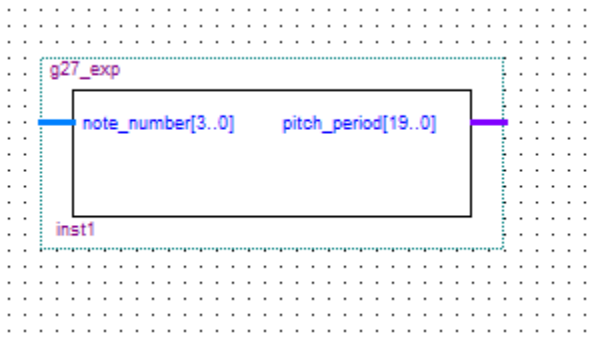
ECSE 323 Digital System Design

Lab #2 – g27_exp (exponentiator circuit)

Group 27

Lulan Shen (260449509)
Loren Lugosch (260404057)



Design of an exponentiator circuit using VHDL.

Since the frequencies of the notes of the 12-tone musical scale increase exponentially, their corresponding periods decrease exponentially. The first component we needed to design for our music box was a circuit which could produce these periods given a note encoded in 4 bits.

We wrote our exponentiator VHDL code using a single select statement representing a 16-to-1 multiplexor, where 12 of the 16 possible inputs represent notes, and 4 inputs are invalid. We calculated the appropriate outputs based on the equation in the lab description using the following C program:

```
#include<math.h>
#include<stdio.h>
#include<stdlib.h>

int main()
{
	/* given binary N from 0000 to 1111,
	 * return 764451*2^(-N/12) in binary */

	float n;
	float period;
	for(n = 0; n < 12; n++){
		float two = 2.0;
		period = 764451.0 * pow(two, (-n/12.0));
		//period = ceil(period);
		printf("period = %d\n", (int)period);
	}
}
```

We set the outputs of the select statement to the results of our program (converted to binary). The simulation results for the circuit are below.

Functional Simulation Results:

Note_number starts at 0000, ends at 1111, and is incremented by 1. Note_number is counted every 10 ns and the end time is 160ns, testing all 16 possible input patterns. The testing results are shown below.