Design Project

Pseudo-Random Bit Sequence Generator

Prepared by:
Steven Blair
Alex Kapustka
Jeffrey Lee
Alex Wleklinski

For Access to Files Not Included with This Report
(i.e., Schematics, Layouts, Waveforms, etc.) See:
https://github.com/red-135/ECE482_DesignProject
(Files Maintained by Steven Blair, smblair2@illinois.edu)

ECE 482
Digital IC Design
December 7, 2016

The specifications state that the system accepts four inputs (e.g., a clock signal [CLK] that causes the circuit to compute the next random bit, a reset signal [RST] that resets the circuit to an initial state, and two select signals [C1 and C2] that determine the appropriate output driver for a given capacitive load) and provides one output (e.g., an output signal [D] that provides one bit from a random bit sequence). In order to operate on these signals, the system was partitioned into several modules, including:

Three Schmitt Triggers: To speed up transitions on the slowly varying reset and select signals, a Schmitt trigger was constructed to buffer the input signals and distribute the buffered signals throughout the system. The Schmitt trigger was constructed using two inverters with additional pull-up and pull-down lines connected in a feedback configuration. Transistor sizes were set to yield a hysteresis loop that was narrow, symmetric, and centered—yielding a quick switching speed with the additional advantages of small short circuit power consumption.

A Pulse Generator: To ensure that the system functions correctly regardless of startup state, the system had to be reset to an initial state at the beginning of operation; however, to prevent the system from being reset in every clock cycle due to a constantly applied reset signal, a pulse generator had to be used to produce a one clock cycle reset pulse on every rising edge of the reset signal. This was accomplished using a finite state machine that remembers if the reset signal was asserted in the prior clock cycle and that asserts its output only when the reset signal was not applied in the prior clock cycle but is applied in the current clock cycle.

A Linear Feedback Shift Register: To generate the random bit sequence, a linear feedback shift register (LFSR) was constructed consisting of a shift register with the input of the first register connected to the outputs of the other registers through an XOR gate. To maximize speed and robustness, a complementary static CMOS implementation of an XOR gate was used, but to ensure optimal performance, a variety of register architectures were examined. Ultimately, a true single-phase clock (TSPC) register was selected over other registers (e.g., static complementary CMOS, C^2MOS, pulse registers) since it balanced between area, power, speed, and robustness.

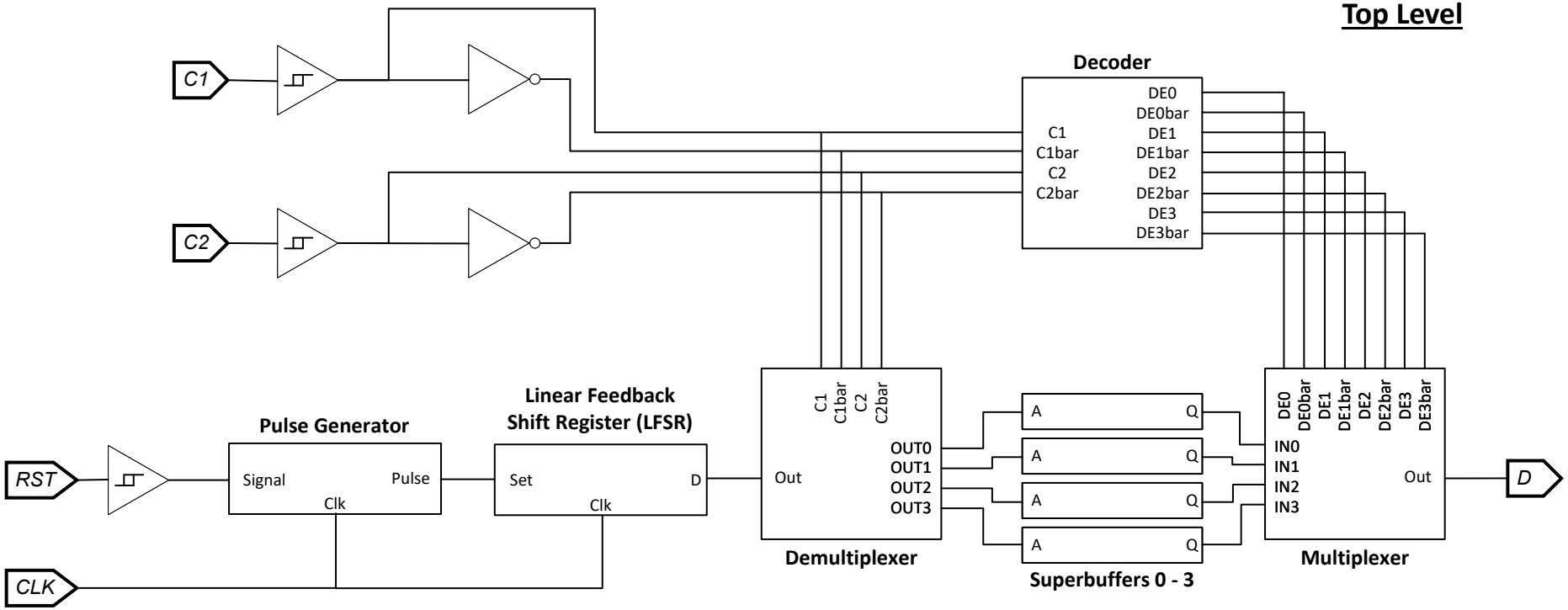
Four Superbuffers: To drive large capacitive loads while minimizing supply bounce, four different drive chains were constructed to correspond to four different ranges of capacitive loads. Each drive chain took the form of a four-stage superbuffer with the first stage being sized to minimum size and with following stages being scaled by a scaling factor that was set based on the capacitive load (i.e., a larger load required a larger scaling factor). The number of stages and the scaling factor were initially set based on ideal superbuffer design but were later refined based on simulations to ensure that the superbuffer designated for a range of capacitive loads could barely drive the largest capacitive load in the range in a single clock cycle, slowing the current drawn by the superbuffer and reducing supply bounce.

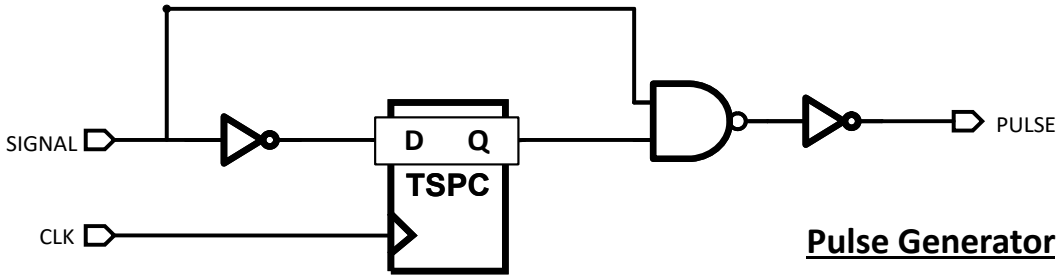
A Demultiplexer: To reduce power consumption, our team attempted to minimize unnecessary switching in the drive chains by connecting the random bit sequence from the linear feedback shift register through a demultiplexer to the input of the appropriate drive chain and by pulling the inputs of the other drive chains up using a pull-up transistor (ensuring that one drive chain sees a signal that is switching and that other drive chains don't begin switching due to capacitive coupling to floating nodes).The demultiplexer took the form of four branches of two series pass transistors connected to the select signals so that a single branch conducted at a time. Minimum size inverters were used since the linear feedback shift register had no problem driving current through the demultiplexer.

A Multiplexer and A Decoder: To eliminate contention and reduce power consumption at the output, our team connected the correct drive chain to the output and disconnected the other drive chains from the output using a multiplexer and a decoder. The multiplexer took the form of four branches of transmission gates connected to signals from the decoder so that a single branch conducted at a time, and the decoder took the form of a set of NAND gates and inverters. To ensure that the multiplexer did not excessively slow the signals from the superbuffers, the size of a transmission gate connected to a superbuffer was set proportional to the size of the last stage of the superbuffer.

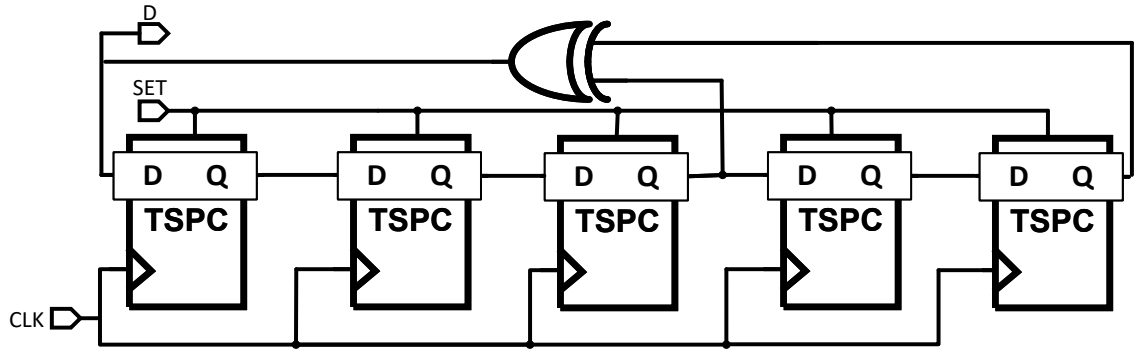
During design and implementation, three steps were followed. First, schematics were produced at the module and system levels and were simulated to ensure correct operation (using Virtuoso Schematic and ADE). Second, layouts were produced and further simulations were executed to ensure minimal changes to performance (using Virtuoso Layout and ADE). Third, all relevant results were collected for presentation (using extracted netlists and HSPICE).

Top Level

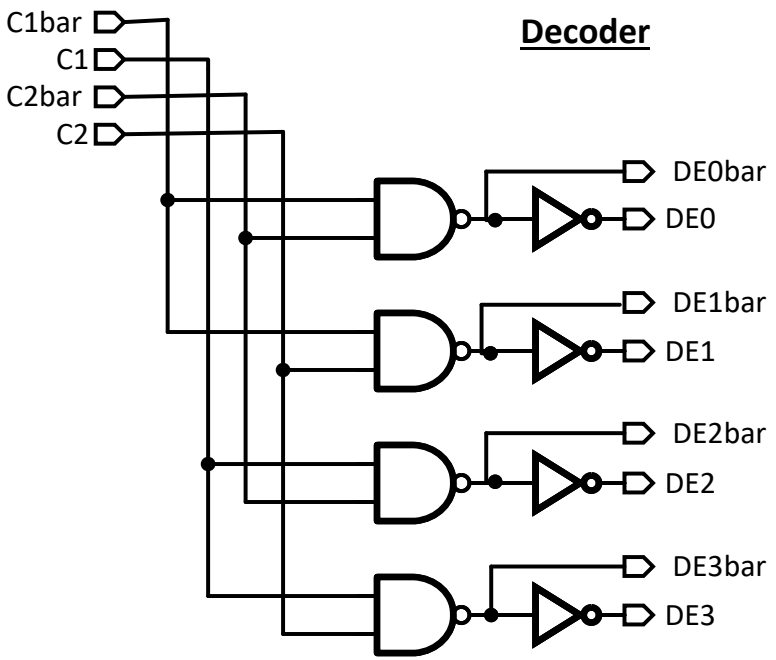




Pulse Generator

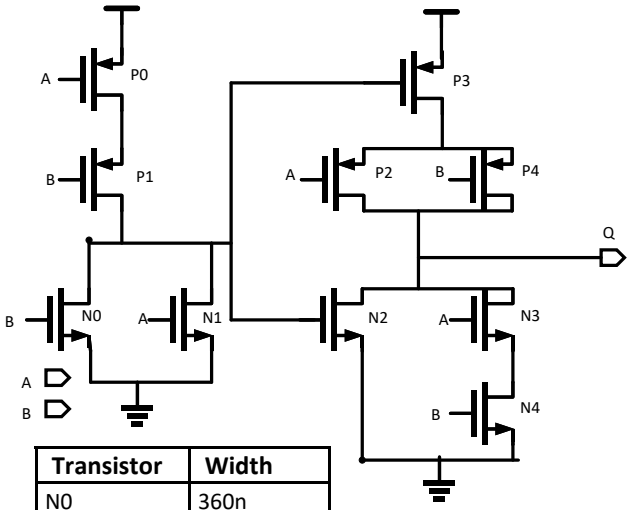


Linear Feedback Shift Register (LFSR)



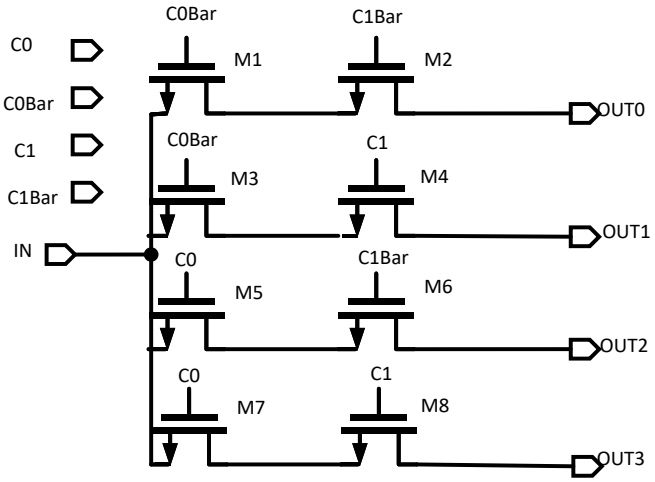
Decoder

See Appendix for Superbuffer Design….



Transistor	Width
N0	360n
N1	360n
N2	360n
N3	720n
N4	720n
P0	1.44u
P1	1.44u
P2	1.44u
P3	1.44u
P4	1.44u

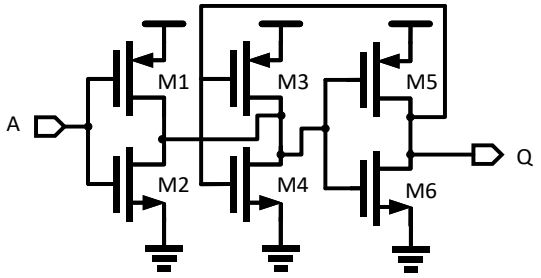
XOR2



Demultiplexer

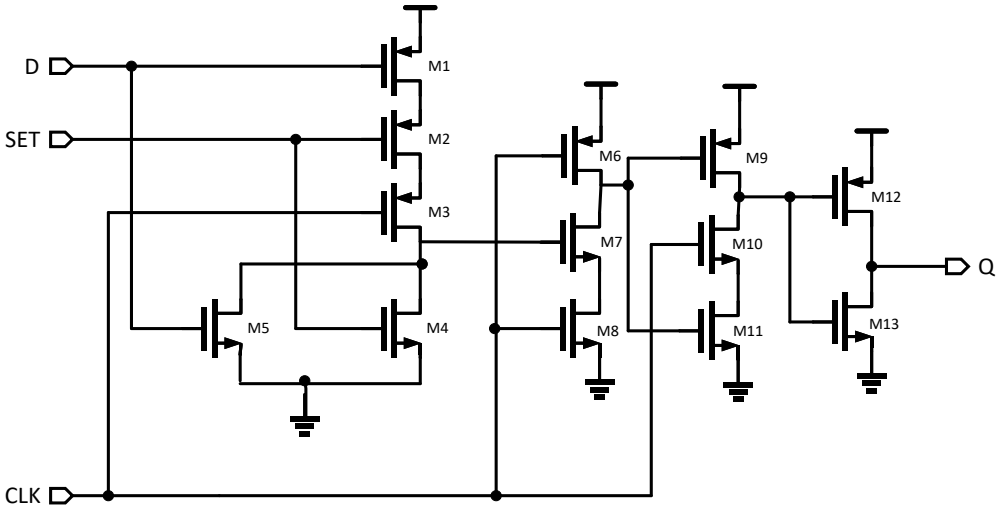
Transistor	Width
P0-P2	720nm
P3	2.88um

Schmitt Trigger



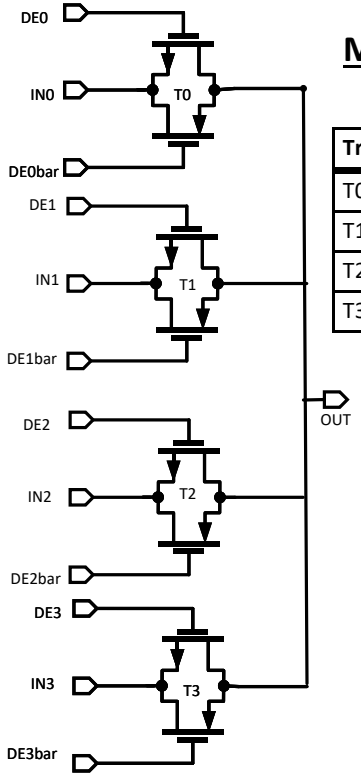
Transistor	Width
M1	2.46u
M2	540n
M3	720n
M4	360n
M5	1.44u
M6	720n

Register with Set (TSPC)



Transistor	Width
M1	2.16u
M2	2.16u
M3	2.16u
M4	360n
M5	360n
M6	720n
M7	720n
M8	720n
M9	720n
M10	720n
M11	720n
M12	720n
M13	360n

Multiplexer



Transistor	Width
T0	15.42u
T1	45u
T2	77.76u
T3	98.88u

Figure 1: Functional Performance Across Capacitive Loads (In SS Corner)

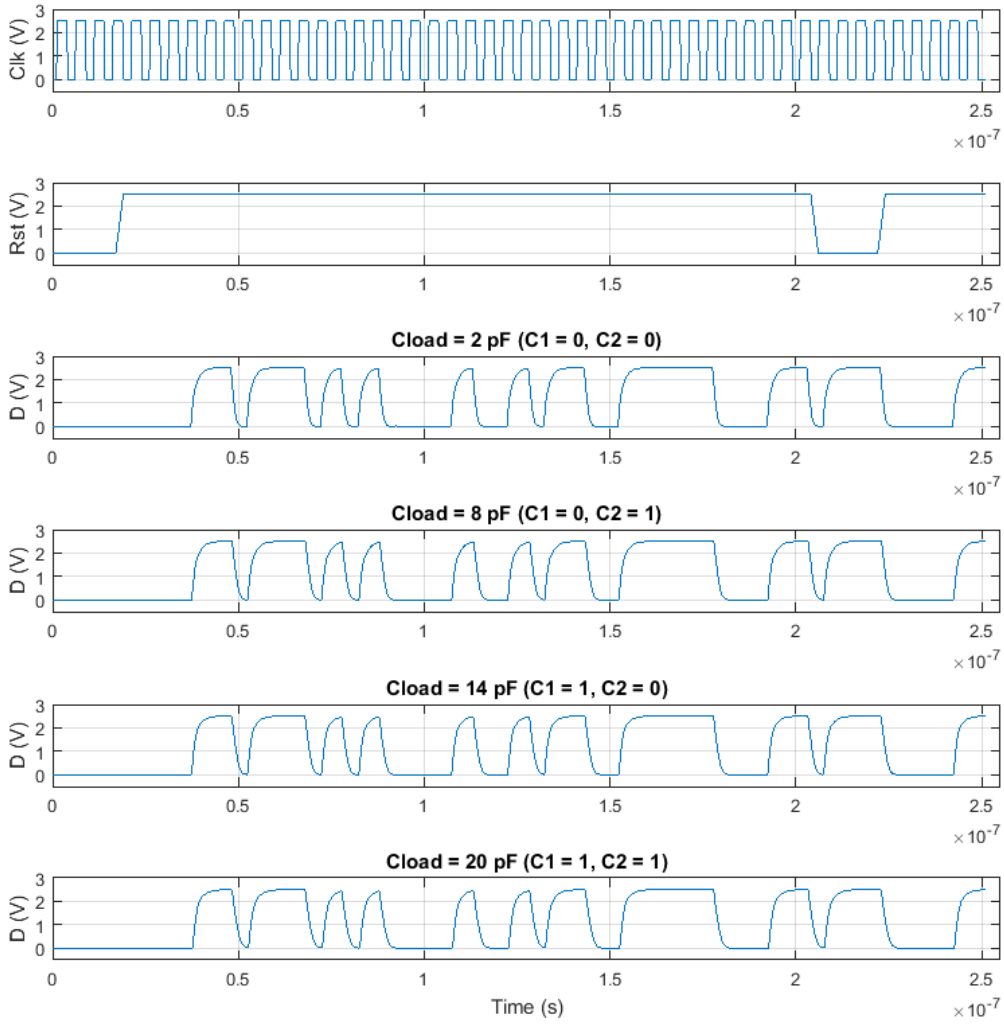


Table 1: Power Dissipation Across Capacitive Loads (In TT Corner)

Capacitive Load	Average Power
2 pF (C1=0, C2=0)	1.042 mW
8 pF (C1=0, C2=1)	2.789 mW
14 pF (C1=1, C2=0)	4.604 mW
20 pF (C1=1, C2=1)	6.340 mW

The power consumption of this system is dominated by the switching power required to drive the parasitic capacitances in the superbuffer connected to the output as well as the capacitive load at the output. The switching power consumed by the clock as well as the static and short-circuit power consumed by the transistors in the system contributed a small but meaningful amount of power, while the power consumed by the reset and select signals was negligible. As expected based on these observations, there is a large change in the power consumed by the circuit as the capacitive load is varied from its largest value (consuming maximum power) to its smallest value (consuming minimal power).

Figure 2: Supply Bounce Across Capacitive Loads (In FF Corner)

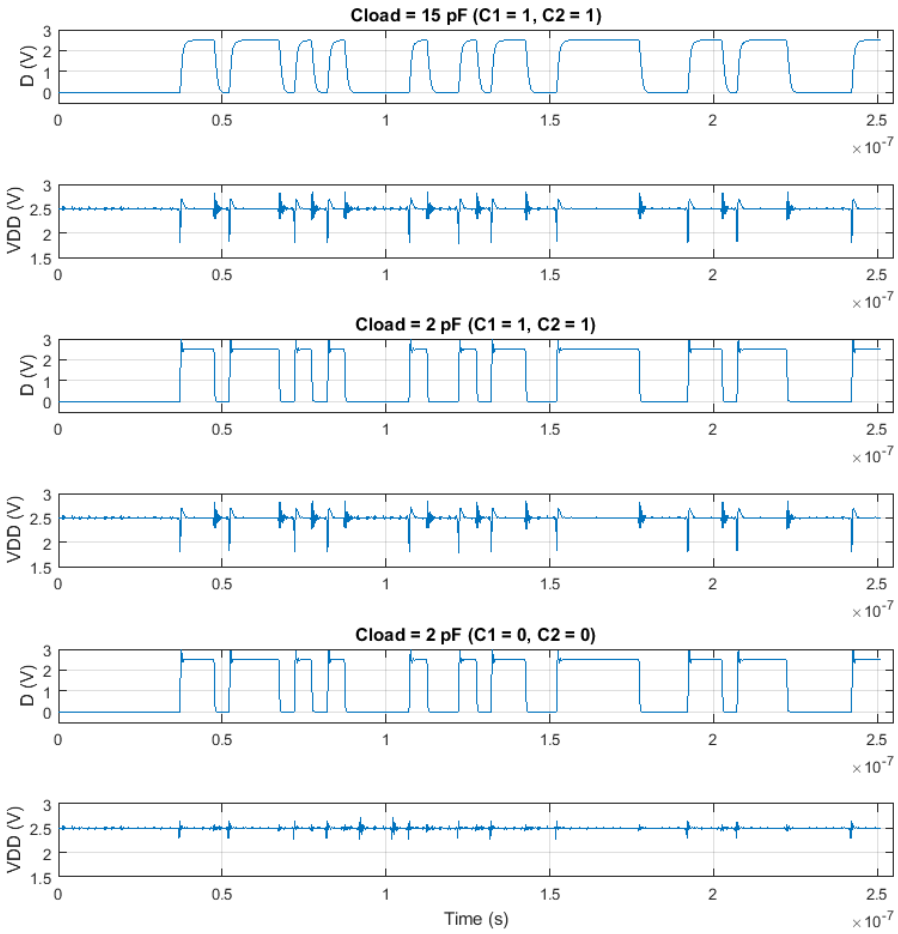


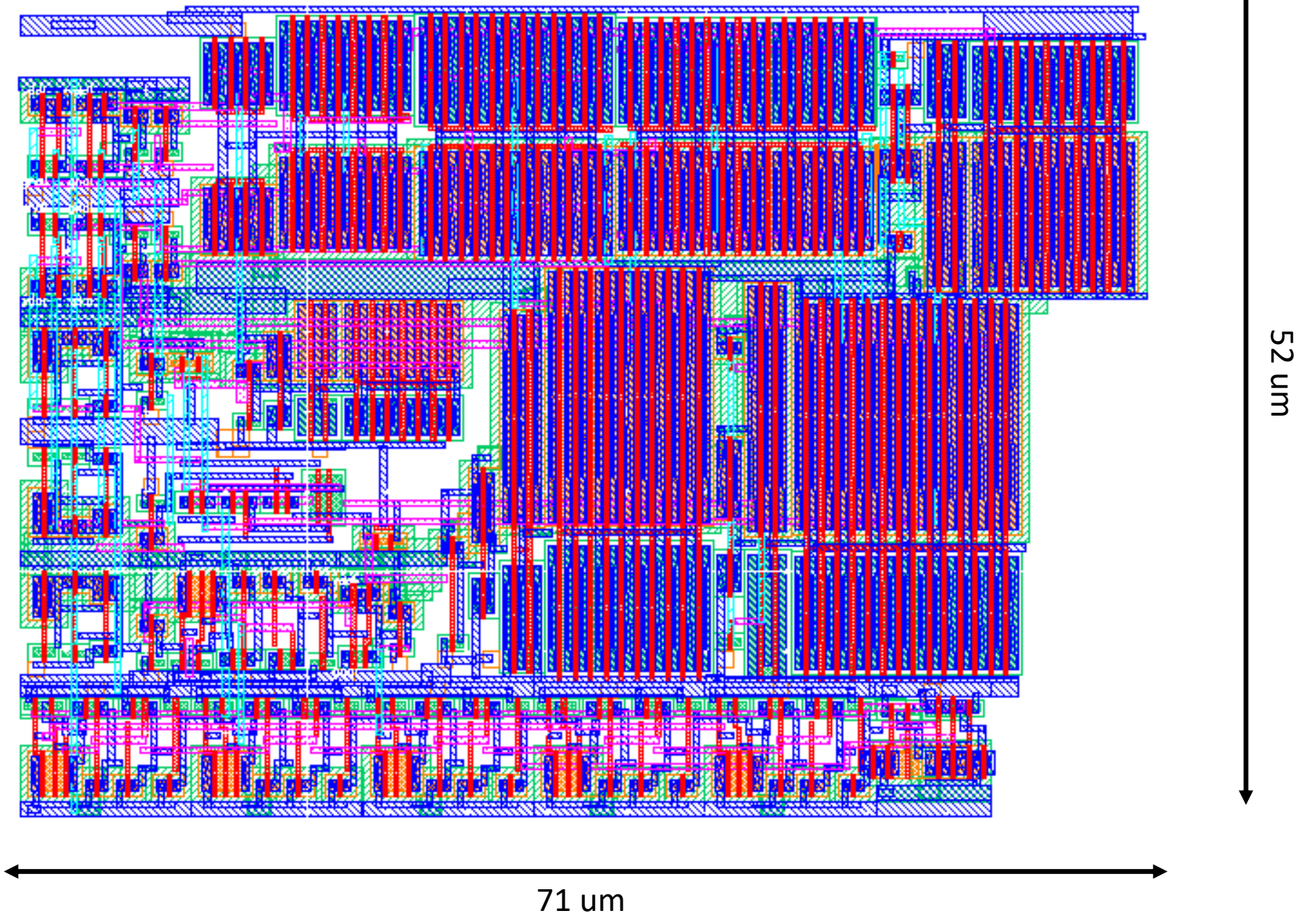
Table 2: Maximum Peak-to-Peak Supply Bounce (In FF Corner)

Capacitive Load	Maximum Peak-to-Peak Supply Bounce
15 pF (C1=1, C2=1)	1.0516 V
2 pF (C1=1, C2=1)	1.2795 V
2 pF (C1=0, C2=0)	0.4523 V

The voltage across an inductor is proportional to the time derivative of the current through the inductor. While very little current is supplied from VDD or dumped to GND when the output signal is constant, a large amount of current is moved when the output signal switches since every parasitic capacitor in the superbuffer connected to the output as well as the capacitive load at the output must be charged from GND to VDD or discharged from VDD to GND. When two inductors were introduced between the power supply and the supply rails of the system, this change in current induces a voltage drop across the inductors that changes the voltages of VDD and GND.

When the superbuffers drive the maximum designed load, they supply the minimum amount of current needed to achieve timing closure, but as the load is decreased below the maximum load, current is moved to or from the load faster than is necessary to achieve timing closure. As a result, for the maximum load, there is a minimum change in current which induces a small change in the power rails, and for loads that are below the maximum load, there is a larger change in current that induces a larger change in the power rails. The data presented for the system corroborates this theory.

Full Layout



Statements of Contribution

Steven Blair: At a high level, I was primarily responsible for maintaining the library of circuit symbols and schematics that were used in the final design; producing the top level schematic for the final design; running simulations on the final design to verify performance; and teaching and debugging the various tools that our team during the project used that were not used in normal classroom assignments (i.e., Virtuoso Schematic and ADE). At a lower level, I produced schematics for two pulse registers that were considered for selection as our final register, refined the schematic for the TSPC register that was selected as our final register, and produced schematics for the Schmitt trigger and decoder module; additionally, I produced layouts for the Schmitt Trigger and NAND2 gate. Toward the end of the project, I found the final sizes for the superbuffers and multiplexer that achieved the minimum necessary sizing for timing closure, and I contributed to the final report by generating waveforms and diagrams for the report, producing writeups for the report, and putting the report together prior to printing.

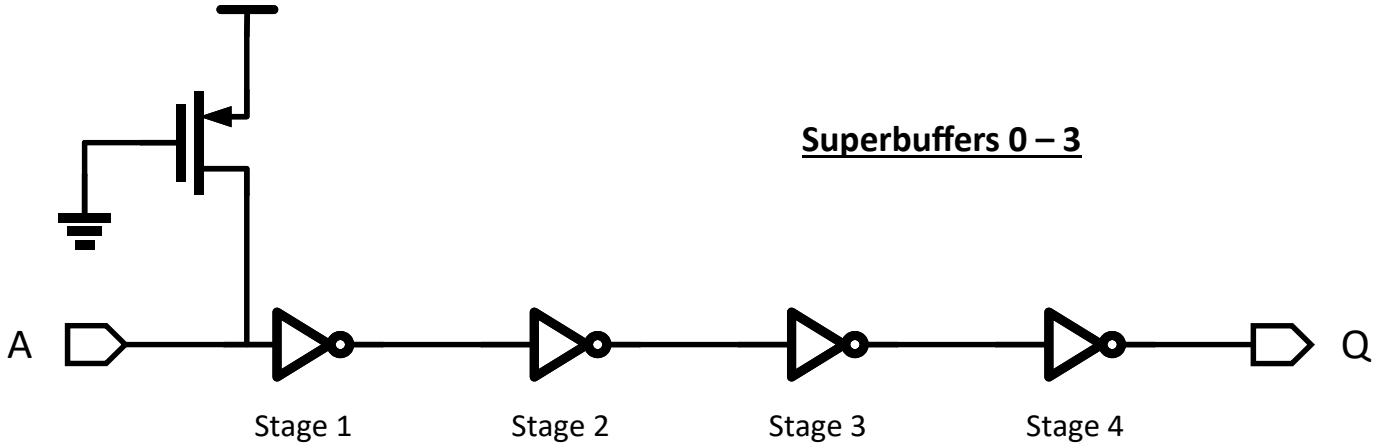
Alex Kaputska: When the team examined the various register designs before deciding which one to use for the final design, I created the schematic for a complementary static CMOS design and ran several simple simulations to verify the performance of the design. When we moved on to the other components of the design, I did the calculations and schematics for the four superbuffers used for the four different capacitive load ranges. When it came time to do layouts, I did the initial layouts for the multiplexer and demultiplexer which needed to be tweaked later when the design changed. When we moved onto the simulations, I worked with Jeffrey to simulate our extracted netlist with all four loads over every operating corner, verifying correct operation and exporting the plots needed for the main part of the report. Finally, when we had to draw the schematics for the report in Microsoft Visio, I did the TSPC register and the decoder then gathered all of the transistor-level schematics together so that Steven and Alex could do the final layout for the report.

Jeffrey Lee: During the schematic production part of the project, everyone produced a different register design so that we could compare the designs and select the best design for implementation in the linear feedback shift register. While I worked on a C^2MOS register, we decided that the C^2MOS register consumed too much power and decided on a different register instead. During the layout production part of the project, I produced initial layouts for each of the superbuffers based on designs that my group members gave me. When the report was being put together, I collected the extracted the netlists and ran the simulations needed to acquire plots for the report, and I created transistor-level schematics for the demultiplexer, multiplexer, and XOR modules as well as gate-level schematics for the pulse generator module.

Alexandra Wleklinski: I contributed schematics and layouts for the XOR gate, TSPC register, pulse generator, and linear feedback shift register; produced layouts for many other blocks in the system including the final implementations of the superbuffers, multiplexer, demultiplexer, linear feedback shift register, and decoder; and contributed to the routing of the top-level system. The most challenging task of layout was resizing the superbuffers and incorporating multipliers and fingers in the layout to get a good fit for the top-level system. Additionally, I helped in verifying and debugging correct operation throughout the project and especially during the transition from schematic to layout.

Appendix 1

Superbuffer Sizing



Transistor Widths in Superbuffers by Stage

	Superbuffer 0	Superbuffer 1	Superbuffer 2	Superbuffer 3
Stage 1—PMOS	720n	720n	720n	720n
Stage 1—NMOS	360n	360n	360n	360n
Stage 2—PMOS	2.52u	3.60u	4.32u	4.68u
Stage 2—NMOS	1.26u	1.80u	2.16u	2.34u
Stage 3—PMOS	8.82u	18.00u	25.92u	30.42u
Stage 3—NMOS	4.44u	9.00u	12.96u	15.24u
Stage 4—PMOS	30.90u	90.00u	155.52u	197.76u
Stage 4—NMOS	15.42u	45.00u	77.76u	98.88u
PMOS-to-NMOS	2	2	2	2
Scaling Factor be-	3.5	5.0	6.0	6.5
Pull-Up Width	360n	360n	360n	360n