Design a circuit that initializes a register to a random value

When btn[0] is pressed the T flip flop changes to state '1' so Rand4 cycles thru values; when btn[0] is pressed again the T flip flop changes back to state '0' so Rand4 halts displaying a random value.

Architecture rand4 of rand4 is

```
signal qs: STD_LOGIC_VECTOR(3 downto 0);
begn
  process(clk, clr)
  begin
    if clr = '1' then
      qs <= "0001";
    elsif clk'event and clk = '1' then
      qs(3) <= qs(0) xor qs(3);
      qs(2 downto 0) <= qs(3 downto 1);
    end if;
  end process;
  q <= qs;
end rand4;
```

The T-FlipFlop

```
entity T_flipflop is
  port(clk: in STD_LOGIC;
       clr: in STD_LOGIC;
       q: out STD_LOGIC);
end T_flipflop;
architecture T_flopflop of T-flipflop is
  signal qs: STD_LOGIC;
  begin
    process(clk, clr)
    begin
      if (clr = '1') then
        qs <= '0';
      elsif clk'event and clk = '1' then
        qs <= not qs;
      end if;
    end process;
    q <= qs;
  end T_flopflop;
```

Clock_Pulse

```
clkdiv
```

```
uint[3:0] → debounce4
```

Note: All buttons input now goes thru the debounce4 component
**Fibonacci Sequence:** \( F(n+2) \leq F(n+1) + F(n) \)

- **Fibonacci Registers**
  ```vhdl
  process(clr, clk) -- fn
  begin
    if (clr = '1') then
      fn <= "00000000000000";
      fn1 <= "00000000000001";
    elsif (clk'event and clk = '1') then
      fn <= D0;
      fn1 <= D1;
    end if;
  end process;
  ```

- **Details of “plus” process**
  ```vhdl
  process(fn1) -- plus
  begin
    if (fn1 < fib_max) then
      D1 <= fn + fn1;
      D0 <= fn1;
    else
      D0 <= "00000000000000";
      D1 <= "00000000000001";
    end if;
  end process;
  ```

  ```vhdl
  constant fib_max: STD_LOGIC_VECTOR(13 downto 0) := "10011100001111";
  ```

- **Exercises**
  1. Implement on the BASYS-2 board a 12 bit random number register circuit.
  2. Implement on the BASYS-2 board the Fibonacci Sequence as follows: Use `fib_top.vhd` (Listing 7.33) but substitute for `fib.vhd` a circuit design based on figure 7.54 (see this handout); do not use the code in listing 7.32.

```vhdl
constant fib_max: integer := 9999;
use IEEE::STD_LOGIC_UNSIGNED.all;
```