WANG, Tiffany 260684152
YE, Frank 260689448
GROUP 08 - MacroHard EE

G08 RANDU CIRCUIT

The goal of the lab is to describe a random number generator circuit using VHDL, to be familiar with the use library design entities and to use ROM modules to implement look-up tables. The random number generator circuit will be based off the **linear congruential generator algorithm**:

$$R = mod\ (a*SEED + b,\ c)$$

In the following report, we will present and justify our design through the display of our test results.

TABLE OF CONTENT

1. Circuit Description

Circuit Function:
As described in the introduction of the report, the goal of the lab is to design a basic random generator circuit based on the algorithm:

$$R = mod(a * SEED + b, c)$$

where a, b, c are constants, SEED is an arbitrary initial value and R is the random number output. To obtain different values of R, for the same a, b, c parameters one must replace SEED with R and and re-compute the output.
The parameters we implemented for this design is based on the IBM RANDU function:

- a = 65539
- b = 0
- c = 2^{31}

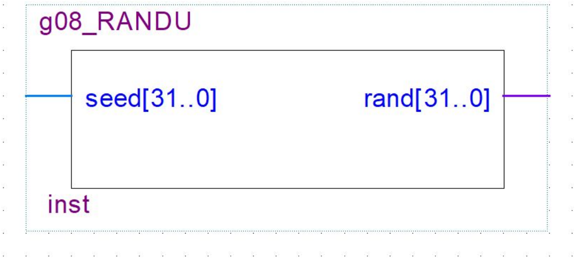
2. VHDL Description

The entity g08_RANDU takes in a 32-bit input SEED that will generate a 32 bit output SEED.
The representation of the decimal number 65539 in binary is 10000000000000011, which is $2^{16} + 2^1 + 2^0$. Therefore, the operation of *X*65539* can be simply computed through a series of shifts and additions. Namely,

$$X * 65539 = ShiftLeft\_16(X) + ShiftLeft\_1(X) + X$$

For our circuit design, since we set parameter b to 0, the last operation before the final output is to perform a A mod 2^{31} calculation. This works out to simply retaining the N - 31 MSB, N being the number of input bits in the operation. In this case, A = X * 65539 mentioned previously. Thus, we end up with a 32 bit output.

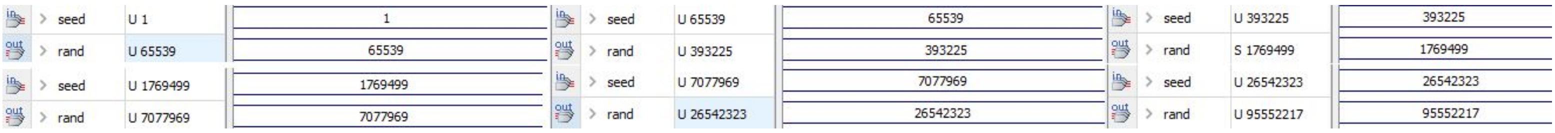
INPUT: seed[31..0]
OUTPUT: rand[31..0] *rand = mod(65539 * seed , 2^{31})*



RANDU circuit schematic

3. Testing

Since testing the 2^{32} possible inputs is impractical, the circuit was tested by choosing an initial seed value of 00000000000000000000000000000001, running the simulation, then using the first output as the seed value for the second simulation. The simulation was run 5 times, and the result was verified for every run.



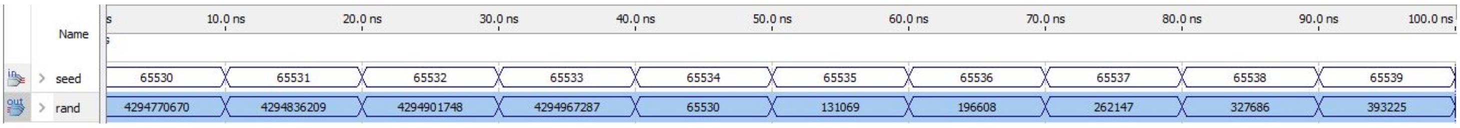
```
seed  U 1         1        | seed  U 65539     65539     | seed  U 393225    393225
rand  U 65539     65539    | rand  U 393225    393225    | rand  S 1769499   1769499
seed  U 1769499   1769499  | seed  U 7077969   7077969   | seed  U 26542323  26542323
rand  U 7077969   7077969  | rand  U 26542323  26542323  | rand  U 95552217  95552217
```

Moreover, due to its poor design, the following equation should hold if the RANDU circuit was designed correctly.

$$mod(R_n - 6\ R_{n-1} + 9R_{n-2}\ ,\ 2^{31}) = 0$$

We have (95552217 - 6 * 26542323 + 9 * 7077969) % 2^{31} = 0 % 2^{31} = 0

As well, a wrap around of the modulo is expected occur when the seed exceeds 2^{16}, as shown in our test results.



```
Name  s   10.0 ns     20.0 ns     30.0 ns     40.0 ns     50.0 ns  60.0 ns  70.0 ns  80.0 ns  90.0 ns  100.0 ns
seed  65530       65531       65532       65533       65534       65535    65536    65537    65538    65539
rand  4294770670  4294836209  4294901748  4294967287  65530       131069   196608   262147   327686   393225
```

4. Advantages and Limitations

The obvious advantage of this circuit is its simple implementation. The modulo and multiplication circuits required a lot less circuitry due to the fact that for those operations, we could merely match the correct input bits, e.g. when multiplying by a power of 2. The lack of physical of shift registers prevents problems such as gate delays and clock implementations. Hence, we were able to write VHDL code to perform all operations using the lpm add modules.

However, due to its simplicity, the numbers generated by this circuit are easily predictable, as indicated by the above equation. Hence, this design could not be used for serious applications.

5. References

[1] Clark J., *Lab #2 - Combinational Circuit Design with VHDL*, ECSE 323 Digital System Design, Fall 2017

[2] Altera. LPM Quick Reference Guide, December 1996

[3] Altera. lpm_add_sub Megafunction User Guide, March 2007

[4] Altera. lmp_rom Megafunction User Guide, March 2005