CMOS Project Report
Delay Measurement Device

Nathan Dwek – Ilias Fassi Fihri

May 31, 2016

Contents

1 Introduction

1.1 Specifications

The goal of this project is to design and simulate a delay measurement device using components available through the AMS C35 0.35 µm CMOS process. In order to measure the length of a long cable, a voltage pulse is applied to one end of the cable and the length is deduced from the time it takes to reach the other end.

The output of the device must scale with the delay between two voltage pulses, which can range between 10 ns and 500 ns. The pulses themselves are chosen to be 10 ns long, and the device must be able to drive a 10 pF capacitive load.

1.2 Basic Block Diagram

To achieve this, the device is organized as shown in figure 1.

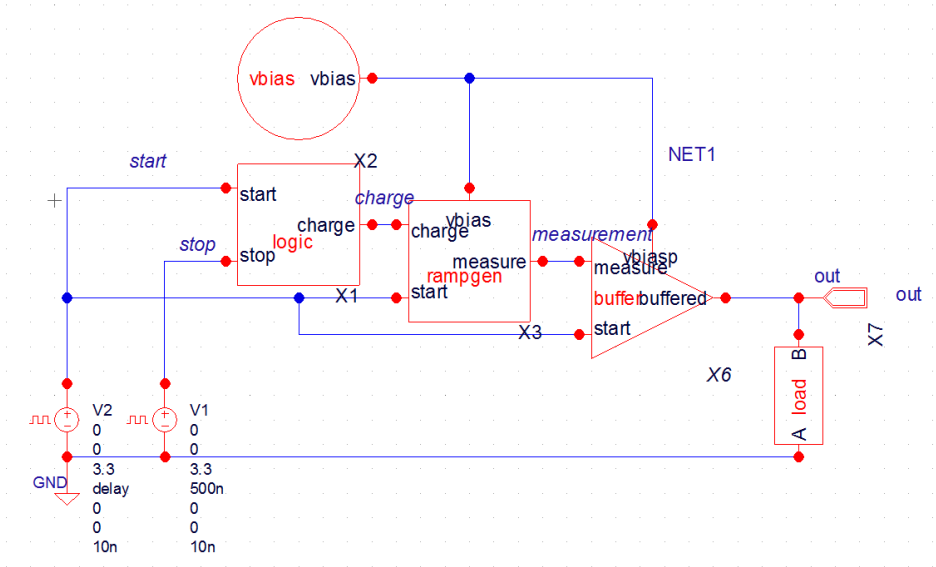

Figure 1: Block diagram overview of the delay measurement device.

At the core is the block `rampgen` which generates a rising ramp on its analog output `measure` when the digital input `charge` is *LO*, maintains its output when `charge` is *HI*, and resets it to ground when the digital input `start` is *HI*.

It is the controlled by the first block, `logic`, which controls `charge` based on the succession of pulses on its digital inputs `start` and `stop` and some internal logic.

The output of `rampgen` is buffered by the output stage `buffer` which is able to drive the specified load of 10 pF.

In the next three sections, the design of each of these blocks is reviewed. Then, in the last section, the operation curves of the whole are presented.

2 Internal Logic

This blocks controls its output `charge` based on the voltage pulses received on its inputs `start` and `stop`, so that the next block `rampgen` can control its output `measure` based only on the current value of its inputs.

2.1 Derivation of a Truth Table

First, we notice that the system needs to have a memory. Indeed, after a pulse has occured, and when both inputs are *LO*, the output `charge` can still be *HI* or *LO* depending on the last pulse. Given the logic that needs to be implemented and the fact that there are two separate pulse inputs we choose to implement the memory using an RS-latch. This choice will prove to be adequate at the end of this section. Intuitively, we choose to put `start` on the `set` of the latch and `stop` on the `reset`.

Furthermore, we cannot immediately reset `measure` to ground upon receiving a pulse on `stop`, because this wouldn't give enough time for the output stage to properly latch the final value[1], or more generally, wouldn't leave time for any reading device to use the measurement. For this reason, we choose to reset the whole device at the rising edge of a pulse on `start`, and to start the measurement at the falling edge of the pulse. To correctly measure the delay, the measurement must then be stopped at the falling edge of the pulse on `stop`. The output voltage is thus stabilized at its final value and available for reading from the end of the `stop` pulse until the beginning of the next `start` pulse.

With that it mind, it is possible to derive a first logical function, where `q` is the output of the RS-latch:

$$\mathsf{charge}(\mathsf{start}, \mathsf{stop}, \mathsf{q}) = \mathsf{q} \cdot \overline{\mathsf{start}} + \mathsf{stop} \tag{1}$$

[1] In our case the output stage doesn't even have a memory so this doesn't really apply.

2.2 NAND / NOR Implementation

This function must be adapted based on the following implementation details:

- As will be shown in next section, charge should be active-low because it drives a PMOS switch.
- NAND and NOR gates are the most directly available and require less transistors than AND and OR gates.
- $\overline{q}$ is directly available from the RS-latch.

Using de Morgan's law on the first term of the NOR which appears when taking account that charge should be active low, we find:

$$\begin{aligned}\overline{\texttt{charge}} &= \texttt{NOR}(\texttt{q} \cdot \overline{\texttt{start}}, \texttt{stop}) \\ &= \texttt{NOR}(\texttt{NOR}(\overline{\texttt{q}}, \texttt{start}), \texttt{stop})\end{aligned}$$

This final expression allows to implement the required logic using only one RS-latch and two NOR gates (rather than for example four NOT , one NAND and one NOR gate if we simply inverted function 1), as shown on figure 2.

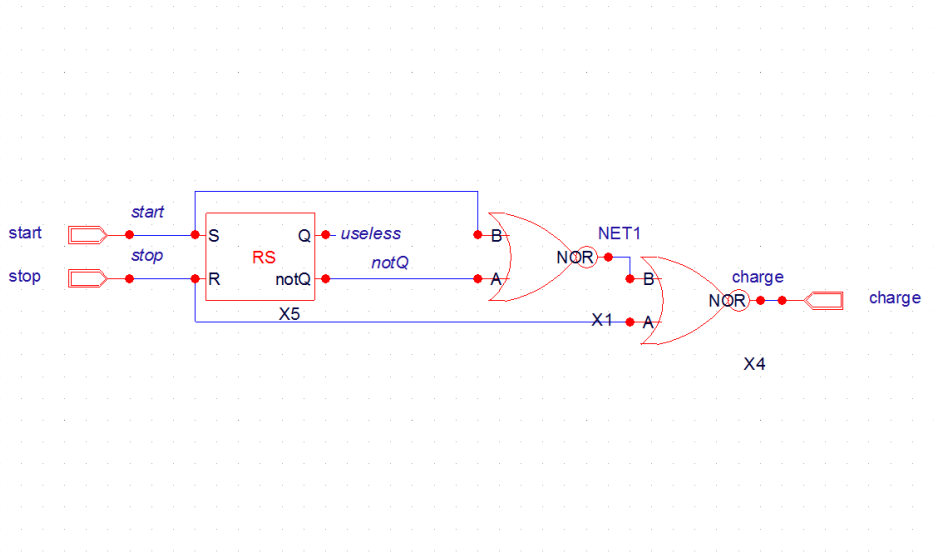

Figure 2: Gate circuit of `logic` .

2.3 Sizing

The logical gates are built using minimal length transistors. The width should theoretically progressively increase from input gates to output gates so that the output gate is able to drive the parasitic input capacitance of the next block and so that every gate in the circuit is able to drive the gate at its output. However, it turns out that minimally-sized transistors are able to correctly drive `rampgen`, so the whole logic circuit is made of minimally-size transistors.

3 Generation of the Measurement

`rampgen` generates a rising ramp on its output `measure` or maintains it based on its first input `charge`. The second input `start` allows to reset the output to ground.

3.1 Principle of Operation

To generate a voltage ramp on command, we choose to charge a capacitor through a switch with a DC current source.

We choose to split the switching and current generation responsibilities between two transistors. This allows to simulaneously achieve good switching behaviour which requires a low area transistor to reduce the gate parasitic capacitances, and good current source behaviour which requires a long transistor (and thus wide for a given I_{ds} and V_{ov}) to reduce g_{ds}.

We choose to reference `measurement` to ground so that its scales positively with the measured delay. This means that one of the capacitor terminals is grounded, and that the current source uses a PMOS. This choice is arbitrary, but will be taken into account when designing the output stage in order to maximize the output range.

The ramp switching is done using a PMOS as well because it allows to put the source on the drain of the current source which has relatively stable voltage. This gives better results than using an NMOS, because in that case the voltage a the source of the switch is actually the output ramp which should cover the widest possible range of voltages.

Finally, the output is shorted to ground by an NMOS switch controlled by `start`. `start` is thus active-high while `charge` is active-low, which corresponds to what we use in the previous section.

Figure 3 shows the circuit designed according to these considerations.

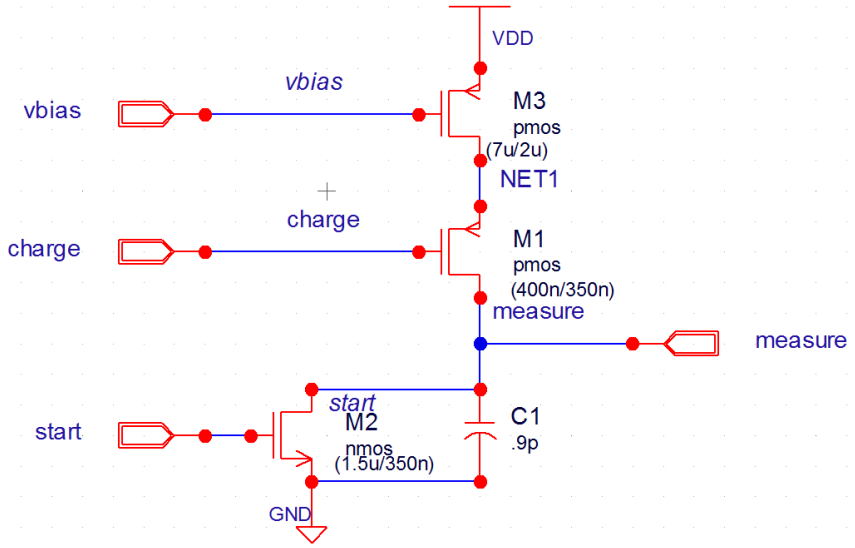

Figure 3: `rampgen` Circuit.

3.2 Sizing

`M3` should be long in order to minimize its g_{ds} since it acts as a current source. Its width, in combination with the capacitance of `C1` determines the slope of the output ramp. The slope is chosen so that the device reaches the limits of saturation for the longest required delay of 500 ns. The width also controls the sizing of `M1` which has the same i_{ds} as `M3` .

On the other hand, `M1` should be as small as possible for two reasons. It should be small enough so that it can be correctly driven by `logic` , and it should be as small as possible in order to minimize C_{gd}. $C_{gd_{\texttt{M3}}}$ tends to reproduce the falling and rising edges of `charge` on `measure` , which introduces small steps at the beginning end the end of the output ramp. This is undesired and increasing `C1` can also reduce this effect.

Based on this, the reasoning that led to the sizing of `M1` , `M3` and `C1` is as follows. `M3` 's length is set to 2 µm in order to get a good current source behaviour. A minimally sized transistor can still conduct a reasonable amount of current so we try to use one for `M1` . `C1` is then set to a value which is big enough to satisfyingly reduce the effect of $C_{gd_{\texttt{M3}}}$. Next, $i_{ds\texttt{M3}}$ is set by choosing `M3` 's width so that the ramp reaches the upper limit of the output range for a 500 ns delay.

Finally, we check that `M1` is able to deliver this final value of i_{ds}. A

minimally sized transistor is able to deliver the current required by `M3` so the sizing of this part of the ciruit is finished.

For `M2`, we don't care about linearity so we use a minimal length transistor that is wide enough to reset `measure` to ground in 10 ns, the length of a pulse, so that the output is properly initialized for the next measurement.