CPET-233 Digital Systems Design
Fall 2019

Congratulations! You have just been hired by the National Security Agency (NSA) to design a new encryption key generation system. Every day NSA generates a new encryption key that they transmit to every secure communication system in use by the United States military. This key is used by the encryption algorithms to ensure data communication that, if intercepted, cannot be decrypted. The system you design is an important part of our nations's security.

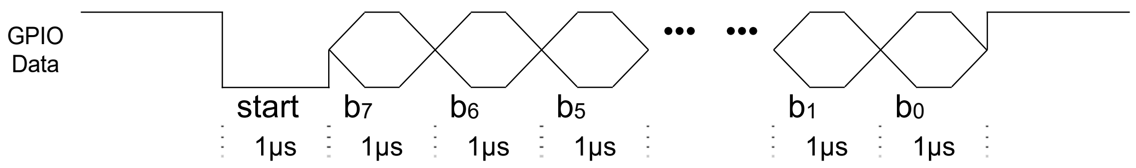
A block diagram of the system is shown on the next page. Your system will generate and transmit an 8-bit encryption key.

The system works as follows

1. The user inputs an 8-bit seed value for the 8-bit random number generator (RNG)
2. The seed is locked in with the pushbutton (PB) input
3. The RNG is then clocked EXACTLY 30 times (this is controlled by the counter)
4. The random number is then loaded into an 8-bit shift register
5. The shift register is clocked at a 1 us rate as its serial output data is put on the General Purpose I/O pin (GPIO). Prior to the 8 data bits being transmitted, a 1us low pulse should be transmitted (see waveform below)
6. After 9 bits have been transmitted, the system returns to idle.

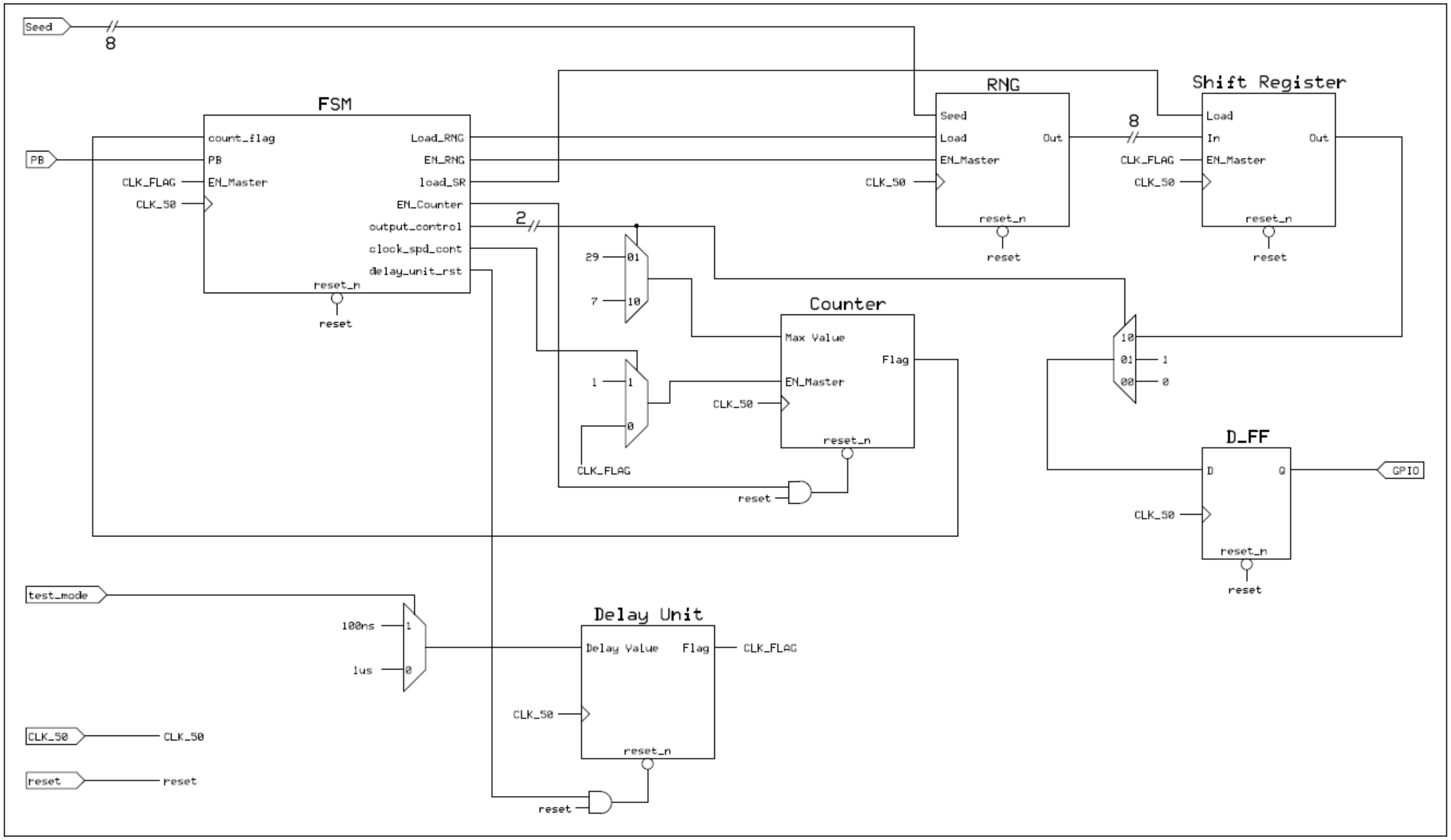
A state machine should be used to enable each step of the process.

Communication Protocol: The signal that will be used for transmission will remain in the high state until a transmission is started. At the start of a transmission, the line will drop low and stay low for 1 us. After that, a new data bit will be output every 1 us. After 8 bits are transmitted, the line returns to the high state.



*Note b_7-b_0 are the serial data bits being shifted out of the 8-bit shift register

System Block Diagram:



System Components

- **Finite State Machine –** this will take the system through the sequence of events. In each state its outputs control the other components. Design this first and start with a state transition diagram and output table.
- **RNG –** this is an 8-bit random number generator that was designed for a homework assignment. The only difference is this one has an enable that will be active for exactly 30 clock cycles.
- **Shift Register –** this is an 8-bit shift register with a parallel load. If it is not in the load state, it shifts. Since its output only goes out on the GPIO port when in the transmit state, you do not have to worry about it shifting data in other states.
- **Delay Unit –** this is the same circuit that has been used in previous labs. For test mode it generates a flag every 100 ns and for operational mode it generates a flag every 1 us.
- **Counter –** this counter is programmable. Its enable and max-value are determined by the state. In the “generate random number” state it counts 30 times at the system clock rate. In the transmit state, it counts 8 times at a 1 us rate
- **Mux and D_FF –** these components determine what the outgoing data is and synchronizes it. The outgoing data should be high in all but two states, start and transmit. In the start state the outgoing data is low for 1us. In the transmit state the outgoing data is the serial data from the shift register.

Procedure:

1. Write the code to implement the encryption key generation system.
2. Simulate your design using the test bench provided. YOU MUST SIMULATE FIRST. Obtain a signoff when your simulation works
3. Assign pins and download your design to the DE0-CV board.
4. Verify operation of the design using an oscilloscope. Obtain a signoff for a working board.

Signoffs and Grade

Name________________________________

Component	Signoff	Date	Time
Functional Simulation (50 pts)			
Working Board (50 pts)			

===

Component	Received	Possible
Signoffs		100
Total		100