know that in reality this is not the case. The threshold voltages and current factors of the transistors can vary because of random dopant fluctuations. Even the resistors which are assumed to be nominally identical may not be so. This results in the two branches of each stage of the amplifier to be slightly different from each other. This means that the differential output will not be zero even if the input difference is zero, i.e, the amplifier has an input referred offset. Whether this is a serious problem or not depends on the extent of mismatch and so we need to quantify the input referred offset voltage. The mismatch of the stages after the first one affect the input referred offset to a much lesser extent because of gain preceding those stages and hence we consider only the effect of mismatches in the first stage.

Consider the first stage of the amplifier as shown below. The DC operating points are also mentioned, as they affect the extent of the effect of mismatch.

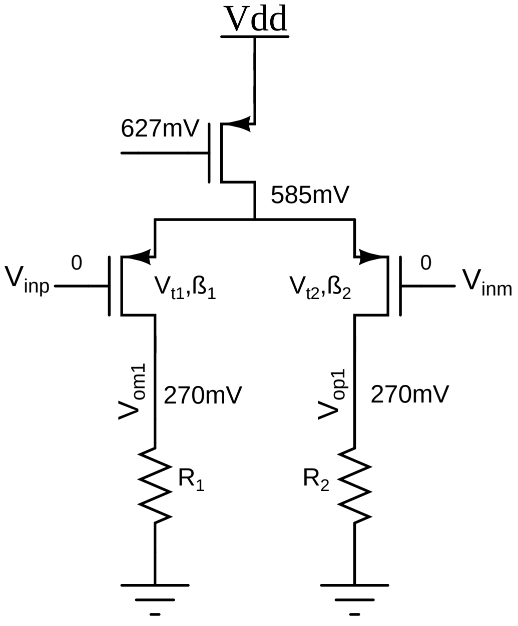


Figure 3.5: Amplifier: First stage

We want to determine the input voltage for which the output voltage is zero. Let us assume that the threshold voltage, current factor and resistance value of the first branch are given by V_{t1}, β_1 and R_1 and let the corresponding values for the second branch be V_{t2}, β_2 and R_2, where $V_{t2} = V_{t1} + \Delta V_t$, $\beta_2 = \beta_1 + \Delta\beta$ and

$R_2 = R_1 + \Delta R$. Let the current in the two branches be I_1 and $I_2 = I_1 + \Delta I$. Since $V_{off} = V_{gs1} - V_{gs2}$, we have

$$V_{off} = \sqrt{\frac{2I_1}{\beta}_1} + V_{t1} - \sqrt{\frac{2I_2}{\beta}_2} - V_{t2}$$

Simplifying, we get

$$= \sqrt{\frac{2I_1}{\beta}} \left(1 - \sqrt{\frac{1 + \frac{\Delta I}{I_1}}{1 + \frac{\Delta\beta}{\beta}}}\right) - \Delta V_t$$

Using that $\frac{\Delta I}{I}$ & $\frac{\Delta\beta}{\beta} << 1$ and that $I_1 R_1 = I_2 R_2$, we get

$$V_{off} = \frac{1}{2}\sqrt{\frac{2I_1}{\beta}} \left(\frac{\Delta R}{R} + \frac{\Delta\beta}{\beta}\right) - \Delta V_t$$

Hence,

$$\sigma^2\left(V_{off}\right) = \left(\frac{V_{gs} - V_t}{2}\right)^2 \left(\sigma^2\left(\frac{\Delta R}{R}\right) + \sigma^2\left(\frac{\Delta\beta}{\beta}\right)\right) + \sigma^2\left(\Delta V_t\right)$$

Using the 130nm UMC parameters for mismatch, we have $\sigma\left(\Delta V_t\right) = \frac{A_{VT}}{\sqrt{WL}}$ where $A_{VT} = 4mV\mu m$, $\sigma\left(\frac{\Delta\beta}{\beta}\right) = \frac{A_\beta}{\sqrt{WL}}$ where $A_\beta = 1.5\%\mu m$. Neglecting resistor mismatch and using these values along with DC operating point information and sizes, we get $3\sigma(V_{off}) = 30$ mV. So the offset will be less than or equal to 30 mV with 99.7% probability. But if the offset voltage is actually 30 mV, then the output will be saturated and will not change upon arrival of the pulse. Even if the offset is anywhere greater than 12 mV (23% probability), the output will saturate to $V_{dd} = 1.2$ V. Clearly, we cannot use this amplifier as it is. We need to cancel the effect of the offset. There are a few alternatives, and we discuss them and the chosen method is explained thereafter.

One method is to use AC coupling capacitors at the output which remove the DC offset and only the AC signal is passed. While this does remove the effect of the DC offset, the problem here is that if one of the intermediate stages goes into saturation, the transient signal may not be seen at the output at all. To correct this, we need to put coupling capacitors between each stage. DC bias can be applied at each stage using a resistive divider. The corner frequency of the coupling network should be smaller than the lower frequency of interest. This relatively low frequency results in a large capacitance value between each stage.

So, this method, although attractive in its simplicity, takes up a large amount of area. Another alternative is to use a feedback circuitry which senses the output voltage difference for zero input and changes the input DC voltage so as to bring the output voltage closer to zero. Here also, we have two alternatives: a completely analog feedback circuit or a digital circuit. In the analog alternative, the output is amplified and a voltage controlled current source along with a resistor is used to change the input DC voltage so as to bring the output closer to zero. The drawback of this scheme is that is consumes a large amount of power. The digital alternative as explained below is both low power and compact.

The offset cancellation circuitry implemented in our design is shown in Fig. 3.6 and Fig. 3.7.

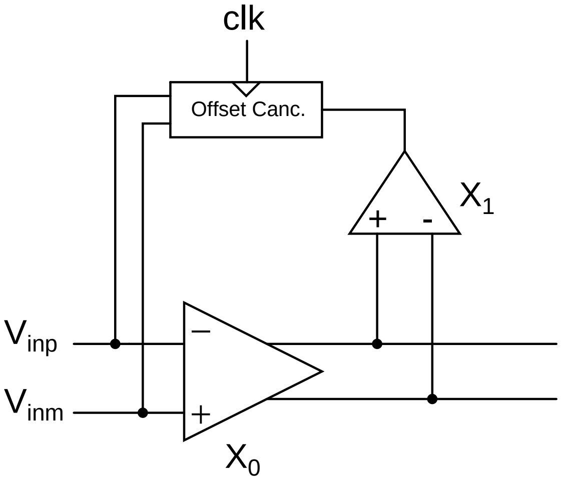


Figure 3.6: Offset cancellation

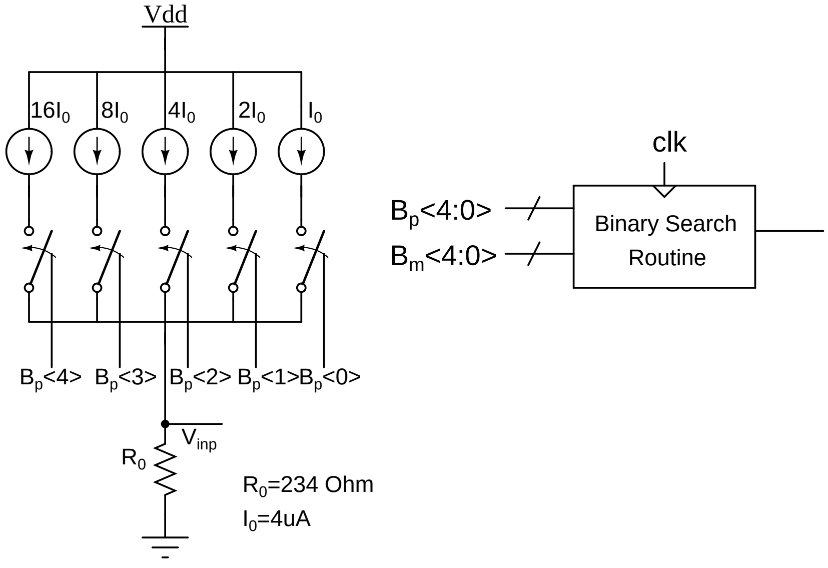


Figure 3.7: Offset cancellation circuitry

The offset cancellation circuitry aims to change the input DC value depending on the offset. If the differential DC output voltage is positive (negative) for zero input, we increase the DC voltage at the negative (positive) input of the amplifier, so as to make the differential DC output less positive (negative). We use a sequential digital circuit to increase the DC voltage at the desired terminal in steps so as to finally converge at a voltage which makes the output difference as close to zero as possible. A comparator is used to determine if the differential output voltage is positive or negative. In each clock cycle we determine whether the positive input or the negative input of the amplifier has to be increased. In each cycle the amount by which voltage of any input node is increased is reduced by a factor of 2. The voltage at the amplifier input is changed by changing the amount of current being passed through the resistor R_0 shown in Fig. 3.7 which sets the DC operating point at the input. Since our target is to bring down the input referred offset to less than 1 mV (3σ value), we choose 5 bits of resolution for the binary search routine. This gives us an offset of 3σ =0.94 mV. The offset cancellation circuitry runs at a slow clock frequency of $10kHz$ since the causes for offset are mainly process mismatch and temperature, which are either time invariant or vary at a much lower frequency.

I_0 and R_0 are chosen so that the full scale voltage corresponds to $3\sigma(V_{off})$=30 mV. So, $I_0R_0 \simeq$1 V. If we fix the LSB of current as I_0=$4\mu A$ we get $R_0 = 250$ Ω. The binary search routine was written in Verilog and synthesized using *Design Vision* and laid out using *Encounter*. It occupies an area of 80μm× 40μm. For testing purposes, we apply an external offset to the positive terminal and vary it from -30 mV to 30 mV and determine the differential output voltage. The input referred offset is calculated by dividing this by the gain of the amplifier. We find that the input referred offset has been reduced to a maximum value of around 1 mV, which was the resolution targeted.

3.2 Phase locked loop

The DLL and the TDC in this design use a 250MHz clock. The complete system has to be integrated with digital time stamping circuitry and other digital blocks. These blocks are implemented on an FPGA and run at a clock frequency of less than 50MHz. So, the clock input to our system can be expected to be at the most 50MHz. So, to be able to run the DLL and the TDC, we need to increase the clock frequency to 250MHz. For this purpose, we need a PLL which increases the clock frequency by a factor of 5. The basic blocks in a PLL are shown in Fig. 3.8. The PFD and the charge pump can be the same as used in the DLL. We need to design a frequency divider and a voltage controlled oscillator (VCO).

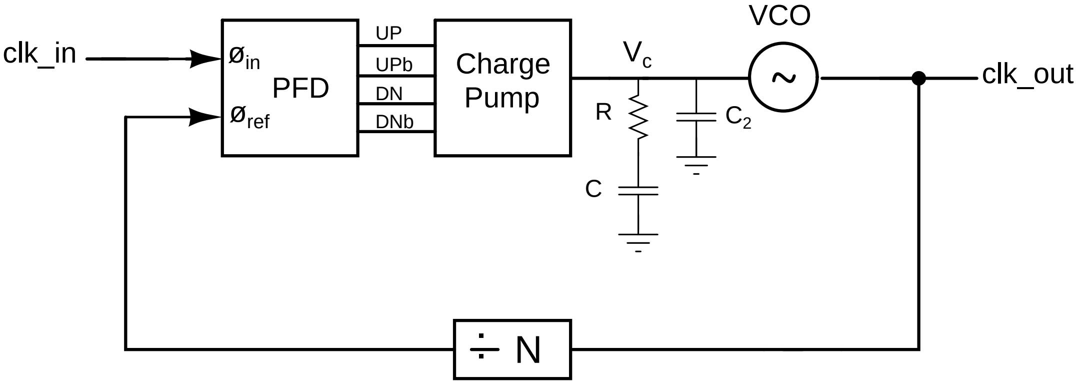


Figure 3.8: Phase locked loop

3.2.1 Voltage controlled oscillator

As mentioned above, the PFD and the charge pump are the same as that used in the DLL. The current input of the charge pump will be changed as needed. The VCO generates an output square wave whose frequency is proportional to the input control voltage. The VCO is implemented as a ring oscillator with odd number of inverter stages connected in a loop. Since each voltage controlled delay unit discussed in Section 2.3 consists of two inverting stages, we cannot use an integral number of such delay elements to realize the VCO since the ring oscillator must have an odd number of inverting stages. So, we keep the last stage as half of the voltage controlled delay unit i.e. comprising of only one inverting stage. A total of 15 delay elements and half an element at the end are connected in a loop as shown in Fig. 3.9. If they delay of each delay element is T, the frequency of oscillation is given by

$$f_{VCO} = \frac{1}{31T}$$

We want this frequency to be $250MHz$, hence giving delay of each delay stage as $T = 129ps$. A parameter to quantify for the VCO is its gain, k_{VCO} given by $k_{VCO} = \frac{\delta f_{VCO}}{\delta V_c}$. Simulations show $k_{VCO} \simeq 800MHz/V$.

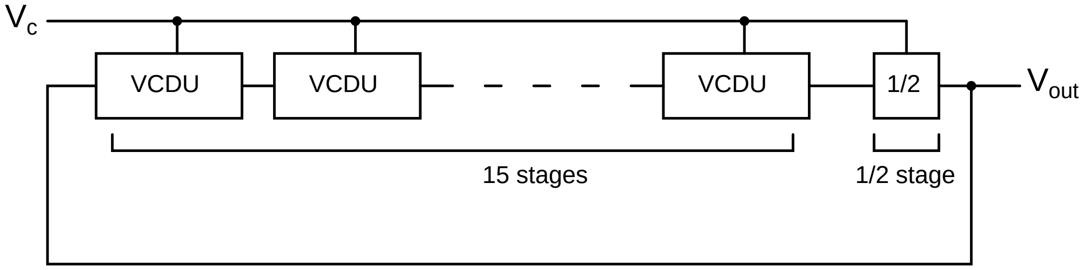


Figure 3.9: Voltage controlled oscillator

3.2.2 Frequency divider

The frequency divider must divide the frequency of the VCO output by a factor of $N = 5$ and output a square wave with the divided frequency. If N is a power of 2, the implementation is very simple and it just comprises of $log_2 N$ flip flops in series. Even when N is not a power of 2 but still an even number, a Johnson counter with $N/2$ stages can be used as a frequency divider. When N is odd, as is the case with our design, we need a little bit more elaborate circuitry. A direct approach for all N is to use a counter to count $\lfloor N/2 \rfloor$ cycles of the input clock and set the output clock to 1 during this duration and keeping it zero for the remaining $N - \lfloor N/2 \rfloor$ cycles. A simpler implementation was used in this design. Note that for odd N, the duty cycle of the output will not be 0.5.

As shown in Fig. 3.10 the frequency divider consists of N D flip flops in series. We consider here the case with $N = 5$.The 5-tuple of their outputs is initialized to $Q_5, Q_4, Q_3, Q_2, Q_1 = 100...0$ with only one of the outputs being one. In each clock cycle, the 1 is shifted through the flip flop chain in a cyclic manner. For N=5, the states are 10000 → 01000→00100→00010→00001→10000 and so on. We see that any set of three consecutive outputs is 1 for three cycles and zero for two cycles and repeats with a period of 5 cycles. So, we choose Q_4, Q_3 and Q_2 since Q_4 goes high with the first clock cycle and hence the output clock will start with the start of the input clock. So, the output clock is assigned as $Q_4{+}Q_3{+}Q_2$, where addition stands for boolean addition. The output clock has a 3:2 duty cycle. This is not a problem since the PLL is insensitive to the duty cycle of the fed back clock signal and corrects only for the total time period of clock.

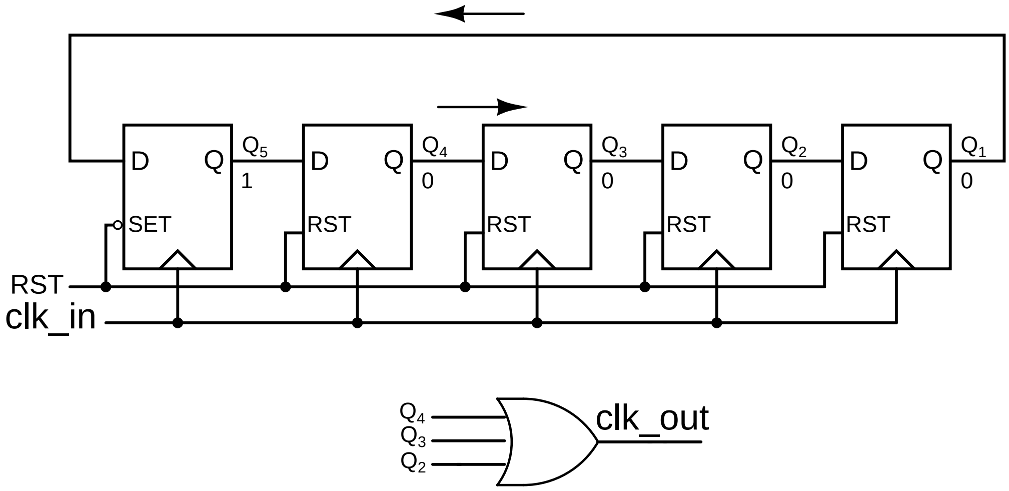


Figure 3.10: Frequency divider

3.2.3 Loop filter

The loop filter as shown in Fig. 3.11 is used to ensure the stability of the PLL. Unlike a DLL, the PLL is not a first order system and has multiple poles. So, we have to ensure stability by making sure that the phase transfer function looks like a first order system near the unity gain frequency. For this design, we have f_{ref}=50 MHz, $f_{out} = 250MHz$, k_{VCO}=800 MHz/V. The loop bandwidth should be significantly less than f_{ref} and we choose the bandwidth to be $\frac{f_{ref}}{50}$ = 51MHz.

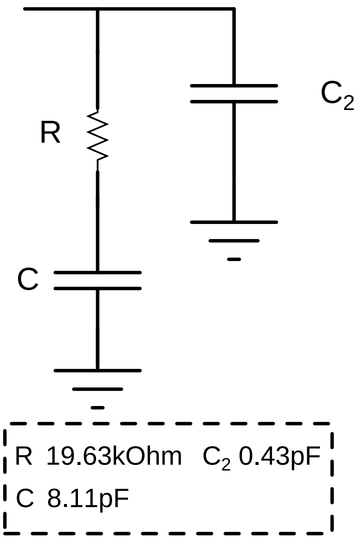


Figure 3.11: Loop filter

If the charge pump current is given by I_0, then we have,

$$w_{u,loop} = \frac{I_0 R k_{VCO}}{N} = 2\pi\times5\times10^6 rad/s$$

. Choosing $I_0 = 10\mu A$, we get $R = 19.63k\Omega$. The zero which ensures that the system looks like a first order system near the unity gain frequency is at a frequency $z_0 = \frac{1}{2\pi RC}$ Hz. This zero should be at least a few times lesser than the loop bandwidth. We choose it to be 5 times smaller, and we get the value of capacitance as $C = 8.11pF$. The closed loop bandwidth of the PLL is 1 MHz and the unity gain frequency is 25MHz. The reference frequency of 50MHz is hence attenuated by a factor of 4. To further attenuate the reference feed through, we add an additional pole between the dominant pole and the unity gain frequency. Somewhat arbitrarily, we choose the location of the new pole to be at 15 MHz. So, we get $\frac{1}{2\pi R\frac{CC_2}{C+C_2}}$ =15MHz. From this we get, C_2 =0.57 pF. This is comprised of both the explicit capacitor and the input capacitance of the VCO. Using a capacitance density of $10fF/\mu m^2$, we get the input capacitance as 140 fF. So, we need a capacitance of C_2 =0.43 pF.

Note that a single PLL/DLL feeds clock and control voltages to all the systems on a given chip. So, the area of the PLL/DLL is effectively divided amongst all systems on the chip. The PLL was simulated for an input clock of 50 MHz and the input output waveforms are shown in Fig. 3.12. Note that the time axis of both have been shifted so as to align the rising edges of the input and output clock. This can be done since the phase offset of the PLL is not of concern to us.

We can see that the output frequency is 5 times the input frequency i.e. a time period of 4 ns. We also plot the time period of the output as it finally locks to a value of 4 ns. The plot is shown in Fig. 4.1.

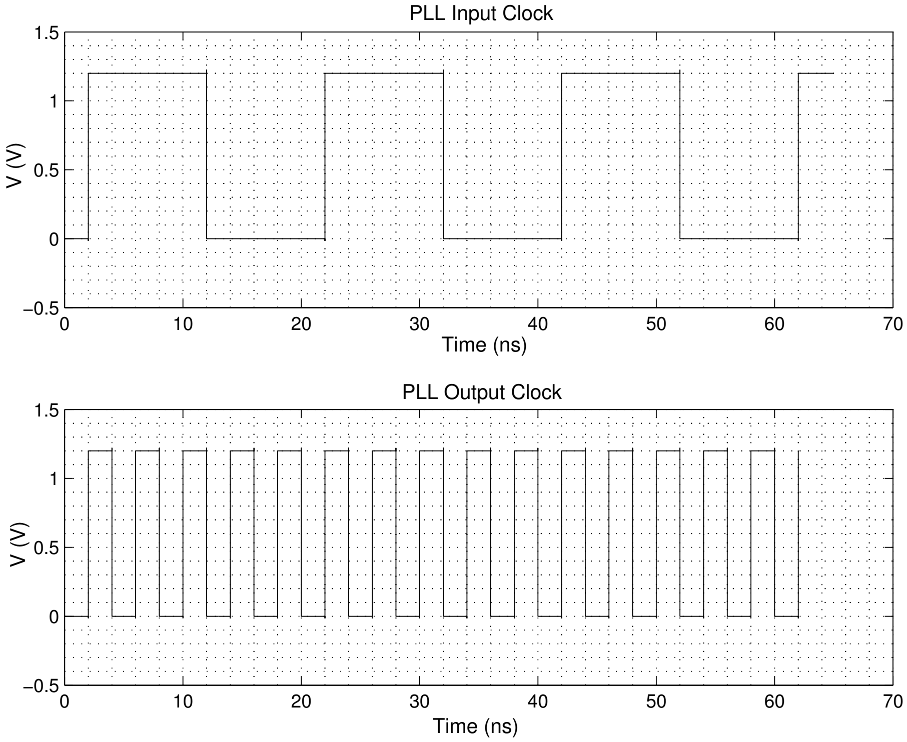


Figure 3.12: Input and output waveforms of the PLL after locking.

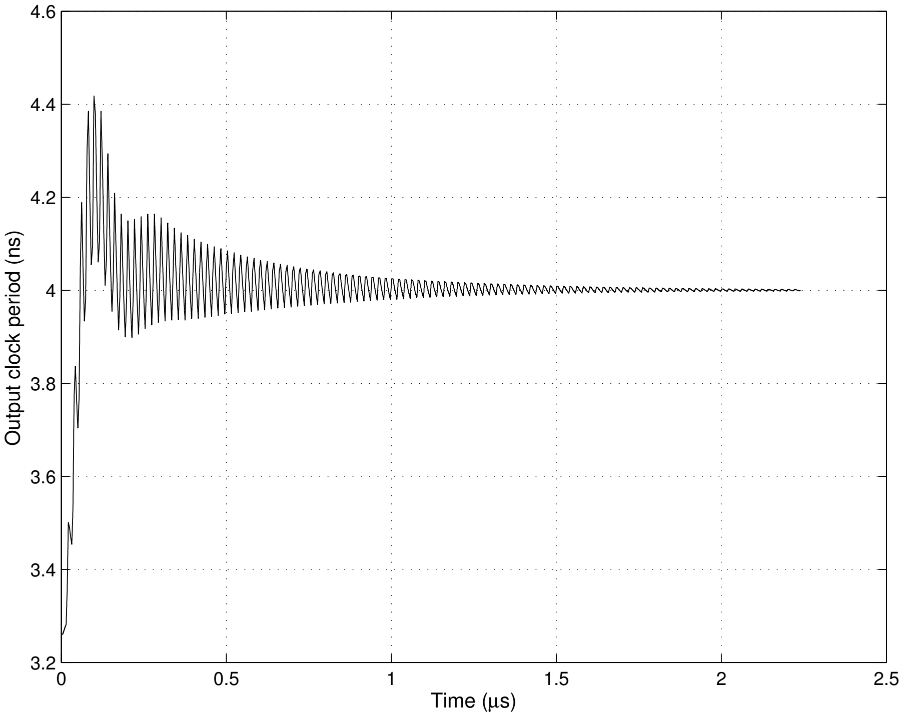


Figure 3.13: Locking of output time period.

CHAPTER 4

Layout & Simulation Results

Layout of the complete design was completed and post layout simulation results are presented here. Care had to be taken in the layout of the TDC and the DLL because the delays depend on the parasitic capacitances due to routing of metal lines. The layout was made as symmetric as possible w.r.t the delay chain. Fig. **??** shows the layout diagram of the TDC, DLL and the amplifier circuitry.

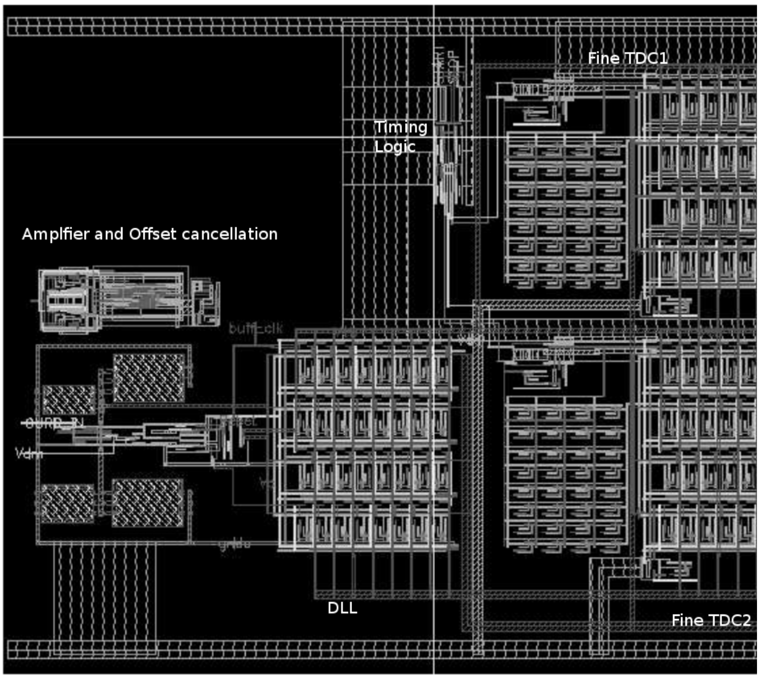


Figure 4.1: Layput diagram. X axis is 1 mm long.

We begin by presenting the results of the DLL. The delays of each delay element are plotted in Fig. 4.2. We can see that the maximum deviation is about 0.02 LSB and the variance is less than 0.003 LSB. So the delay elements are almost identical till post layout stage. Mismatch between devices will of course increase the variance of delays according to calculations in Section 2.5.

The DLL has a steady state offset because the charge pump needs a sufficiently high voltage for some minimum amount of time. If the locking offset is too high, the delay of each element will no lock to the desired value of 125 ps. We see in

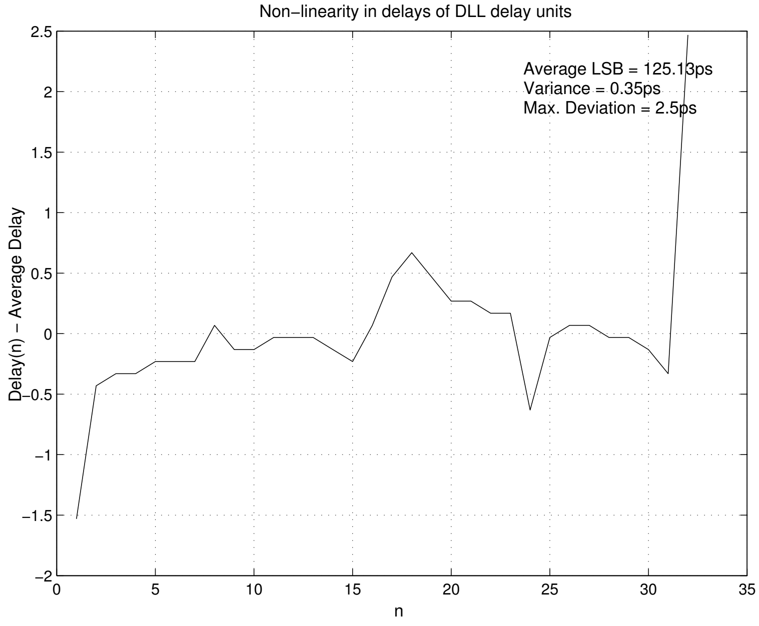


Figure 4.2: Delay of each delay element of the DLL after locking.

Fig. 4.3 that the delays lock to exactly 125 ps for tt and ff corner whereas for ss corner there is some locking offset. Some characteristics of the DLL are tabulated in Table 4.

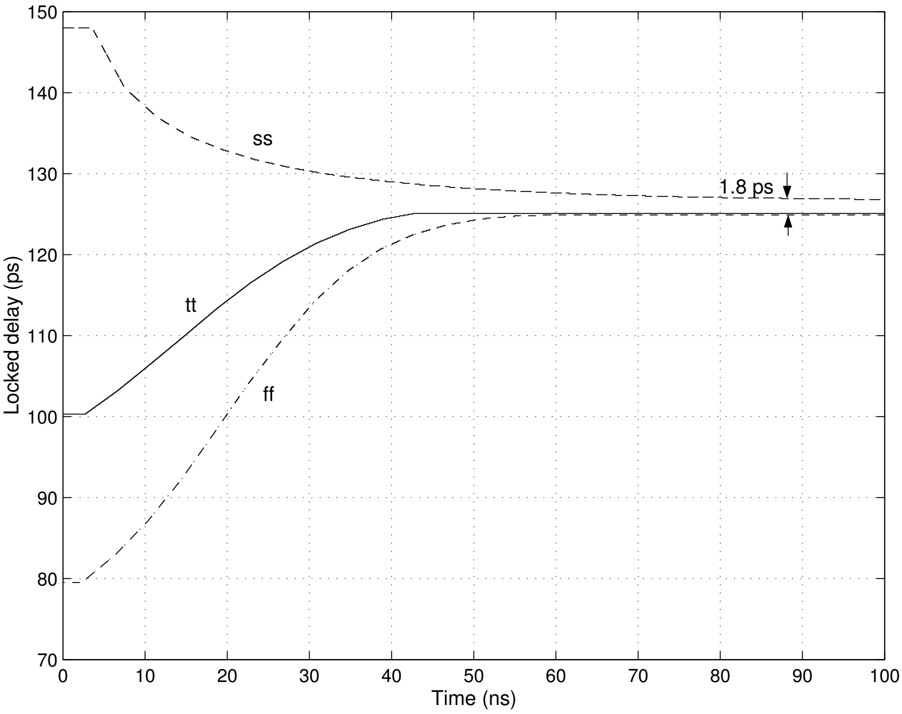


Figure 4.3: DLL locking at different process corners.

Table 4.1: DLL Characteristics

Parameter	Value
Input clock	$250MHz$
Average delay per stage	$125.12\,ps$
Variance of delay	$0.35\,ps$
Power consumed	$2\,mW$
Area	$0.12\,mm^2$
Peak to peak jitter	$2.1\,ps$
Locking offset	$0.12\,fs/125\,ps$

The TDC was tested by varying the time interval between its inputs and plotting the output. The input time difference was varied from 0 to 4 ns in steps of 6 ps. Hence a total of about 667 points were collected. The output is shown in Fig. 4.4 along with input which is just a straight line with slope 1. The output code, as expected has more or less uniform bin widths. Some characteristics of the TDC are tabulated in Table 4.

Table 4.2: TDC Characteristics

Parameter	Value
Resolution	$125\,ps$
Range	$81\,\mu s$
Area	$0.24\,mm^2$
DNL$_{max}$,INL$_{max}$	$0.1\,LSB$,$0.4\,LSB$

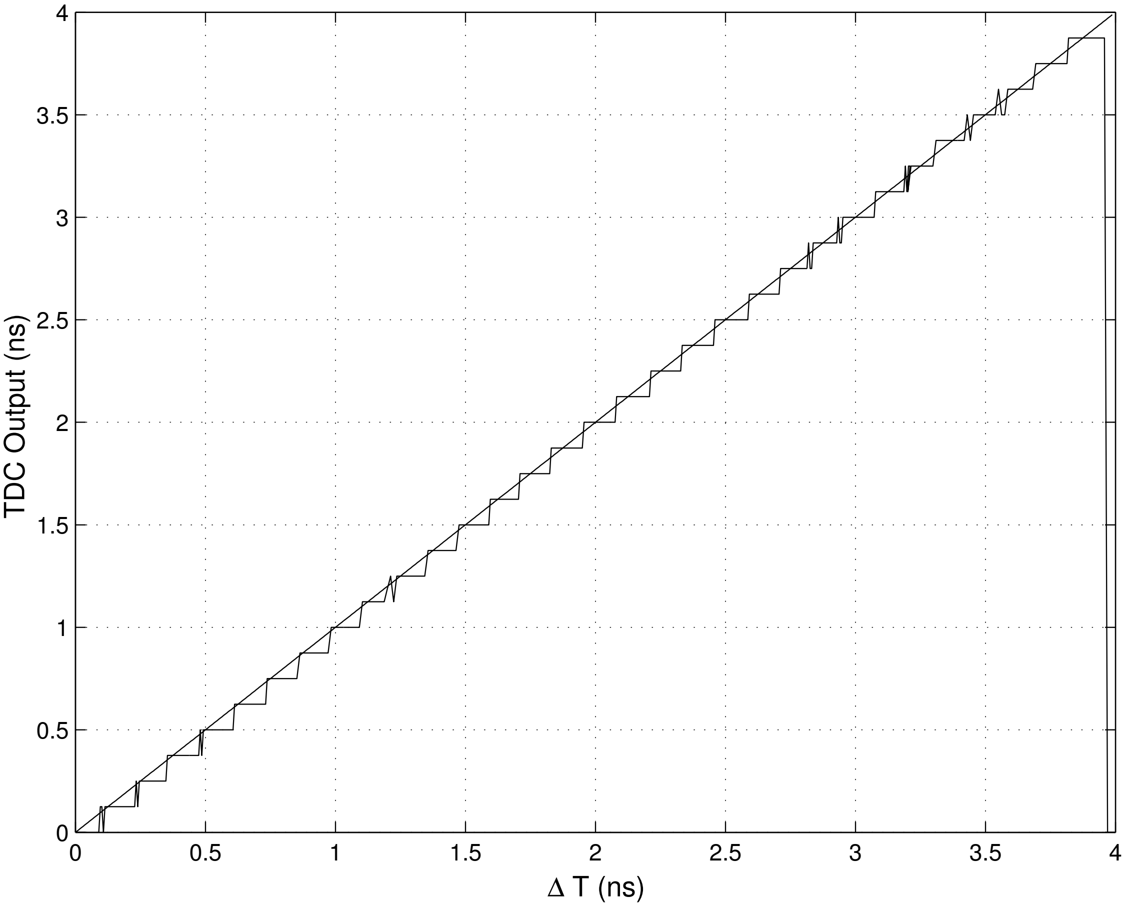


Figure 4.4: Output code of the TDC.

CHAPTER 5

Conclusions and Future Work

In this project, a front end circuitry for neutrino detectors was designed. Although the target was this particular application, the designed systems can be used for other purposes with little modifications. The DLL, TDC, amplifier and the latching circuitry work correctly in the post layout simulations. The PLL was simulated at transistor level and works as expected. Offset cancellation circuit for the amplifier was also designed. The digital back end was synthesized and routed from Verilog code using automated CAD tools.

We also developed a model for the delay of a delay unit. This was used to calculate the variance in delays due to random mismatch between devices. Simulations confirmed the correctness of the model. This model helped us to quantify the DNL and INL of the TDC.

5.1 Future work

While the system designed here fulfills many requirements of the target application, there are still some additions that can be made to increase the functionalities. The output code of the TDC has glitches due to the metastability of the flip flops, which was reduced to a lot of extent using modified flip flops. Still, some way around this can be found to reduce the non-linearity in the TDC output code. The possible directions for future work are briefly explained below.

5.1.1 Single TDC architecture

While two TDCs were used here to measure the total time, we can do the same measurement by using only one TDC and by using it twice. Appropriate digital logic has to be designed to ensure correct operation for all cases of input time interval (i.e. <1 clock cycle or more). A design was developed on paper and can be taken ahead and simulated. This will not only reduce area by half but will also probably improve the linearity of the output code.

5.1.2 Programmability

The gain of the amplifier and the threshold levels of the latch in this design are not under external control. If they can be made digitally programmable, it would give the user more flexibility. The input impedance of the system can also be made programmable. All these features will probably increase the parasitic capacitances at many nodes. The actual design might have to be appropriately changed to meet the speed and accuracy criterion. It would also be a good feature if the programming data can be sent serially through one pin.

5.1.3 TDC calibration and testing scheme

While the TDC designed here has constant resolution independent of PVT variations, routing drops of the control signals might cause the resolution to be different for different TDCs. If a method can be found to calibrate the TDC at the start so as to determine the average LSB size, it would reduce errors in measurement. A self calibrating architecture would be really helpful. Also, if we can provide for testing the circuit by using signals generated from the circuit itself, it would avoid the need for high frequency measurement equipment.

5.1.4 Integration of ADC

The amplifier output can be fed to an ADC in order to digitize the output waveform. This will help us to store the shape of the input spike. A high speed and low power ADC is required for this purpose.

REFERENCES

1. **Abas, M.**, **G. Russell**, and **D. Kinniment** (2007). Built-in time measurement circuits: a comparative design. *Computers & Digital Techniques, IET*, **1**(2), 87–97.

2. **INO** (2008). India-based neutrino observatory ino. Technical report. URL `www.imsc.res.in/~ino/OpenReports/INOReport.pdf`.

3. **Li, G.** and **H. Chou** (2007). A high resolution time-to-digital converter using two-level vernier delay line technique. *Nuclear Science Symposium Conference Record*, **1**(11), 276–280.

4. **Zhou, J.**, **J. Liu**, and **D. Zhou** (2001). Reduced setup time static d flip flop. *Electronics Letters*, **37**(5), 279–280.