EE343 CMOS Mixed Signal Design
Lab 3: Charge Redistribution DAC Layout and Simulation

Jeffrey Cool 006442018
Dr. M. Wagdy

11-10-2015

Contents

1 Introduction

Presented here is a detailed account of the design, layout and simulation of a 4 bit serial DAC within the Microwind and DSCH environments. Figure 1 shows the charge redistribution circuit which provides 4 bit serial digital to analog conversion. Table 1 summarizes the required switching signals. The integrated circuit presented is laid out in $0.12\mu m$ CMOS technology, and consists of the switched capacitor circuit and a control unit which provides parallel to serial data conversion and sequential control of the switches.

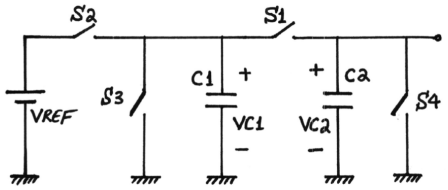

Figure 1: Charge Redistribution Circuit

Table 1: Control Signals

PHASE 1	$S_1 = 0$ $S_2 = bit$ $S_3 = \overline{bit}$ $S_4 = 0(1)^*$
$^*S_4 = 1$ only during clock cycle 1	
PHASE 2	$S_1 = 1$ $S_2 = 0$ $S_3 = 0$ $S_4 = 0$

2 Parallel to Serial Data Conversion

An easy way to accomplish parallel to serial conversion is to use a multiplexer. The 4 bit DAC requires a 4 to 1 MUX, which can be realized with 10 transistors using pass gates, as shown in Figure 2. The two inverters are used to restore the logic levels (NPGs pass logic 1 as $V_{DD} - V_{TN}$ and PPGs pass logic zero as V_{TP}) and to provide logical inversion.

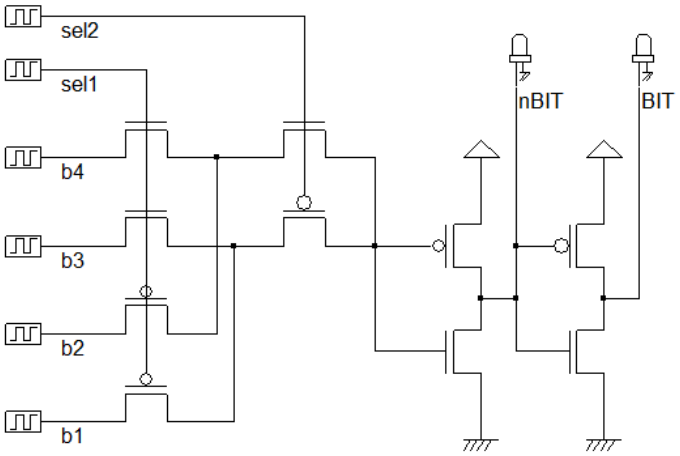

Figure 2: 4 to 1 MUX schematic

The layout for the MUX was done manually in Microwind. The aspect ratios of the NMOS and PMOS transistors are 5 and 10 respectively, with $L = 2\lambda$. Note the dotted line around the transistors–this is the option mask which is used here to specify high speed (low threshold voltage) devices to minimize the negative effects of the pass transistors and to speed up the conversion.

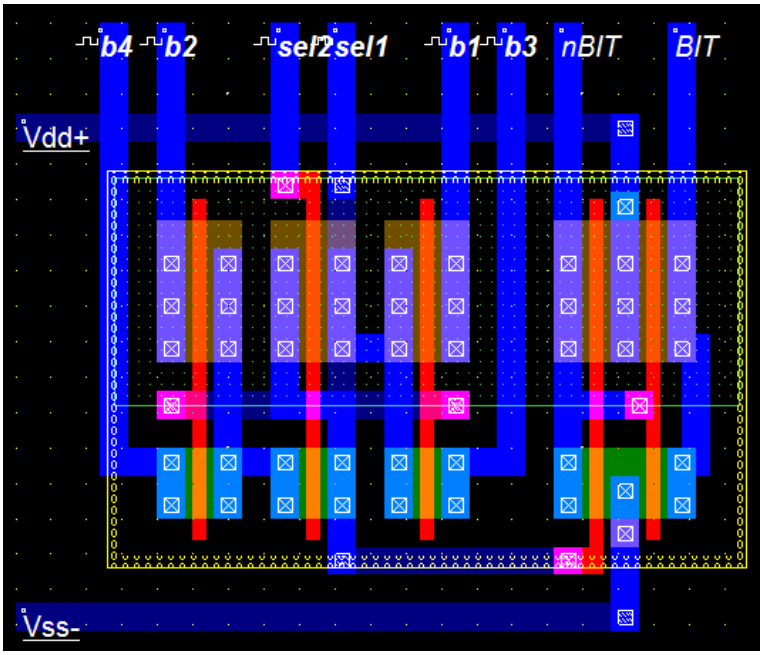

Figure 3: 4 to 1 MUX Layout

Proper operation of the multiplexer is verified in the timing diagram below:

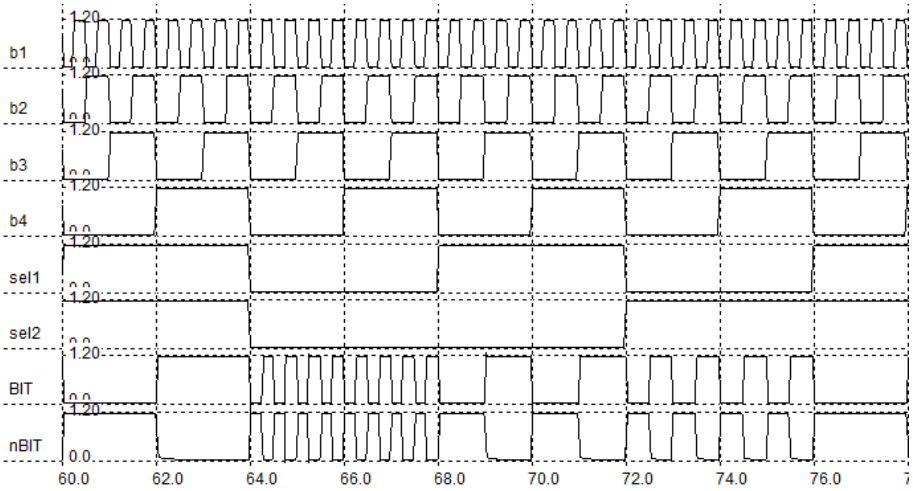

Figure 4: 4 to 1 MUX timing diagram

3 Controlling the MUX: A 2 Bit Dynamic Counter

As seen in Table 1, switches S_2 and S_3 are logically determined by the current input bit. Since the DAC must input the LSB (b_4) first, the control inputs to the multiplexer can be generated using a two bit down counter (3 to 0).

3.1 Edge Detector Circuit

The counter presented is based on a dynamic D-latch circuit, which requires an edge detector circuit to operate. Figure 5 shows a logic diagram for an edge detector. The output is high when the two inputs

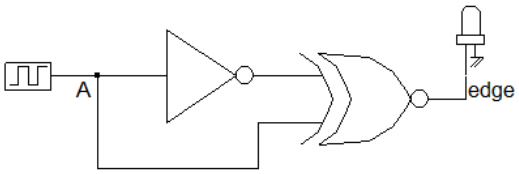

Figure 5: Edge Detector Logic Diagram

are the same, and this only occurs during a small interval due to the short delay of the inverter. A simple NXOR gate can be realized using pass gates and two inverters.

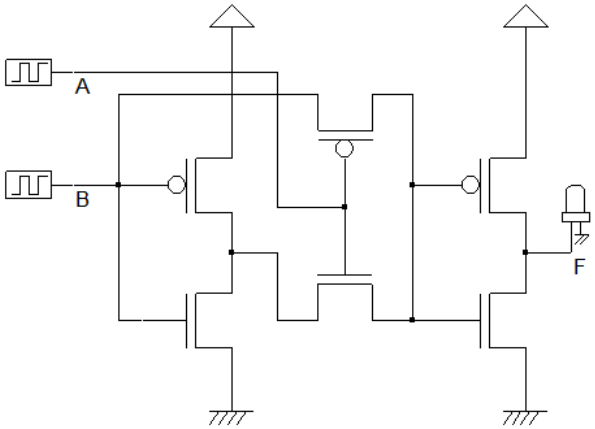

Figure 6: NXOR Gate CMOS Realization

Realizing the edge detector is simply a matter of adding another inverter. The delay inverter shown Figure 7 must have longer channel MOS devices in order to provide sufficient delay.

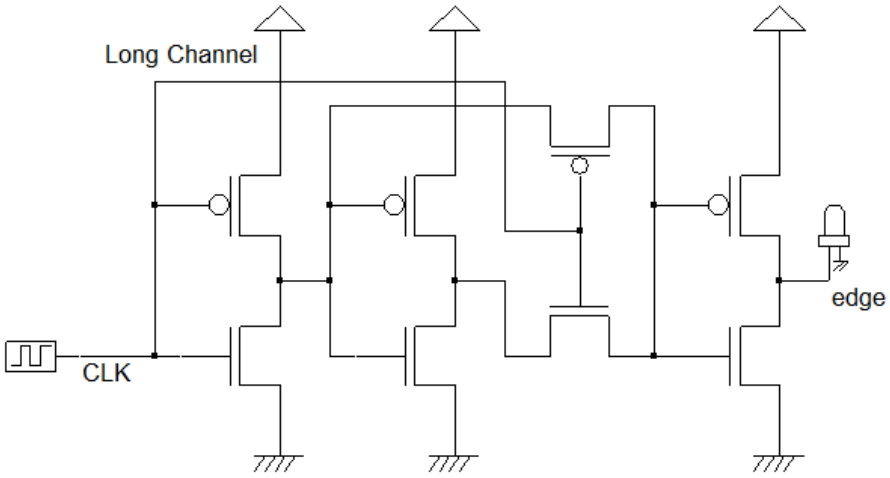

Figure 7: CMOS Edge Detector Schematic

The edge detector was laid out manually to allow for fine control over MOS characteristics. The delay

inverter has twice the length of the others, and it is specified as a low leakage device (slower). The other transistors are specified as low threshold voltage devices using the option layer–this has the effect of reducing delay in the inverters and allowing the PGs to pass better logic levels. It would seem that

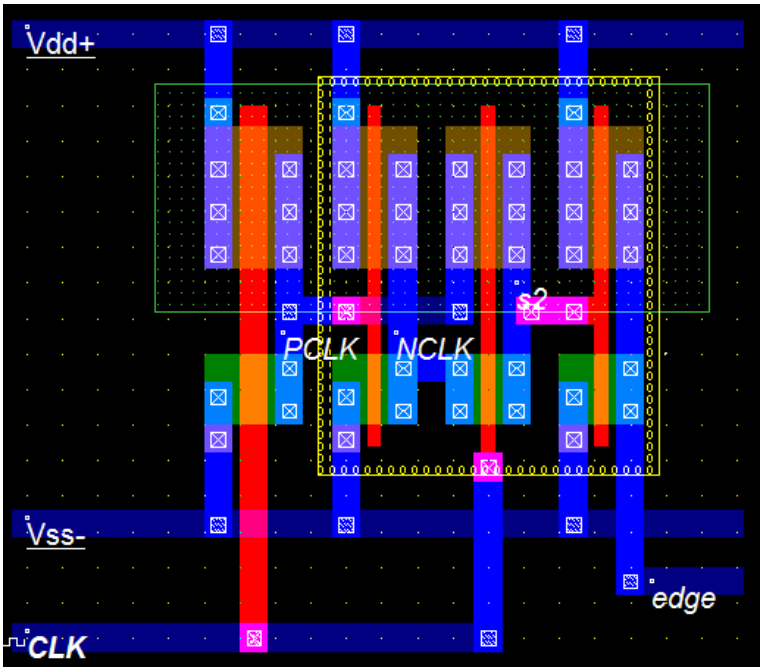

Figure 8: Edge Detector Layout

this circuit should detect both positive and negative edges, which is not desirable since an additional latch would be needed for the counter; however, the circuit will act as a positive edge detector with a reasonably good noise margin if the input into the NMOS PG (NCLK) has a faster edge than the signal going into the PMOS PG (PCLK). This was the goal of the layout design shown in Figure 8. Result of BSIM4 simulation is shown below.

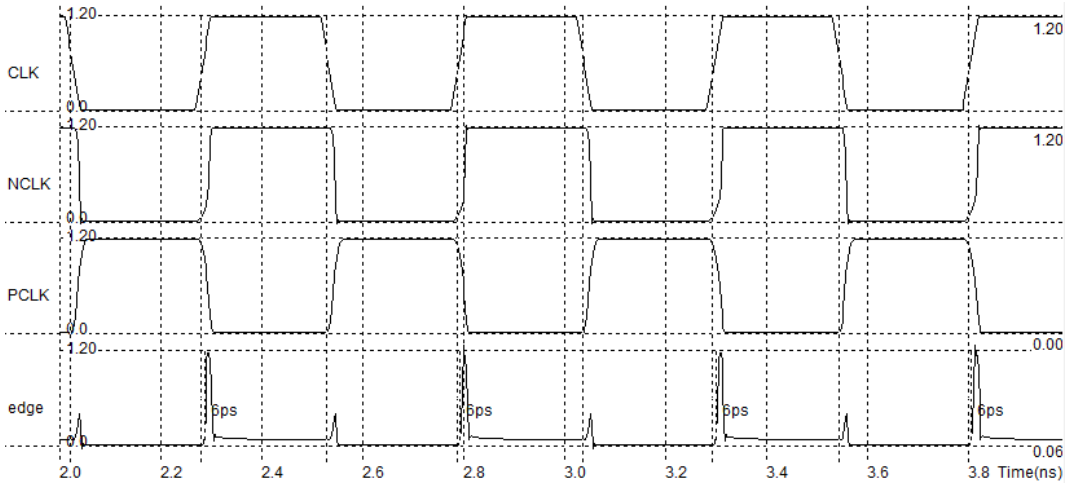

Figure 9: Edge Detector Timing Diagram

3.2 Dynamic D-Latch

The dynamic D-latch is an attractive option because it only requires five transistors: 1 NMOS PG and 2 inverters (a buffer). The layout is shown below.

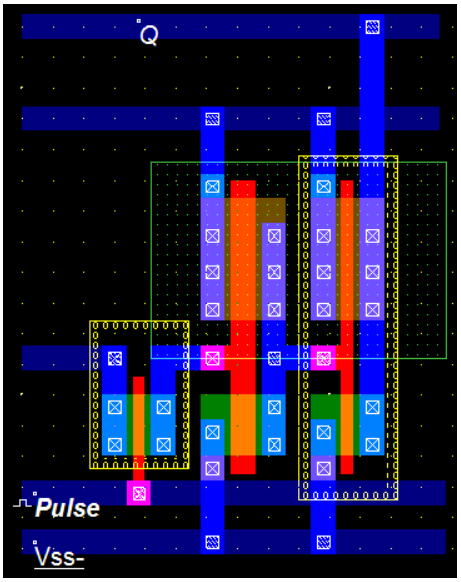

Figure 10: Dynamic D Latch Layout

Notice that the options mask does not surround the the first inverter. The gate of the first inverter is the storage node, so it is made to be a low leakage type, while the pass gate benefits from lower threshold voltage, as does the output inverter.

3.3 Synchronous Counter

The complete counter is shown in Figure 11, and the BSIM4 simulation results are shown in figure 12

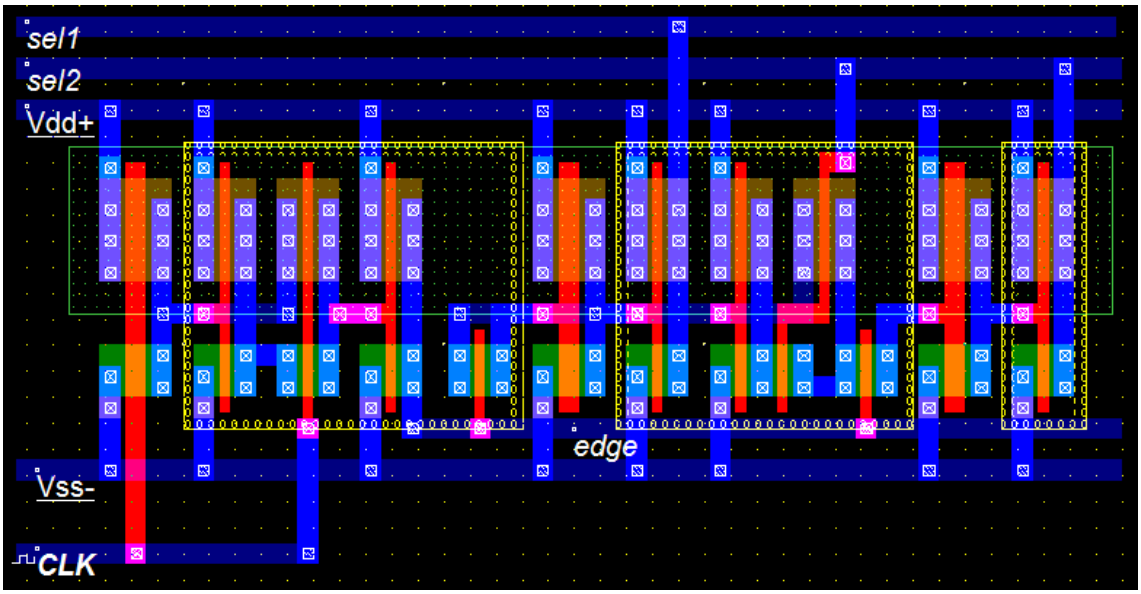

Figure 11: Synchronous Counter Layout

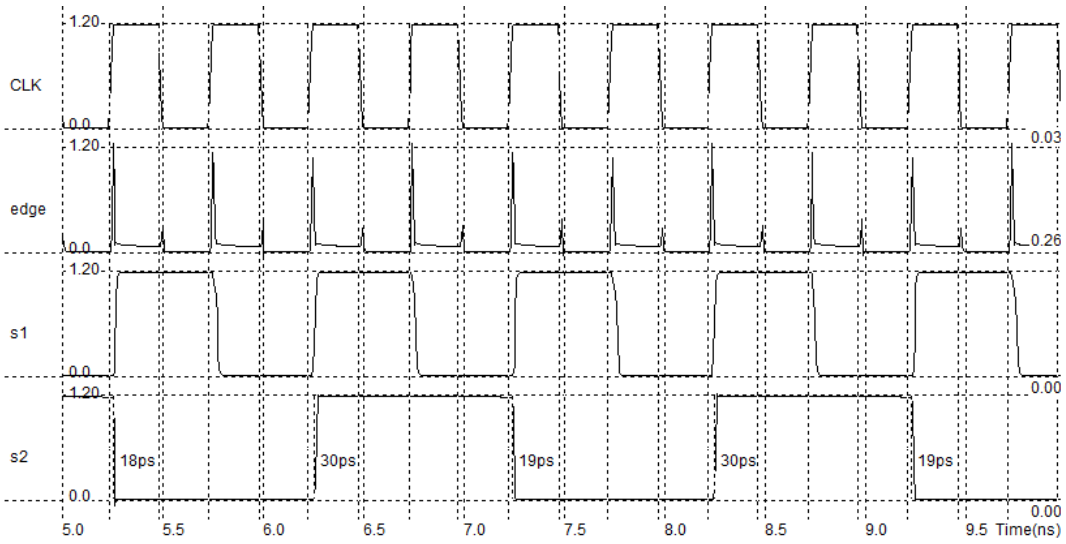

Figure 12: Counter Timing Diagram

4 Switching Control

The four switches S_i must be properly controlled to give D to A conversion. The logical expressions from which the switching signals are derived are summarized below.

$$
\begin{aligned}
S_1 &= \overline{CLK} \\
S_2 &= CLK \cdot Bit = \overline{\overline{CLK} + \overline{Bit}} \\
S_3 &= CLK \cdot \overline{Bit} = \overline{\overline{CLK} + Bit} \\
S_4 &= CLK \cdot Sel_1 \cdot Sel_2
\end{aligned}
$$

There is one caveat: simply using logical inputs such as the master clock and the outputs of the counter and multiplexer will produce unacceptable outputs due to mismatched delay and signal overlap. A work around is in order.

4.1 Non-overlapping Clock Generator

The non-overlapping clock generator described here is based on simple logical combinations of the original clock signal and a delayed version of it. The delay unit is essentially a slow buffer. The input transistor is made to have a small aspect ratio to limit the current that it can source. The output transistor is made to have a large aspect ratio and a relatively large gate area to increase its input capacitance, and allow for sourcing of larger currents. The circuit diagram is shown in figure 13, and the layout is shown in figure 14.Note that options mask is again invoked to specify that the long channel inverter is to have a thicker gate oxide to further reduce the current that it can source. This circuit can provide delays on the order of 100ps.

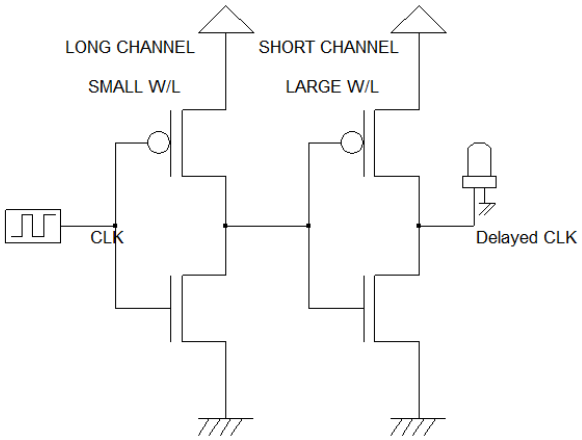

Figure 13: Delay Unit Schematic

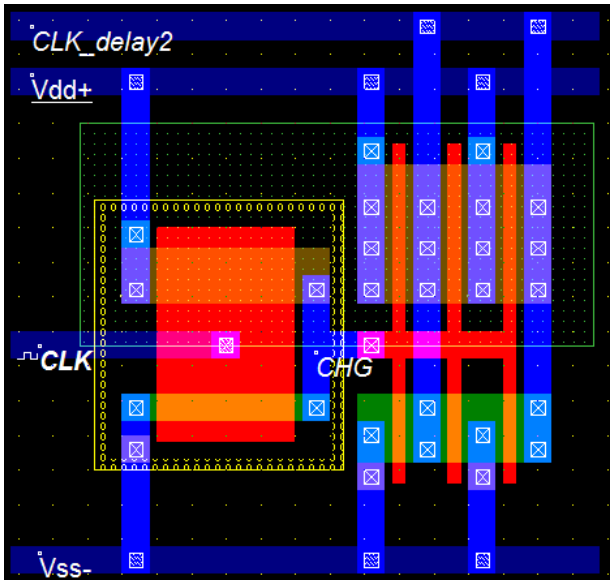

Figure 14: Delay Unit Layout

The logical combinations of the clock and the delayed clock required are easy to determine by looking at a plot of the two signals, as shown in figure 15. A pair of non-overlapping clocks can therefore be generated using the following logic:

$$\phi_1 = CLK \cdot CLKdel$$
$$\phi_2 = \overline{CLK} \cdot \overline{CLKdel} = \overline{CLK + CLKdel}$$

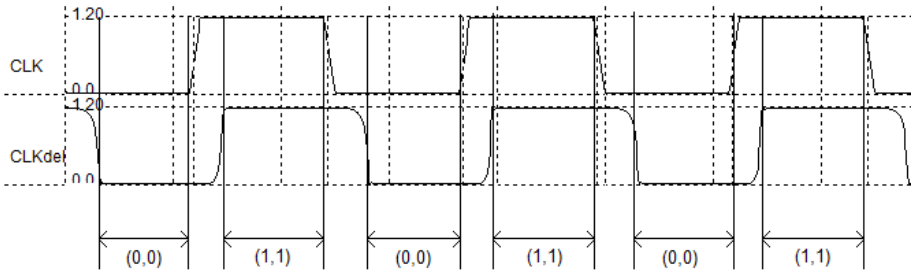

Figure 15: Plot for Determining Non-overlapping Clock Generation

The layout used is shown in figure 16, and the timing diagram is shown in 17. Note that ϕ_2 will be used as S_1.

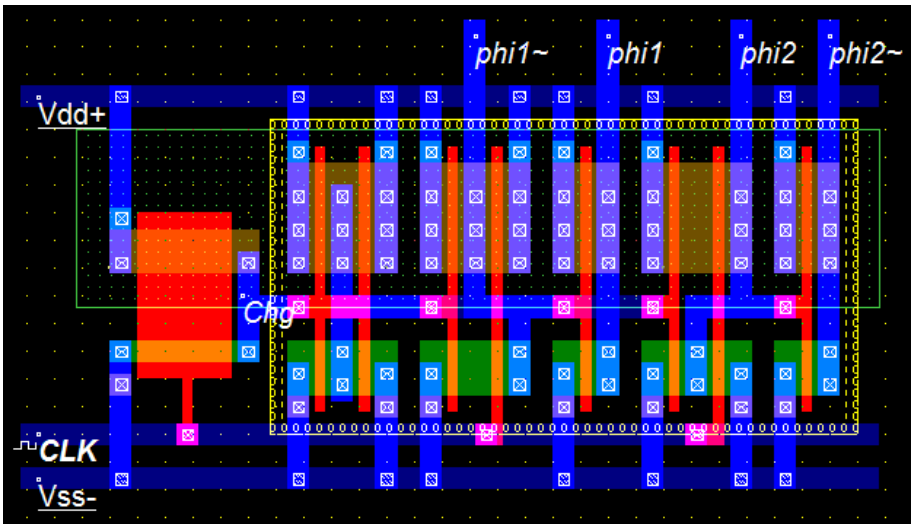

Figure 16: Non-overlapping Clock Generator Layout

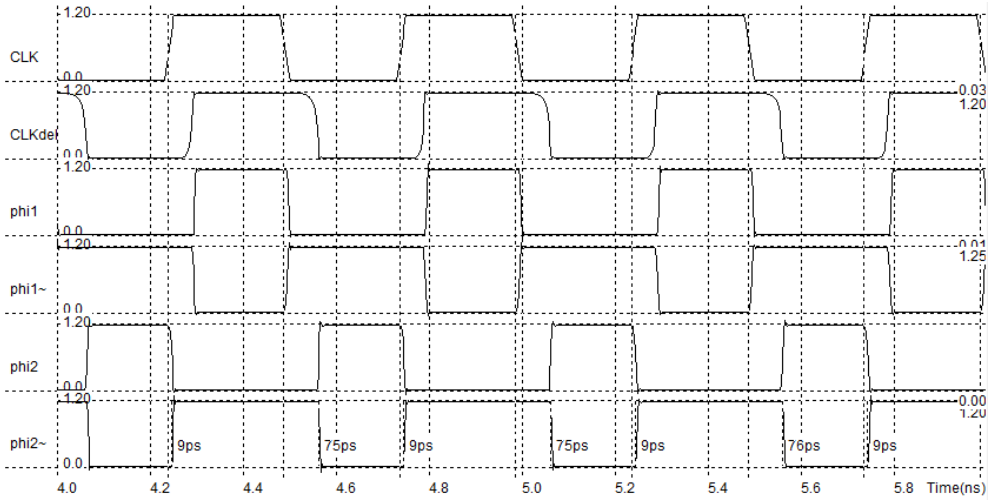

Figure 17: Non-overlapping Clock Generator Timing Diagram

4.2 Switch Logic

The modified logic expressions are summarized below.

$$S_1 = \phi_2$$
$$S_2 = \phi_1 \cdot Bit = \overline{\overline{\phi_1} + \overline{Bit}}$$
$$S_3 = \phi_1 \cdot \overline{Bit} = \overline{\overline{\phi_1} + Bit}$$
$$S_4 = \phi_1 \cdot Sel_1 \cdot Sel_2$$

Since the inverted counterparts of both clock signals are available, layout area can be saved by implementing the nor logic version, as shown in figure 18.

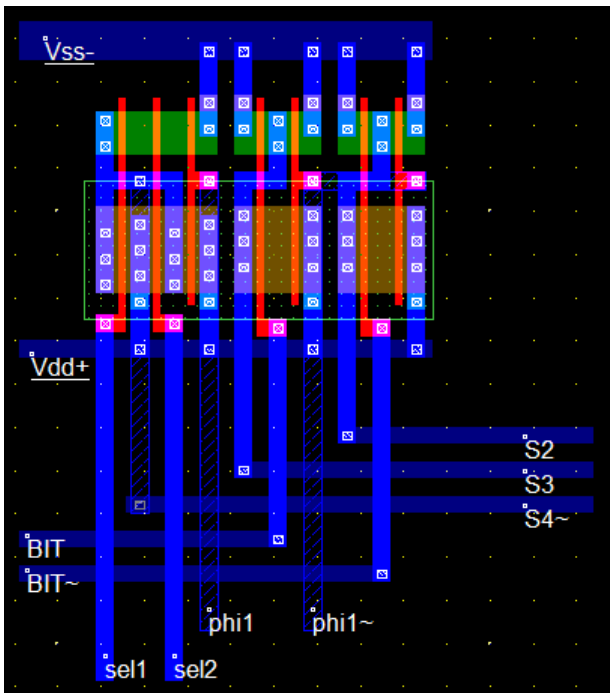

Figure 18: Layout of Switching Control Logic

5 Charge Redistribution Circuit

The charge redistribution circuit requires two capacitors and four switches. MOSFET capacitors are a good option since the gate oxide is very thin, resulting in more capacitance per unit area than other types of integrated capacitors. The layout in figure 19 shows two large area MOSFET caps (C_1 and C_2) laid out using a simple interdigitization scheme, connected together and to Vdd and Vss through our TG switches. The capacitance of each was calculated in Microwind to be just under 40fF

6 Sampling Circuit

Connecting all of the modules together and simulating reveals that the conversion from digital to analog is complete on the second clock phase of the last clock cycle (one half of a clock cycle before S_4 turns on). At this time, it is desirable to sample the voltage across C_1 or C_2 Since the capacitors require some time to charge up, the delay circuit used earlier will be used again. If delay and overlap were not a problem, the logic for the sampling signal would be:

$$sample = \overline{sel_1} \cdot \overline{sel_2} \cdot \phi_2 \tag{1}$$

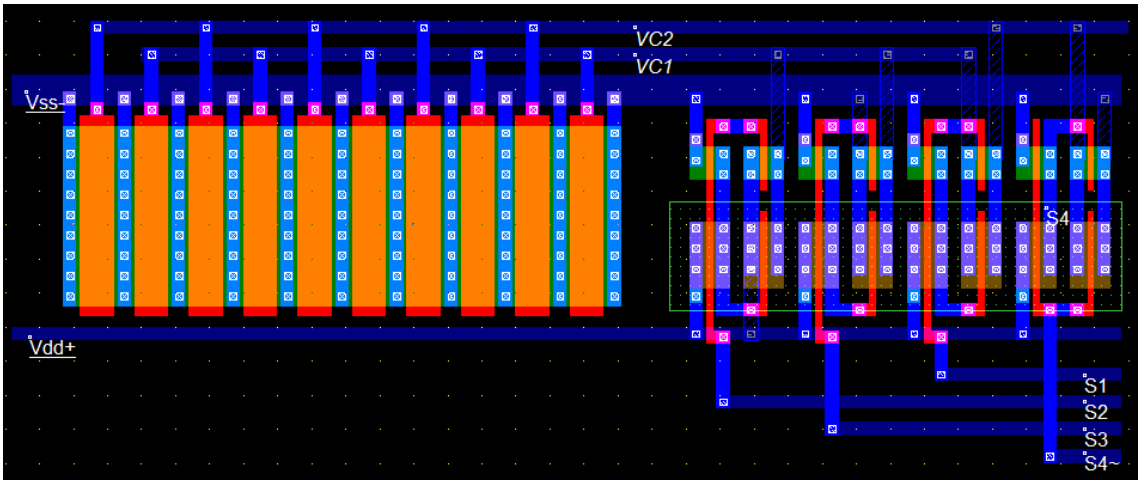

Figure 19: Switched Capacitor Circuit Layout

However, The voltage across the capacitor should be allowed to settle before sampling, so the sample pulse should be delayed:

$$sample' = sample(delayed) \tag{2}$$

This introduces another problem: The sampling pulse now hangs into clock phase 1, when C_2 is discharged, so the output will drop by some amount depending on the delay used. To solve this, the sampling control signal can be made from the logical and of the delayed control signal (sample') and S_1 to ensure that it can only occur during clock phase 2.

$$sample'' = sample' \cdot S_1 \tag{3}$$

The layout for generating this signal is shown in figure 20.

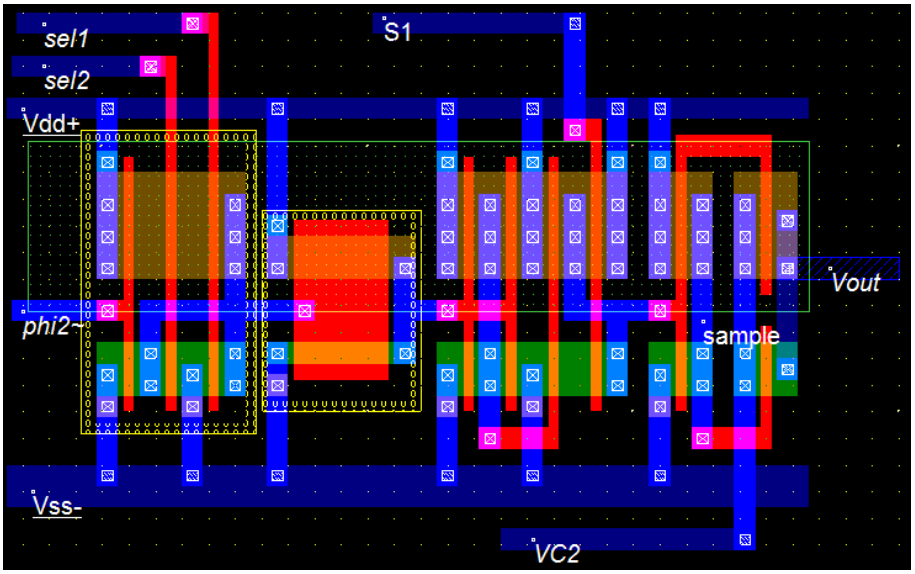

Figure 20: Layout of Sampling Circuit

7 The Complete DAC

The layout for the complete DAC is shown in figure 21, and the timing diagram showing the system's response to an input of 1011 is shown in figure 22

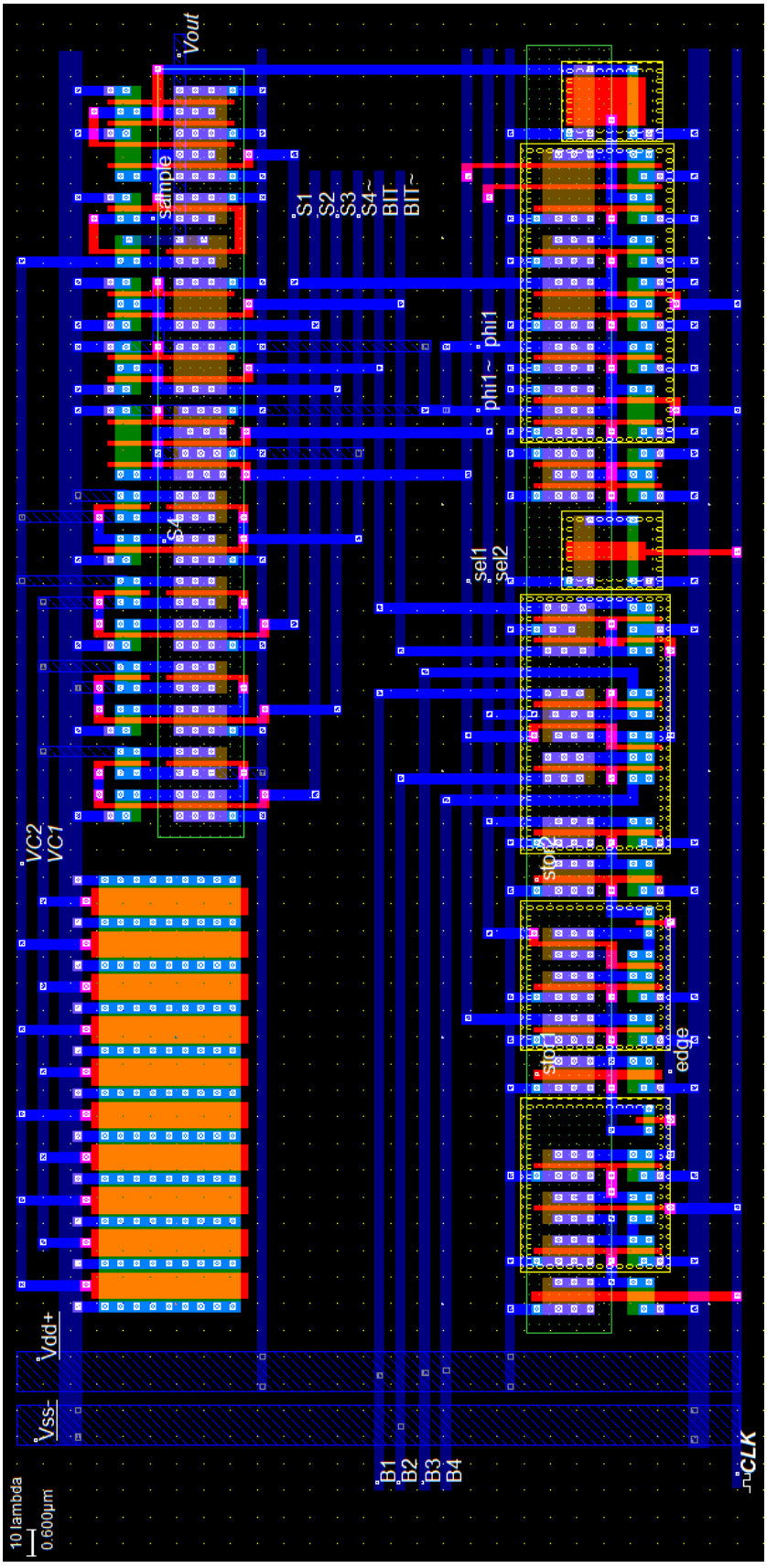

Figure 21: Complete DAC layout

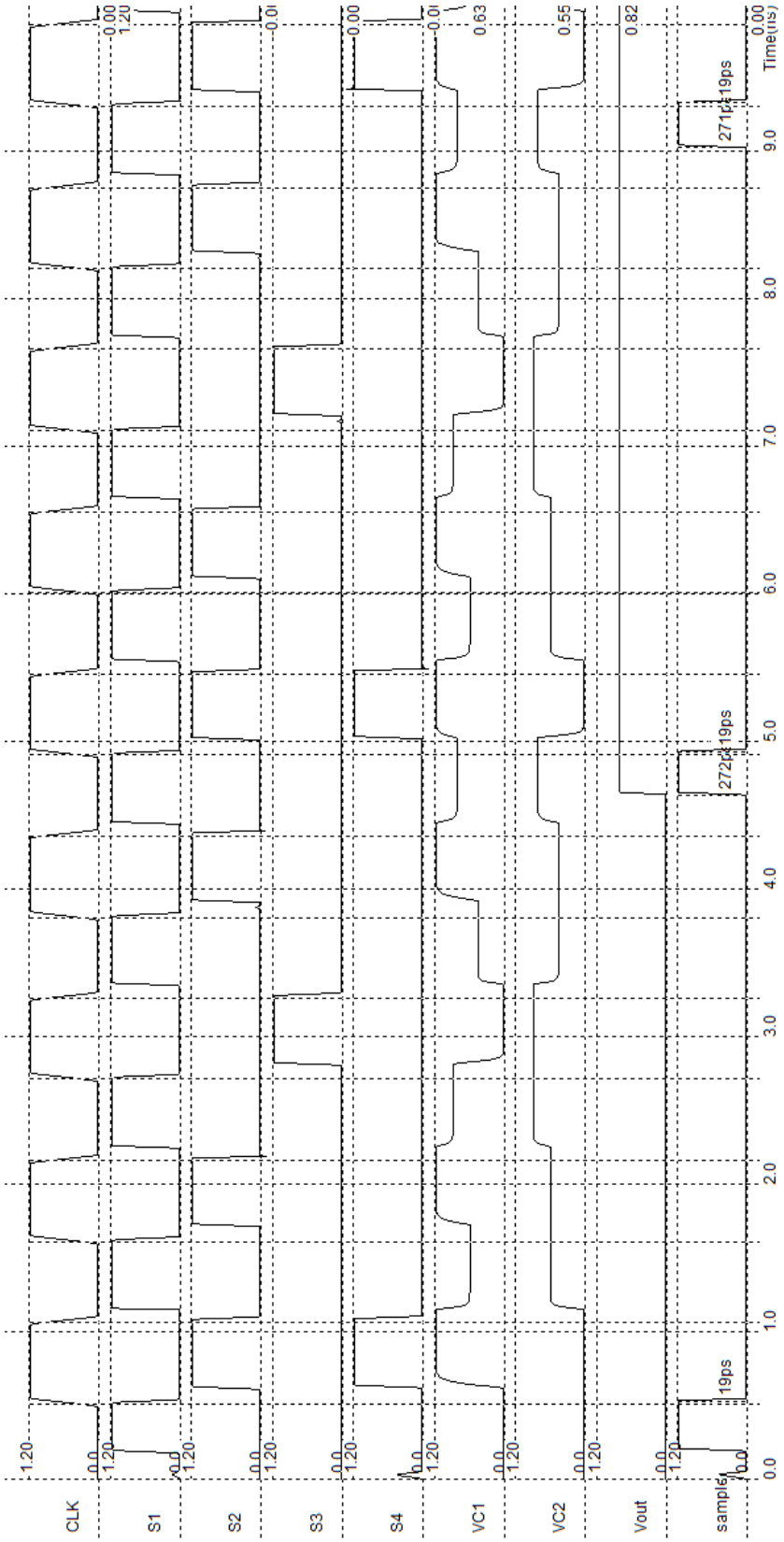

Figure 22: DAC timing diagram with input 1011

7.1 Analysis

The DAC was tested using BSIM4 simulation for all possible inputs (binary 0 through 15) by alternately connecting the B_i inputs to the Vdd or Vss bus. The raw results are summarized in the plot in figure 23.

The response appears to be very linear, but the system could benefit from gain compensation. Dividing

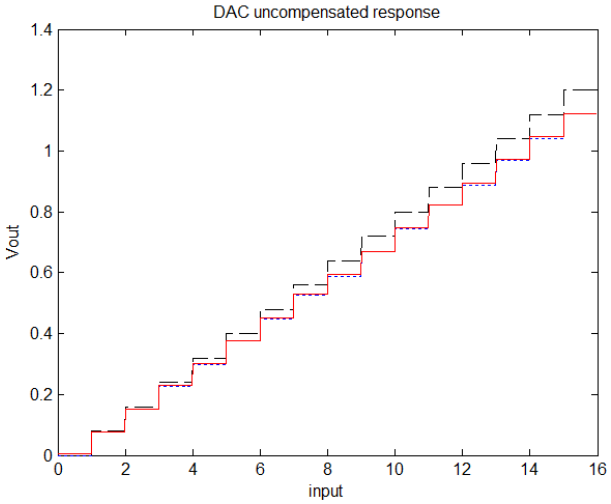

Figure 23: Raw BSIM4 Simulation Results

the ideal output when the input is 15 by the experimentally observed output gives a compensation gain of 1.0705. The plot in figure 24 shows the output with gain compensation. As can be seen, simple gain

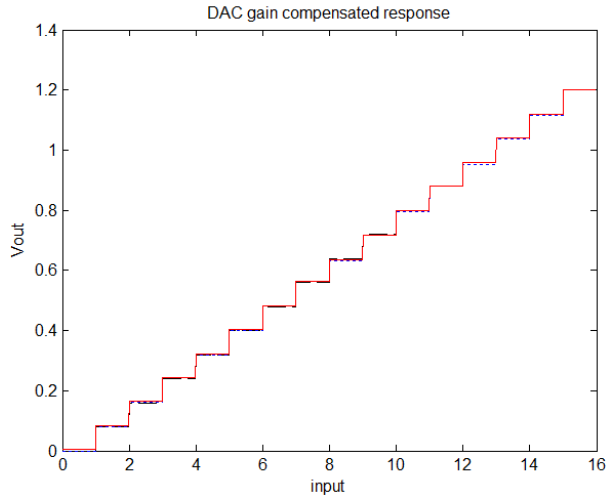

Figure 24: Simulation Results with Compensation

compensation gives very good results, making the performance of this DAC is close to ideal.

8 Conclusion

The design, layout and simulation of the four bit serial DAC based on MOS capacitors, TG switches and a dynamic counter was demonstrated to be successful. Several benefits of the design presented over a more straightforward approach (i.e. automatic layout): 1. By generating and assembling logic gates and latches manually, layout area can be conserved. 2. Using a synchronous counter instead of a ripple counter to control the parallel to serial conversion becomes beneficial as the number of bits increases. 3. The use of non-overlapping clocks removes glitches from the control signals. One important consideration which was not addressed was output loading. The output node had a capacitance of near 0.5fF. This is just over one percent of the capacitance of the capacitors in the charge redistribution circuit. If a larger load capacitance is to be encountered, two changes can be made to accommodate: 1. the capacitors C1 and

C2 can be increased in value–the downside is increased area. 2. Due to their larger size, the capacitors would take longer to charge, so the switches S_i must be made to have longer widths to increase current flow, which will also increase layout area. Also, the edge detector used has a somewhat compromised noise margin (small negative edge pulses show at the output). A more robust rising edge detector can be made in the same way that the non-overlapping clock signals were generated, at the expense of a few additional transistors.