Chapter 1
INTRODUCTION

In most experiments in which time is involved, it is necessary to develop estimates of time, frequency and measurement errors from a series of time measurements between the clocks of a number of computers and ancillary devices interconnected by some kind of computer network. However, time is not a physical quantity, such as mass, nor can it be measured relative to an absolute frame of reference, such as velocity. The only way to measure time in our universe is to compare the reading of one clock, which runs according to its own timescale, with another clock, which runs according to a given timescale, at some given instant or epoch. The errors arise from the precision of time comparisons and the accuracy of frequency estimates between the timescales involved.

The usual data collected during a performance run of some experiment might include time offsets, time delays, frequency offsets and various error statistics. While time offsets between two clocks can be measured directly; frequency offsets can be estimated only from two or more time offsets made over some time interval in the experiment. In practice, a sequence of time comparisons can be performed over the lifetime of the experiment and the instantaneous frequency estimated either in real time with a recurrence relation, or retrospectively with a polynomial fit to the data. Estimating time and frequency errors in real time has been studied by a distinct subspecies of physicists who have made a career of the technology involved. Various

means including autoregressive models, Kalman filters and simple weighted-average algorithms are used extensively by national standards laboratories to model cesium-clock ensembles. These techniques have been adapted to computer network and transmission engineering problems as well. This memorandum explores issues in performing experiments of this type and summarizes various techniques found useful in practice.

The time-to-voltage converter (TVC) discussed in this thesis has an application in laser radar. In laser radar a short light pulse is sent to the target, and the time between sending the pulse and receiving the reflected pulse is measured. Laser rangefinding devices are used in various industrial measurement applications such as measuring the profile of hot surfaces in steel factories and dimensions of ship blocks in shipyards. Time-to-Voltage

Converter (Time interval measurement) with nanosecond accuracy has many applications, including lifetime measurements in nuclear physics, laser time-of-flight measurements, frequency and phase measurements in telecommunications. Basically, time-to-voltage conversion can be obtained using short time interval measurements. This thesis will discuss time measurements with 6-25 ns resolution and low power consumption. There are number of ways to implement time-to-analog voltage converter (time interval measurement) but the goal of this research is to find the simplest possible method for conversion. So, it is a trade off between getting a high resolution and a simple implementation circuit. Another aim of this project is to get familiar with custom design flow and different industry standard state-of-the-art EDA (CAD) tools which can be very useful in designing a large system with millions of transistors in it.

Chapter 2 describes the main idea of Time-to-Voltage Converter and it also explains some of the very important issues that should be taken care of while implementing TVC (Time-to-Voltage Converter).

Chapter 3 is about the very important component called Width Generator used in the TVC circuit.

Chapter 4 explains the circuitry of TVC in detail.

A conclusion chapter basically summaries the thesis and also gives some discussions on the future design.

Chapter 2
General Concept

Parot and Arai: Principal methods of Time Interval Measurement in the subnanoscrond regime were reviewed by **Parot.** A Time-to-Digital converter technique, in which input signals are recorded to memory cells at a specific time interval, has been developed by **Arai,** for use in nucleon time-of-flight measurement. However, its measurement resolution is limited to around 1 ns by the time resolution of delay lines used in the circuit. In principle, the advantage of near picosecond resolution is provided by a time-to-voltage conversion (TVC) technique, in which a converted voltage can be stored in analog memory.

Many TVC based time interval measuring instruments and systems have already been developed in several application fields.

Concept of Time-of-Flight (TOF) Laser Range-finding:

The concept of Time-to-Voltage Converter (TVC) presented here can be used in Laser Radar. As shown in Figure 2.1, In a laser radar a short light pulse is sent to the target and the time between sending the pulse and receiving the reflected pulse is measured. This is called Time-of-Flight Measurement (TOF). Laser rangefinding devices are used in various industrial measurement applications such as measuring the profile of hot surfaces in steel factories and dimensions of ship blocks in shipyards. Typically, the measured distances are from some tens of meters to hundred meters and the required resolution is in mm/cm-class.

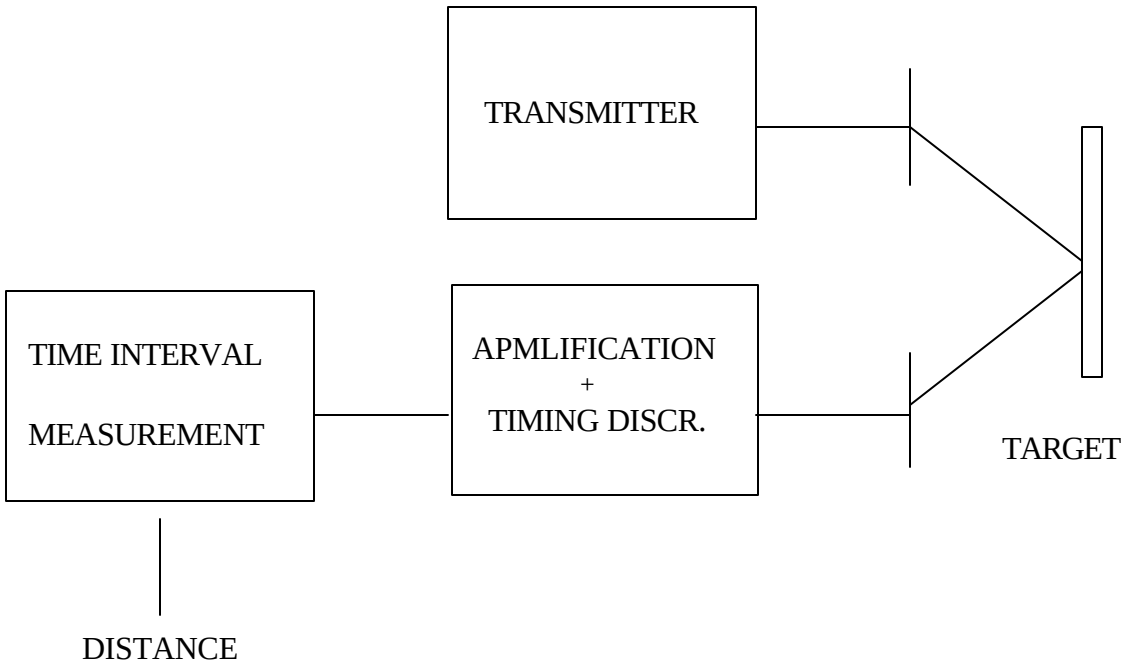


Figure 2.1: Time-of-Flight Laser Range Finder.

The Time-of-Flight laser range-finding method can be used to solve a variety of distance measurement problems in industry, e.g., the measurement of levels in silos, the automatic control of robots and manipulators, and dimensions control in mechanical and construction industries. Time-of-Flight distance measurement involves calculation of the distance of an object from the flight time of a light pulse, e.g., from a laser diode transmitter, to the object and back to the receiver. In industrial application the measurement range is usually from some tens of cms to a few hundred meters, which means that the duration of the time intervals to be measured are from less than 1 ns to a few microseconds.

The basic unit of a time-of-flight distance measurement device is that responsible for measuring the time interval, in which accuracy and measurement time are important parameters. The measurement accuracy required in industry depends on the application, being generally from about 10 ps (distance 1.5 mm) to about 100 ps (1.5 cm). For stable objects the measurement time can be several seconds, but for scanning measurements or moving objects it has to be considerably shorter.

Example of StopWatch:

A time interval measurement is a measurement of the elapsed time between some designated START phenomena and a later STOP phenomena. This is in contrast to real-time observations (time of day) used in our day-to-day living to schedule meetings or transportation, in astronomical observations and for celestial navigation among other things. One might make a time interval measurement with a mechanical stopwatch as when timing a track meet or other sporting event or in making time and motion studies. With increased speed of the timed object as when timing automobiles or airplanes the timed interval becomes shorter and shorter until the human factor involved in determining when to start and when to stop the measuring device, a stopwatch or clock for instance, begins to introduce significant error. Mechanical, optical, or electrical transducers or a combination of all were developed to reduce this error. Finally with advances in many scientific fields, mechanical and electrical time measurements were required which were beyond the resolution of a mechanical stopwatch. This led to the development of a time interval measuring device, in essence an electronic stopwatch.

In a time interval measurement, clock pulses are accumulated for the duration the main gate is open. The gate is opened by one event, START and closed by the other, STOP. Minimum time measurement is much less than possible with a stopwatch. Also resolution and accuracy are much greater than attainable with a stopwatch.

Some typical time measurements that might be made are:

- Propagation delay of integrated circuits
- Radar Ranging
- Nuclear and Ballistic Time of Flight
- Pulse Measurement
- Cable Measurement
- Delay Line Measurement

Other Applications:

Time interval measurement can also be made on any physical phenomena that can be translated into appropriate electrical signals. Transducers such as photo electric cells, magnetic pickups, strain gauges, micro-switches, bridge wire systems, or thermistors can be used to translate physical events into the electrical start and stop signals required for a time interval measurement.

Important Issues:

The resolution of conventional time interval technique is determined by its "clock" frequency. A clock frequency of 1MHz gives 1microsec resolution, 100 MHz gives 10 ns resolution, 500 MHz gives 2 ns resolution and so on.

Three important specifications are sometimes overlooked when considering time interval measurement.

1. Minimum Time Interval.
2. Minimum Dead Time.
3. Minimum Pulse Width.

1. The **minimum time interval** or minimum range specification is the minimum time between start and stop pulses. For a single shot (the time between a single pair of start and stop pulses) measurements this time much be one or more clock periods.
2. The **minimum dead time** is the time from a stop pulse to the acceptance of the next start pulse. Typical dead time specification is 150 ns for 10 MHz clock frequency. Since this is single shot measurement, this specification is ignored.
3. The **minimum pulse width** is the shortest pulse width for STOP and START signal. The typical minimum pulse width for a 50 MHz clock frequency is 10 ns or the period of half a cycle.

Metastabilty in Time-Interval Measurement:

The START and STOP inputs (input for Width Generator) are asynchronous with respect to the circuit CLK so there is a possibility of metastability in the flip-flops clocked by CLK. If a

START or STOP pulse arrives near the rising edge of CLK, the propagation delay of the corresponding flip-flop can increase and the time interval can be erroneous.

To reduce the chances of metastability, it is a good practice to make asynchronous inputs synchronous to the circuit clock but by doing that you are slowing down the speed of the inputs but TVC explained in this thesis runs at 100 MHz and that is why time required to convert asynchronous input into synchronous input (delay) can be ignored. A Width Generator, one of the most important blocks of the TVC circuit is an interface block to the outside world. A Width Generator takes asynchronous inputs and it deals with metastability issues and that is why rest of this chapter talks about metastability and synchronization. Synchronous circuits are generally slower than the asynchronous circuits. An asynchronous circuit is one whose input, states and outputs can sampled or changed independently of nay clock reference. Asynchronous circuits lie at the heart of every synchronous circuit. The basic R-S latch is an asynchronous circuit, whereas the edge-triggered D flip-flop, constructed from several such latches, is synchronous. The J-K master/slave flip-flop falls into something of a gray area because of its onescatching behavior.

Synchronous versus Asynchronous Inputs: Even a supposedly synchronous circuit like the D flip-flop can have asynchronous inputs such as PRESET and CLR. These set the output (PRESET) and reset it (CLR) whenever they are asserted, independent of the clock. Synchronous inputs are active only while the clock edge or level is active; at all other times, changes on the inputs are not noticed by the memory element. Asynchronous inputs, on the other hand, take effect immediately and are independent of the clock. Glitches make asynchronous inputs extremely dangerous and should be avoided whenever possible. A glitch

on the logic that drives an asynchronous input can cause a flip-flop to be cleared or set when no state change was called for. It is good design practice to choose components that have only synchronous inputs.

The Problem of Asynchronous Inputs: Sometimes asynchronous inputs cannot be avoided-for example, when a signal must pass from the outside world into the synchronous system. An example must be reset signal, triggered by an operator pressing a push-button. It is particularly dangerous to fan out an asynchronous inputs to many points in the clocked systems: if the input changes close to the clock event, it may be seen at some flip-flop but not others, leading to an "impossible" state.

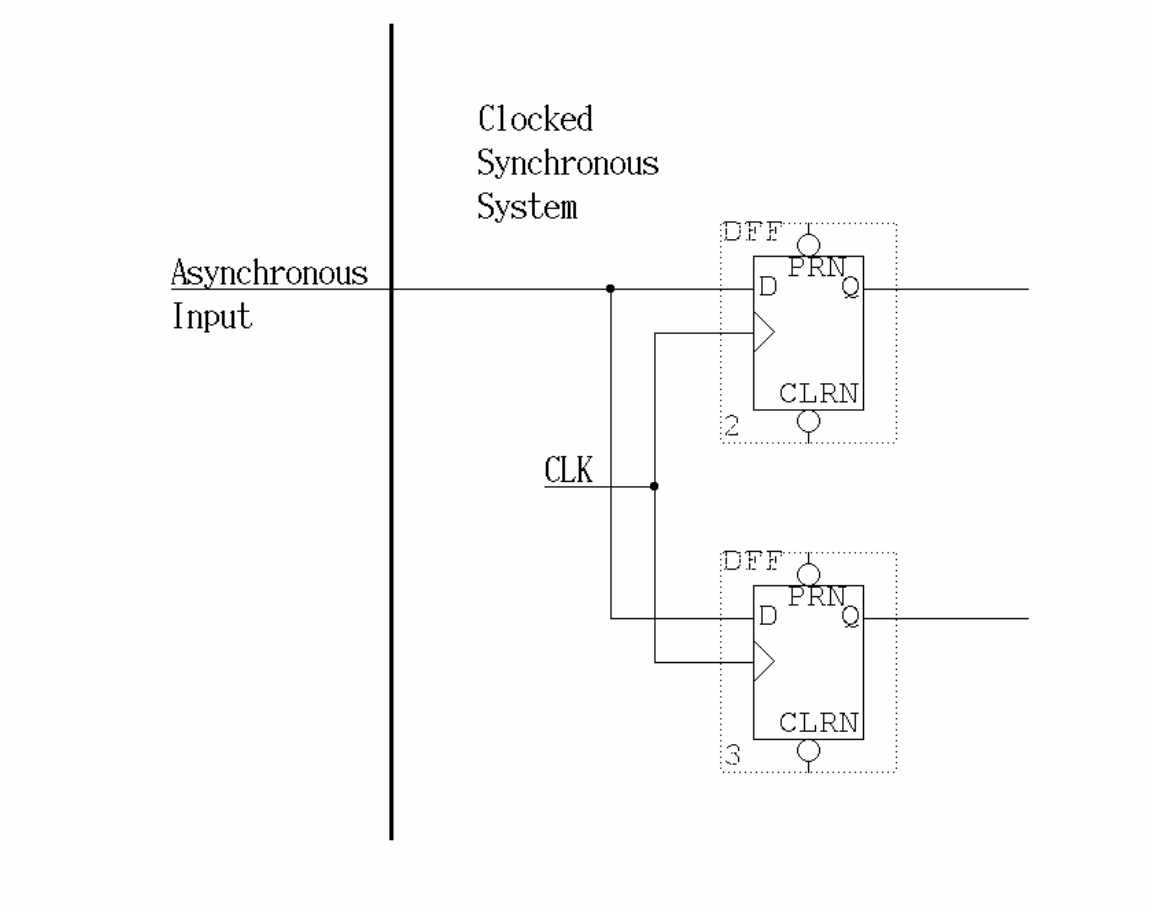


Figure 2.2 Incorrect Fan-out to Multiple Flip-flops

An incorrect circuit for handling an asynchronous input is shown in Figure (2.2). Two positive edge-triggered D flip-flop are driven by the same asynchronous inputs. One would expect both devices to hold the same state, yet because of different wiring and other internal delays, one flip-flop is set while the other remains reset. The assumption that both flip-flops hold the same state is now invalid. The timing waveform in Figure (2.3) tells the sad tale.

The better way to deal with and asynchronous signal is to synchronize it to the clocked system. This synchronization is accomplished by placing a single D flip-flop between the input source and the rest of the system. The proper circuit is shown in Figure (2.4). The

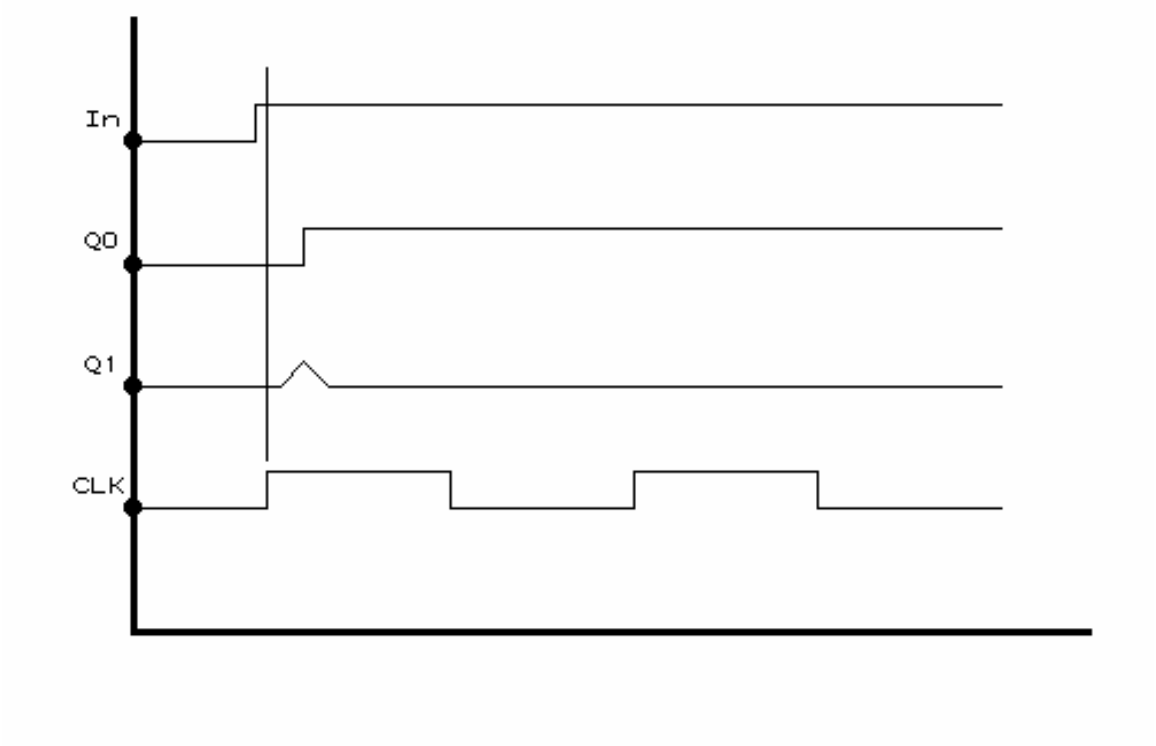


Figure 2.3 Outputs of two flip-flops when their input changes close to the clock edge.

flip-flop's output Q will change only in relation to the clock and can be properly fanned out and distributed to other points in the circuit in a synchronous manner.

Metastability and Synchronizer Failure:

What if setup and hold times of the synchronizer flip-flop are not met by the asynchronous signal? Under such condition output of the synchronizer is undefined. Normally, choosing the synchronizer flip-flop from the fastest available logic family, with the shortest possible setup and hold times can minimize this problem. Unfortunately, the problem can not be eliminated completely. The behavior of this flip-flop is worse than unpredictable: it can result in input values injected into the system that cannot be interpreted as either a '1' or '0'.

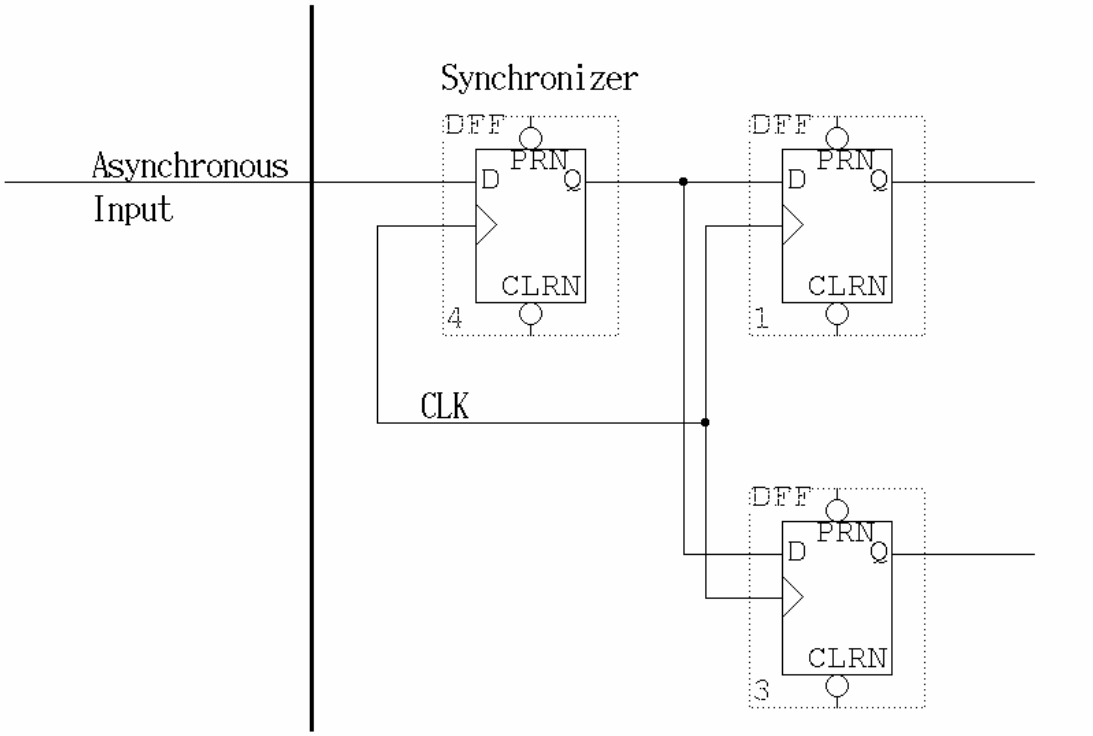


Figure 2.4 Correct Synchronization

Figure (2.3) gives a hint of this: Q1 exhibits a partial transition that falters back to '0'. This "in-between" voltage is called the metastable state. Under the right (or wrong) conditions, the flip-flop can hang in this state indefinitely, a so-called synchronizer failure.

An Analogy for Understanding Metastability:

Figure 2.5 provides a useful analogy for describing the nature of synchronizer failure. The states of the flip-flops are represented by two flat regions separated by a steep slope. The flat parts represent the stables states, logic 0 and logic 1. For the purpose of this analogy, assume the state of the flip-flop by the ball in one plateau or the other. To change the

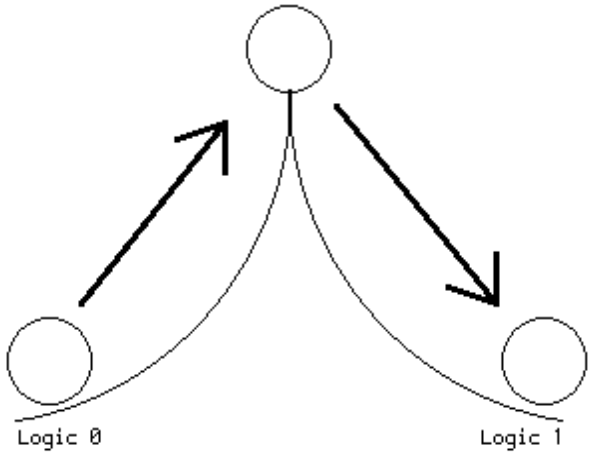


Figure 2.5 Analogy for explaining synchronizer failure

state, energy must be exerted to push the ball up and over the slope to the other side. When setup and hold time constraints are met, there is sufficient energy to cause the state change. If these constraints are not met, three cases are possible, two that yield acceptable behavior and one that does not. In the first case, there is not enough energy to get the ball over the summit, and it rolls back and the state is not changed. In the second case the energy might be just

enough to get the ball over the top and state changes from ‘0’ to ‘1’. Both of these are acceptable outcomes. However, there is a small probability that just enough energy is imparted that the ball ca be push up the slope but remains tottering at the top, not able to return to one or the other side of the stable states. This is the metastable state. Theoretically, a flip-flop can remain in the metastable state. However, thermal disturbances and asymmetries in signal delays within the transistor-level implementation of the flip-flop usually make is settle in one state or the other in some period of time.

Reducing the Chance of Synchronizer Failure:

The only way to recover from synchronizer failure is to reset the entire circuit. While the probability of synchronizer failure can be made small, it can never be eliminated as long as there are asynchronous inputs. One way to reduce the probability of synchronizer failure is to lengthen the system’s clock period. This gives the synchronizer flip-flop more time to make its decision to enter a stable state. The longer the clock period, the lower the probability of failure. Unfortunately, this is not an adequate solution for high-performance systems in which a fast clock is critical. A second strategy places two synchronizers in series between the asynchronous inputs and the rest of the synchronous system. Both the flip-flops must be metastable before the synchronization fails, an even with low probability. A third strategy does away with the clock altogether and follows a timing strategy that is independent of the speed of the individual circuits.

Chapter 3
Width Generator

A Width Generator (Pulse Generator) is a very important and useful component of the circuit. This block actually deals with the outside asynchronous world. It captures asynchronous time inputs and converts them into two complementory digital pulses which will become the controlling signals for the gates of transistors. A Width Generator also deals with the metastability issues discussed in the previous chapter. This chapter explains a circuit and functioning of With Generator. Time interval measurement is actually done here.

Conceptualization:

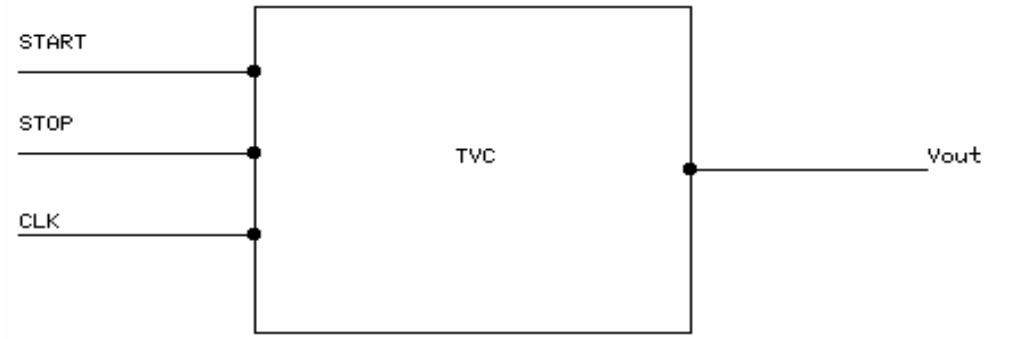


Figure 3.1 TVC as a Black Box

Figure 3.1 shows Time-to-Voltage Converter (TVC) as a black box which has three inputs START, STOP and CLK and one output Vout. Above diagram can be modified as shown in Figure 3.2.

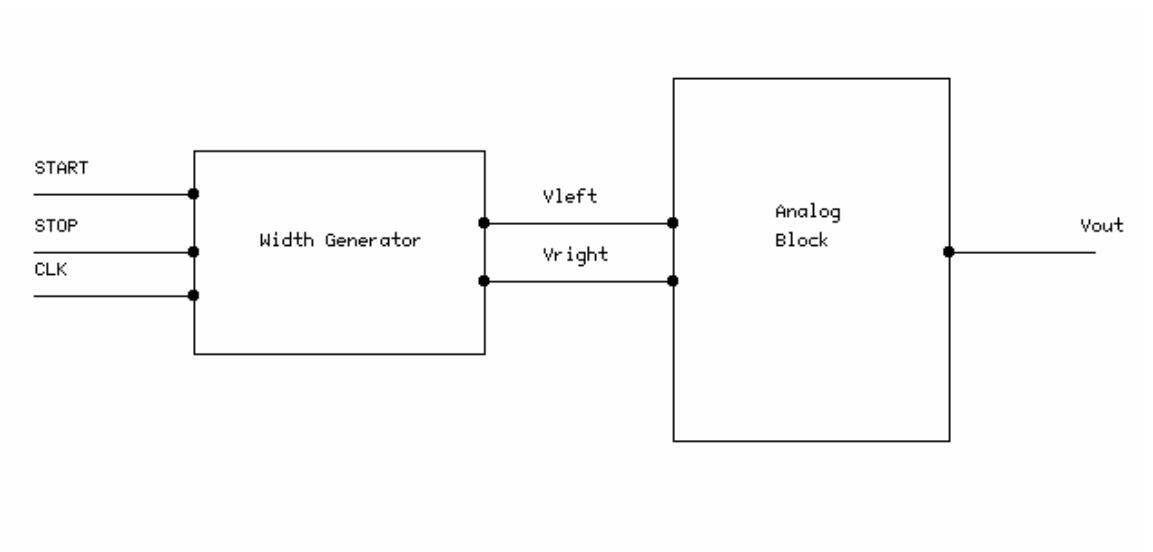


Figure 3.2 Width Generator as a Separate Block

Assume Width Generator as a black box. It has three inputs START, STOP and CLK and two complementary outputs Vleft and Vright are needed. It is understood that START pulse occurs earlier than the STOP pulse. This can be understood from the example of Radar. In the application of Radar, a signal is being sent to the target for the detection (Sending Pulse) and that signal is reflected back and is received by Radar (Receiving Pulse). By measuring the time difference between these two signals, the distance of the target is determined. Thus, it is obvious

that Sending Pulse always occurs first, without Sending Pulse, Receiving Pulse does not exist. Width Generator works on the same principle. Thus, START pulse occurs first. The function of a

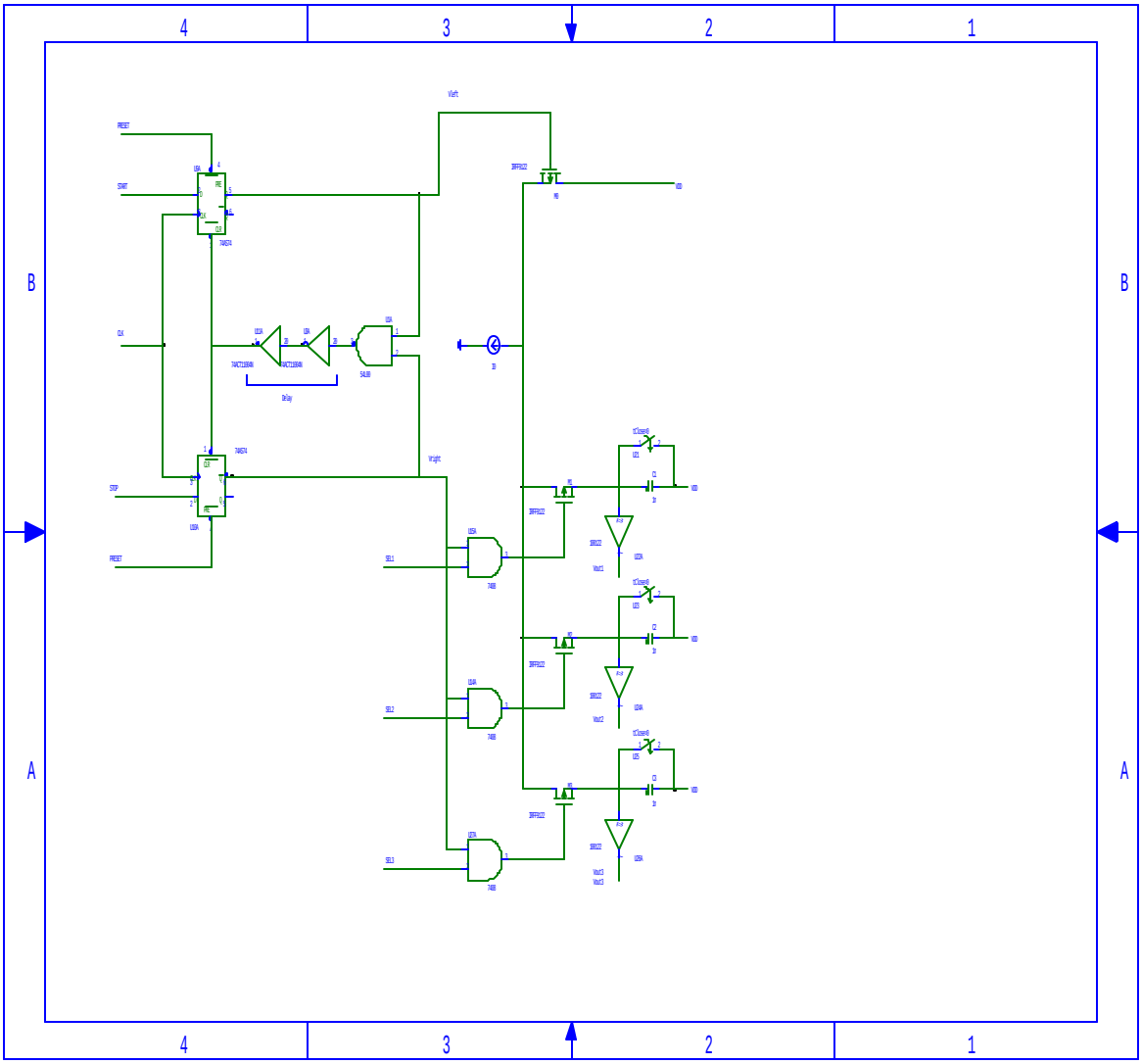
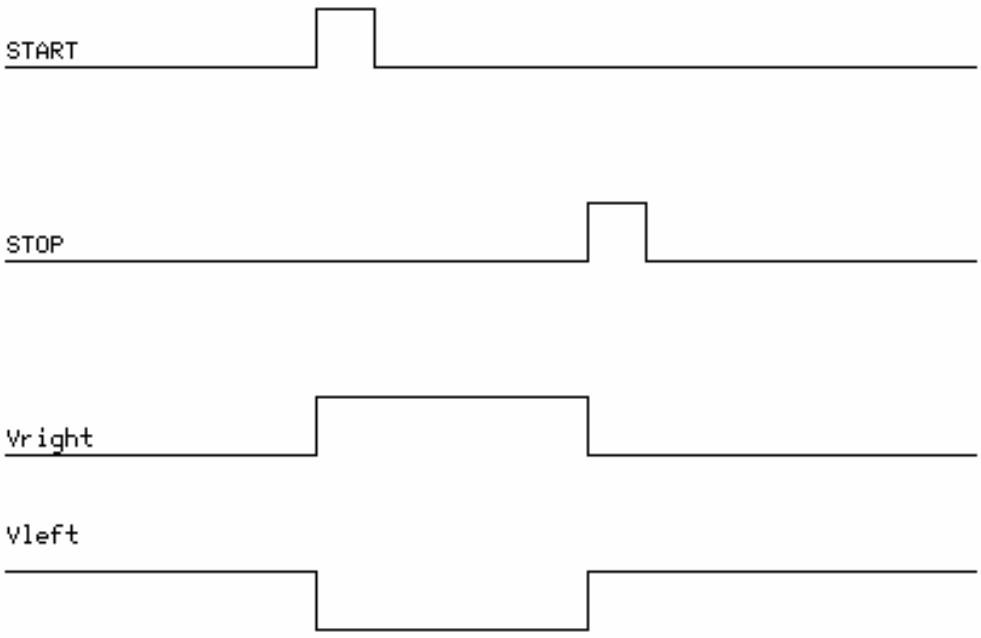

4
3
2
1
B
A
RESET
START
CLK
STOP
Vleft
Vright
Delay
SEL1
SEL2
SEL3

Figure 3.5 Three Channel Time-to-Voltage Converter.

Width Generator is to provide control pulses (Vleft and Vright) for the gates of the transistors M0 and M1. These control pulses are complementary.



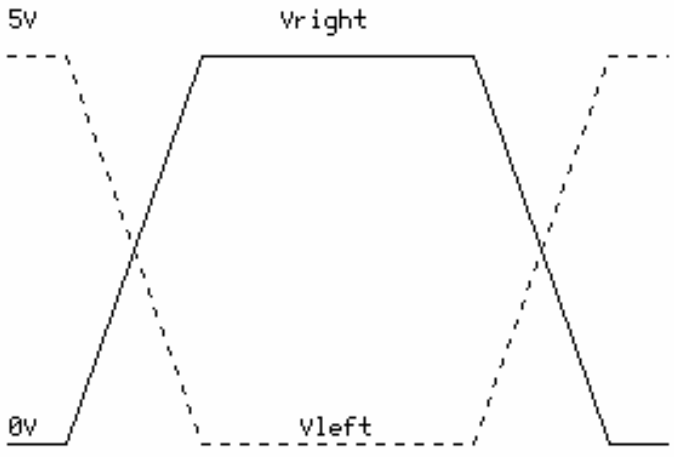


Figure 3.4 Expected Timing Diagram.

In reality this is an asynchronous circuit. That means START and STOP pulses don't depend on CLK. Whenever START pulse is received, Vleft goes high and remains high until STOP pulse is detected. Vright is the complement of Vleft. Since, it is always a good design practice to make any asynchronous signals synchronous to the clock, figure(3.3) shows the circuit diagram of single channel TVC with the synchronizer flip-flops at the input and Figure(3.4) shows the timing diagram.

General Flow: General design flow for any kind of digital design is

- Architecture
- Behavioral Code (RTL)
- Compilation
- Test Bench (RTL)
- Test Bench Compilation
- Synthesis
- Simulation

Architecture: Since, Width Generator is a single, small block, there is not much work in deciding the architecture. Also, there is no hierarchy in the block. All the components are flat (at the same level). The design criteria is that the Width Generator block is to be designed which has three inputs START, STOP (START pulse occurs earlier than STOP) and CLK which is 200 MHz and two complementary outputs Vleft and Vright. Figure (3.2) shows black box with three inputs START, STOP and CLK and two outputs Vleft and Vright.

Behavioral Code (RTL): From the above architecture and design criteria, code for Width Generator is written in VHDL. VHDL is a hardware description language
mainly for digital electronic systems. It arouses out of the United States government's Very High-Speed Integrated Circuits (VHSIC) program. In the course of this program, it became clear that there was a need for a standard language for describing the structure and function of integrated circuits (ICs). Hence the **V**HSIC **H**ardware **D**escription **L**anguage (VHDL)

was developed. Nowadays, research is going on for Analog VHDL. Figure (3.4) shows RTL level VHDL code for Width Generator.

```
--#################################################################
--File_Name: widthgen_final_async.vhd
--Chirag Patel.
--01/10/1999.
--Description:
--    This code is written by assuming this as an asynchronous
--    design. Any of the inputs does not depend on clock.
--    Clock should run at 100 MHz. It is assumed that START
--    pulse is always received first.

--#################################################################



Library IEEE;
USE IEEE.std_logic_1164.all;

-----------------------------------------------------------------
----------------------Entity Declaration-----------------------
-----------------------------------------------------------------

ENTITY WIDTHGEN is
     port
     (START     : in std_logic;
      STOP : in std_logic;
      CLK  : in std_logic;
      Vleft     : out std_logic;
      Vright    : out std_logic
       );

end WIDTHGEN;

-----------------------------------------------------------------
-------------------------Architecture-----------------------
-----------------------------------------------------------------

architecture FINAL of WIDTHGEN is
  signal Vleft1    : std_logic  := '0';

  begin

P1: PROCESS(START, STOP)
  begin

 if START = '1' then
```

```
  Vleft1 <= '1';
 elsif  STOP = '1' then
  Vleft1 <= '0';
  end if;

end PROCESS P1;

  Vright <= NOT Vleft1;
   Vleft <= Vleft1;

end FINAL;
```

Figure 3.4 RTL Code for Width Generator (Asynchronous)

A VHDL code for Width Generator is written in two manners. First one is as an asynchronous design which is shown in Figure 3.4 and the second is as a synchronous design as shown in Figure 3.5.

In real world this is an asynchronous event. In RADAR application, a sending signal is being sent to the target (START) and the receiving signal (STOP) is received back upon finding a target regardless of the clock. It is clear from the architecture section of the code (Figure 3.4) that any the of the inputs does not depend on CLK. This is an asynchronous design. First of all, asynchronous designs are always faster than synchronous designs. The concept of Time Interval Measurement is also implemented in RADAR applications. Now, for a while assume that this is a synchronous design and also assume signal START as a sending signal from RADAR and STOP as a receiving signal back from target. Because of synchronous to the CLK, even if sending signal is ready to go, it has to wait for a CLK to go high (if the transition is set to occur at rising edge of the CLK) and similarly, even if RADAR has already received signal back from the target upon finding it, one can not see it until the next CLK pulse. This introduces a delay, which is very dangerous in this kind of applications. Since, there are certain disadvantages of an asynchronous design as discussed in chapter 2

and also for this thesis purpose the circuit clock runs at 100 MHz which is fast enough to create a very small delay. That is why we will continue with a synchronous design.

```
--#################################################################
--File_Name: widthgen_final_sync.vhd
--Chirag Patel.
--01/10/1999.
--Description:
--    This code is written by assuming this as a synchronous
--    design. Both the inputs depend on the clock as it is
--    clear from PROCESS P1.
--    Clock should run at 100 MHz. It is assumed that START
--    pulse is always received first. RESET port is added on
--    the entity because for synchronous designs reset input
--    is very important.

--#################################################################


Library IEEE;
USE IEEE.std_logic_1164.all;

-----------------------------------------------------------------
----------------------Entity Declaration-----------------------
-----------------------------------------------------------------

ENTITY WIDTHGEN is
     port
     (START     : in std_logic;
      STOP : in std_logic;
      CLK  : in std_logic;
      RESET     : in std_logic;
      Vleft     : out std_logic;
      Vright    : out std_logic
       );

end WIDTHGEN;

-----------------------------------------------------------------
--------------------------Architecture-----------------------
-----------------------------------------------------------------

architecture FINAL of WIDTHGEN is
  signal Vleft1      : std_logic  := '0';
```

```
  signal reset_d     : std_logic;
  signal start_d     : std_logic := '0';
  signal stop_d      : std_logic := '0';

     begin

RST : PROCESS(CLK)
       begin

-- reset
if (CLK'event and CLK = '1' ) then
   reset_d  <= RESET;
   start_d <= START;
   stop_d <= STOP;

  end if;
end PROCESS RST;

P1: PROCESS(CLK, start_d, stop_d)
     begin
 if (CLK'event and CLK = '1' ) then

    if start_d = '1' then
  Vleft1 <= '1';
 elsif (CLK'event and CLK = '1' ) then

 if stop_d = '1' then
  Vleft1 <= '0';
      end if;
   end if;

-- if (reset = '0') then

--  START1 <= '0';
 -- STOP1 <= '0';
--end if;

end if;
end PROCESS P1;

   Vright <= NOT Vleft1;
   Vleft <= Vleft1;

--START <= START1;
--STOP <= STOP1;

 end FINAL;
```

Figure 3.5 Synchronous Design.

Compilation: For compilation two different tools have been used just to make sure that code compiles ok. ModelTech's MTI and IKOS System's VOYAGER. Some of the tools in industry are user friendly and they sometimes could not catch some of the errors.

Test Bench: This is a very important part in design flow. This Test Bench is behavioral VHDL code. This is a place where you can tell the circuit to behave in a manner you want. Figure (3.6) shows the concepts of the Test Bench.

In this case, UUT (Unit Under Test) is Width Generator and TB is the VHDL Test Bench shown in Figure (3.7). The Test Bench is like a wrapper around a UUT. Input

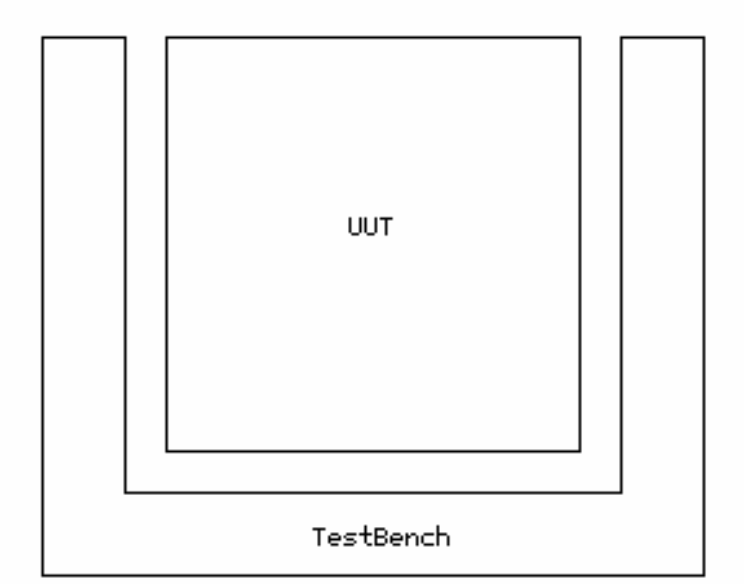


Figure 3.6 UUT and TestBench.

Ports can be forced by the Test Bench and output signals can be views by the Test Bench. In complex ASIC world, actually Wrapper (Ring) file, Force file and Monitor files are

written separately and basically one Test Bench file calls all the above files to come into affect. The Wrapper file is written to make all the ports of UUT visible to the Test Bench, the Force file is written to force the inputs of UUT and the Monitor file is written in order to view the output signals of UUT.

```
--#################################################################
--File_Name: tb_widthgen_final.vhd
--Chirag Patel.
--01/10/1999.

--Description:
--    Rightnow this testbench only generates two signals clock
--    and reset. Rest of the inputs are force from force file.
--    This force file called Widthgen.do. Some of the lines
--    are commented out from this testbench because it is
--    redundant at this stage. If you do not want to use do
--    file, please uncomment those lines.
--#################################################################


Library IEEE;
USE IEEE.std_logic_1164.all;

-----------------------------------------------------------------
------------------Entity Declaration for TestBench-----------
-----------------------------------------------------------------

entity TB_WIDTHGEN is
   end TB_WIDTHGEN;


-----------------------------------------------------------------
--------------------Architecture for TestBench---------------
-----------------------------------------------------------------

architecture TEST of TB_WIDTHGEN is

--Component Declaration

component WIDTHGEN port
            (START   : In std_logic;
```

```
          STOP    : In std_logic;
             CLK     : In std_logic;
          RESET   : in std_logic;
          Vleft   : Out std_logic;
             Vright  : Out std_logic
             );
 end component;

--Signals at top level (infact there is no hierarchy...)

signal       START   :   std_logic;
signal       STOP    :   std_logic;
signal       CLK     :   std_logic := '0';
signal       RESET   :  std_logic := '0';
signal       Vleft   :   std_logic;
signal       Vright  :   std_logic;

  begin

--Reset the device

RESET <= '1' after 30 ns;

--CLK generation...
  CLK <= not CLK after 10 ns;

  --START <= '0';
  --STOP <= '0';

--wait for 80 ns;
--P2 : PROCESS

 --begin
--wait for 80 ns;

--if (CLK'event and CLK = '1' ) then
--  START <= '1';
--wait for 50 ns;
--  START <= '0';
--end if;
--end PROCESS P2;

--P3 : PROCESS
--begin
 --wait for 120 ns;

--if (CLK'event and CLK = '1' ) then
--  STOP <= '1';
--wait for 50 ns;
 -- STOP <= '0';
```

```
--end if;
--end PROCESS P3;

UUT: WIDTHGEN port map
                    (START    => START,
                STOP     => STOP,
                CLK      => CLK,
                     RESET => RESET,
                     Vleft    => Vleft,
                Vright   => Vright
                     );

 end TEST;

-----------------------------------------------------------------
----------------Configuration for TestBench------------------
-----------------------------------------------------------------

--This Configuration cfg_TEST is loaded into Simulator.

configuration cfg_TEST of TB_WIDTHGEN is
  for TEST
 end for;
end cfg_TEST;
```

Figure 3.7 Test-Bench.

Since, this is a small block, only a main Test Bench and the Force file are written. Figure (3.8) shows the Force file (Non-VHDL Stimulus File). The main Test Bench generates a clock signal (200 MHz) and the Force file forces the inputs START and STOP.

```
-----------------------------------------------------------------
-----
--File_Name: Widthgen_final.do
--Chirag Patel.
--01/10/1999.
-----------------------------------------------------------------
-----


-----------------------------------------------------------------
-----
--Description:
```

```
--    This is basically Non-VHDL Stimulus file for Width
Generator.
--    For MTI --(ModelTech.), it is .do file but for other
simulators
--    it can be with other --extensions.
--    This file basically forces signals START and STOP.
--    CLK is generated by TestBench.
--    Initially it loads all the signals into waveform viewer.
--    The main purpose of this file is to save time in writing a
--    seperate testbenchs for synchronous and asynchronous
designs.
-----------------------------------------------------------------
------


--add wave loads signals (ports) into waveform viewer.
add wave -logic /tb_widthgen/start
add wave -logic /tb_widthgen/stop
add wave -logic /tb_widthgen/clk
add wave -logic /tb_widthgen/uut/vleft1
add wave -logic /tb_widthgen/vleft
add wave -logic /tb_widthgen/vright
add wave -logic /tb_widthgen/uut/RESET
add wave -logic /tb_widthgen/RESET

force START 0
force STOP 0
run 200

force START 1
run 20

force START 0
run 100

force STOP 1
run 20

force STOP 0
run 100


--End of the Stimulus..
```

Figure 3.8 Force (Stimulus) File.

Test Bench Compilation: MTI and VOYAGER tools are used for compilation.

Synthesis: A Synthesis is a process of converting RTL Behavioral VHDL (or Verilog) code into real gates using provided technology. As a result of Synthesis process, gate-level structural VHDL (Netlist) code is generated. Synopsys's Design Analyzer is used for Synthesis. Figure (3.9) shows the gate -level structural VHDL code.

```
library IEEE;
library sla30000;

use IEEE.std_logic_1164.all;
use sla30000.components.all;

entity WIDTHGEN is

   port( START, STOP, clk : in std_logic;  Vleft, Vright : out
std_logic);

end WIDTHGEN;

architecture SYN_FINAL of WIDTHGEN is

   component iv1
      port( yn : out std_logic;  a : in std_logic);
   end component;

   component lnr1
      port( q, qn : out std_logic;  r, s : in std_logic);
   end component;

   signal Vleft_port, n10 : std_logic;

begin
   Vleft <= Vleft_port;

   u10 : iv1 port map( yn => Vright, a => Vleft_port);
   Vleft1_reg : lnr1 port map( q => Vleft_port, qn => n10, r =>
STOP, s =>
                           START);

end SYN_FINAL;
```

Figure 3.9 Structural VHDL From Synopsys Design Analyzer.

Simulation: This is a heart of whole process because here you can see your circuit doing actual intended functions. Modeltech's MTI Simulator is used for Simulation. Figure (3.10) and Figure (3.11) shows simulation results for synchronous and asynchronous circuits respectively.

This is also a Timing Diagram. It is already known that START pulse comes first, so whenever START occurs Vleft goes high and remains high until the STOP comes. The STOP pulse pulls Vleft back to low and Vright is just a complement of Vleft.

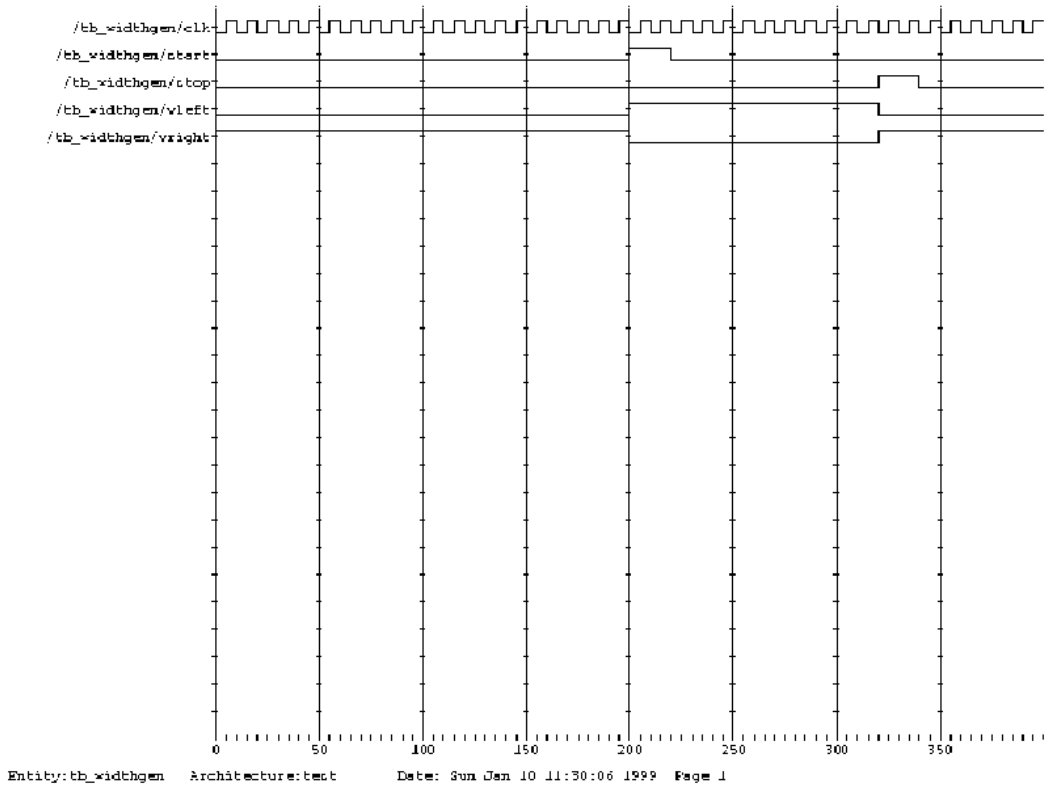


Figure 3.10 Simulation Results (Asynchronous Circuit)

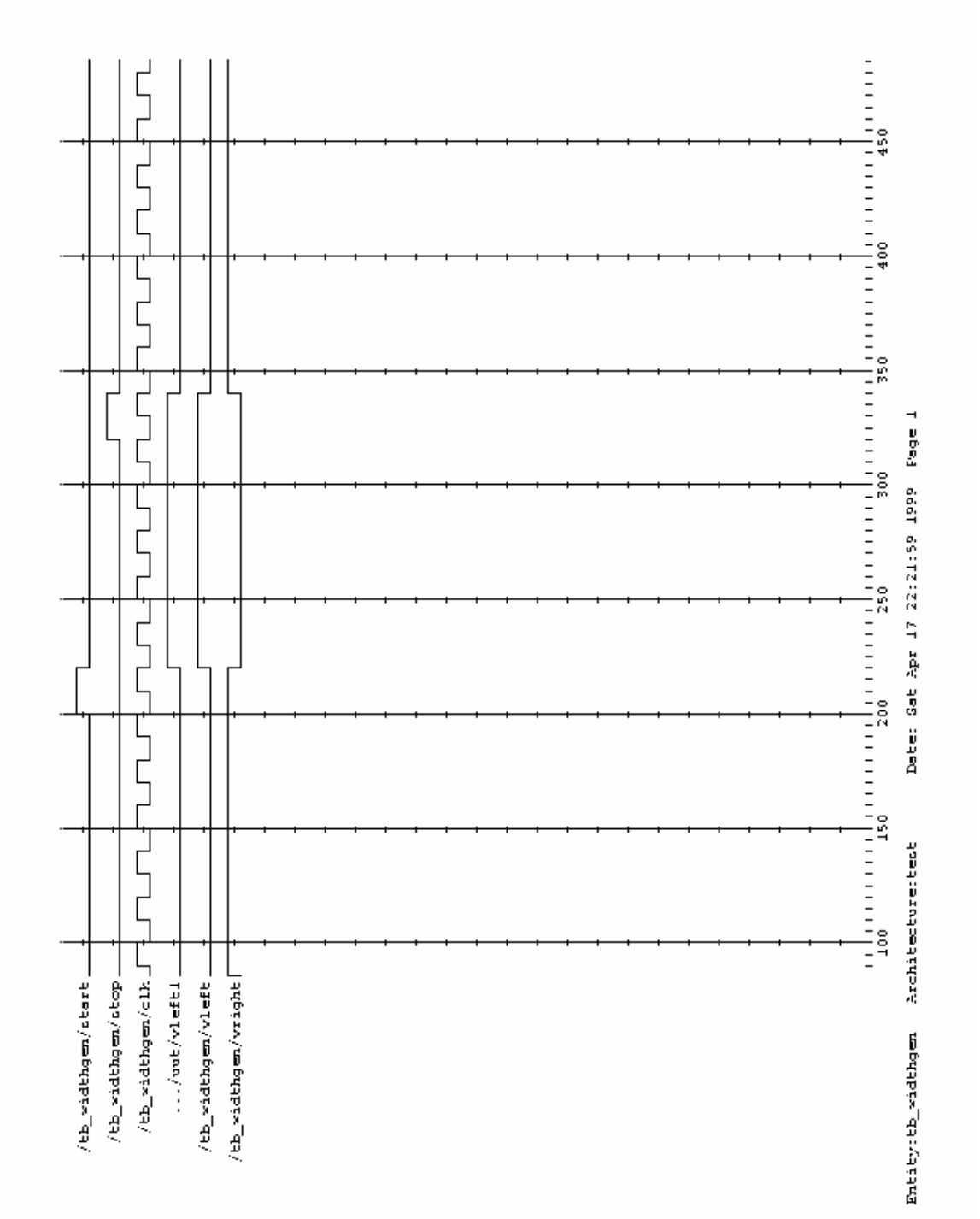


Figure 3.11 Simulation Results (Synchronous Circuit)

Chapter 4
Time-to-Analog Voltage Converter

Introduction: The digitalization of physical dimensions plays a major role in the progress of electronics. In order to control processes, supervise and regulate the course of events and for several additional measurement tasks converter systems are necessary, which convert the measured physical values into digital or analog values. This Time-to-Voltage Converter is developed for the measurement of time differences with resolutions in the nanosecond range.

This design, which belongs to the family of TDCs (Time-to-Digital Converter), converts a measured value into a digital value via the in-between value of the time difference. They accomplish the same as an ADC (Analog-to-Digital Converter), which uses the electrical voltage of the in-between values. While ADCs have been employed by the industry for decades, TDCs and their opportunities are still rather new.

Analog Procedure: Analog converter procedures work in 2 phases. In the first phase the measured time difference is converted into digital pulses (Time-to-Amplitude Conversion). In a second step an analog voltage is stored in analog memory. In regards to the technical opportunities of today this is the older technique. Although it allows very high resolutions to be measured down to several Pico seconds, it does however demand many requirements and restrictions. These requirements of course have their price. Special care is needed when constructing a circuit and

when choosing analog elements. The following diagram shows the correlation and the sequence of the measurement.

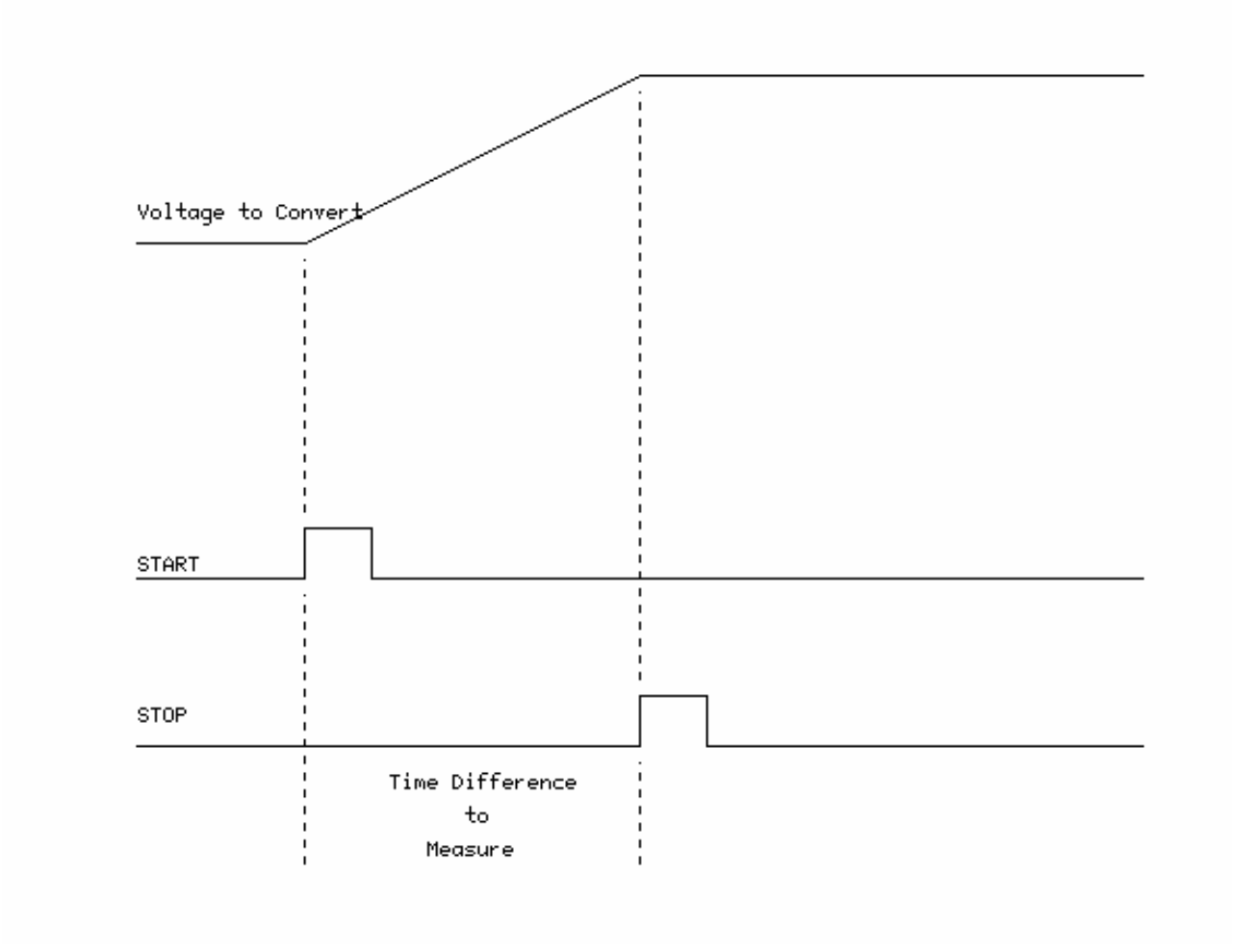


Figure 4.1 Principal functions of an analog TDC

Digital delay time TVCs on the other hand work without any analog components. They are part of the innovations that have been made possible by the major technical progress in semiconductor technology in the last couple of years. They use the chip-internal measurement

times of simple logical gates (i.e. inverters) for fine quantification of the time difference. Due to the great speed accelerations, especially in the CMOS sector, it is possible to implement such TVCs on CMOS processes today, so that the resolutions in the lower Pico second area can be realized. Efficient, power saving and non-the-less inexpensive complete systems integrated in one chip are the result. A measurement time TVCs can be divided into 2 groups:

· Absolute delay time TVCs.

· Relative delay time TVCs.

Absolute Delay Time TVCs: This type of TVC uses the absolute delay time of simple internal logical elements for the fine quantification of the time difference. In other words, one determines how many basic delay times, i.e. from inverters, the measured time difference consists of. Figure 4.2 displays the principal set-up. Clever circuit set-ups, redundant circuit elements and special layout methods on the chip enable the exact reconstruction of the number of basic delay times. This accessible resolution during this process is strictly dependent of the optional basic delay time on the chip. Resolutions in the area of 80-100 ps can be realized by a simple set-up of the measuring core and the use of the most modern CMOS processors.

Absolute delay time TVCs have the following additional advantages:

- The delay time of the inverter can be precisely adjusted and stabilized at a specific temperature within certain limits via appropriate evaluation with phase lock loops (PLLs).

- The resolution can be improved via a dexterous varied set-up of the measurement circuit. All sensible factors result in 2-4, so that the resolution can be improved up to 20-30 ps with the help of absolute delay time TVCs.

This type of chip belong to the group of universal TDCs, which can be adapted to (almost) any type of task

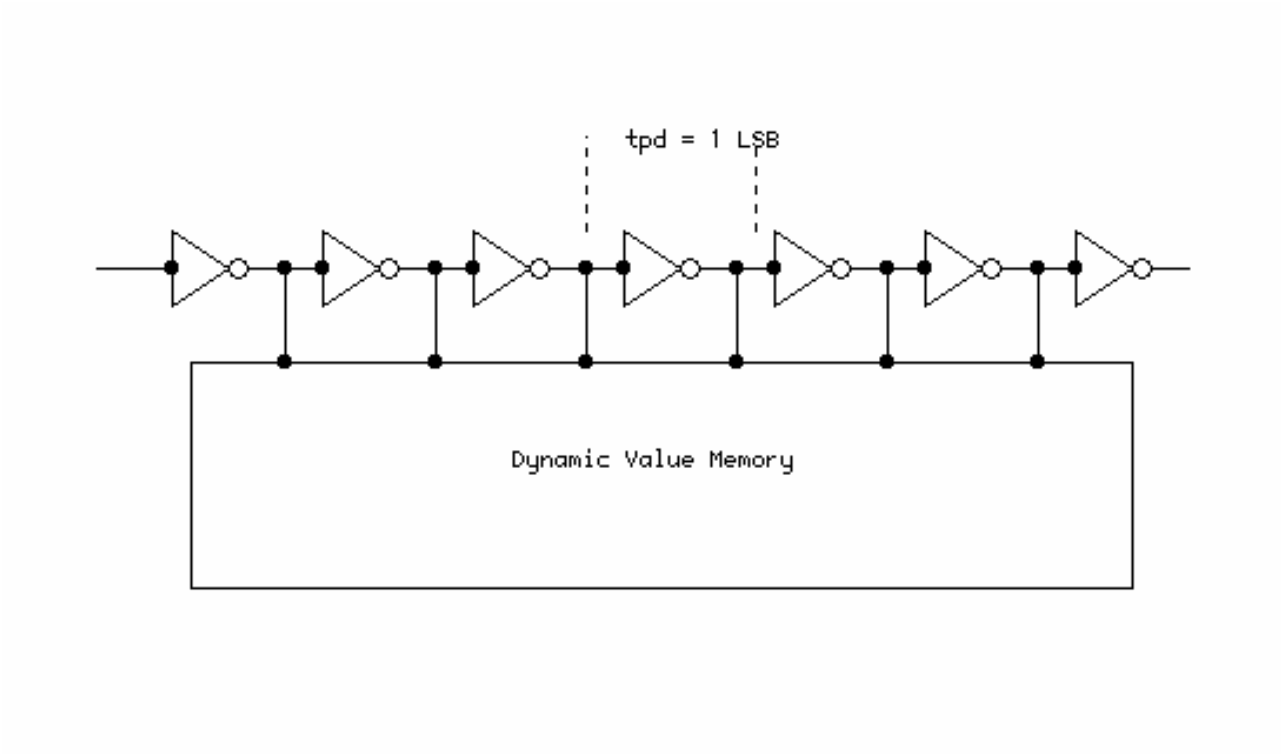


Figure 4.2 Principal Set-up of an Absolute Delay Time TVC

Relative Delay Time TVCs: While the absolute procedure only permits the available resolution to be bound to the speed of the semiconductor processes, this binding for the relative procedure is not quite as strict. As the name explains, relative delay time differences are used between two logical elements for fine quantification. Figure 4.3 displays the principal correlation. With the help of a corresponding circuit set-up, the resolution becomes identical to the difference between the two running time's tpd1 and tpd2. Fields are reached that lie far under the min. delay time on a chip via this procedure. Principally, any type of high resolutions can be realized, but limits do exist regarding precision.

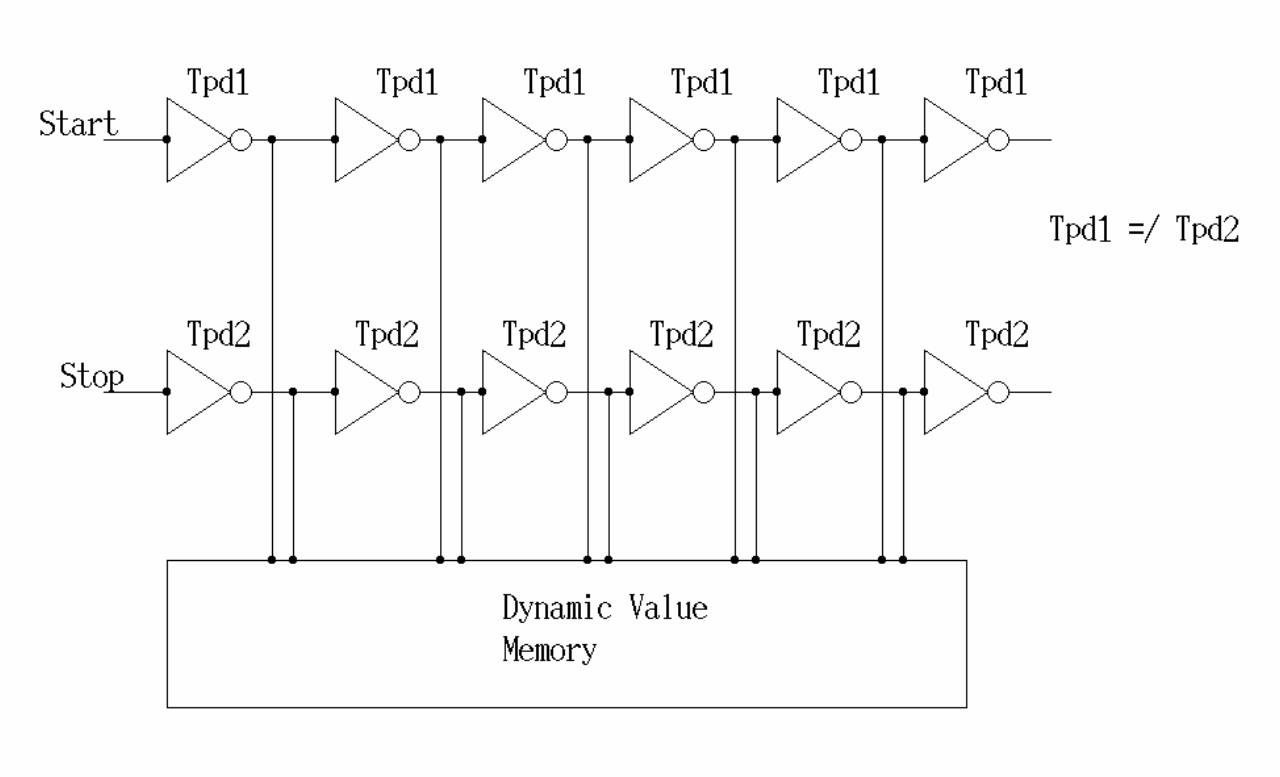


Figure 4.3 Principal Set-up of Relative Delay Time TVCs

These limits occur when one only veers from the quantification errors on to other error sources. Based on previous experience, this procedure enables a sensible realization of app. 1/5 of the delay time as the resolution. This permits, given the use of modern CMOS processors, the realization of a resolution with app. 10-15 ps seconds. Other features, such as limited differential non-linearity predestine this procedure for several measurement tasks. Wherever these special parameters play a major role, this type of TDC should be the first choice. Several opportunities for absolute delay time TDCs have not yet been made possible, or only in a limited manner. It is not possible to realize a quartz exact adjustment of the resolution with simple chip-related elements. Multi-hit TDCs only have limited double pulse resolution, since this measuring procedure possesses a conversion time, resulting in a relatively long lag time. In

conclusion, the relative delay time TDCs belong to a special solutions group that can be applied anywhere their advantages are useful.

Measuring range: The measuring ranges, which are necessary for the realization of various applications, vary greatly. They begin in the range of several nanoseconds for some applications in the research area (i.e. high energy physics), and end in the range of few milliseconds for industrial applications (i.e. position controllers). Even for larger time differences a resolution in the Pico second range is important. The requested dynamic has a range of almost 30 bit!!!!!! Today's TDCs can fulfil all these requirements comfortably. They work with 2 measuring ranges, which are basically very different.

Basic measuring range: In the basic measuring range the entire time difference is measured with the high speed measuring unit of the TDCs. The starting signal starts the unit and begins with the quantification, and the stop signal saves the momentary internal state, from which the measured time difference can be reconstructed.

Time differences of up to app. 10-15 µs can be measured in this measuring range. In addition, the noise effects and other analogue effects become noticeable, so that the standard deviation of the results is slightly increased. If the above mentioned rules are followed, the standard deviation can be kept under 1 LSB. For time differences < 5 µs the standard deviation ranges from 0.5 0.7 LSBs. Depending on the chips and the measuring mode used, a min. measuring difference exists below which a measurement cannot take place. The TDC-GP1 of acam-messelectronic offers measuring modes where exactly 0sec can be measured, and where it is not necessary to

clearly define Start and Stop as such, so that Stop can arrive before Start and still be measured correctly.

Calibration and Adjustment Methods: The delay time of logical gates of circuit is a very imprecise matter. The value can vary greatly. The resulting delay time is dependent upon process variations, temperature and voltage variations. Only variations in temperature and voltage can cause changes in the delay time > 50 %. In order to compensate these variations 2 methods have been established

· Software calibration

· Direct adjustment of the delay time via hardware methods

The hardware methods can also be divided into 2 methods

· Resolution Lock

· Resolution Adjust

Software calibration: The time difference is measured here (first) with an unknown resolution. Calibration values are then generated relative to the measuring event (best when done directly after a measurement). These calibration values help to norm the measuring value. The calibration values are generated while measuring time differences of known values against unknown resolutions. These known time differences can i.e. be easily be derived from a quartz clock. A quartz clock is a very suitable and reliable source regarding it's absolute precision and the jitter. Figure 4.5 displays the correlation graphically.

The unknown measuring value Vmess can be found directly on the measuring line, just like the two calibration values Vcal1 and Vcal2. This line is characterized by a gradient which corresponds with the resolution and an offset set at point zero. With the help of 2 reference measurements Tcal1 and Tcal2 one can determine the gradient and the offset, and in conclusion the unknown time difference Tmess via 2 marks on the measuring line. These two marks on the line offer the following results:

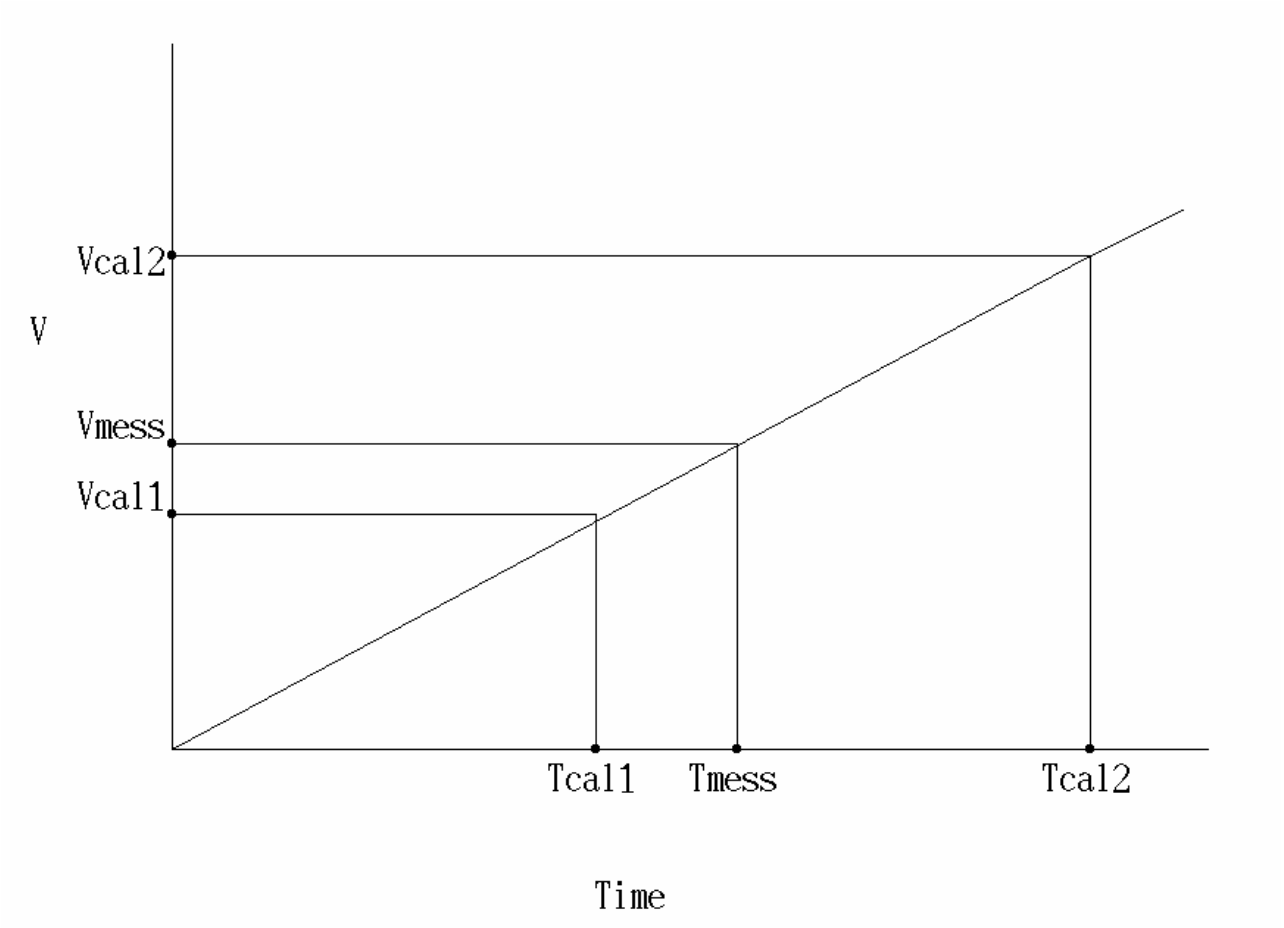


Figure 4.5 Graphical Correlation

Tmess = [(Tcal2-Tcal1) * (Vmess-(2*Vcal1-Vcal2)) (Vcal2-Vcal1)] ÷ (Vcal2-Vcal1)

The gradient : (Tcal2-Tcal1) (Vcal2-Vcal1)

And the offset : (2*Vcal1-Vcal2)

Digital delay time TDCs have the pleasant feature that the measuring line is in reality a straight line. A precision problem due to integral non-linearity, which is a standard problem with analog TDCs, does not exist. The integral non-linearity lies far below 1 LSB. This simplifies the necessary math of a calibration calculation.

Resolution Lock: During this method both the PLL and the measuring circle are located on the same chip, although they do not belong to the same circuit. Similar construction of the delay elements enables similar actions regarding voltage and temperature dependency via matching effects. A full suppression of temperature dependency however cannot be achieved. A good circuit construction and a good layout permit a suppression factor of app. 100. This method can stabilize the resolution at app. 30 PPM/K due to the unregulated temperature dependency of a CMOS Process at app. 3000 PPM/K. During operation in the Resolution Lock Mode it is not possible to work without calibration measurements. The resolution is not entirely independent of the temperature, and this can lead to obvious errors during greater temperature variations and longer measurement times. Differences occur from chip to chip regarding the matching of the PLL and the measuring unit. As a result, identical adjustments on different chips can lead to different resolutions is the measurement circuit. This method is suitable to make resolution variations more independent of temperature, so that the calibration runs can be drastically reduced up to i.e. 1 time per 5 minutes. For some applications, which depend on high measuring rates, this is a big help.

Introduction: A Time-to-Voltage Converter (TVC) has been designed which measures time intervals over a linear range of 17ns to 33ns with an accuracy of 0.5ns. The range is adjustable to longer intervals with a corresponding loss in accuracy. The measurement is stored as an analog voltage on a capacitor. An additional feature of the circuit is the implementation of an analog memory, which allows a series of Time-to-Voltage conversions to be performed in rapid succession. Close matching between the memory channels can allow a total accuracy of 0.3ns or less. This circuit has direct application to instrumentation for high-energy physics experiments (e.g. the Fermilab Tevatron or the Superconducting Super Collider). It also could be used in phase-locked loops, laser remote sensing (lidar), and time-of-flight detectors (medical tomography).

The monolithic TVC/analog memory circuit contains eight memory channels.

Block Diagram:

Figure 4.1 shows the block diagram of Time-to-Voltage Converter. It contains two high level components. Digital circuitry and an analog circuitry. A digital part is also called as Width Generator. A Width Generator will not be discussed in detail here as it has been already discussed in the previous chapter. Figure 4.2 shows the schematic diagram of Width Generator.

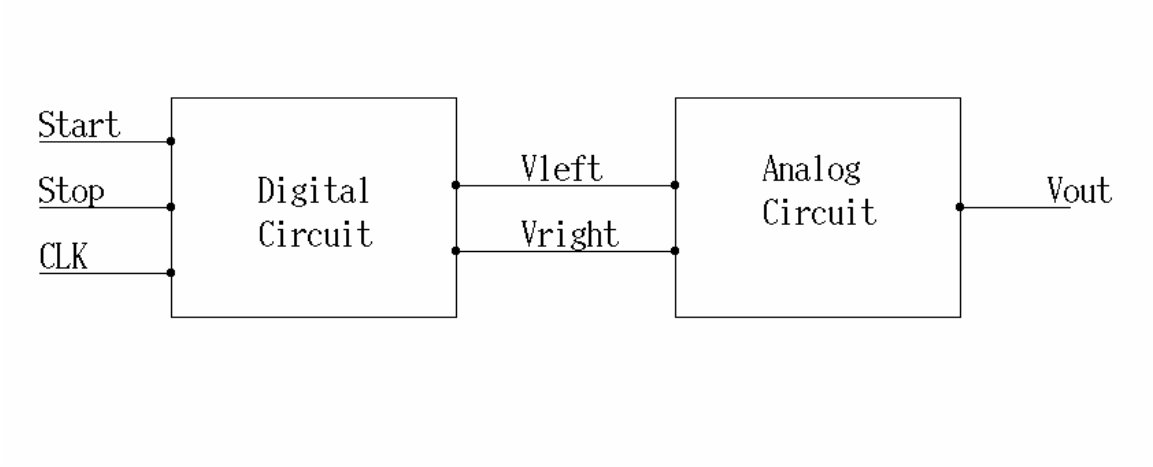


Figure 4.1 Block Diagram

Digital Part (Width Generator):

As shown in the schematic diagram of a Width Generator CLK runs at 250 MHz. CLR input is added in order to reset the D-latches. The first pair of D-latches (U66 and U67) is used for the purpose of synchronization. After these latches everything will be synchronized to the CLK. By using these double synchronizer latches, one may slow down the speed of the data propagation but at the same time one can reduce a great amount of risk of metastability. A metastability issue is already discussed before. As these D-latches are reset by active low reset signal, CLR is low initially for few nanoseconds and then it forced high. A gated clock is used for the synchronizer latch in order to latch proper values of START and STOP pulses. As seen form Figure 4.2 two inverters are added to the CLK signal before it goes to an AND gate because it is not a good practice to feed an AND gate with a CLK signal as an input aspecially when it is running at a very high speed like 250MHz. Figure 4.2 is an outcome of PSPICE Schematic Capture Tool. Basically, a circuit can be viewed as two parts; upper and lower parts. An upper part generates

Vright and lower part generates Vleft. Delays are matched so that Width Generator gives two complementary pulses at the output.

A simulation is done in PSPICE and figure 4.3 shows the timing diagram of Width Generator. Vright and Vleft are the output voltage pulses and as shown in waveform they are complimentary. Also, Vright goes high before Vleft goes low.

A figure 4.4 and 4.5 show some more simulation results of START, STOP, CLK, Vright and Vleft signals to get better picture.

Appendix A shows the output of PSPICE (Circuit Description) and PSPICE code for Width Generator.

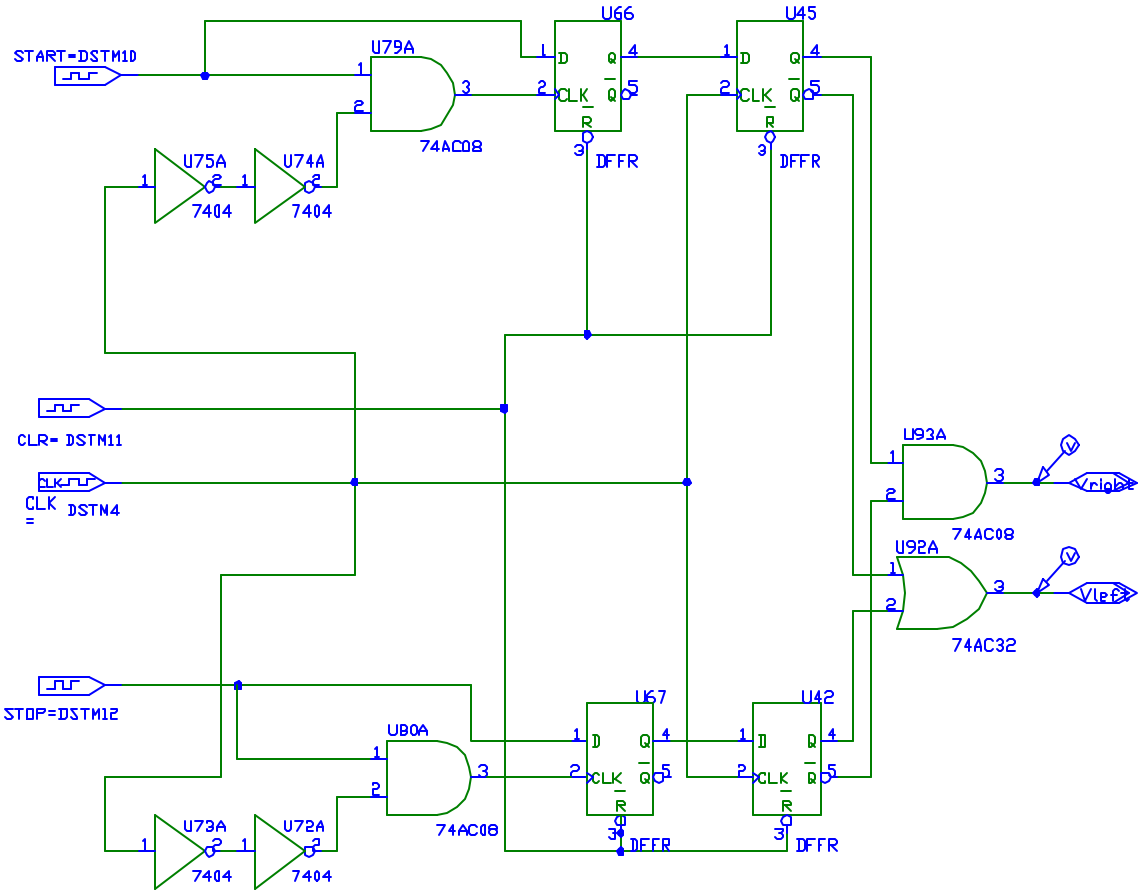


Figure 4.2 Digital Part (Width Generator)

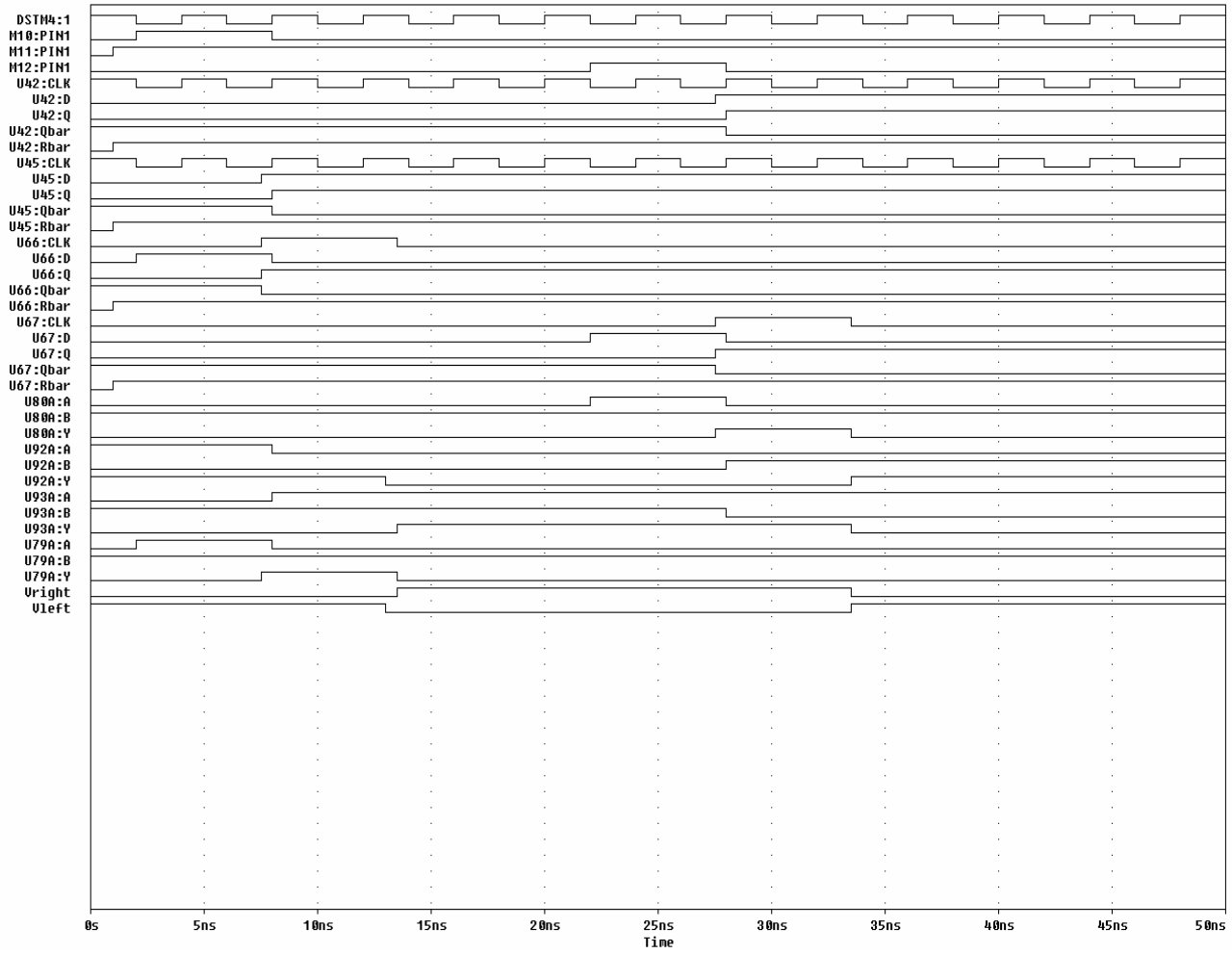


Figure 4.3 Timing Diagram of Width Generator

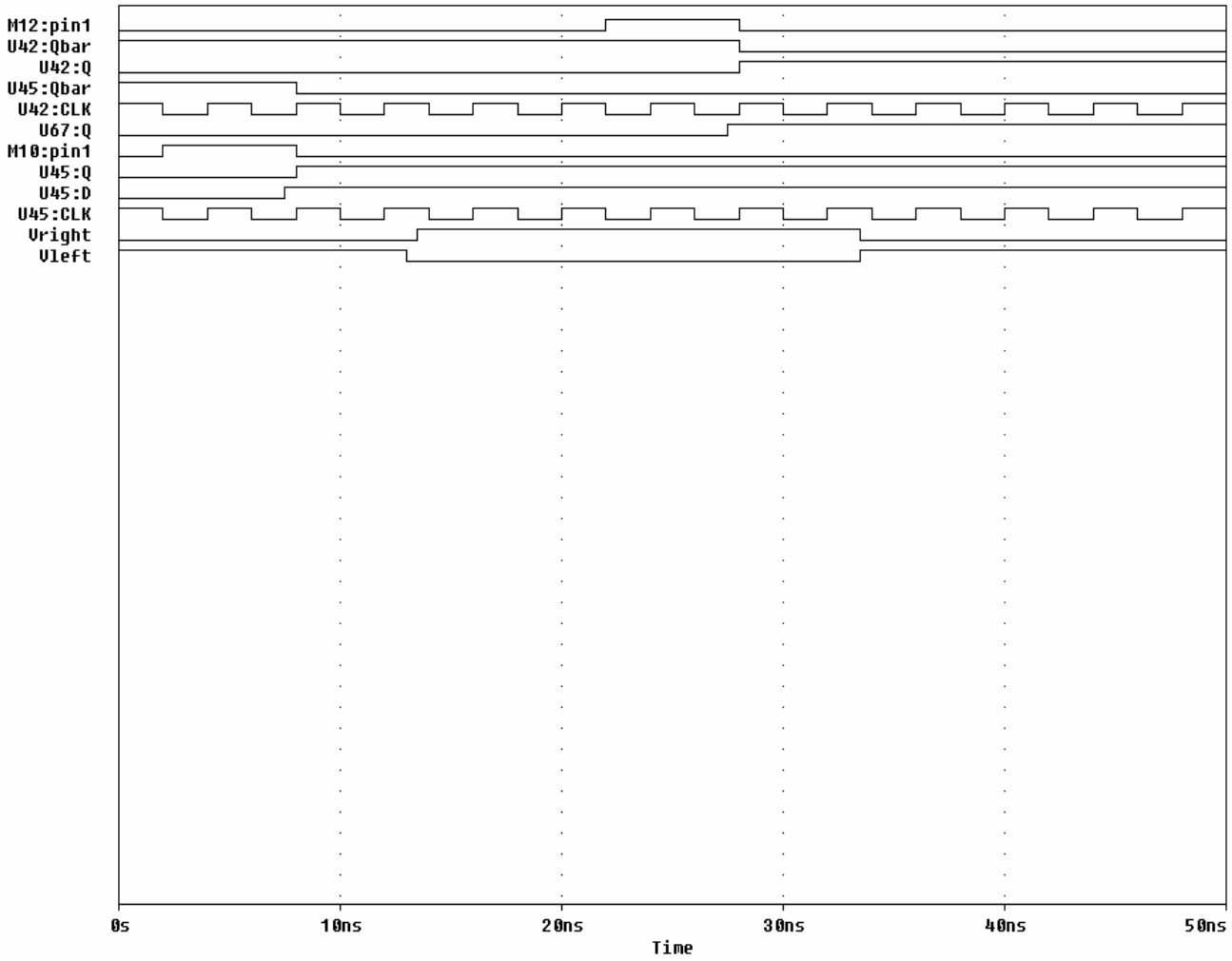


Figure 4.4 Timing Diagram (START, STOP, CLK, Vleft and Vright)

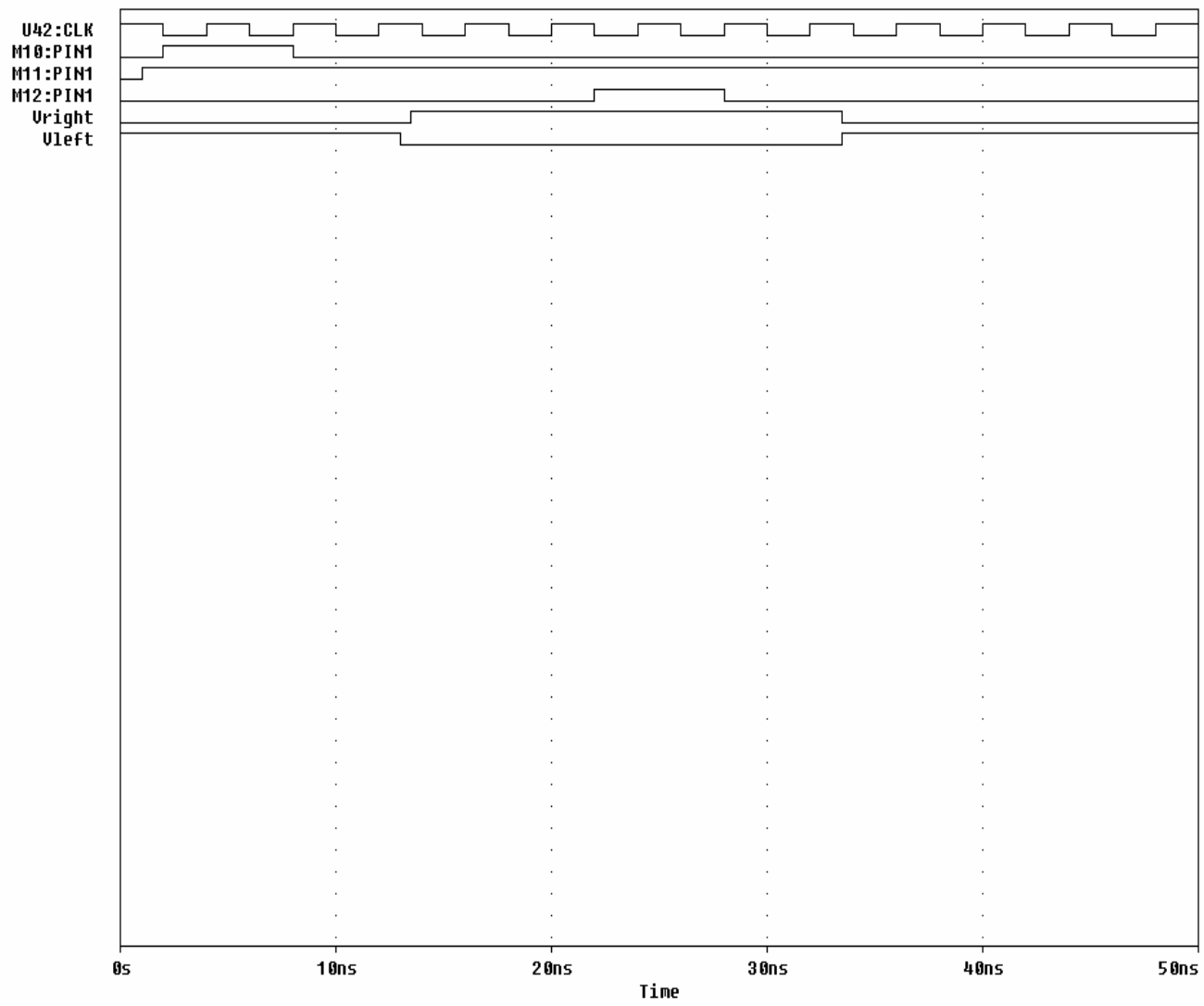


Figure 4.4 More Simulation Results for Width Generator.

Analog Part:

For the purpose of comparison with the actual circuit output, one dummy circuit model for analog part is made and simulated in PSPICE. Figure 4.5 shows that dummy model. Two inputs Vleft and Vright are forced as shown in figure 4.6 between 0V and 5V. They are complimentary pulses between 0V and 5V. Initially, a switch across the capacitor is closed and open again to precharge the capacitor to 5V. Two input Vleft and Vright are forced exactly as they are output from the digital part (Width Generator).

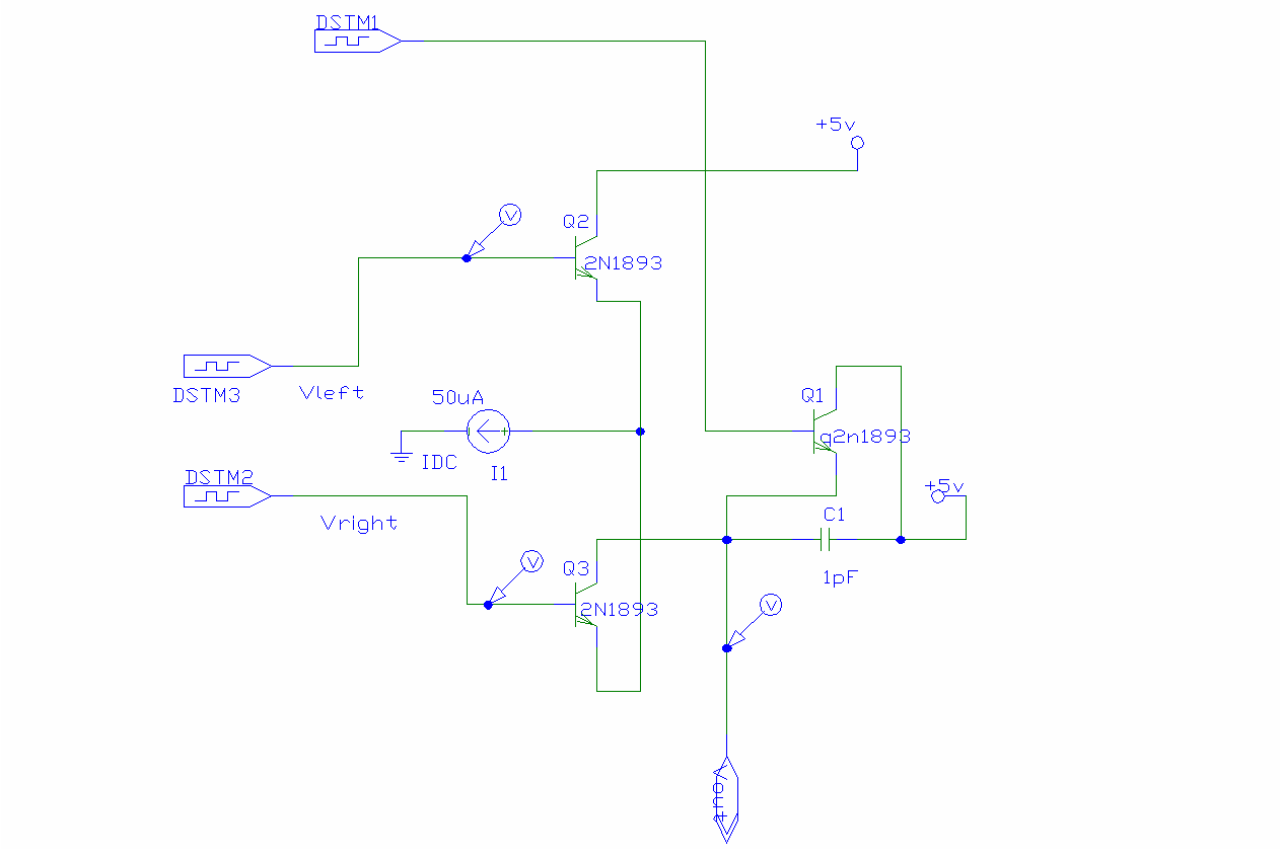


Figure 4.5 Analog Part of TVC.

A measured output analog voltage Vout obeys equation:

Vout = Io/C * t

Where t is a time being measured and also an amount of time for which both the transistors Q2 and Q3 are ON.

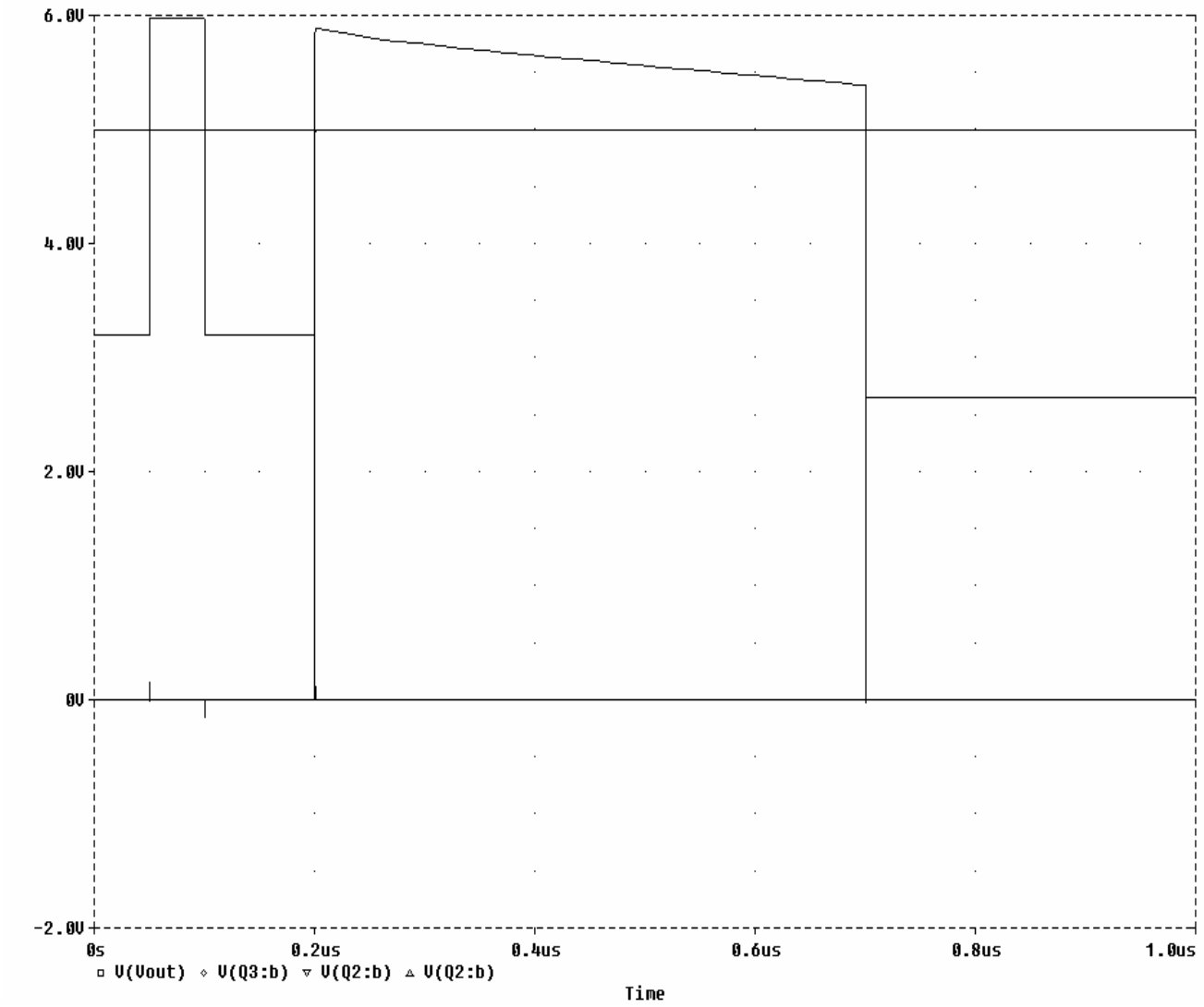


Figure 4.6 Simulation Results.

As shown in Figure 4.6, Vout initially precharged to 6V and then when two complementary pluses Vleft and Vright arrive, it starts decreasing. Figure 4.9 shows more zoomed in picture. As seen from Figure 4.7 Vout drops from about 5.9V to 5.4V in about 500ns(t). Since, this is just a dummy circuit, it runs at slower speed and that is why time interval (t) is about 500ns.

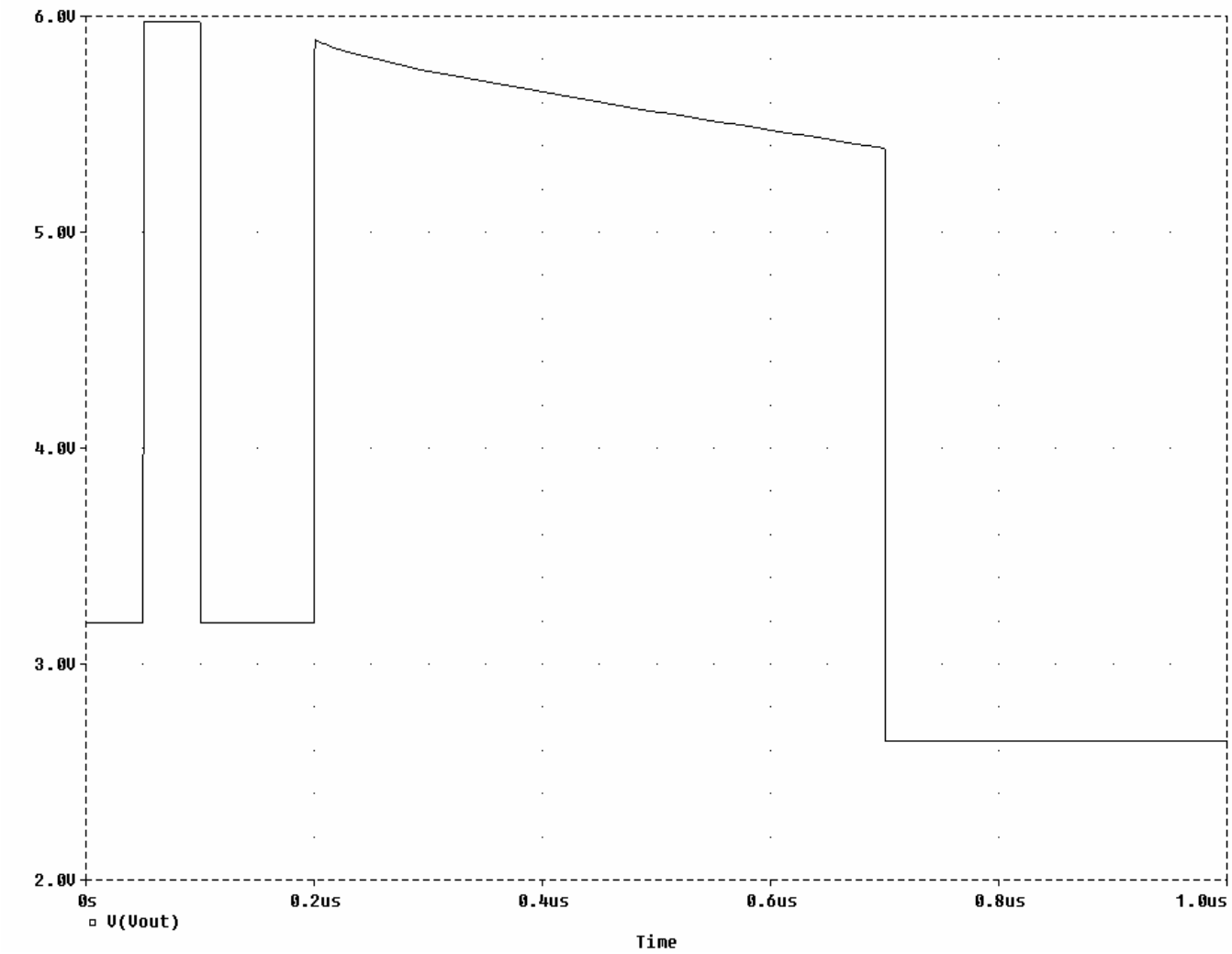


Figure 4.7 Simulation Results.

Figure 4.7 shows only Vout where simulation is run until 1us. Figure 4.8 shows both the inputs to the analog part Vleft and Vright. As shown they are opposite to each other. In really, these signals are never like these as they are the outputs from Width Generator they may be little bit noisy. Appendix B shows PSPICE output and netlist PSPICE code.

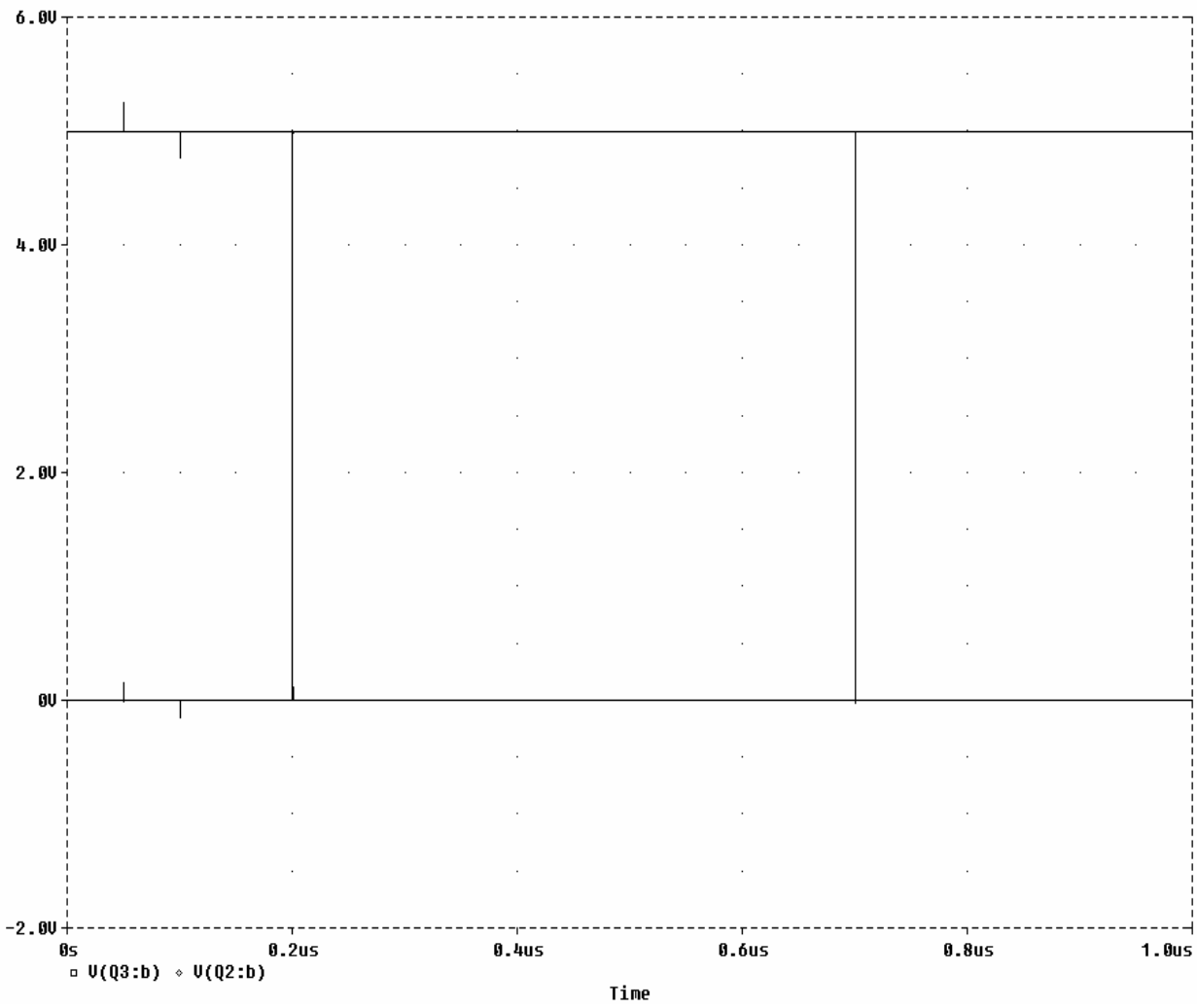


Figure 4.8 Inputs to the Analog Circuit.

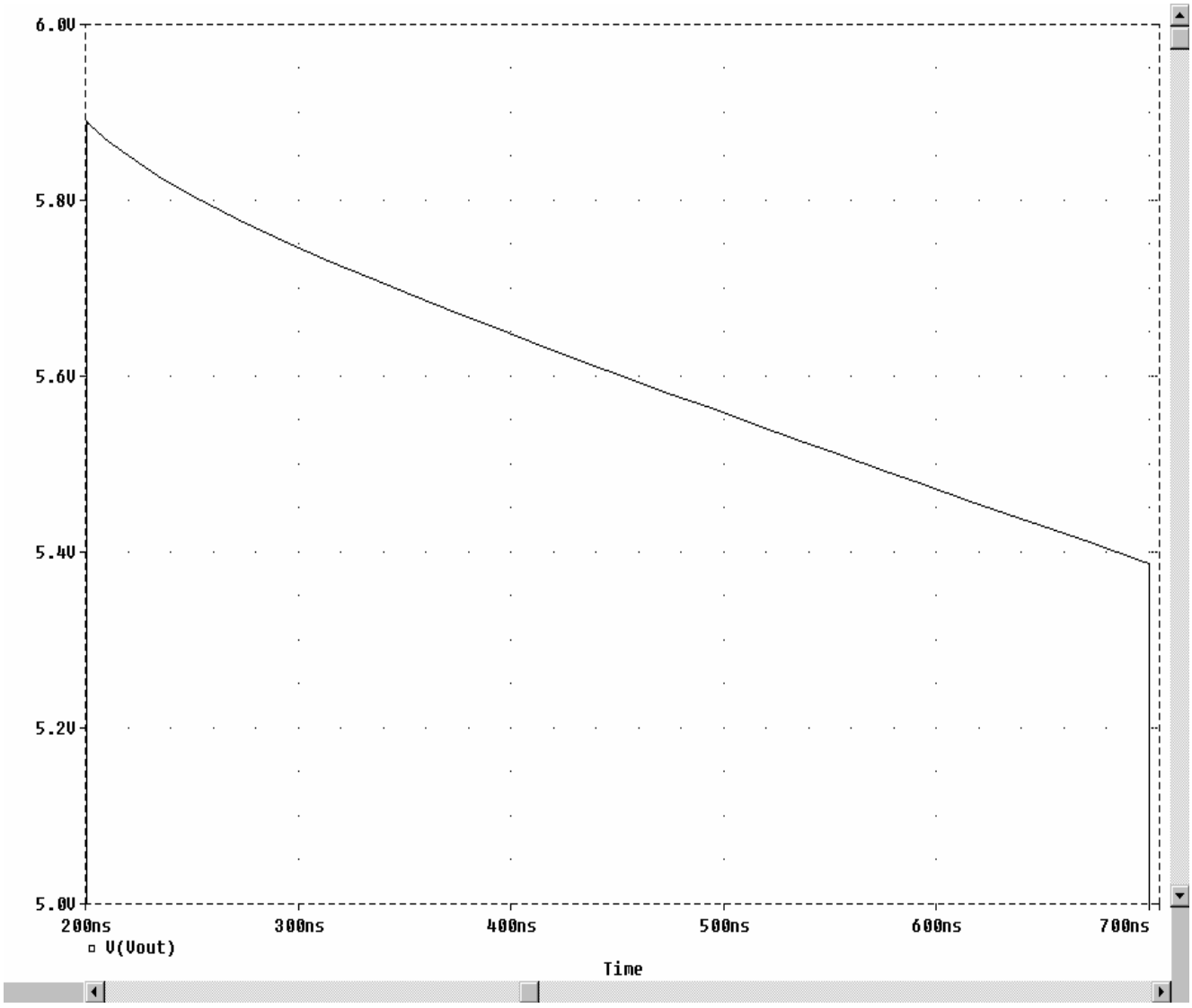


Figure 4.9 Linear Decrease in Voltage at the Capacitor.

Time-to-Voltage Converter (TVC):

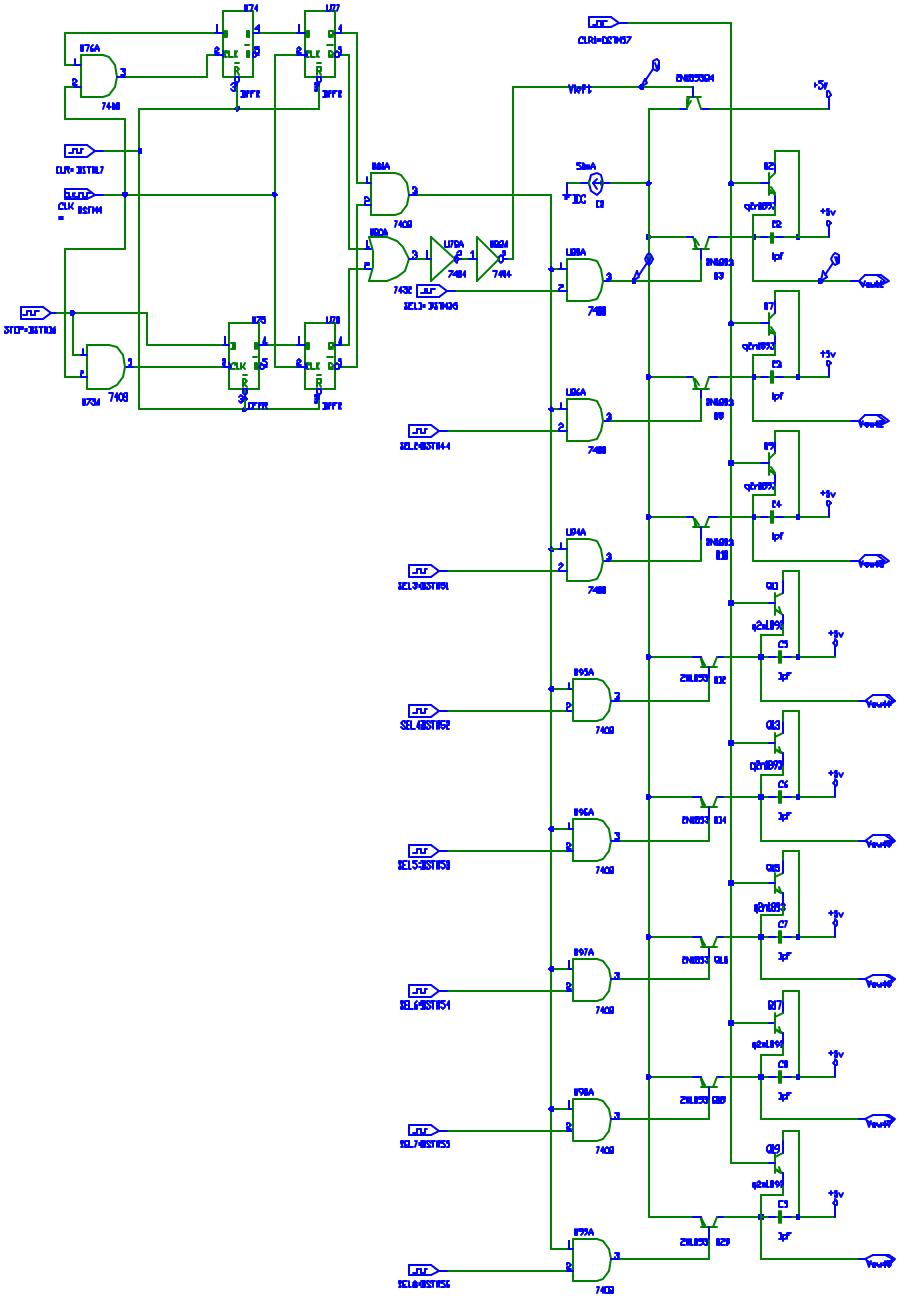


Figure 4.10 8-Channel TVC

Basic Design Considerations: A schematic diagram of the system configuration is shown in Figure 4.10. The circuit inputs are the digital START and STOP signals which mark the beginning and end of the time interval to be measured with a CLK of 250 MHz. These signals are fed to a Width Generator that generates a digital pulse which is ON for the duration of the time interval. Eight other digital signals SELECT1-SELECT8 are used to enable one of the eight storage capacitors (C1-C8) via the AND gates. The complementary outputs of the Width Generators are then used to steer the current Io from M0 to Mn and then back to M0. The resulting voltage on the capacitor will be proportional to the time of charging, and will obey the equation:

$Vc = Io/C * t$

where t is the time interval being measured and C = C1 = C2 = ….= C8. Once a capacitor has been charged, the next capacitor is set up to receive the next measurement.

Time-to-Voltage Converter: A detailed circuit for single-channel TVC is shown in Figure 4.11. Initially, switch S1 is closed and opened to precharge the capacitor to VDD. After a measurement, the resulting output voltage can be read nondestructively via an appropriate output buffer. Nominal values for Io and C1 are 50 microA and 1pF, respectively.

The voltages at the gate of M0 and M1 are digital signals which switch differentially between ground and VDD. It is important to note that the two transistors M0 and M1 do not operate as MOS analog switches in the linear region. Instead, these transistors operate in the saturated region, and the circuit functions similar to a differential pair. If M0 and M1 were operated as pass transistors, then an uncertainty would exist at the instant of switching: first, if both

transistors were off momentarily, the current source would be connected to a floating node, resulting in a large change in the node voltage; alternatively, if both transistors were on momentarily, a sneak path would exist between the capacitor and Vdd, thus bleeding off of the capacitor. By operating in the saturated

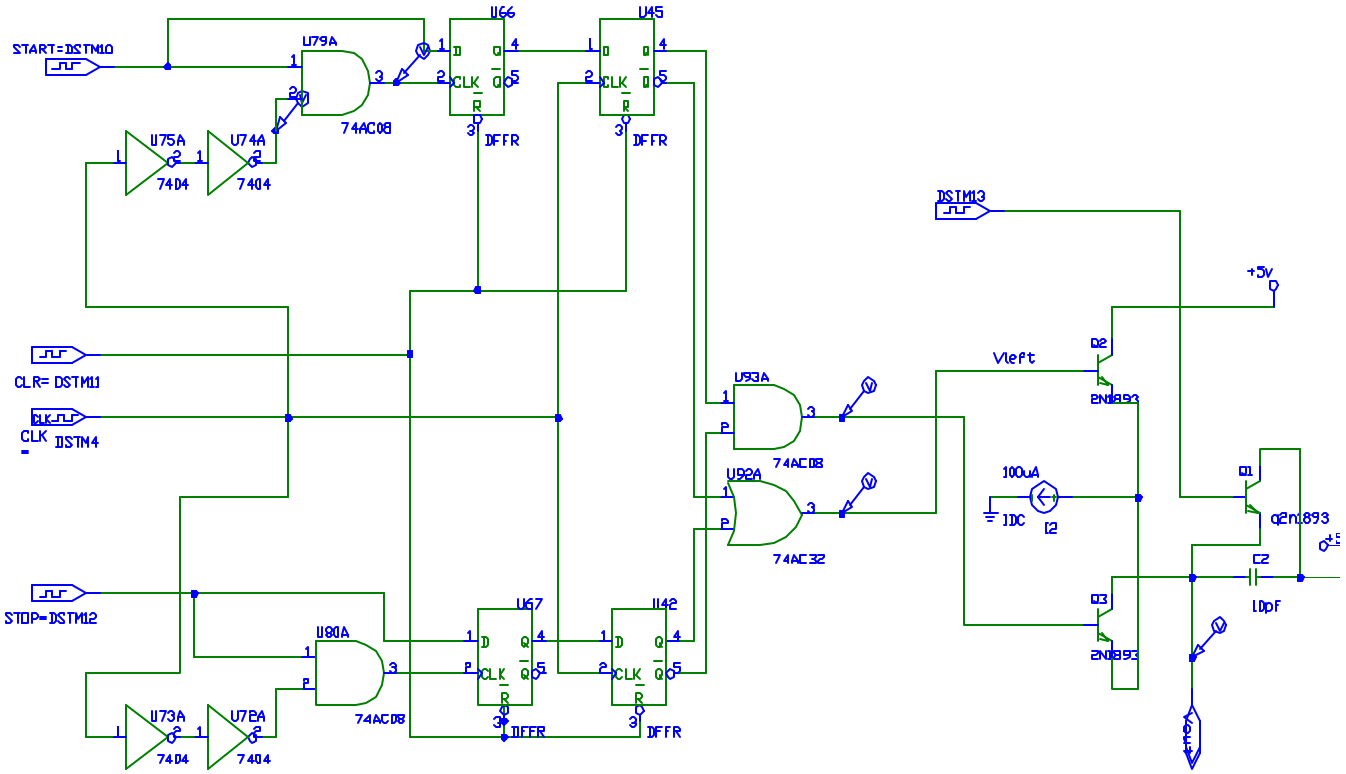


Figure 4.11 Single Channel TVC

region, the first problem still exists, but the second problem is eliminated due to the inherent current-flow characteristics of the saturated transistor. Since, the drain current in saturation can only flow in one direction (unlike the linear transistor, where the current direction depends on the drain-source voltage), no short-circuit condition will exist. Thus, at the beginning of the measurement, the following chain of events takes place in rapid succession:

- Initially, M0 is ON, M1 is OFF, and no current flows to the capacitor.
- then, after the START signal is received, M1 is turned ON before M0 is turned OFF, and the two transistors supply current for a short time.
- finally, M0 is turned OFF completely, and all of the current flows through M1.

The resulting capacitor voltage will contain two components: a linear portion which is proportional to the time interval, and an offset which will depend on the length of the time that both transistors are ON. However, as long as stable set of overlapping gate pulses can be generated for M0 and M1 by the Width Generator, the offset will be constant and reproducible. The measurement range of the circuit is limited by two different effects. The minimum measurable time will be determined by the smallest time difference between START and STOP that can be resolved by the Width Generator, which is about 25ns. It should be noted that maximum measurement interval could be arbitrarily increased or decreased by adjusting Io, which would cause an inverse change in time resolution as well. Alternatively, the measurement range could be increased by initially charging the storage capacitor to a higher voltage.

Several parasitic elements are important in the design of the TVC. The capacitance at the sources of the transistors is most significant. This capacitance (cumulatively referred to as Cp) consists of the source-bulk junctions of M0 and M1, the drain-bulk junction of the current source, and the interconnections wire between these three circuit elements. Other important parasitics include the gate-source and gate-drain overlap capacitance and the channel charge. Figure 4.12 shows the simulation results of single channel TVC.

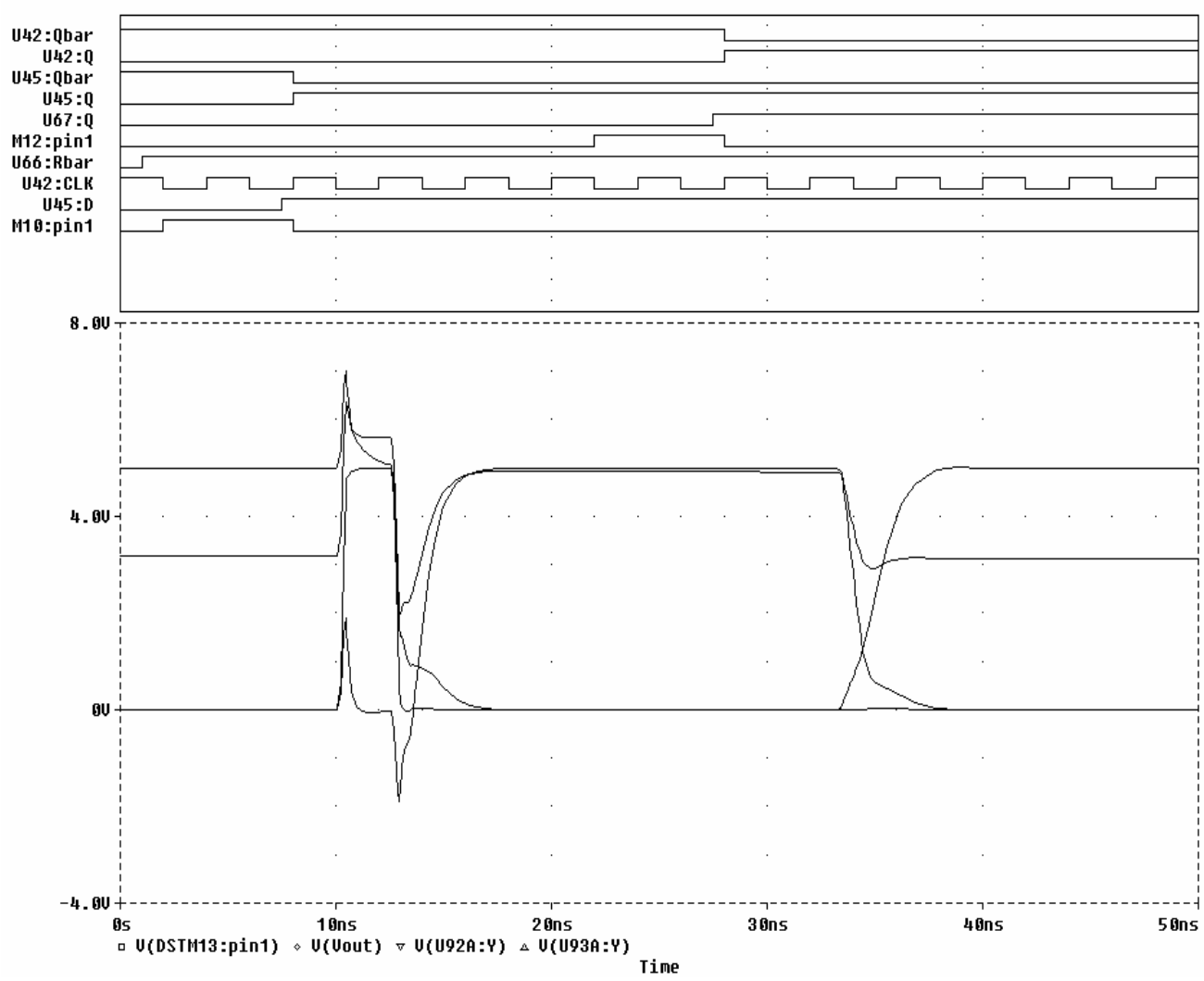


Figure 4.12 Simulation Results of Single Channel TVC.

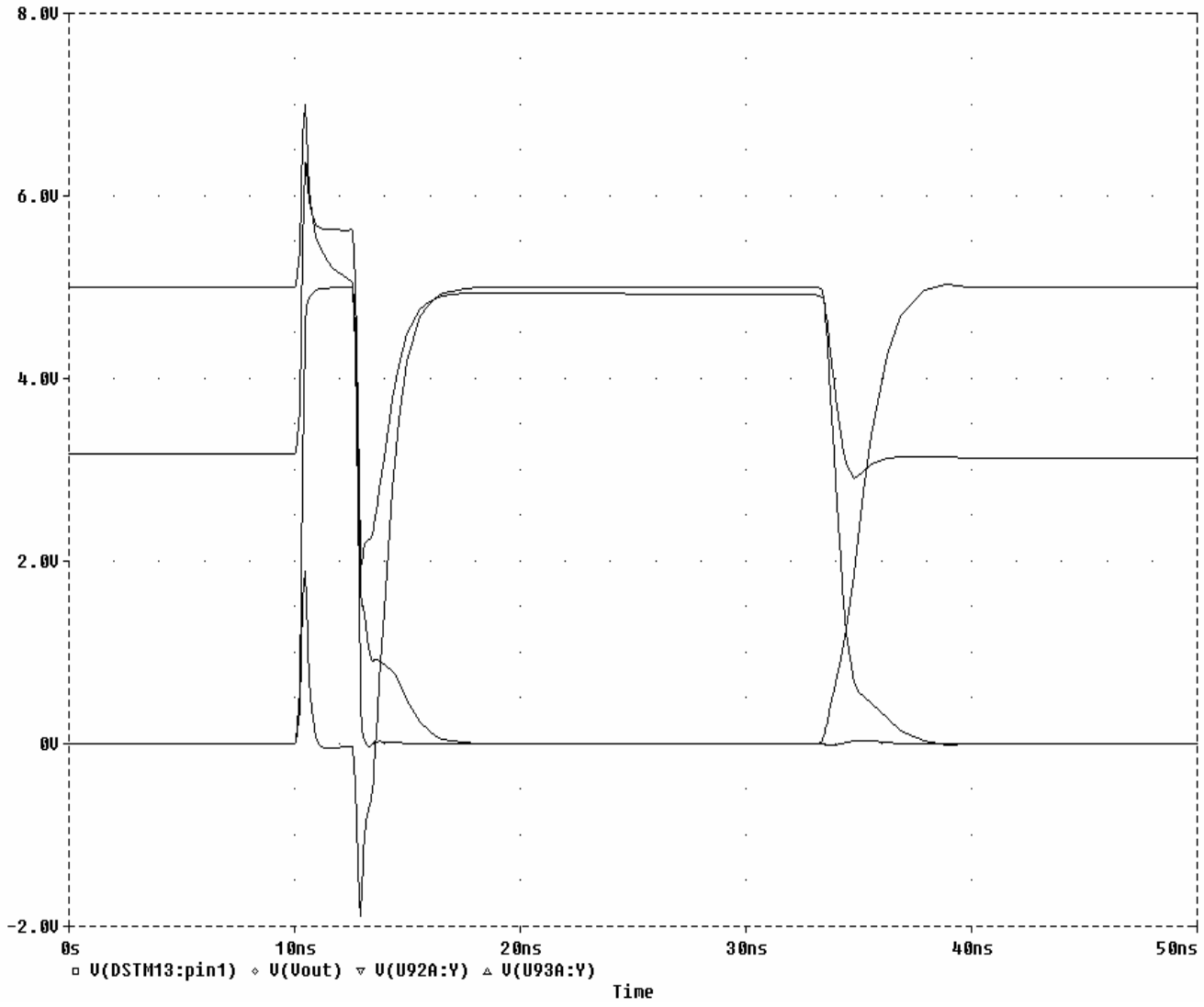


Figure 4.13 Vout, Vleft and Vright (Single-Channel TVC)

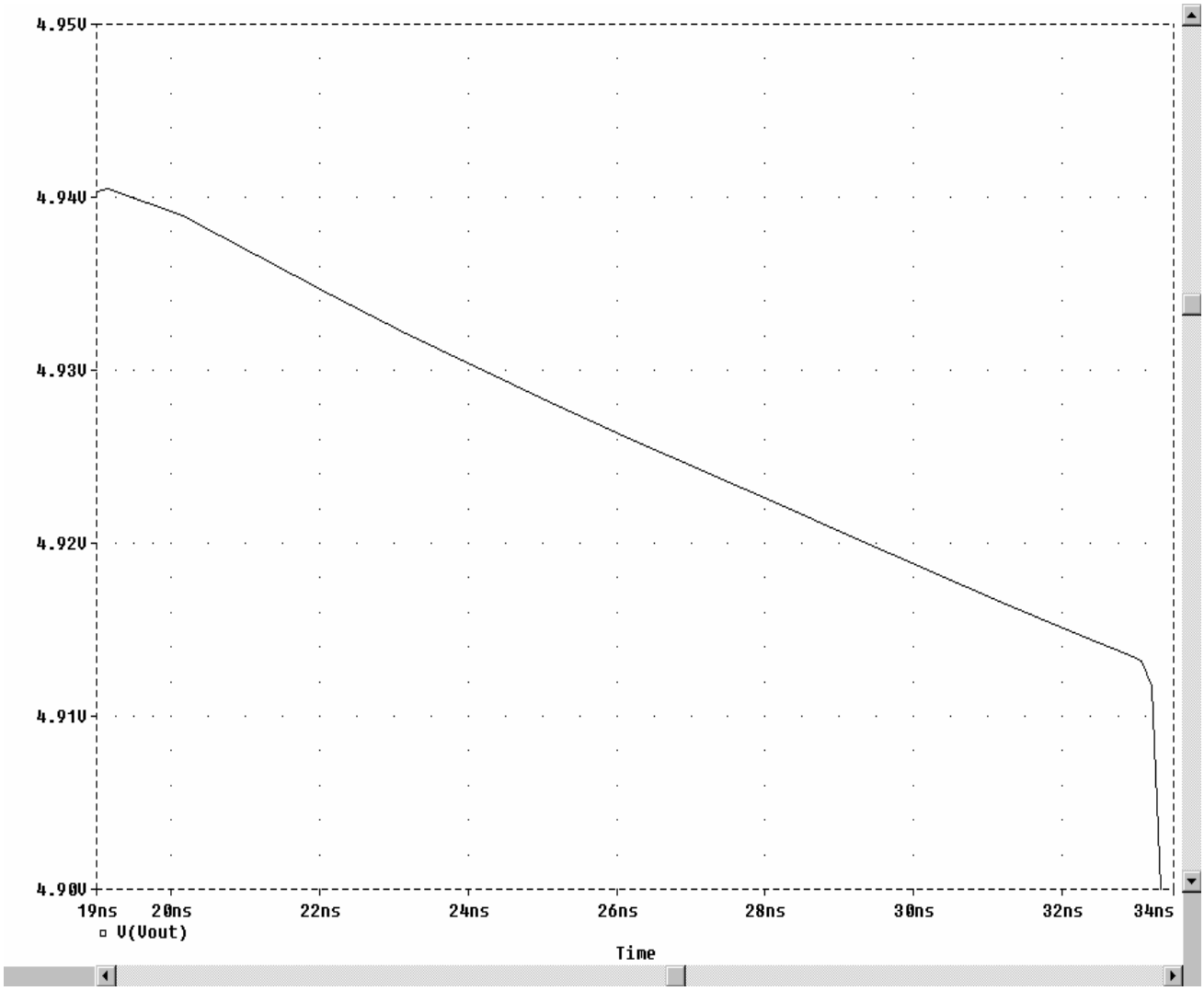


Figure 4.14 Zoomed-in Vout

As shown in Figure 4.13, two inputs Vleft and Vright are not ideal inputs as discussed earlier but they do have some noise and that is because this is very high-speed (250Mhz) circuit. Figure 4.14 shows the linear curve of decreased voltage across the capacitor. A linear range is 19ns to 34ns.

A large capacitor can also slow down the circuit by limiting the slew rate at the source node (source node a point where two transistors and current source is connected). This in turn will cause the transient effects due to switching to take longer to die out. Thus, the size of the capacitor must be minimized because it will affect the linearity of the time measurement for small intervals due to the nonlinear charging of the storage capacitor during the transient. This minimization is accomplished by carefully controlling the transistor source areas and interconnect area at source node.

Analog Memory:

This small section gives useful description for 8-channel TVC. In other words, it gives hints for the future work. The 8-channel analog memory is easily implemented by adding more transistor-capacitor combinations on the right side of single-channel version as shown in Figure 4.10. Thus, in theory, an arbitrary number of TVC's can be integrated into a single system, limited by the added parasitic capacitance at source node. SELECT1-SELECT8 are used to select among the channels. The matching of components between channels is important in order to maintain a constant level of performance across the system. However, it is more critical for certain components to be matched than others. In particular, The hold capacitors must be matched as closely as possible because variations in the absolute value of the capacitor will have

a first-order effect on the output voltage. Capacitor matching can be achieved through the use of various layout techniques, including common-centroid geometry. Matching of the other components is less critical as the effects will be second order. In the case of switching transistors, process variations will cause changes in W, L and Vt of transistors. However, since the transistor current is always set by Io, the variations will cause a change in Vgs; this will in turn cause fluctuations on Vs, but will not affect accuracy. Threshold (Vt) variations can also affect the logic at the gates of the switching transistors. These variations can cause a shift in the logic levels of the gate, corresponding to an uncertainty of the precise instant when the gate will switch from high to low. In order to reduce this uncertainty, the logic which drives the current is always overdriven so that the rise times will be as short as possible. This reduces any logic skew to a minimum.

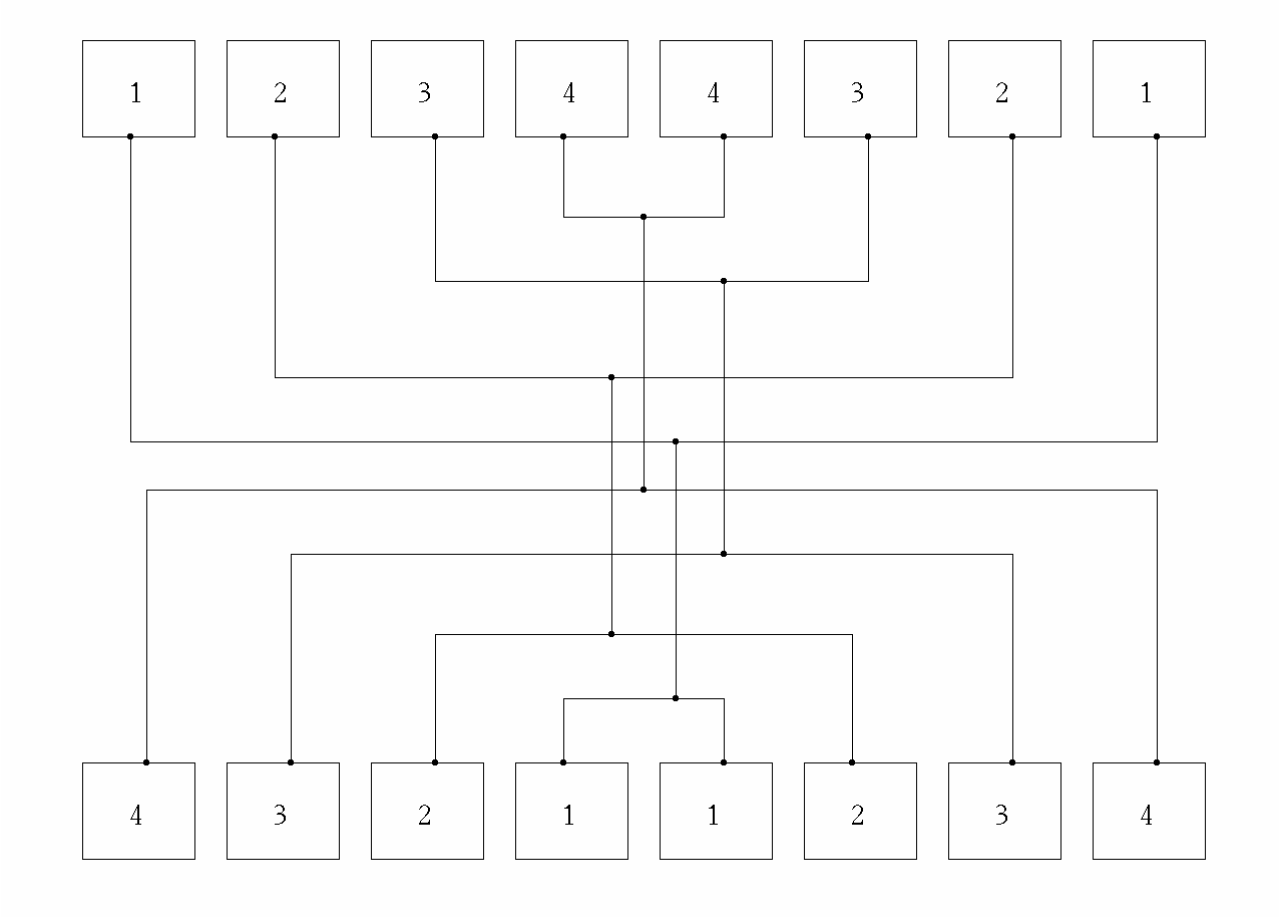


Figure 4.15 Common-centroid layout of the storage capacitors.

Applications:

A conversion from Time-to-Voltage Converter (TVC) to Time-to-Digital Converter (TDC) is very simple and it can be achieved by putting A/D converter at the end of TVC. Some of the applications below may describe TDC as a major source.

The Tracing Back of Physical Values to a Time Difference:

Time-to-Digital Converters are very universal products, which is also why the spectrum of applications is so broad. A TDC can be applied in all areas where physical values can be traced

back to time differences, and this occurs far more often than one would expect. Several examples where this takes place are:

· Tracing back the difference between 2 or more marks to a time difference by measuring the time of flight (light, microwave, sound).

· Tracing back the flowing amount of gas or liquid to a time difference by measuring the ultrasonic time of flight and considering the Doppler effect.

· Tracing back the temperature to a time difference by measuring the time consistency of a RC-unit with a temperature dependent resistor.

· Tracing back a weight to a time difference by measuring the time consistency of a RC-unit with a temperature dependent capacitor.

The digitalization of physical values take place - as always in the measuring technique - via an in-between value, which is time in the case of the TDCs. Figure 10 displays the correlation in a graphic manner.

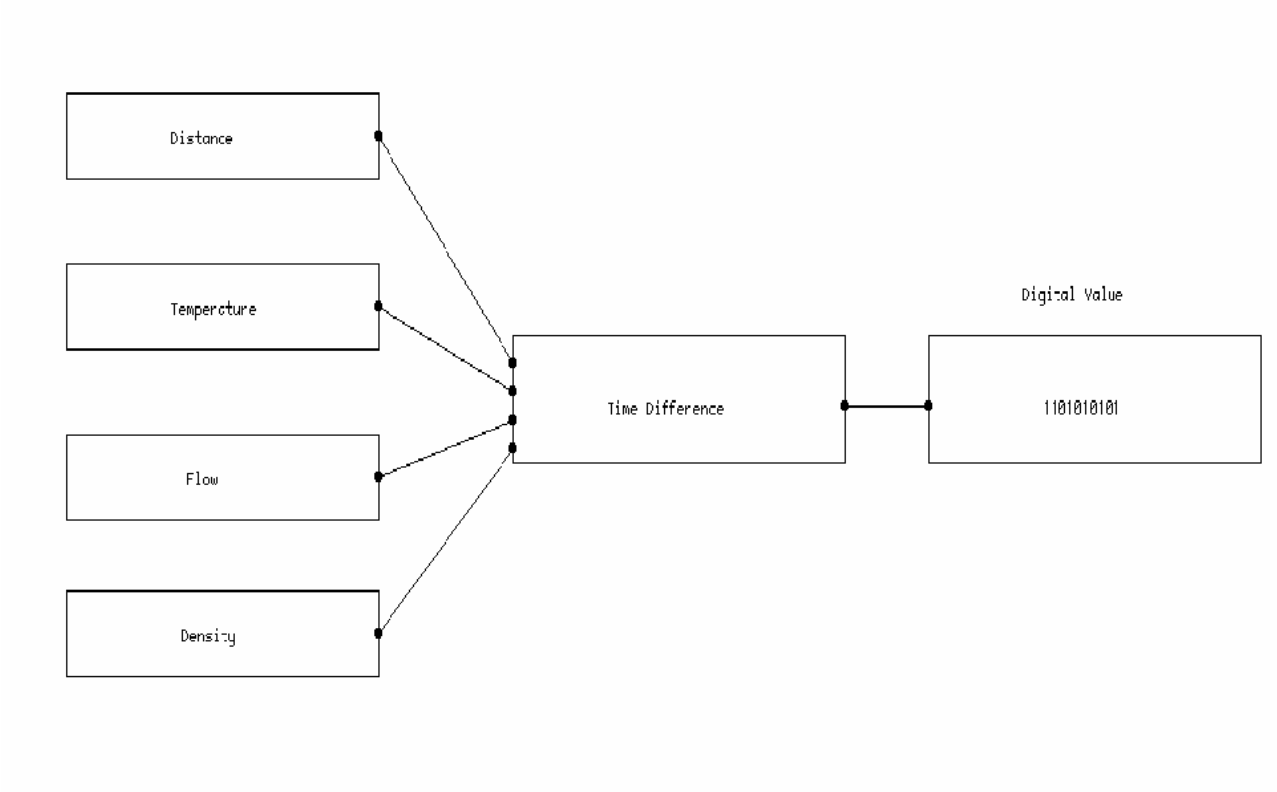


Figure 10 Graphical Correlation

Ultrasonic Flow Measurement:

Technical solutions without mechanical moving parts are a guarantee for high reliability, little need for maintenance and a long life. Measuring procedures have been established, which are built on the knowledge of flow measurement techniques of fluids and gasses. One of these procedures is the measurement of the amount of flow via ultrasound. Fig. 13 displays the principal course of such a

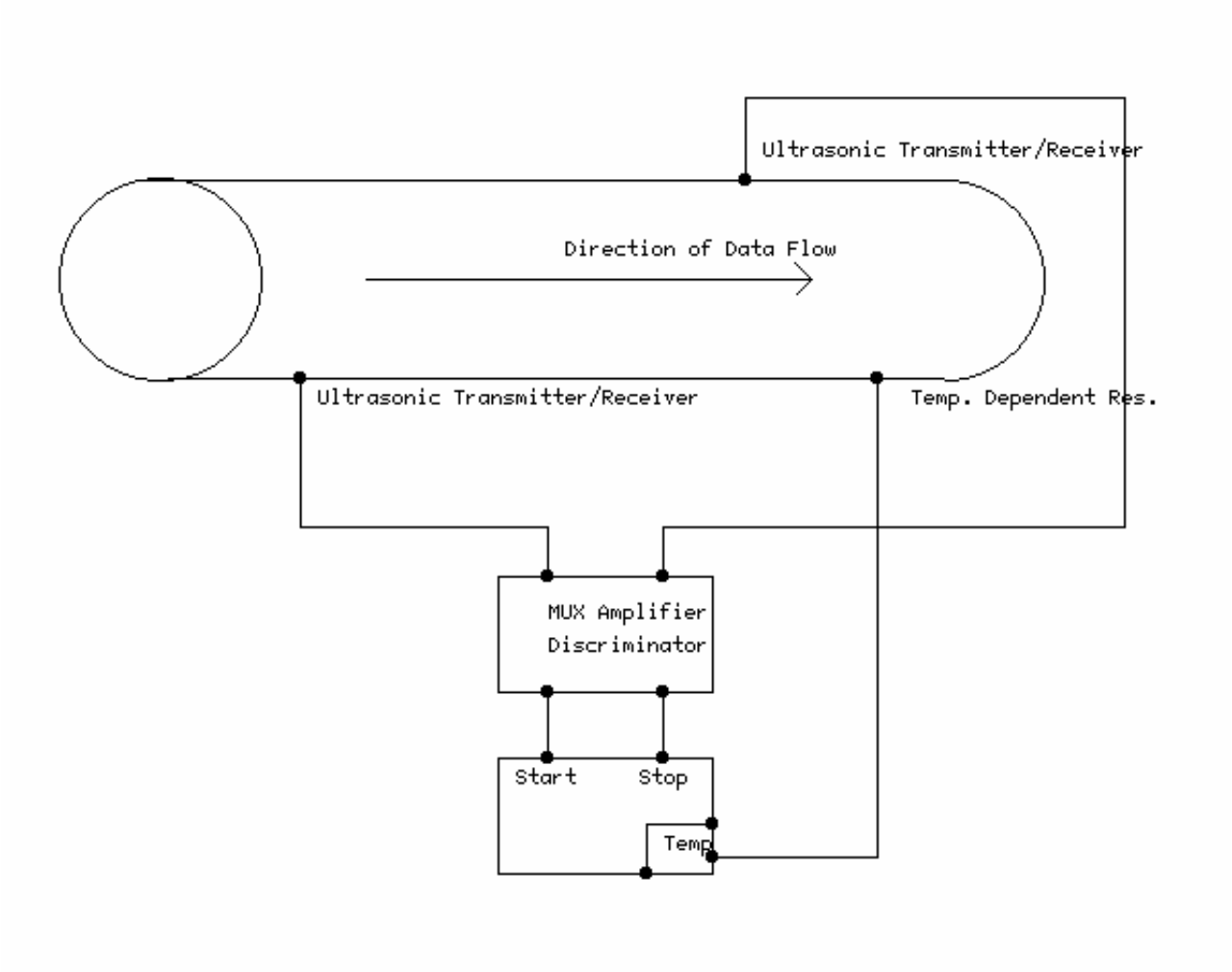


Figure 9 Ultrasonic Flow Measurement

measurement is sent out in the flow direction of the medium and a second against the flow direction. The flight time of the ultrasonic impulse between transmitter and receiver is then measured. Piezos are generally used as ultrasonic transmitters and receivers. They are suited excellently for both functions. Due to the Doppler effect a time difference, which is a measure

for the flow, emerges according to flow speed of the medium. TDCs are applied to measure the flight time of the ultrasonic impulse. Typical characteristics for this application are the rather large time differences, ranging from app. 20 µs - 1 ms, the time in which the ultrasonic impulse is on it's way. The flight time differences due to the Doppler effect only amount to few nanoseconds. This small time difference with simultaneously high measuring time must have a resolution of up to 1-2 %. This results in requirements for the resolution ranging from 100 ps or less. In order to reach this resolution one has to fall back to the generation of averages. The basic resolution of the TDCs is much higher than with individual measurements, but the systematic errors are not permitted to exceed this value. Otherwise, malicious errors will add up over time. The necessary dynamic area of measurements results in 22-24 Bit. These high dynamics, which create several problems for some AD converters, is part of everyday business for TDCs and can be conquered without any problems. One generally works with the measuring procedure of the predivider, which permits the measurement of large time differences while the resolution remains the same. There is no process limits in the dynamic area. If one only relies on the relative variation of two measurements, like it is the case with flow measurements, one can use customary quartz oscillators as predivider clocks, since the frequency of the oscillator is brought out by generating differences.

If one wants to measure the absolute time with precision, which is better than the resolution, one would need to use a frequency norm as a predivider clock at app. 20 Bit, so that measurements with an absolute precision of up to 30 Bits can be made possible. A good example which displays the type of precision one would be able to measure time with: If an AD converter

would exist, which has a similar dynamic range and identical precision, it would be able to measure voltage with 5nV precision in a measurement area of 5V. Such an AD-converter does not exist. Modern TDCs are however capable of measuring this in such a situation.

In addition, permissible small power consumption plays a major role for flow techniques. Much equipment is battery operated, so that a long-time flawless operation without battery change is indispensable, and this can be realized with CMOS TDCs. The ultrasonic speed in the medium that needs to be measured is dependent upon temperature. A regular measurement of the time is therefore necessary. Since a temperature measurement can easily be traced back to a time difference measurement, it becomes apparent to have a TDC take care of the temperature measurements.

Magnetostrictive Positioning:

The contact-less registration of positions, driving movements etc. are a required method for procedure automation, so that re-registration can take place in the actual situation of the system. Often precision in the µm range is needed. The systems need to be robust, with little required maintenance and (of course) inexpensive. Several years ago a method was established, where ultrasonic delay time on a wire was used to determine a position.

A figure 14 shows the principal construction of such positioning systems.

A wire is strung over the entire length of the positioning equipment. A strong permanent magnet is situated at a certain position.

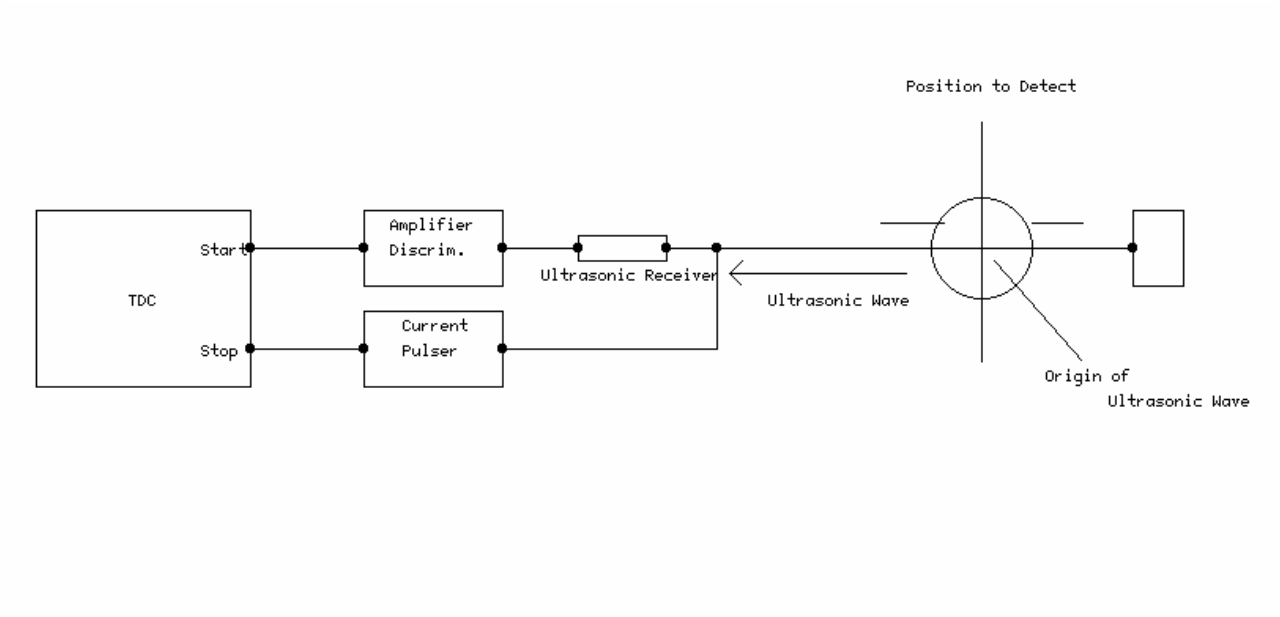


Figure 12 Magnetostrictive Positioning

A wire is strung over the entire length of the positioning equipment. A strong permanent magnet is situated at a certain position.

A current pulse is sent by wire at regular intervals (i.e. 1k/sec) with several amperes and a length of several microseconds. This high current pulse results in a magnetic field around the wire, which changes rapidly. The position where the magnetic field of the permanent magnet is situated, an energy is generated that winds around the wire and generates an ultrasonic wave, which extends into both directions of the wire (at app. 2,700 m/sec) with ultrasonic speed. In the dead side the ultrasonic impulse is absorbed, whereas the "active" side detects the impulse via ultrasonic receiver, which also converts it into a digital receiver impulse. The time between the start of the pulse and receiving of the receiver impulse is measured via TDC so that the

location of the permanent magnet can be determined. In cases where the permanent magnet moves with the positioned parts (i.e. a milling cutter) the equipment can be positioned precisely. This procedure permits positioning precision of 2-5 µm, which equals an ultrasonic delay time of 800 ps - 2 ns, which is exactly the range that a TDC has to offer. With a high resolution, which is located slightly above the requirements, TDCs are ideal for time difference measurements. In addition, these chips are extremely small in comparison to past solutions, are ten times as precise while using only 1/10 of the power. All of these features contribute to the increase in distribution of TDCs.

Laser Distance Measurements: A laser is best used for measuring all types of differences. Due to its precise focus, it's independence regarding expanding speed of environmental influences and several additional advantages; it offers modern meter measures for longer distances as well. Especially with short laser pulses, the possibility of using high performance power (several Watt) without endangering people is given. Its large operation range consists of scanners for surfaces or field guide observation, up through distance regulators ranging from a few meters up to several kilometers. TDCs can naturally be used in classical methods of determining the flight time of laser impulses. Also, other methods of distance determination via laser can easily be carried out by a TDC, i.e. the measurement of a phase of modulated signals, with little power consumption and without great expense. Light or more general electromagnetic waves are - as we all know (till the contrary is proven) - the fastest thing nature has to offer at the moment. Any type of information or energy transport can take place at the most with the speed of light. Measurements of the speed of light in a vacuum results in speeds of 299,792.485

km / sec. If one plan to measure distances covered by this speed precisely with a delay time procedure, the result will be very small measured times. If one wants to achieve precision similar to a normal measuring tape that is set at 1 mm, then this corresponds with a light delay time of 3 ps. If one measures the distance in the reflection procedure, the time is increased due to doubled distance up to 6 ps. This time difference is far lower than what digital delay time TDCs can presently offer regarding resolution. Laser distance measurement therefore expects to be a real challenge for TDCs regarding resolution and precision. Luckily it in not as difficult as it seems:

Often resolutions in the range of several centimeters are sufficient for typical laser applications, so that we are already in the familiar area. One can often fall back on averaged results, so that justifiable averaged rates can achieve the requested precision. Also, the time difference measurements are not the only source of error. The rest of the signal path incl. measuring distances results in 30-50 ps standard deviation during good measurement conditions and good circuit construction. The standard deviation is therefore in the same range as today's good TDCs, so that even a 5 ps TDC can't achieve a breakthrough, but can only minimize sources for errors, without drastically improving the entire result.

While considering all of these conditions TDCs are ideal for the laser distance measuring techniques via pulse delay time measurement and are also increasingly applied in this area.

Chapter 5
Conclusion

This thesis first describes how important the time measurement is nowadays. An evolution in time measurement techniques from resolution of seconds few years ago to nanoseconds and picoseconds nowadays has attracted lots of researchers.

A chapter two describes some of the important issues in designing TVC, which could be very useful. Next, a Width Generator, a very important component in TVC circuitry which captures digital time signals and converts them into two digital complementary pulses which can fed to and analog circuit. Finally chapter 4 explains the whole TVC circuit design.

A Time-to-Voltage Converter circuit designed here has some tradeoffs. After days of research and circuit design, I came up with a very simple circuit for TVC, which solves the same purpose as commercial TVCs. In order to get simplicity, I have lost some of the accuracy and resolution. A TVC designed here has resolution of about 300-400ns (basically it is in the range of few 100's of ns) compare to most of the TVCs nowadays with few 10's of resolution. On the other hand it has very less complexity and easy implementation. This also opens a door for future work to get resolution of 10's ns with this implementation of circuit design by massaging a circuit a bit more.

Thus, a successful design of a high-precision low-power Time-to-Voltage Converter and analog memory has been presented here. A CMOS time-integration circuit based on saturation-region current switching performs very well, demonstrating that CMOS is well suited for this type of application. The TVC circuit presented here works at 40MHz clock but it could be increased to 100MHz or more to finer resolution. The TVC architecture is also expandable to implement the analog memory. Future versions of the chip could be expanded to 16 or more channels, limited by the added parasitic capacitance at one critical node. The measured time resolution of 8-channel TVC is in few nanoseconds.

The Time-to-voltage Converter whose design and construction is described here is intended as the time interval measurement unit of a laser-range finder employing the time-of-flight method. For short range applications these range finders may have a semiconductor laser diode as the light transmitting device. The method can be used for verity of distance measurement purposes in industry.

APPENDIX A

Below is the output of PSPICE and also a PSPICE netlist code for Width Generator.

**** 06/05/99 13:17:20 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\63digital.sch

**** CIRCUIT DESCRIPTION

****
******

* Schematics Version 8.0 - July 1997
* Sat Jun 05 13:14:20 1999

** Analysis setup **
.tran 5ns 50ns
.LIB "C:\cpp\lib\Breakout.lib"
.LIB "C:\cpp\lib\Eval.lib"
.LIB "C:\cpp_setup.lib"
.LIB "nom.lib"
.STMLIB "C:\cpp\Projects\tvc.stl"
.STMLIB "tvc_pre.stl"
.STMLIB "tvc_pre_1.stl"
.STMLIB "tvc_pre_2.stl"
.STMLIB "tvc_pre_2_sync_100mhz.stl"

* From [SCHEMATICS NETLIST] section of msim.ini:
.lib "C:\MSim_8\UserLib\64msdout.lib"
.lib "C:\MSim_8\UserLib\64msdnor.lib"
.lib "C:\MSim_8\UserLib\64msdinp.lib"

```
.lib "C:\MSim_8\UserLib\64msdhig.lib"
.lib "C:\MSim_8\UserLib\m63z818.lib"
.lib "C:\MSim_8\UserLib\SIBUFIF.lib"
.lib "C:\MSim_8\UserLib\BT8IF.lib"
.lib "C:\MSim_8\UserLib\BT8RIF.lib"
.lib "C:\MSim_8\UserLib\BT8HIF.lib"
.lib "C:\MSim_8\UserLib\BT4RIF.lib"
.lib "C:\MSim_8\UserLib\BT4IF.lib"
.lib "C:\MSim_8\UserLib\BIBUFIF.lib"
.lib "C:\MSim_8\UserLib\BD4CIF.lib"
.lib "C:\MSim_8\UserLib\BT4HIF.lib"
.lib "C:\MSim_8\UserLib\BD8RCIF.lib"
.lib "C:\MSim_8\UserLib\BD8HCIF.lib"
.lib "C:\MSim_8\UserLib\BD8CIF.lib"
.lib "C:\MSim_8\UserLib\BD4RCIF.lib"
.lib "C:\MSim_8\UserLib\IBUFIF.lib"
.lib "C:\MSim_8\UserLib\BD4HCIF.lib"
.lib "nom.lib"

.INC "63digital.net"

**** INCLUDING 63digital.net ****
* Schematics Netlist *

.EXTERNAL OUTPUT Vleft
.EXTERNAL OUTPUT Vright


X_U92A         $N_0001 $N_0002 Vleft $G_DPWR $G_DGND 74AC32 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U93A         $N_0003 $N_0004 Vright $G_DPWR $G_DGND 74AC08 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U75A         $N_0005 $N_0006 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U79A         $N_0007 $N_0008 $N_0009 $G_DPWR $G_DGND 74AC08 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U45        $N_0011 $N_0005 $N_0010 $N_0003 $N_0001 $G_DPWR $G_DGND
DFFR
X_U42        $N_0012 $N_0005 $N_0010 $N_0002 $N_0004 $G_DPWR $G_DGND
DFFR
X_U67        $N_0014 $N_0013 $N_0010 $N_0012 $N_0015 $G_DPWR $G_DGND
DFFR
X_U80A         $N_0014 $N_0016 $N_0013 $G_DPWR $G_DGND 74AC08 PARAMS:
```

```
+ IO_LEVEL=0 MNTYMXDLY=0
X_U72A        $N_0017 $N_0016 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U73A        $N_0005 $N_0017 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U66        $N_0007 $N_0009 $N_0010 $N_0011 $N_0018 $G_DPWR $G_DGND
DFFR
U_DSTM10        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0007
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 2ns 1
+ 8ns 0
+ 7000ns 0
+ 7200ns 0
+ 15000ns 0
U_DSTM12        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0014
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 22ns 1
+ 28ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM11        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0010
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 1ns 1
+ 100ns 1
+ 300ns 1
+ 15000ns 1
U_DSTM4        STIM(1,1) $G_DPWR $G_DGND $N_0005 IO_STM IO_LEVEL=0
+ 0 0
+ +0ns 1
+REPEAT FOREVER
```

```
+ +2ns 0
+  +2ns 1
+ ENDREPEAT
X_U74A         $N_0006 $N_0008 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0

**** RESUMING 63digital.cir ****
.INC "63digital.als"



**** INCLUDING 63digital.als ****
* Schematics Aliases *

.ALIASES
X_U92A         U92A(A=$N_0001 B=$N_0002 Y=Vleft PWR=$G_DPWR
GND=$G_DGND )
X_U93A         U93A(A=$N_0003 B=$N_0004 Y=Vright PWR=$G_DPWR
GND=$G_DGND )
X_U75A         U75A(A=$N_0005 Y=$N_0006 PWR=$G_DPWR GND=$G_DGND )
X_U79A         U79A(A=$N_0007 B=$N_0008 Y=$N_0009 PWR=$G_DPWR
GND=$G_DGND )
X_U45          U45(D=$N_0011 CLK=$N_0005 Rbar=$N_0010 Q=$N_0003
Qbar=$N_0001
+  PWR=$G_DPWR GND=$G_DGND )
X_U42          U42(D=$N_0012 CLK=$N_0005 Rbar=$N_0010 Q=$N_0002
Qbar=$N_0004
+  PWR=$G_DPWR GND=$G_DGND )
X_U67          U67(D=$N_0014 CLK=$N_0013 Rbar=$N_0010 Q=$N_0012
Qbar=$N_0015
+  PWR=$G_DPWR GND=$G_DGND )
X_U80A         U80A(A=$N_0014 B=$N_0016 Y=$N_0013 PWR=$G_DPWR
GND=$G_DGND )
X_U72A         U72A(A=$N_0017 Y=$N_0016 PWR=$G_DPWR GND=$G_DGND )
X_U73A         U73A(A=$N_0005 Y=$N_0017 PWR=$G_DPWR GND=$G_DGND )
X_U66          U66(D=$N_0007 CLK=$N_0009 Rbar=$N_0010 Q=$N_0011
Qbar=$N_0018
+  PWR=$G_DPWR GND=$G_DGND )
U_DSTM10         DSTM10(PIN1=$N_0007 )
U_DSTM12         DSTM12(PIN1=$N_0014 )
U_DSTM11         DSTM11(PIN1=$N_0010 )
U_DSTM4         DSTM4(PWR=$G_DPWR GND=$G_DGND 1=$N_0005 )
X_U74A         U74A(A=$N_0006 Y=$N_0008 PWR=$G_DPWR GND=$G_DGND )
```

```
_    _(Vleft=Vleft)
_    _(Vright=Vright)
_    _($G_DGND=$G_DGND)
_    _($G_DPWR=$G_DPWR)
.ENDALIASES


**** RESUMING 63digital.cir ****
.probe


.END



**** 06/05/99 13:17:20 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

 * C:\cpp\Projects\63digital.sch


 ****     Digital Gate MODEL PARAMETERS

**********************************************************************
******



            D_04          D_AC32        D_AC08
     TPLHMN    4.800000E-09    1.000000E-09    1.000000E-09
     TPLHTY   12.000000E-09    5.500000E-09    5.500000E-09
     TPLHMX   22.000000E-09    8.500000E-09    8.500000E-09
     TPHLMN    3.200000E-09    1.000000E-09    1.000000E-09
     TPHLTY    8.000000E-09    5.000000E-09    5.500000E-09
     TPHLMX   15.000000E-09    7.500000E-09    7.500000E-09



**** 06/05/99 13:17:20 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

 * C:\cpp\Projects\63digital.sch
```

**** Digital Edge Triggered FF MODEL PARAMETERS

****
******

D_PLD_EFF
TPCLKQLHMN 0
TPCLKQLHTY 0
TPCLKQLHMX 0
TPCLKQHLMN 0
TPCLKQHLTY 0
TPCLKQHLMX 0
TPPCQLHMN 0
TPPCQLHTY 0
TPPCQLHMX 0
TPPCQHLMN 0
TPPCQHLTY 0
TPPCQHLMX 0
TWCLKLMN 0
TWCLKLTY 0
TWCLKLMX 0
TWCLKHMN 0
TWCLKHTY 0
TWCLKHMX 0
TWPCLMN 0
TWPCLTY 0
TWPCLMX 0
TSUDCLKMN 0
TSUDCLKTY 0
TSUDCLKMX 0
TSUPCCLKHMN 0
TSUPCCLKHTY 0
TSUPCCLKHMX 0
THDCLKMN 0
THDCLKTY 0
THDCLKMX 0
TSUCECLKMN 0
TSUCECLKTY 0

TSUCECLKMX 0
THCECLKMN 0
THCECLKTY 0
THCECLKMX 0

**** 06/05/99 13:17:20 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\63digital.sch

**** Digital IO MODEL PARAMETERS

****

IO_STM IO_STD IO_AC IO_PLD
INLD 5.000000E-12
DRVL 0 104 18.7
DRVH 0 96.4 24.5
AtoD1 AtoD_STD AtoD_AC
AtoD2 AtoD_STD_NX AtoD_AC_NX
AtoD3 AtoD_STD AtoD_AC
AtoD4 AtoD_STD_NX AtoD_AC_NX
DtoA1 DtoA_STM DtoA_STD DtoA_AC
DtoA2 DtoA_STM DtoA_STD DtoA_AC
DtoA3 DtoA_STM DtoA_STD DtoA_AC
DtoA4 DtoA_STM DtoA_STD DtoA_AC
TSWHL1 1.373000E-09 477.000000E-12
TSWHL2 1.346000E-09 477.000000E-12
TSWHL3 1.511000E-09 477.000000E-12
TSWHL4 1.487000E-09 477.000000E-12
TSWLH1 3.382000E-09 470.000000E-12
TSWLH2 3.424000E-09 470.000000E-12
TSWLH3 3.517000E-09 470.000000E-12
TSWLH4 3.564000E-09 470.000000E-12
TPWRT 100.000000E+03 100.000000E+03 100.000000E+03 100.000000E+03

JOB CONCLUDED

TOTAL JOB TIME .70

APPENDIX B

Below is the PSPICE output and PSPICE netlist code for single-channel TVC.

**** 06/05/99 13:29:36 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_2ns_6_5.sch

**** CIRCUIT DESCRIPTION

****
******

* Schematics Version 8.0 - July 1997
* Sat Jun 05 12:51:52 1999

** Analysis setup **
.tran 5ns 50ns
.LIB "C:\cpp\lib\Breakout.lib"
.LIB "C:\cpp\lib\Eval.lib"
.LIB "C:\cpp_setup.lib"
.LIB "nom.lib"
.STMLIB "C:\cpp\Projects\tvc.stl"
.STMLIB "tvc_pre.stl"
.STMLIB "tvc_pre_1.stl"
.STMLIB "tvc_pre_2.stl"
.STMLIB "tvc_pre_2_sync_100mhz.stl"

* From [SCHEMATICS NETLIST] section of msim.ini:
.lib "C:\MSim_8\UserLib\64msdout.lib"
.lib "C:\MSim_8\UserLib\64msdnor.lib"
.lib "C:\MSim_8\UserLib\64msdinp.lib"
.lib "C:\MSim_8\UserLib\64msdhig.lib"

```
.lib "C:\MSim_8\UserLib\m63z818.lib"
.lib "C:\MSim_8\UserLib\SIBUFIF.lib"
.lib "C:\MSim_8\UserLib\BT8IF.lib"
.lib "C:\MSim_8\UserLib\BT8RIF.lib"
.lib "C:\MSim_8\UserLib\BT8HIF.lib"
.lib "C:\MSim_8\UserLib\BT4RIF.lib"
.lib "C:\MSim_8\UserLib\BT4IF.lib"
.lib "C:\MSim_8\UserLib\BIBUFIF.lib"
.lib "C:\MSim_8\UserLib\BD4CIF.lib"
.lib "C:\MSim_8\UserLib\BT4HIF.lib"
.lib "C:\MSim_8\UserLib\BD8RCIF.lib"
.lib "C:\MSim_8\UserLib\BD8HCIF.lib"
.lib "C:\MSim_8\UserLib\BD8CIF.lib"
.lib "C:\MSim_8\UserLib\BD4RCIF.lib"
.lib "C:\MSim_8\UserLib\IBUFIF.lib"
.lib "C:\MSim_8\UserLib\BD4HCIF.lib"
.lib "nom.lib"

.INC "tvc_final_2ns_6_5.net"

**** INCLUDING tvc_final_2ns_6_5.net ****
* Schematics Netlist *

.EXTERNAL OUTPUT Vout


X_U93A        $N_0001 $N_0002 $N_0003 $G_DPWR $G_DGND 74AC08 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U75A        $N_0004 $N_0005 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U74A        $N_0005 $N_0006 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U79A        $N_0007 $N_0006 $N_0008 $G_DPWR $G_DGND 74AC08 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U45        $N_0010 $N_0004 $N_0009 $N_0001 $N_0011 $G_DPWR $G_DGND
DFFR
X_U42        $N_0012 $N_0004 $N_0009 $N_0013 $N_0002 $G_DPWR $G_DGND
DFFR
X_U67        $N_0015 $N_0014 $N_0009 $N_0012 $N_0016 $G_DPWR $G_DGND
DFFR
X_U80A        $N_0015 $N_0017 $N_0014 $G_DPWR $G_DGND 74AC08 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U72A        $N_0018 $N_0017 $G_DPWR $G_DGND 7404 PARAMS:
```

```
+ IO_LEVEL=0 MNTYMXDLY=0
X_U73A          $N_0004 $N_0018 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U66           $N_0007 $N_0008 $N_0009 $N_0010 $N_0019 $G_DPWR $G_DGND
DFFR
U_DSTM10        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0007
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 2ns 1
+ 8ns 0
+ 7000ns 0
+ 7200ns 0
+ 15000ns 0
U_DSTM12        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0015
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 22ns 1
+ 28ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM11        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0009
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 1ns 1
+ 100ns 1
+ 300ns 1
+ 15000ns 1
U_DSTM4         STIM(1,1) $G_DPWR $G_DGND $N_0004 IO_STM IO_LEVEL=0
+ 0 0
+ +0ns 1
+REPEAT FOREVER
+ +2ns 0
+  +2ns 1
```

```
+ ENDREPEAT
Q_Q1         +5v $N_0020 Vout Q2N1893
Q_Q2         +5v $N_0021 $N_0022 Q2N1893
I_I2         $N_0022 0 DC 100uA
Q_Q3         Vout $N_0003 $N_0022 Q2N1893
X_U92A       $N_0011 $N_0013 $N_0021 $G_DPWR $G_DGND 74AC32 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
C_C2         Vout +5v  10pF
U_DSTM13     STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0020
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 10ns 1
+ 12.5ns 0
+ 215ns 0
+ 15000ns 0

**** RESUMING tvc_final_2ns_6_5.cir ****
.INC "tvc_final_2ns_6_5.als"



**** INCLUDING tvc_final_2ns_6_5.als ****
* Schematics Aliases *

.ALIASES
X_U93A       U93A(A=$N_0001 B=$N_0002 Y=$N_0003 PWR=$G_DPWR
GND=$G_DGND )
X_U75A       U75A(A=$N_0004 Y=$N_0005 PWR=$G_DPWR GND=$G_DGND )
X_U74A       U74A(A=$N_0005 Y=$N_0006 PWR=$G_DPWR GND=$G_DGND )
X_U79A       U79A(A=$N_0007 B=$N_0006 Y=$N_0008 PWR=$G_DPWR
GND=$G_DGND )
X_U45        U45(D=$N_0010 CLK=$N_0004 Rbar=$N_0009 Q=$N_0001
Qbar=$N_0011
+  PWR=$G_DPWR GND=$G_DGND )
X_U42        U42(D=$N_0012 CLK=$N_0004 Rbar=$N_0009 Q=$N_0013
Qbar=$N_0002
+  PWR=$G_DPWR GND=$G_DGND )
X_U67        U67(D=$N_0015 CLK=$N_0014 Rbar=$N_0009 Q=$N_0012
Qbar=$N_0016
+  PWR=$G_DPWR GND=$G_DGND )
```

```
X_U80A         U80A(A=$N_0015 B=$N_0017 Y=$N_0014 PWR=$G_DPWR
GND=$G_DGND )
X_U72A         U72A(A=$N_0018 Y=$N_0017 PWR=$G_DPWR GND=$G_DGND )
X_U73A         U73A(A=$N_0004 Y=$N_0018 PWR=$G_DPWR GND=$G_DGND )
X_U66          U66(D=$N_0007 CLK=$N_0008 Rbar=$N_0009 Q=$N_0010
Qbar=$N_0019
+  PWR=$G_DPWR GND=$G_DGND )
U_DSTM10         DSTM10(PIN1=$N_0007 )
U_DSTM12         DSTM12(PIN1=$N_0015 )
U_DSTM11         DSTM11(PIN1=$N_0009 )
U_DSTM4         DSTM4(PWR=$G_DPWR GND=$G_DGND 1=$N_0004 )
Q_Q1           Q1(c=+5v b=$N_0020 e=Vout )
Q_Q2           Q2(c=+5v b=$N_0021 e=$N_0022 )
I_I2           I2(+=$N_0022 -=0 )
Q_Q3           Q3(c=Vout b=$N_0003 e=$N_0022 )
X_U92A         U92A(A=$N_0011 B=$N_0013 Y=$N_0021 PWR=$G_DPWR
GND=$G_DGND )
C_C2           C2(1=Vout 2=+5v )
U_DSTM13         DSTM13(PIN1=$N_0020 )
_    _(Vout=Vout)
_    _(+5v=+5v)
_    _($G_DPWR=$G_DPWR)
_    _($G_DGND=$G_DGND)
.ENDALIASES


**** RESUMING tvc_final_2ns_6_5.cir ****
.probe


.END


**** Generated AtoD and DtoA Interfaces ****

*
* Analog/Digital interface for node $N_0003
*
* Moving X_U93A.U1:OUT1 from analog node $N_0003 to new digital node
$N_0003$DtoA
X$$N_0003_DtoA1
+ $N_0003$DtoA
+ $N_0003
```

```
+ $G_DPWR
+ $G_DGND
+ DtoA_AC
+      PARAMS: DRVH=  24.5   DRVL=  18.7   CAPACITANCE=   0
*
* Analog/Digital interface for node $N_0020
*
* Moving U_DSTM13:OUT1 from analog node $N_0020 to new digital node
$N_0020$DtoA
X$$N_0020_DtoA1
+ $N_0020$DtoA
+ $N_0020
+ $G_DPWR
+ $G_DGND
+ DtoA_STM
+      PARAMS: DRVH=   0     DRVL=   0     CAPACITANCE=   0
*
* Analog/Digital interface for node $N_0021
*
* Moving X_U92A.U1:OUT1 from analog node $N_0021 to new digital node
$N_0021$DtoA
X$$N_0021_DtoA1
+ $N_0021$DtoA
+ $N_0021
+ $G_DPWR
+ $G_DGND
+ DtoA_AC
+      PARAMS: DRVH=  24.5   DRVL=  18.7   CAPACITANCE=   0
*
* Analog/Digital interface power supply subcircuits
*

X$DIGIFPWR 0 DIGIFPWR


**** 06/05/99 13:29:36 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

 * C:\cpp\Projects\tvc_final_2ns_6_5.sch


 ****     BJT MODEL PARAMETERS
```

****
******

Q2N1893
NPN
IS 2.079000E-15
BF 167.2
NF 1
VAF 100
IKF 1.088
ISE 24.360000E-15
NE 1.357
BR 1.502
NR 1
IKR .3028
ISC 264.600000E-15
NC 1.545
NK .8271
RC .9069
CJE 58.120000E-12
VJE .5
MJE .441
CJC 45.290000E-12
VJC .5
MJC .31
TF 738.100000E-12
XTF 24.93
VTF 10
ITF 2.376
TR 1.000000E-06
XTB 1.5

**** 06/05/99 13:29:36 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_2ns_6_5.sch

**** Digital Input MODEL PARAMETERS

****
******

```
          DINSTM          DIN74AC
      FILE DSO_DTOA       DSO_DTOA
    FORMAT    6             6
  TIMESTEP  100.000000E-12  100.000000E-12
    S0NAME 0              0
     S0TSW  500.000000E-12  700.000000E-12
     S0RLO     .5          1
     S0RHI    1.000000E+03  100.000000E+03
    S1NAME 1              1
     S1TSW  500.000000E-12  700.000000E-12
     S1RLO    1.000000E+03  100.000000E+03
     S1RHI     .5          1
    S2NAME X              X
     S2TSW  500.000000E-12  700.000000E-12
     S2RLO     .429        104
     S2RHI    1.16         100
    S3NAME R              R
     S3TSW  500.000000E-12  700.000000E-12
     S3RLO     .429        104
     S3RHI    1.16         100
    S4NAME F              F
     S4TSW  500.000000E-12  700.000000E-12
     S4RLO     .429        104
     S4RHI    1.16         100
    S5NAME Z              Z
     S5TSW  500.000000E-12  700.000000E-12
     S5RLO    1.000000E+06  200.000000E+03
     S5RHI    1.000000E+06  200.000000E+03
```

**** 06/05/99 13:29:36 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_2ns_6_5.sch

**** Digital Gate MODEL PARAMETERS

****

	D_04	D_AC08	D_AC32
TPLHMN	4.800000E-09	1.000000E-09	1.000000E-09
TPLHTY	12.000000E-09	5.500000E-09	5.500000E-09
TPLHMX	22.000000E-09	8.500000E-09	8.500000E-09
TPHLMN	3.200000E-09	1.000000E-09	1.000000E-09
TPHLTY	8.000000E-09	5.500000E-09	5.000000E-09
TPHLMX	15.000000E-09	7.500000E-09	7.500000E-09

**** 06/05/99 13:29:36 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_2ns_6_5.sch

**** Digital Edge Triggered FF MODEL PARAMETERS

****

	D_PLD_EFF
TPCLKQLHMN	0
TPCLKQLHTY	0
TPCLKQLHMX	0
TPCLKQHLMN	0
TPCLKQHLTY	0
TPCLKQHLMX	0
TPPCQLHMN	0
TPPCQLHTY	0

TPPCQLHMX 0
TPPCQHLMN 0
TPPCQHLTY 0
TPPCQHLMX 0
TWCLKLMN 0
TWCLKLTY 0
TWCLKLMX 0
TWCLKHMN 0
TWCLKHTY 0
TWCLKHMX 0
TWPCLMN 0
TWPCLTY 0
TWPCLMX 0
TSUDCLKMN 0
TSUDCLKTY 0
TSUDCLKMX 0
TSUPCCLKHMN 0
TSUPCCLKHTY 0
TSUPCCLKHMX 0
THDCLKMN 0
THDCLKTY 0
THDCLKMX 0
TSUCECLKMN 0
TSUCECLKTY 0
TSUCECLKMX 0
THCECLKMN 0
THCECLKTY 0
THCECLKMX 0

**** 06/05/99 13:29:36 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_2ns_6_5.sch

**** Digital IO MODEL PARAMETERS

****
******

```
          IO_STM         IO_STD         IO_PLD         IO_AC
 INLD                                  5.000000E-12
 DRVL    0              104            18.7
 DRVH    0              96.4           24.5
AtoD1              AtoD_STD                AtoD_AC
AtoD2              AtoD_STD_NX                AtoD_AC_NX
AtoD3              AtoD_STD                AtoD_AC
AtoD4              AtoD_STD_NX                AtoD_AC_NX
DtoA1 DtoA_STM       DtoA_STD                DtoA_AC
DtoA2 DtoA_STM       DtoA_STD                DtoA_AC
DtoA3 DtoA_STM       DtoA_STD                DtoA_AC
DtoA4 DtoA_STM       DtoA_STD                DtoA_AC
TSWHL1                1.373000E-09            477.000000E-12
TSWHL2                1.346000E-09            477.000000E-12
TSWHL3                1.511000E-09            477.000000E-12
TSWHL4                1.487000E-09            477.000000E-12
TSWLH1                3.382000E-09            470.000000E-12
TSWLH2                3.424000E-09            470.000000E-12
TSWLH3                3.517000E-09            470.000000E-12
TSWLH4                3.564000E-09            470.000000E-12
 TPWRT  100.000000E+03  100.000000E+03  100.000000E+03  100.000000E+03


          IO_AC_DTOA
 TPWRT  100.000000E+03



**** 06/05/99 13:29:36 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

 * C:\cpp\Projects\tvc_final_2ns_6_5.sch


 ****     INITIAL TRANSIENT SOLUTION       TEMPERATURE =   27.000 DEG C


**********************************************************************
******
```

NODE VOLTAGE NODE VOLTAGE NODE VOLTAGE NODE VOLTAGE

(+5v) 4.3534 (Vout) 3.1781 ($G_DGND) 0.0000

($G_DPWR) 5.0000 ($N_0003) 466.6E-06

($N_0020) .0025 ($N_0021) 4.9981

($N_0022) 4.3398 (X$$N_0003_DtoA1.DGND_OL) 416.6E-06

(X$$N_0003_DtoA1.DPWR_OH) 4.9994 (X$$N_0021_DtoA1.DGND_OL) 416.5E-06

(X$$N_0021_DtoA1.DPWR_OH) 4.9982

DGTL NODE : STATE DGTL NODE : STATE DGTL NODE : STATE DGTL NODE : STATE

($N_0019) : 1 ($N_0003$DtoA) : 0 ($N_0004) : 1

(N_0021DtoA) : 1 ($N_0009) : 0 ($N_0010) : 0

($D_HI) : 1 ($N_0015) : 0 ($N_0016) : 1 ($N_0020$DtoA) : 0

($N_0005) : 0 ($N_0006) : 1 ($N_0011) : 1 ($N_0012) : 0

($N_0017) : 1 ($N_0018) : 0 ($N_0001) : 0 ($N_0002) : 1

($N_0007) : 0 ($N_0008) : 0 ($N_0013) : 0 ($N_0014) : 0

VOLTAGE SOURCE CURRENTS
NAME CURRENT

X$DIGIFPWR.VDPWR -5.202E-03
X$DIGIFPWR.VDGND -1.050E-04

TOTAL POWER DISSIPATION 2.60E-02 WATTS

JOB CONCLUDED

TOTAL JOB TIME 1.31

APPENDIX C

Below is the PSPICE output and PSPICE netlist code for 8-channel TVC.

**** 05/30/99 22:33:55 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_100mhz_TEST1.sch

**** CIRCUIT DESCRIPTION

****
******

* Schematics Version 8.0 - July 1997
* Sun May 30 22:33:53 1999

** Analysis setup **
.tran 20ns 1000ns
.LIB "C:\cpp\lib\Breakout.lib"
.LIB "C:\cpp\lib\Eval.lib"
.LIB "C:\cpp_setup.lib"
.LIB "nom.lib"
.STMLIB "C:\cpp\Projects\tvc.stl"
.STMLIB "tvc_pre.stl"
.STMLIB "tvc_pre_1.stl"
.STMLIB "tvc_pre_2.stl"
.STMLIB "C:\cpp\Projects\tvc_final.stl"

* From [SCHEMATICS NETLIST] section of msim.ini:
.lib "C:\MSim_8\UserLib\64msdout.lib"
.lib "C:\MSim_8\UserLib\64msdnor.lib"

```
.lib "C:\MSim_8\UserLib\64msdinp.lib"
.lib "C:\MSim_8\UserLib\64msdhig.lib"
.lib "C:\MSim_8\UserLib\m63z818.lib"
.lib "C:\MSim_8\UserLib\SIBUFIF.lib"
.lib "C:\MSim_8\UserLib\BT8IF.lib"
.lib "C:\MSim_8\UserLib\BT8RIF.lib"
.lib "C:\MSim_8\UserLib\BT8HIF.lib"
.lib "C:\MSim_8\UserLib\BT4RIF.lib"
.lib "C:\MSim_8\UserLib\BT4IF.lib"
.lib "C:\MSim_8\UserLib\BIBUFIF.lib"
.lib "C:\MSim_8\UserLib\BD4CIF.lib"
.lib "C:\MSim_8\UserLib\BT4HIF.lib"
.lib "C:\MSim_8\UserLib\BD8RCIF.lib"
.lib "C:\MSim_8\UserLib\BD8HCIF.lib"
.lib "C:\MSim_8\UserLib\BD8CIF.lib"
.lib "C:\MSim_8\UserLib\BD4RCIF.lib"
.lib "C:\MSim_8\UserLib\IBUFIF.lib"
.lib "C:\MSim_8\UserLib\BD4HCIF.lib"
.lib "nom.lib"

.INC "tvc_final_100mhz_TEST1.net"

**** INCLUDING tvc_final_100mhz_TEST1.net ****
* Schematics Netlist *

.EXTERNAL OUTPUT Vout1
.EXTERNAL OUTPUT Vout2
.EXTERNAL OUTPUT Vout3
.EXTERNAL OUTPUT Vout4
.EXTERNAL OUTPUT Vout5
.EXTERNAL OUTPUT Vout6
.EXTERNAL OUTPUT Vout7
.EXTERNAL OUTPUT Vout8


X_U73A        $N_0001 $N_0002 $N_0003 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U74        $N_0006 $N_0004 $N_0005 $N_0007 $N_0008 $G_DPWR $G_DGND
DFFR
X_U75        $N_0001 $N_0003 $N_0005 $N_0009 $N_0010 $G_DPWR $G_DGND
DFFR
X_U76A        $N_0006 $N_0002 $N_0004 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
```

```
X_U77        $N_0007 $N_0002 $N_0005 $N_0011 $N_0012 $G_DPWR $G_DGND
DFFR
X_U78        $N_0009 $N_0002 $N_0005 $N_0013 $N_0014 $G_DPWR $G_DGND
DFFR
X_U79A       $N_0015 $N_0016 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U80A       $N_0012 $N_0013 $N_0015 $G_DPWR $G_DGND 7432 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
U_DSTM16     STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0001
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 500ns 1
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM17     STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0005
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 50ns 1
+ 100ns 1
+ 300ns 1
+ 15000ns 1
U_DSTM4      STIM(1,1) $G_DPWR $G_DGND $N_0002 IO_STM IO_LEVEL=0
+ 0 0
+ +0ns 1
+REPEAT FOREVER
+ +20ns 0
+  +20ns 1
+ ENDREPEAT
X_U82A       $N_0016 $N_0017 $G_DPWR $G_DGND 7404 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
C_C2         Vout1 +5v  1pF
C_C3         Vout2 +5v  1pF
C_C4         Vout3 +5v  1pF
Q_Q7         +5v $N_0018 Vout2 Q2N1893
Q_Q2         +5v $N_0018 Vout1 Q2N1893
```

```
C_C5        Vout4 +5v  1pF
Q_Q9        +5v $N_0018 Vout3 Q2N1893
C_C6        Vout5 +5v  1pF
Q_Q11       +5v $N_0018 Vout4 Q2N1893
Q_Q13       +5v $N_0018 Vout5 Q2N1893
C_C7        Vout6 +5v  1pF
Q_Q15       +5v $N_0018 Vout6 Q2N1893
C_C8        Vout7 +5v  1pF
Q_Q4        +5v $N_0017 $N_0019 Q2N1893
I_I2        $N_0019 0 DC 50uA
Q_Q3        Vout1 $N_0020 $N_0019 Q2N1893
Q_Q8        Vout2 $N_0021 $N_0019 Q2N1893
Q_Q10       Vout3 $N_0022 $N_0019 Q2N1893
X_U97A      $N_0023 $N_0024 $N_0025 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U96A      $N_0023 $N_0026 $N_0027 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U95A      $N_0023 $N_0028 $N_0029 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U81A      $N_0011 $N_0014 $N_0023 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U89A      $N_0023 $N_0030 $N_0020 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U86A      $N_0023 $N_0031 $N_0021 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U94A      $N_0023 $N_0032 $N_0022 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
X_U98A      $N_0023 $N_0033 $N_0034 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
Q_Q17       +5v $N_0018 Vout7 Q2N1893
Q_Q12       Vout4 $N_0029 $N_0019 Q2N1893
Q_Q14       Vout5 $N_0027 $N_0019 Q2N1893
Q_Q16       Vout6 $N_0025 $N_0019 Q2N1893
Q_Q18       Vout7 $N_0034 $N_0019 Q2N1893
C_C9        Vout8 +5v  1pF
X_U99A      $N_0023 $N_0035 $N_0036 $G_DPWR $G_DGND 7408 PARAMS:
+ IO_LEVEL=0 MNTYMXDLY=0
Q_Q19       +5v $N_0018 Vout8 Q2N1893
Q_Q20       Vout8 $N_0036 $N_0019 Q2N1893
U_DSTM36        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0030
+ IO_STM
```

```
+ IO_LEVEL=0
+  0ns 1
+ 500ns 1
+ 600ns 1
+ 11000ns 1
+ 12200ns 1
+ 15000ns 1
U_DSTM44        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0031
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 500ns 0
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM51        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0032
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 500ns 0
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM52        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0028
+ IO_STM
+ IO_LEVEL=0
+  0ns 0
+ 500ns 0
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM53        STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0026
```

+ IO_STM
+ IO_LEVEL=0
+ 0ns 0
+ 500ns 0
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM54 STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0024
+ IO_STM
+ IO_LEVEL=0
+ 0ns 0
+ 500ns 0
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM55 STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0033
+ IO_STM
+ IO_LEVEL=0
+ 0ns 0
+ 500ns 0
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM56 STIM(1,1)
+ $G_DPWR $G_DGND
+ $N_0035
+ IO_STM
+ IO_LEVEL=0
+ 0ns 0
+ 500ns 0
+ 600ns 0
+ 11000ns 0
+ 12200ns 0
+ 15000ns 0
U_DSTM57 STIM(1,1)
+ $G_DPWR $G_DGND

+ $N_0018
+ IO_STM
+ IO_LEVEL=0
+ 0ns 0
+ 50ns 1
+ 100ns 1
+ 300ns 1
+ 15000ns 1

**** RESUMING tvc_final_100mhz_TEST1.cir ****
.INC "tvc_final_100mhz_TEST1.als"

**** INCLUDING tvc_final_100mhz_TEST1.als ****
* Schematics Aliases *

.ALIASES
X_U73A U73A(A=$N_0001 B=$N_0002 Y=$N_0003 PWR=$G_DPWR
GND=$G_DGND)
X_U74 U74(D=$N_0006 CLK=$N_0004 Rbar=$N_0005 Q=$N_0007
Qbar=$N_0008
+ PWR=$G_DPWR GND=$G_DGND)
X_U75 U75(D=$N_0001 CLK=$N_0003 Rbar=$N_0005 Q=$N_0009
Qbar=$N_0010
+ PWR=$G_DPWR GND=$G_DGND)
X_U76A U76A(A=$N_0006 B=$N_0002 Y=$N_0004 PWR=$G_DPWR
GND=$G_DGND)
X_U77 U77(D=$N_0007 CLK=$N_0002 Rbar=$N_0005 Q=$N_0011
Qbar=$N_0012
+ PWR=$G_DPWR GND=$G_DGND)
X_U78 U78(D=$N_0009 CLK=$N_0002 Rbar=$N_0005 Q=$N_0013
Qbar=$N_0014
+ PWR=$G_DPWR GND=$G_DGND)
X_U79A U79A(A=$N_0015 Y=$N_0016 PWR=$G_DPWR GND=$G_DGND)
X_U80A U80A(A=$N_0012 B=$N_0013 Y=$N_0015 PWR=$G_DPWR
GND=$G_DGND)
U_DSTM16 DSTM16(PIN1=$N_0001)
U_DSTM17 DSTM17(PIN1=$N_0005)
U_DSTM4 DSTM4(PWR=$G_DPWR GND=$G_DGND 1=$N_0002)
X_U82A U82A(A=$N_0016 Y=$N_0017 PWR=$G_DPWR GND=$G_DGND)
C_C2 C2(1=Vout1 2=+5v)
C_C3 C3(1=Vout2 2=+5v)

C_C4 C4(1=Vout3 2=+5v)
Q_Q7 Q7(c=+5v b=$N_0018 e=Vout2)
Q_Q2 Q2(c=+5v b=$N_0018 e=Vout1)
C_C5 C5(1=Vout4 2=+5v)
Q_Q9 Q9(c=+5v b=$N_0018 e=Vout3)
C_C6 C6(1=Vout5 2=+5v)
Q_Q11 Q11(c=+5v b=$N_0018 e=Vout4)
Q_Q13 Q13(c=+5v b=$N_0018 e=Vout5)
C_C7 C7(1=Vout6 2=+5v)
Q_Q15 Q15(c=+5v b=$N_0018 e=Vout6)
C_C8 C8(1=Vout7 2=+5v)
Q_Q4 Q4(c=+5v b=$N_0017 e=$N_0019)
I_I2 I2(+=$N_0019 -=0)
Q_Q3 Q3(c=Vout1 b=$N_0020 e=$N_0019)
Q_Q8 Q8(c=Vout2 b=$N_0021 e=$N_0019)
Q_Q10 Q10(c=Vout3 b=$N_0022 e=$N_0019)
X_U97A U97A(A=$N_0023 B=$N_0024 Y=$N_0025 PWR=$G_DPWR
GND=$G_DGND)
X_U96A U96A(A=$N_0023 B=$N_0026 Y=$N_0027 PWR=$G_DPWR
GND=$G_DGND)
X_U95A U95A(A=$N_0023 B=$N_0028 Y=$N_0029 PWR=$G_DPWR
GND=$G_DGND)
X_U81A U81A(A=$N_0011 B=$N_0014 Y=$N_0023 PWR=$G_DPWR
GND=$G_DGND)
X_U89A U89A(A=$N_0023 B=$N_0030 Y=$N_0020 PWR=$G_DPWR
GND=$G_DGND)
X_U86A U86A(A=$N_0023 B=$N_0031 Y=$N_0021 PWR=$G_DPWR
GND=$G_DGND)
X_U94A U94A(A=$N_0023 B=$N_0032 Y=$N_0022 PWR=$G_DPWR
GND=$G_DGND)
X_U98A U98A(A=$N_0023 B=$N_0033 Y=$N_0034 PWR=$G_DPWR
GND=$G_DGND)
Q_Q17 Q17(c=+5v b=$N_0018 e=Vout7)
Q_Q12 Q12(c=Vout4 b=$N_0029 e=$N_0019)
Q_Q14 Q14(c=Vout5 b=$N_0027 e=$N_0019)
Q_Q16 Q16(c=Vout6 b=$N_0025 e=$N_0019)
Q_Q18 Q18(c=Vout7 b=$N_0034 e=$N_0019)
C_C9 C9(1=Vout8 2=+5v)
X_U99A U99A(A=$N_0023 B=$N_0035 Y=$N_0036 PWR=$G_DPWR
GND=$G_DGND)
Q_Q19 Q19(c=+5v b=$N_0018 e=Vout8)
Q_Q20 Q20(c=Vout8 b=$N_0036 e=$N_0019)
U_DSTM36 DSTM36(PIN1=$N_0030)

```
U_DSTM44        DSTM44(PIN1=$N_0031 )
U_DSTM51        DSTM51(PIN1=$N_0032 )
U_DSTM52        DSTM52(PIN1=$N_0028 )
U_DSTM53        DSTM53(PIN1=$N_0026 )
U_DSTM54        DSTM54(PIN1=$N_0024 )
U_DSTM55        DSTM55(PIN1=$N_0033 )
U_DSTM56        DSTM56(PIN1=$N_0035 )
U_DSTM57        DSTM57(PIN1=$N_0018 )
_   _(Vout1=Vout1)
_   _(Vout2=Vout2)
_   _(Vout3=Vout3)
_   _(Vout4=Vout4)
_   _(Vout5=Vout5)
_   _(Vout6=Vout6)
_   _(Vout7=Vout7)
_   _(Vout8=Vout8)
_   _(+5v=+5v)
_   _($G_DPWR=$G_DPWR)
_   _($G_DGND=$G_DGND)
.ENDALIASES


**** RESUMING tvc_final_100mhz_TEST1.cir ****
.probe


.END


**** Generated AtoD and DtoA Interfaces ****

*
* Analog/Digital interface for node $N_0036
*
* Moving X_U99A.U1:OUT1 from analog node $N_0036 to new digital node
$N_0036$DtoA
X$$N_0036_DtoA1
+ $N_0036$DtoA
+ $N_0036
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+      PARAMS: DRVH=  96.4   DRVL= 104    CAPACITANCE=   0
```

```
*
* Analog/Digital interface for node $N_0020
*
* Moving X_U89A.U1:OUT1 from analog node $N_0020 to new digital node
$N_0020$DtoA
X$$N_0020_DtoA1
+ $N_0020$DtoA
+ $N_0020
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+      PARAMS: DRVH=  96.4   DRVL= 104    CAPACITANCE=   0
*
* Analog/Digital interface for node $N_0025
*
* Moving X_U97A.U1:OUT1 from analog node $N_0025 to new digital node
$N_0025$DtoA
X$$N_0025_DtoA1
+ $N_0025$DtoA
+ $N_0025
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+      PARAMS: DRVH=  96.4   DRVL= 104    CAPACITANCE=   0
*
* Analog/Digital interface for node $N_0021
*
* Moving X_U86A.U1:OUT1 from analog node $N_0021 to new digital node
$N_0021$DtoA
X$$N_0021_DtoA1
+ $N_0021$DtoA
+ $N_0021
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+      PARAMS: DRVH=  96.4   DRVL= 104    CAPACITANCE=   0
*
* Analog/Digital interface for node $N_0022
*
* Moving X_U94A.U1:OUT1 from analog node $N_0022 to new digital node
$N_0022$DtoA
X$$N_0022_DtoA1
+ $N_0022$DtoA
```

+ $N_0022
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+ PARAMS: DRVH= 96.4 DRVL= 104 CAPACITANCE= 0
*
* Analog/Digital interface for node $N_0027
*
* Moving X_U96A.U1:OUT1 from analog node $N_0027 to new digital node
N_0027DtoA
X$$N_0027_DtoA1
+ N_0027DtoA
+ $N_0027
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+ PARAMS: DRVH= 96.4 DRVL= 104 CAPACITANCE= 0
*
* Analog/Digital interface for node $N_0017
*
* Moving X_U82A.U1:OUT1 from analog node $N_0017 to new digital node
N_0017DtoA
X$$N_0017_DtoA1
+ N_0017DtoA
+ $N_0017
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+ PARAMS: DRVH= 96.4 DRVL= 104 CAPACITANCE= 0
*
* Analog/Digital interface for node $N_0034
*
* Moving X_U98A.U1:OUT1 from analog node $N_0034 to new digital node
N_0034DtoA
X$$N_0034_DtoA1
+ N_0034DtoA
+ $N_0034
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+ PARAMS: DRVH= 96.4 DRVL= 104 CAPACITANCE= 0
*
* Analog/Digital interface for node $N_0018

```
*
* Moving U_DSTM57:OUT1 from analog node $N_0018 to new digital node
$N_0018$DtoA
X$$N_0018_DtoA1
+ $N_0018$DtoA
+ $N_0018
+ $G_DPWR
+ $G_DGND
+ DtoA_STM
+      PARAMS: DRVH=   0      DRVL=   0      CAPACITANCE=   0
*
* Analog/Digital interface for node $N_0029
*
* Moving X_U95A.U1:OUT1 from analog node $N_0029 to new digital node
$N_0029$DtoA
X$$N_0029_DtoA1
+ $N_0029$DtoA
+ $N_0029
+ $G_DPWR
+ $G_DGND
+ DtoA_STD
+      PARAMS: DRVH=  96.4   DRVL= 104    CAPACITANCE=   0
*
* Analog/Digital interface power supply subcircuits
*

X$DIGIFPWR 0 DIGIFPWR


**** 05/30/99 22:33:55 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

 * C:\cpp\Projects\tvc_final_100mhz_TEST1.sch


 ****     BJT MODEL PARAMETERS


**********************************************************************
******
```

Q2N1893
NPN
IS 2.079000E-15
BF 167.2
NF 1
VAF 100
IKF 1.088
ISE 24.360000E-15
NE 1.357
BR 1.502
NR 1
IKR .3028
ISC 264.600000E-15
NC 1.545
NK .8271
RC .9069
CJE 58.120000E-12
VJE .5
MJE .441
CJC 45.290000E-12
VJC .5
MJC .31
TF 738.100000E-12
XTF 24.93
VTF 10
ITF 2.376
TR 1.000000E-06
XTB 1.5

**** 05/30/99 22:33:55 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_100mhz_TEST1.sch

**** Digital Input MODEL PARAMETERS

****
******

```
          DINSTM          DIN74
      FILE DSO_DTOA       DSO_DTOA
    FORMAT    6             6
  TIMESTEP  100.000000E-12  100.000000E-12
    S0NAME 0              0
     S0TSW  500.000000E-12    3.500000E-09
     S0RLO     .5            7.13
     S0RHI    1.000000E+03  389
    S1NAME 1              1
     S1TSW  500.000000E-12    5.500000E-09
     S1RLO    1.000000E+03  467
     S1RHI     .5          200
    S2NAME X              X
     S2TSW  500.000000E-12    3.500000E-09
     S2RLO     .429         42.9
     S2RHI    1.16         116
    S3NAME R              R
     S3TSW  500.000000E-12    3.500000E-09
     S3RLO     .429         42.9
     S3RHI    1.16         116
    S4NAME F              F
     S4TSW  500.000000E-12    3.500000E-09
     S4RLO     .429         42.9
     S4RHI    1.16         116
    S5NAME Z              Z
     S5TSW  500.000000E-12    3.500000E-09
     S5RLO    1.000000E+06  200.000000E+03
     S5RHI    1.000000E+06  200.000000E+03
```

**** 05/30/99 22:33:55 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_100mhz_TEST1.sch

**** Digital Gate MODEL PARAMETERS

****

	D_32	D_04	D_08
TPLHMN	4.000000E-09	4.800000E-09	7.000000E-09
TPLHTY	10.000000E-09	12.000000E-09	17.500000E-09
TPLHMX	15.000000E-09	22.000000E-09	27.000000E-09
TPHLMN	5.600000E-09	3.200000E-09	4.800000E-09
TPHLTY	14.000000E-09	8.000000E-09	12.000000E-09
TPHLMX	22.000000E-09	15.000000E-09	19.000000E-09

**** 05/30/99 22:33:55 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_100mhz_TEST1.sch

**** Digital Edge Triggered FF MODEL PARAMETERS

****

	D_PLD_EFF
TPCLKQLHMN	0
TPCLKQLHTY	0
TPCLKQLHMX	0
TPCLKQHLMN	0
TPCLKQHLTY	0
TPCLKQHLMX	0
TPPCQLHMN	0
TPPCQLHTY	0
TPPCQLHMX	0
TPPCQHLMN	0
TPPCQHLTY	0
TPPCQHLMX	0
TWCLKLMN	0
TWCLKLTY	0

TWCLKLMX 0
TWCLKHMN 0
TWCLKHTY 0
TWCLKHMX 0
TWPCLMN 0
TWPCLTY 0
TWPCLMX 0
TSUDCLKMN 0
TSUDCLKTY 0
TSUDCLKMX 0
TSUPCCLKHMN 0
TSUPCCLKHTY 0
TSUPCCLKHMX 0
THDCLKMN 0
THDCLKTY 0
THDCLKMX 0
TSUCECLKMN 0
TSUCECLKTY 0
TSUCECLKMX 0
THCECLKMN 0
THCECLKTY 0
THCECLKMX 0

**** 05/30/99 22:33:55 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_100mhz_TEST1.sch

**** Digital IO MODEL PARAMETERS

****
******

IO_STM IO_STD IO_PLD
DRVL 0 104
DRVH 0 96.4
AtoD1 AtoD_STD

AtoD2 AtoD_STD_NX
AtoD3 AtoD_STD
AtoD4 AtoD_STD_NX
DtoA1 DtoA_STM DtoA_STD
DtoA2 DtoA_STM DtoA_STD
DtoA3 DtoA_STM DtoA_STD
DtoA4 DtoA_STM DtoA_STD
TSWHL1 1.373000E-09
TSWHL2 1.346000E-09
TSWHL3 1.511000E-09
TSWHL4 1.487000E-09
TSWLH1 3.382000E-09
TSWLH2 3.424000E-09
TSWLH3 3.517000E-09
TSWLH4 3.564000E-09
TPWRT 100.000000E+03 100.000000E+03 100.000000E+03

**** 05/30/99 22:33:55 ***** Win95 PSpice 8.0 (July 1997) ***** ID# 100586 ***

* C:\cpp\Projects\tvc_final_100mhz_TEST1.sch

**** INITIAL TRANSIENT SOLUTION TEMPERATURE = 27.000 DEG C

****

NODE VOLTAGE NODE VOLTAGE NODE VOLTAGE NODE VOLTAGE

(+5v) 2.8674 (Vout1) 2.0693 (Vout2) 2.0693 (Vout3) 2.0693

(Vout4) 2.0693 (Vout5) 2.0693 (Vout6) 2.0693 (Vout7) 2.0693

(Vout8) 2.0693 ($G_DGND) 0.0000 ($G_DPWR) 5.0000

($N_0017) 3.4937 ($N_0018) .0025

($N_0019) 2.8537 ($N_0020) .0900

($N_0021) .0900 ($N_0022) .0900

($N_0025) .0900 ($N_0027) .0900

($N_0029) .0900 ($N_0034) .0900

($N_0036) .0900

DGTL NODE : STATE DGTL NODE : STATE DGTL NODE : STATE DGTL NODE : STATE

($N_0035) : 0 ($N_0017$DtoA) : 1 ($N_0003) : 0

($N_0004) : X ($N_0021$DtoA) : 0 ($N_0009) : 0

($N_0026) : 0 ($N_0010) : 1 (N_0025DtoA) : 0

($D_HI) : 1 ($N_0031) : 0 ($N_0015) : 1 ($N_0018$DtoA) : 0

($N_0032) : 0 ($N_0016) : 0 (N_0034DtoA) : 0

(N_0020DtoA) : 0 ($N_0005) : 0 ($N_0006) : Z

($N_0028) : 0 ($N_0011) : 0 ($N_0012) : 1 ($N_0033) : 0

($N_0001) : 0 ($N_0002) : 1 ($N_0023) : 0 ($N_0007) : 0

($N_0024) : 0 ($N_0008) : 1 (N_0029DtoA) : 0

(N_0027DtoA) : 0 ($N_0013) : 0 ($N_0036$DtoA) : 0

($N_0030) : 1 ($N_0014) : 1 (N_0022DtoA) : 0

VOLTAGE SOURCE CURRENTS
NAME CURRENT

X$DIGIFPWR.VDPWR -1.135E-01
X$DIGIFPWR.VDGND -5.500E-05

TOTAL POWER DISSIPATION   5.68E-01 WATTS

JOB CONCLUDED

TOTAL JOB TIME 1.58

References

1. Dr. J. Starzyk, Professor of Electrical Engineering, Ohio University, Athens, OH, Private Communication regarding the concept and the fundamentals of this thesis.
2. A. Stevens, R. Vanberg and H. Williams, "A Time-to-Voltage Converter and Analog Memory for Colliding Beam Detectors," IEEE Journal of Solid State Circuits, Vol. 24, No. 6, December 1989.
3. "The TDC-Cookbook-A Series for high precision time measurement," ACAM-messelectronics, October 1998.
4. HP-Application Note 162-1: "Time Interval Averaging," 1998.
5. Porat, D. I., "Review of sub-nanosecond time interval measurements," IEEE Trans. Nucl. Sci., Vol. NS-20, pp. 36-51, 1973.
6. Rankinen, R. Maatta, K. Kostamovaara, "Time-to-Digital Conversion with 10ps Single-shot Resolution", Proc. IEEE Melecon, May 22-24, Ljubljana, Yugoslavia.
7. E. Gerds, Van Der Spiegel, R. Vanberg, H. Williams, L. Callewaert, "A CMOS Time-to-Digital Converter IC with 2 Level Analog CAM', IEEE Journal of Solid State Circuits, Vol. 29, No. 9, Sept. 1994.
8. Y. Arai, T. Matsumura, K Endo, "A CMOS 4 Channel $\times$ 1K Time Memory LSI with 1-ns/b Resolution", IEEE Journal of Solid State Circuits, Vol. 27, No. 3, March 1992.
9. A. Rothermel, F. Dell'ova, "Analog Phase Measuring circuit for Digital CMOS IC's", IEEE Journal of Solid State Circuits, Vol. 28, No. 7, July 1993.

10. Elvi Raisanen-Ritosalainen, Time Rahkonen, Juha Kostamovaara, “A 5 mW Time-to-digital Converter Based on a Stabilized CMOS Delay Line”, Dept. of Electrical Engineering, University of Oulu, Oulu, Finland.
11. Technical Memorandum on “Time and Time Interval Measurement with Application to Computer and Network Performance Evaluation”, David L. Mills, Electrical Engineering Dept., University of Delaware, January 1996.
12. Technical Memorandum on “A Kernel Model for Precision Timekeeping”, David L. Mills, Electrical Engineering Dept., University of Delaware, January 1996.
13. J. Kostamovaara, K. Maatta, M Koskinen, R. Myllyla, “Pulsed laser radars with high-modulation-frequency in industrial applications,” Proc. SPIE Laser Radar 7: Advanced Technology for Applications, Vol. 1633, January 992.
14. D. Chu, M. Allen, A. Foster, “Universal Counter Resolves Picoseconds in Time Interval Measurement,’ HP Journal 29, 1988.
15. K. Maatta, J. Kostamovaara, R. Myllyla, ‘Time-to-Digital Converter for Fast, Accurate Laser rangefinding’, Proc. SPIE Industrial Inspection, Vol. 1010, Sept. 1988.
16. T. Rahkonen, “Circuit Techniques and Integrated CMOS Implementations for Measuring Short Time Intervals’, Ph.D. Thesis, University of Oulu, Oulu, Finland, 1993.
17. T. Rahkonen, J. Kostamovaara, ‘Low-power Time-to-Digital and Digital-to-Time Converters for Novel Implementations of Telecommunication Building Blocks’, University of Oulu, Oulu, Finland.

18. Raisanen-Ruotsalainen, T. Rahkonen, J. Kostamovaara, "A Low-power Time Interval Measurement Chip with Subnanosecond Resolution", Proc., 36th Midwest Symp. On Circuits. And Systems, Detroit, USA, August 1993.
19. J. Kostamovaara, R. Myllyla, "Time-to-Digital Converter with an analog interpolation circuit', University of Oulu, Oulu, Finland, May 1996.
20. Marc J. Loinaz, Bruce A. Wooley, "A BiCMOS Time Interval Digitizer Based on Fully-Differential, Current-Steering Circuits', IEEE Journal of Solid State Circuits, Vol. 29, No. 6, June 1994.
21. Ronald Nutt, 'Digital Time Intervalometer', Oak Ridge National Laboratory, Tennessee, The Review of Scientific Instruments, Vol. 39, November, 1968.
22. Randy H. Katz, 'Contemporary Logic Design', The Benjamin/Cummings Publishing Company, Inc., 1994.
23. Kenneth J. Breeding, 'Digital Design Fundamentals', Prentice Hall, NJ, 1989.
24. Zchen Chen, "A VLSI Implementation of Sigma to Delta Converter", Former Student, Electrical Engineering Dept., Ohio University, Athens, OH, 1997.