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).