New Paltz
STATE UNIVERSITY OF NEW YORK

Department of Electrical and Computer Engineering

Design and Verification of System on Chip
EGE593-01

Direct Digital Synthesis

Group Members	Department
Kyle Harkin	EE

Instructor: R. Copeman

Fall 2021

Table of Contents

Objective

The objective of this final project is to design and implement a full direct digital synthesis module which will then interface with an external digital to analog converter.

Introduction

What is Direct Digital Synthesis (DDS)?

Throughout the world, there is need for technologies which transcend both the digital and analog realm. One such technology is that of direct digital synthesis. Direct digital synthesis, or DDS, is the technique of creating analog oscillating signals such as a sine or triangle wave from a fixed frequency reference clock signal. DDS is used in numerous applications, including but not limited to communication systems, audio mixers, function generators, and sound synthesizers.

DDS: Theory

Now that one is familiar with what DDS is, how is it implemented? DDS achieves its goal of modulated outputs varying in relation to the input by means of the DDS equation, as shown in Figure 1.

$$f_{OUT} = \frac{f_C \times M}{2^N}$$

Figure 1. The DDS Equation

As shown in Figure 1, our output frequency is determined from three factors. Those factors are the reference clock, f_C, the tuning word, M, and the bit width of our system, 2^N, where N is the number of bits. Of the three factors, the single one which is variable is the tuning word. Figure 2 shows how one determines the appropriate tuning word for the desired output frequency.

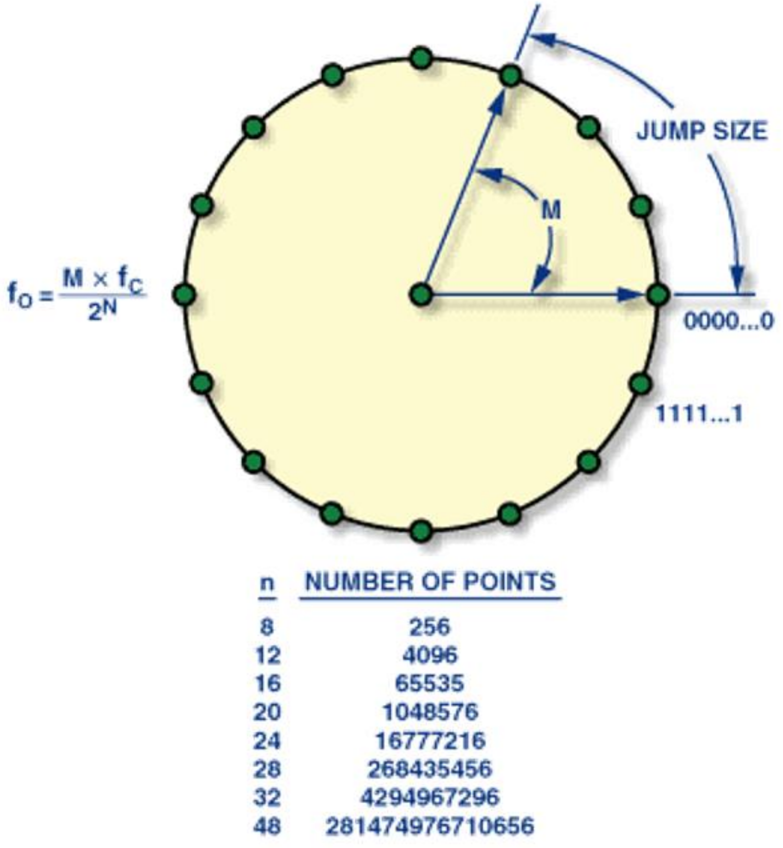
$$\frac{f_{OUT} \times 2^N}{f_C} = M$$

Figure 2. Tuning Word Determination

Therefore, as demonstrated in Figure 2, one can easily determine what tuning word is needed to create the desired output frequency.

DDS: Utilizing Theory

To implement a full DDS system, one must first understand how the theory translates to reality. A graphical representation of how DDS operates is shown in Figure 3.



n	NUMBER OF POINTS
8	256
12	4096
16	65535
20	1048576
24	16777216
28	268435456
32	4294967296
48	281474976710656

Figure 3. DDS Wheel

As demonstrated in Figure 3, DDS is essentially a counter which counts up each successive clock cycle by an amount determined by the tuning word. The resolution, or how many steps our counter takes, is determined by the bit width of the system. Therefore, as the width of the system increases, so too does the resolution and therefore the “detail” of the output. So how exactly are we implementing this DDS system?

Project Technologies

To implement the DDS we will be using the Terasic Technologies DE-10 Nano FPGA. The DE-10 Nano is an Intel Cyclone V device which utilizes the Verilog HDL programming language. Therefore, the Intel Quartus II development suite, in addition to the SignalTap and ModelSim software, will be used to program and analyze our DDS design.

Design and Implementation

Project Overview

A high-level, conceptual, diagram of the required DDS is seen in Figure 4.

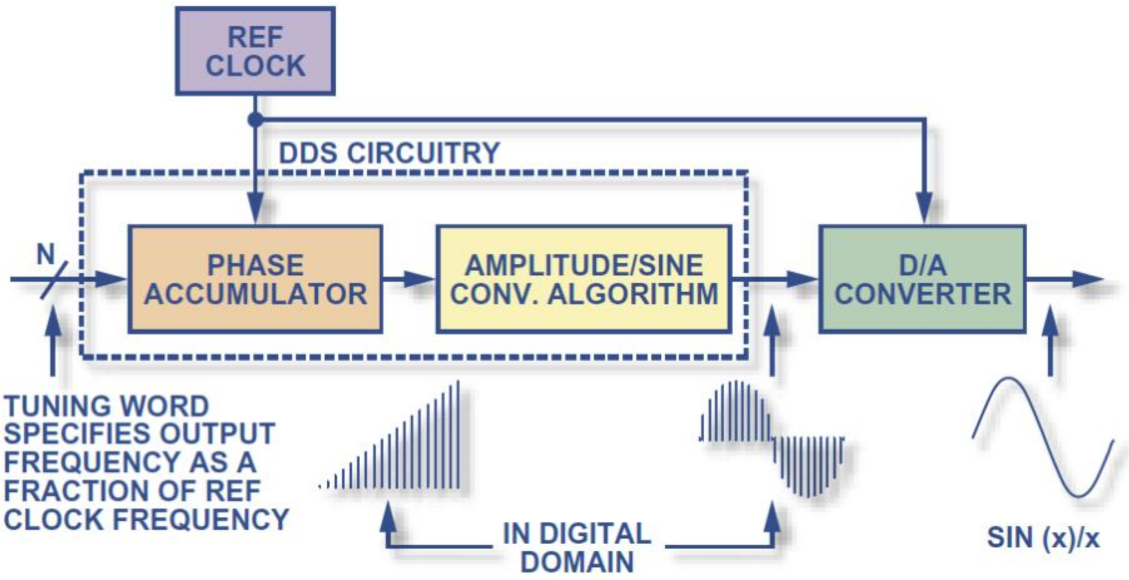

REF CLOCK
DDS CIRCUITRY
N
PHASE ACCUMULATOR
AMPLITUDE/SINE CONV. ALGORITHM
D/A CONVERTER
TUNING WORD SPECIFIES OUTPUT FREQUENCY AS A FRACTION OF REF CLOCK FREQUENCY
IN DIGITAL DOMAIN
SIN (x)/x

Figure 4. DDS Project: Conceptual

As shown in Figure 4, our DDS project is to have the capability of taking a fixed reference clock and outputting the desired digitally constructed waveforms of a chosen frequency to be then accessed by an external digital to analog converter (DAC). The dotted line seen in Figure 4 surrounds the modules which are to be custom made for the DDS.

Schematics in Quartus II

The schematic of our completed DDS is shown in Figures 5-7 as it appears in the Quartus II software.

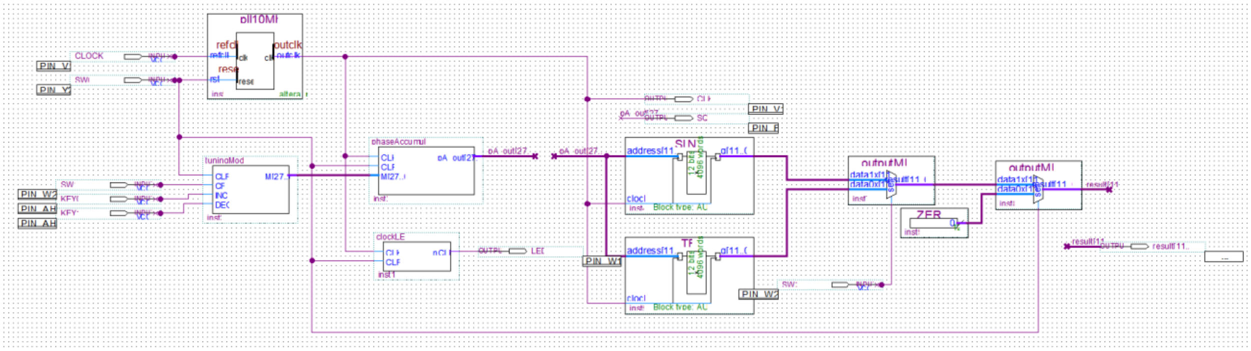


Figure 5. Complete DDS Schematic

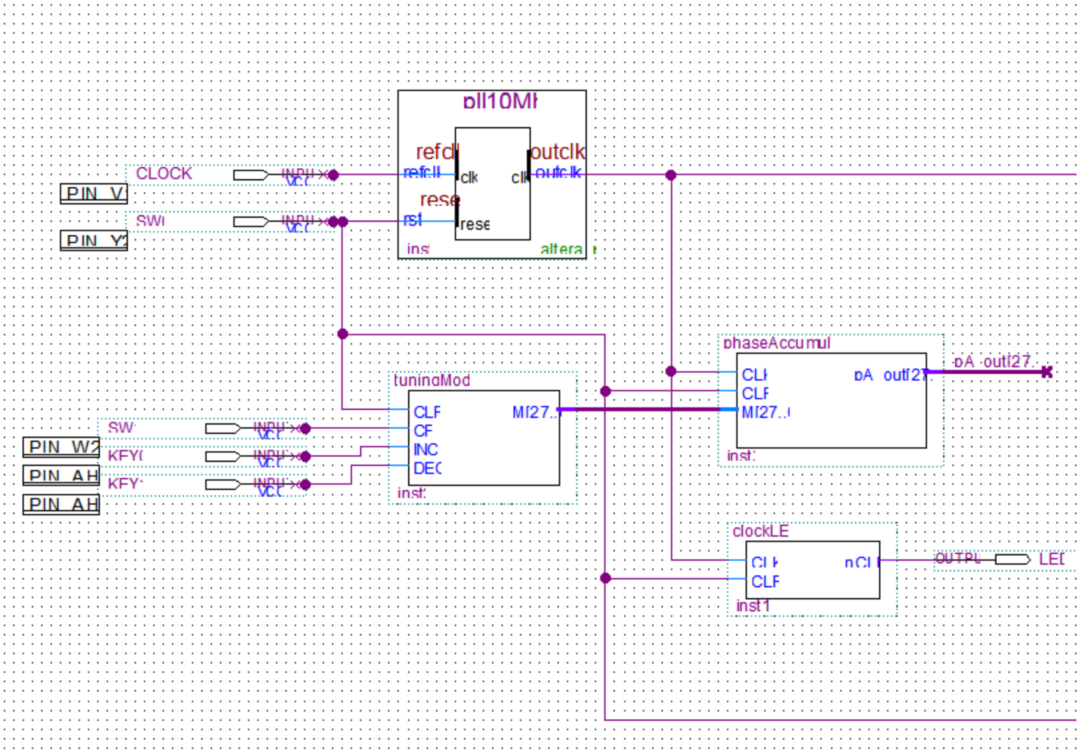


Figure 6. DDS Close-up 1

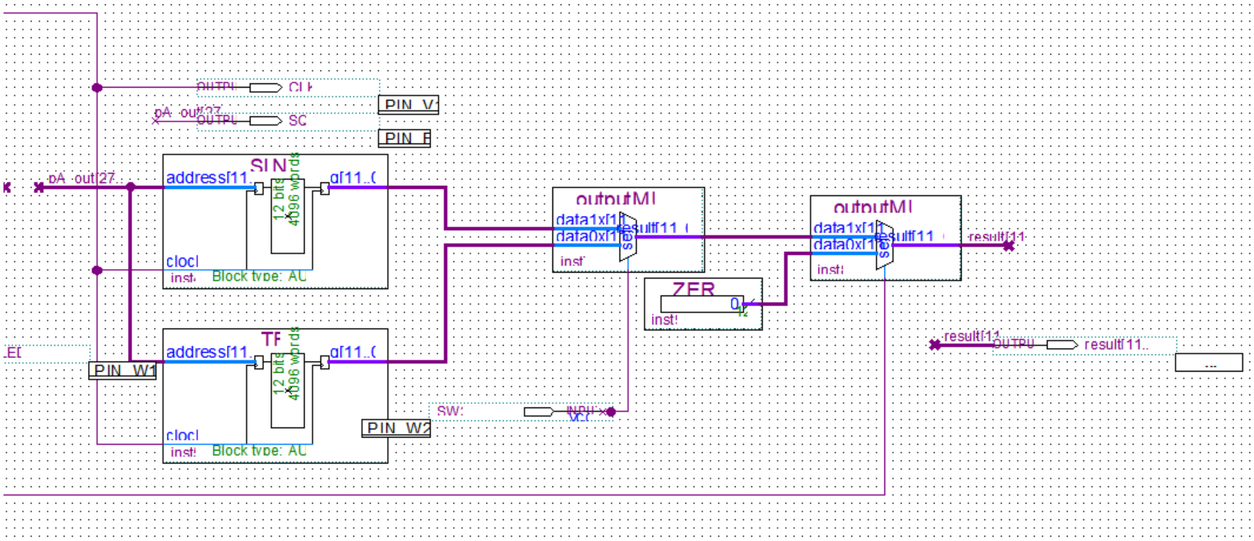


Figure 7. DDS Close-up 2

As shown in Figure 5-7, our DDS system has six inputs and fifteen outputs. We will describe each input, output, and function of the DDS in the following sections.

IP vs Custom Modules

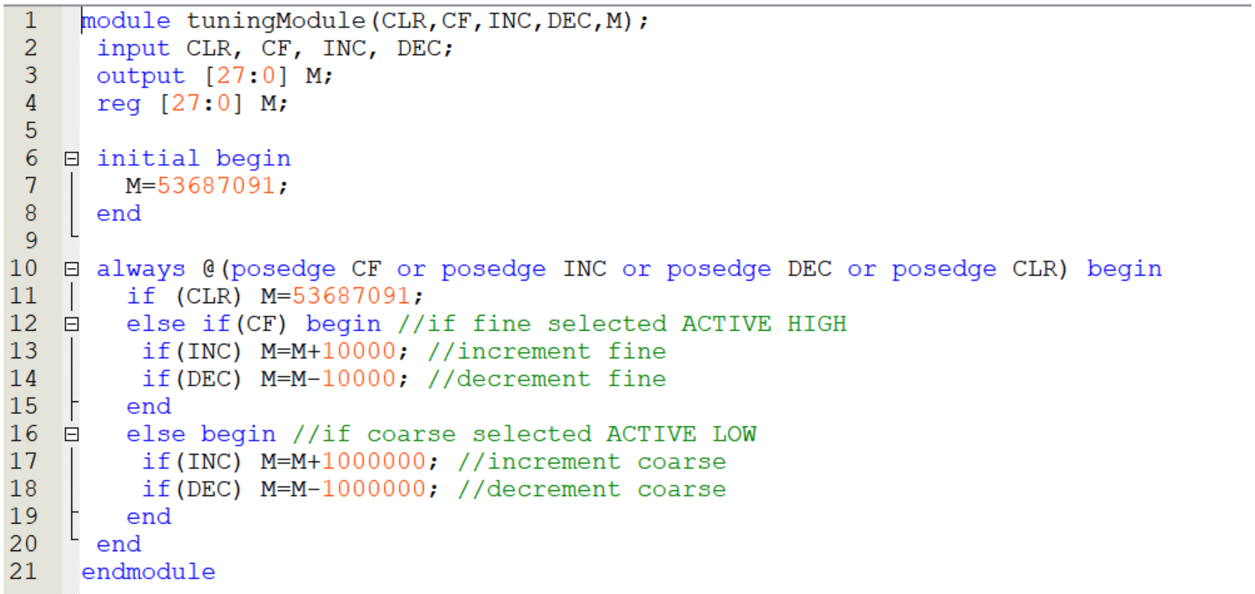
Before describing the DDS design in detail, it is important to note the distinction between module types used. An IP module is one which is pre-made and supplied as a part of the Quartus II Software, these are setup and then used rather than created. Conversely, custom made modules are ones that were designed and written by the user, not supplied. Now, understanding the two types of modules, we can describe the designed DDS.

DDS: Reference Clock

The first module seen in our design is the that of the reference clock, labeled pll10MHz, and clearly seen in Figure 6. This module is an IP phase locked loop (PLL) which takes as its inputs a reference clock and reset, and outputs a new clock. For our design, the pll10MHz module is supplied with the internal 50MHz clock of the DE-10 Nano and converts it to a 10MHz clock to be used throughout our DDS. The reset functionality of the module allows our new clock to be set back to zero, thus restarting our DDS' clock and is assigned to the physical SW0 on the DE-10 Nano.

DDS: Tuning Module

The tuning module of our DDS is clearly seen in Figure 6 and labeled as tuning Module. The purpose of this module is to adjust the value of our DDS tuning word and is a custom module whose implementation can be seen in Figure 8.



```
module tuningModule(CLR,CF,INC,DEC,M);
 input CLR, CF, INC, DEC;
 output [27:0] M;
 reg [27:0] M;

 initial begin
   M=53687091;
 end

 always @(posedge CF or posedge INC or posedge DEC or posedge CLR) begin
   if (CLR) M=53687091;
   else if(CF) begin //if fine selected ACTIVE HIGH
    if(INC) M=M+10000; //increment fine
    if(DEC) M=M-10000; //decrement fine
   end
   else begin //if coarse selected ACTIVE LOW
    if(INC) M=M+1000000; //increment coarse
    if(DEC) M=M-1000000; //decrement coarse
   end
 end
endmodule
```

Figure 8. DDS Tuning Module

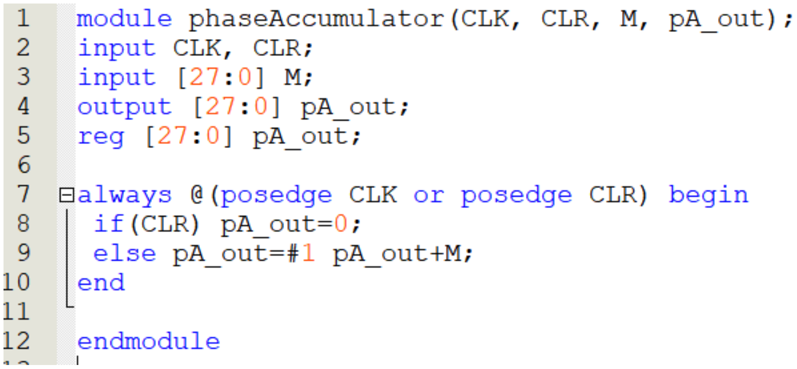
As seen in Figures 6 and 8, this module contains four inputs and one output. The inputs to the module are the clear (CLR), course/fine (CF), increment (INC) and decrement (DEC) pins while the output is the tuning word (M). The CLR function is to set the value of our tuning word back to its original value of 53687091. The value of M was determined via the formula shown in Figure 2 to produce a total DDS output of 0-2MHz given a reference clock of 10MHz and a design bit width of 28 bits. The final tuning word is then output from the module for use in the DDS. The CF function is to provide the user the ability to select between course and fine adjustment of the DDS output frequency. Lastly, the INC and DEC inputs are to allow the user to either increase or decrease the DDS output frequency. A complete listing of the physical components on the DE-10 Nano assigned to these functions are shown in Table 1.

Function	Component
CLR	SW0
CF	SW1
INC	KEY0
DEC	KEY1

Table 1. Functions vs. Components

DDS: Phase Accumulator

The custom phase accumulator module, where the functionality of the DDS Wheel from Figure 3 is implemented, is shown in Figures 6 and 9 labeled as phase Accumulator.



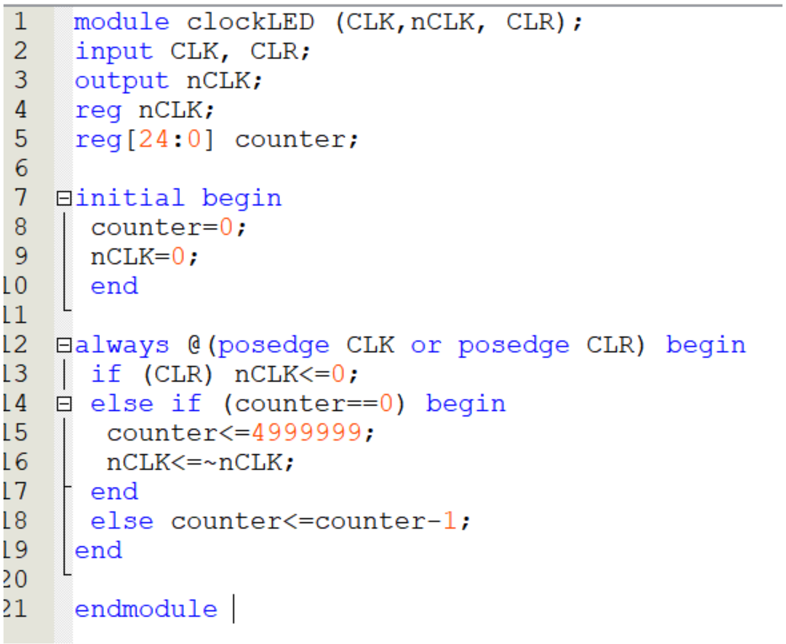
```
module phaseAccumulator(CLK, CLR, M, pA_out);
input CLK, CLR;
input [27:0] M;
output [27:0] pA_out;
reg [27:0] pA_out;

always @(posedge CLK or posedge CLR) begin
 if(CLR) pA_out=0;
 else pA_out=#1 pA_out+M;
end

endmodule
```

Figure 9. Phase Accumulator

As seen in Figures 6 and 9, the phase accumulator module has three inputs and output. The inputs are the clock (CLK), clear (CLR), tuning word (M), and the output is the resulting phase accumulator number (pA_out). The CLK input is the clock generated from the previously discussed reference clock module. The CLR input is to reset the output of the phase accumulator to zero and is tied to the DE-10 Nano component SW0. As discussed in the tuning module, M is the final tuning word after adjustment. The single output of this module, which is further used within the DDS, pA_out, is determined by adding the previous value of the module to the incoming M value. Because this operation takes place every clock cycle and is the same 28-bit width as the tuning word, it is equivalent to the equation shown in Figure 1.

DDS: LED Confirmation

A simple yet important custom module in our DDS system is the clock divider seen in Figures 6 and 10 labeled as clockLED.



```
module clockLED (CLK,nCLK, CLR);
input CLK, CLR;
output nCLK;
reg nCLK;
reg[24:0] counter;

initial begin
 counter=0;
 nCLK=0;
 end

always @(posedge CLK or posedge CLR) begin
 if (CLR) nCLK<=0;
 else if (counter==0) begin
  counter<=4999999;
  nCLK<=~nCLK;
 end
 else counter<=counter-1;
end

endmodule
```

Figure 10. Clock LED

The purpose of the clock divider module is to provide a 1Hz signal to an LED component on the DE-10 Nano to provide visual confirmation that the DDS system is running and is not in the clear state. The inputs to this module are the clock (CLK) and clear (CLR) and the output is the new 1Hz signal (nCLK). This module works by simply counting down to half of the input clock frequency before switching the logic value of the output. Therefore, we end up with a 1Hz clock derived from the input 10MHz. The output from this module (nCLK) is then sent to the LED0 component on the DE-10 Nano.

DDS: Waveform Outputs

As explained in previous sections, the entire purpose of the DDS system is to output varying waveform types from a reference clock. The design shown in Figure 7 allows for constant output of a square wave as well as the choice between a sine or triangle waveform.

Waveform Outputs: Square

The square wave output of our DDS design is accomplished by simply taking the most significant output bit of the 28-bit output of the phase accumulator module. This output is shown in Figure 7 and 11 labeled as SQ. The DE-10 Nano component assigned to this output can be found in the Appendix.

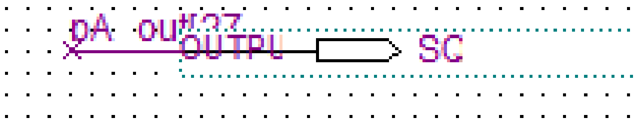


Figure 11. Square Wave Output

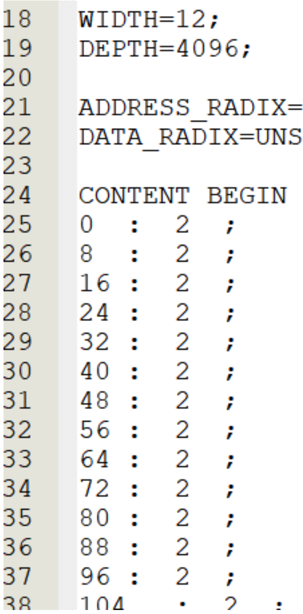
Waveform Outputs: Sine

The sine wave output of our DDS system is accomplished via use of the 1-Port ROM IP module and is shown in Figure 7 labeled as SN. This module takes a clock signal as its input as well as the first 12-bits of the phase accumulator number. Then, this module uses those first 12-bits as an address to look up the stored data value. Therefore, this module makes use of a lookup table correlating values of 0-4096 to an output value. Values for the lookup table were determined via the equation in Figure 12 and rounded to the nearest whole integer, since decimal representations are not possible in Verilog HDL.

$$value = 1.65\sin\left(\frac{\pi x}{2048}\right) + 1.65$$

Figure 12. Sine Lookup Table Equation [1]

Since our output waveform must be between 0 and 3.3 volt, our equation corresponding swings from a midpoint of 2048 input and 1.65 volt with that also as its maximum additional amplitude. A sample of the sine lookup table is shown in Figure 13.



```
WIDTH=12;
DEPTH=4096;

ADDRESS_RADIX=UNS;
DATA_RADIX=UNS;

CONTENT BEGIN
0  :  2  ;
8  :  2  ;
16 :  2  ;
24 :  2  ;
32 :  2  ;
40 :  2  ;
48 :  2  ;
56 :  2  ;
64 :  2  ;
72 :  2  ;
80 :  2  ;
88 :  2  ;
96 :  2  ;
104   :  2  ;
```

Figure 13. Sine Lookup Table Sample

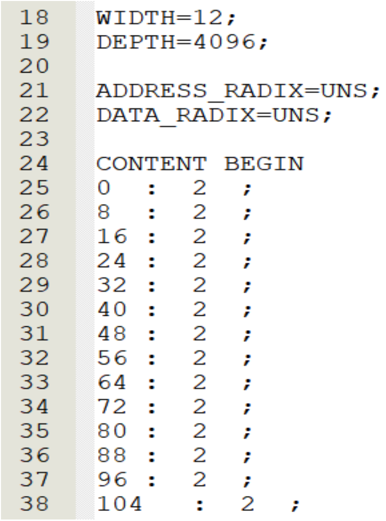
Waveform Outputs: Triangle

Similar to the sine wave output, the triangle wave output uses another 1-Port ROM IP module and separate lookup table, shown in Figure 7 labeled as TR. The equation for determinant of the corresponding lookup values is shown in Figure 14.

$$value = \frac{3.3}{\pi}\arcsin\left(\sin\left(\frac{2\pi x}{2048}\right)\right) + 1.65$$

Figure 14. Triangle Lookup Table Equation [2]

Akin to Figure 12, the triangle wave equation takes our total amplitude being 3.3v and has our wave swing around a midpoint of 1.65v. A sample of the triangle lookup table can be seen in Figure 15.



```
WIDTH=12;
DEPTH=4096;

ADDRESS_RADIX=UNS;
DATA_RADIX=UNS;

CONTENT BEGIN
0   :   2   ;
8   :   2   ;
16  :   2   ;
24  :   2   ;
32  :   2   ;
40  :   2   ;
48  :   2   ;
56  :   2   ;
64  :   2   ;
72  :   2   ;
80  :   2   ;
88  :   2   ;
96  :   2   ;
104     :   2   ;
```

Figure 15. Triangle Lookup Table Sample

Waveform Outputs: Choice and Final Output

In order to be able to chose which output we want to see, our DDS system used two LPM-MUX IP modules, seen in Figure 7 as outputMUX 1 and 2 respectively. The first mux takes the 12-bit output of our two ROM modules and allows selection via the DE-10 Nano component SW2. Then our second mux takes the chosen output and a zero output and allows choice between them via the aforementioned DE-10 Nano clear input component SW0. This second mux ensures that if the clear switch has been enabled throughout the DDS system, then a zero output is guaranteed at the output as a fail-safe measure. The final 12-bit output of our DDS system is then sent to corresponding pins on the DE-10 Nano, found in the Appendix, and shown in Figure 16.

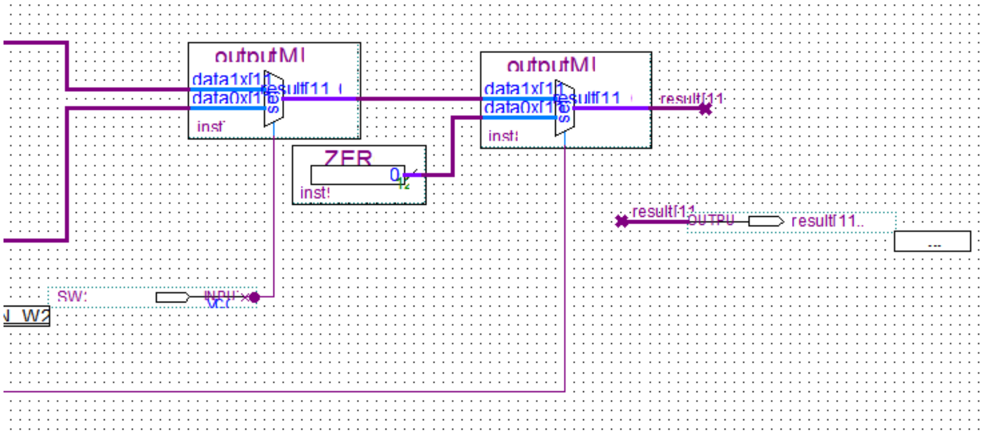
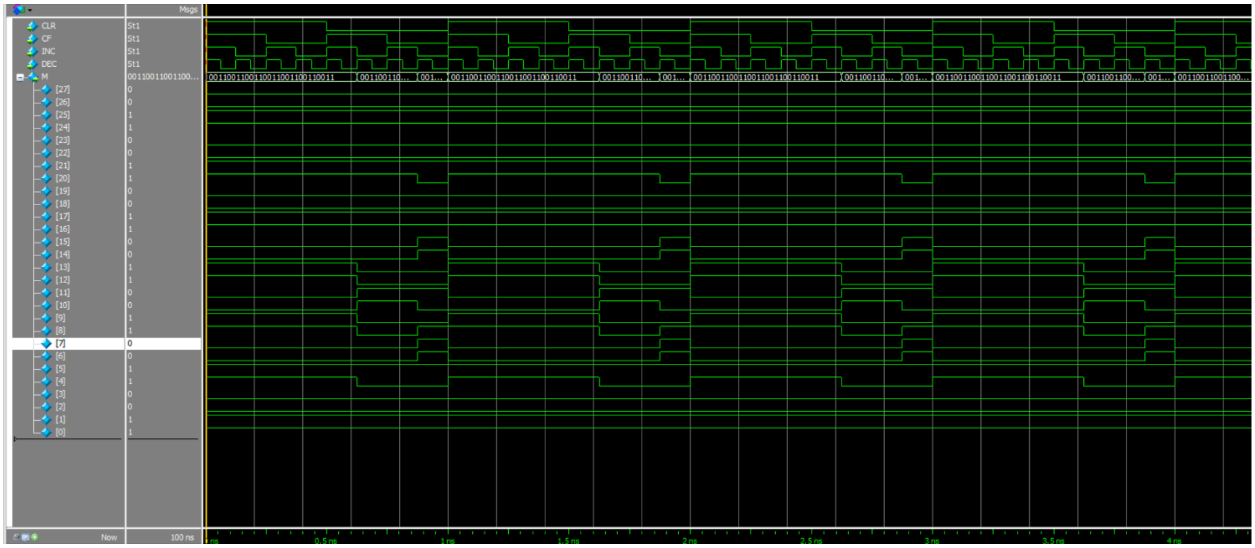


Figure 16. Final Output MUXs

Design Verification

The functionality of our DDS system was confirmed via successful compilation within Quartus II, visual confirmation of the 1Hz blinking LED, and ModelSim simulations. The simulation of the tuning module can be seen in Figure 17.



	Msgs
CLR	St1
CF	St1
INC	St1
DEC	St1
M	00110011001100...
[27]	0
[26]	0
[25]	1
[24]	1
[23]	0
[22]	0
[21]	1
[20]	1
[19]	0
[18]	0
[17]	1
[16]	1
[15]	0
[14]	0
[13]	1
[12]	1
[11]	0
[10]	0
[9]	1
[8]	1
[7]	0
[6]	0
[5]	1
[4]	1
[3]	0
[2]	0
[1]	1
[0]	1
Now	100 ns

Figure 17. Tuning Module Simulation.

One can clearly see in Figure 17 that as INC or DEC are active, the corresponding arithmetic operations are conducted on M. Furthermore, it is clearly demonstrated how when CLR is active, M is reset to its initial value. The simulation of the phase accumulator module is shown in Figure 18.

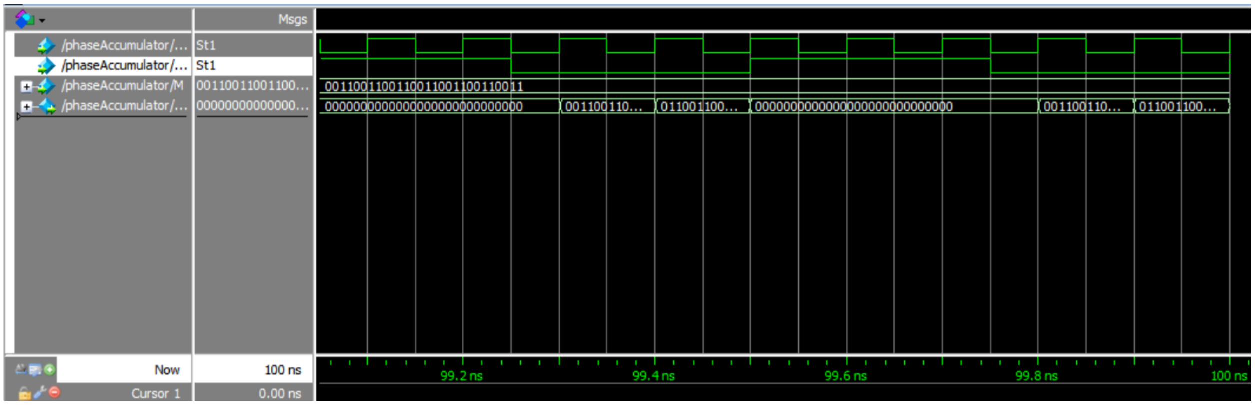

Msgs
/phaseAccumulator/... St1
/phaseAccumulator/... St1
/phaseAccumulator/M 00110011001100...
/phaseAccumulator/... 00000000000000...
Now 100 ns
Cursor 1 0.00 ns
99.2 ns
99.4 ns
99.6 ns
99.8 ns
100 ns

Figure 18. Phase Accumulator Simulation

As shown in Figure 18, the phase accumulator adds to itself as M is held constant and the CLK is oscillating. Furthermore, when CLR is active the phase accumulator is rest back to zero. The simulation of the output LED is shown in Figure 19.

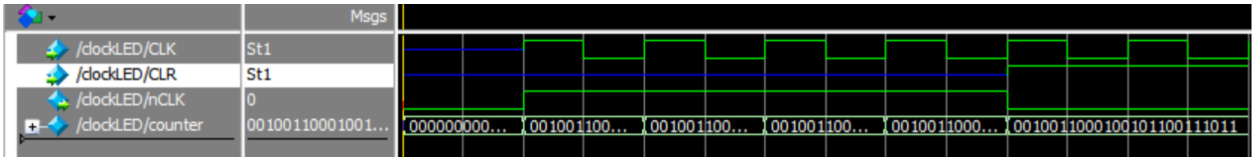


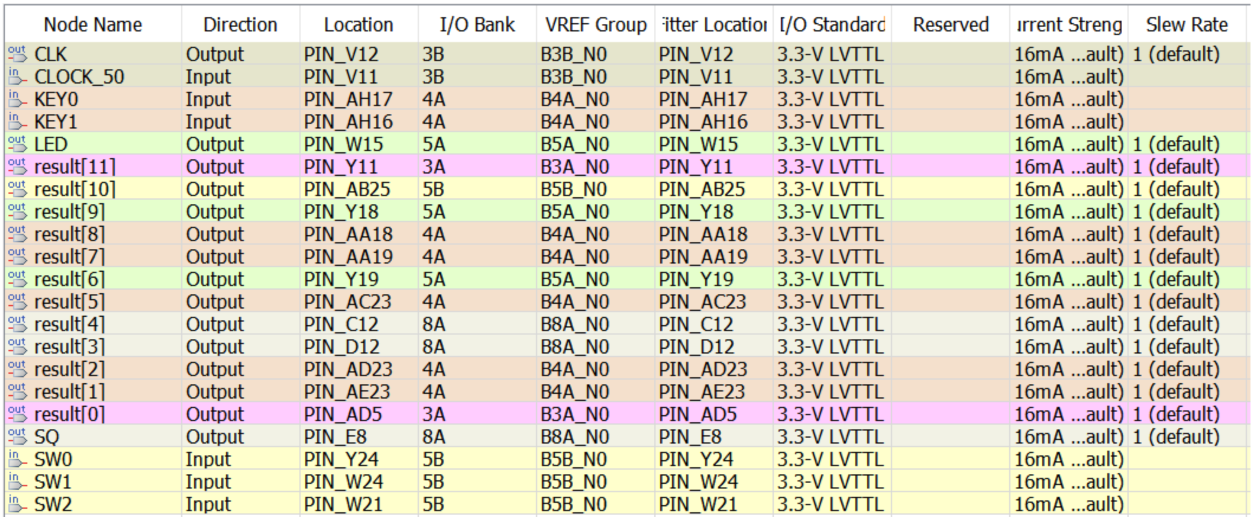
Figure 19. 1Hz LED Simulation

As demonstrated in Figure 19, the LED counter counts and activates nCLK up as CLK is oscillating and the CLR input is not active. However, once the CLR is active the nCLK signal is reset.

Conclusions

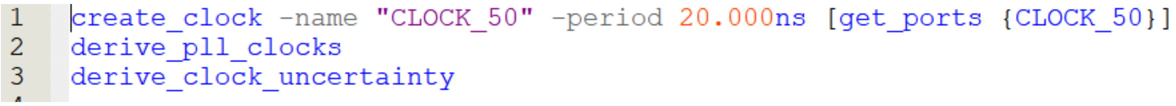
Throughout development of the described DDS system, there were many challenges. The first challenge was learning how to use the various IP modules from professional technical literature provided through the course. Another challenge was how to set up the look up tables and which equations to use. The equations were found through researching the mathematics behind the waveforms themselves and then transforming and applying them to the DDS system. Despite these challenges, a successful implementation of a 0-2MHz DDS system was created and verified using a combination if custom and supplied IP modules within the Quartus II development suite.

Appendix



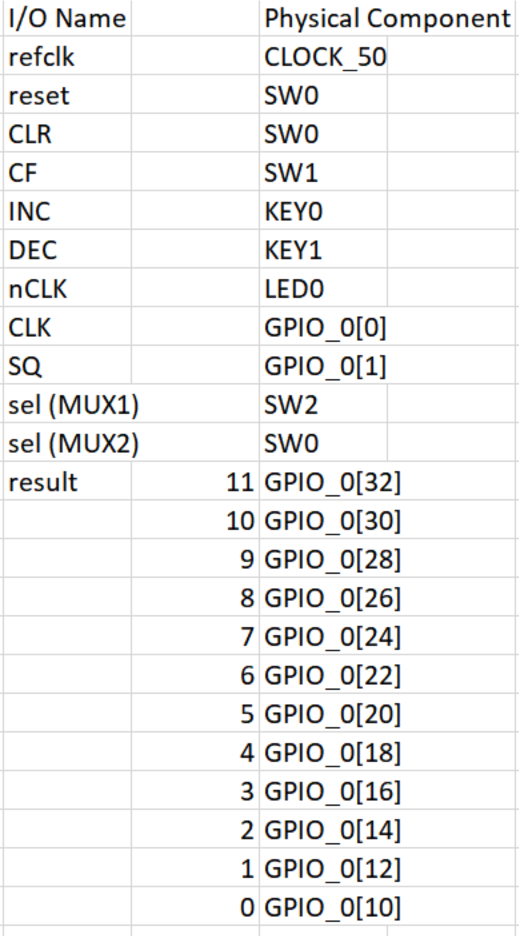
Node Name	Direction	Location	I/O Bank	VREF Group	itter Locatio	I/O Standard	Reserved	rrent Streng	Slew Rate
CLK	Output	PIN_V12	3B	B3B_N0	PIN_V12	3.3-V LVTTL		16mA ...ault)	1 (default)
CLOCK_50	Input	PIN_V11	3B	B3B_N0	PIN_V11	3.3-V LVTTL		16mA ...ault)	
KEY0	Input	PIN_AH17	4A	B4A_N0	PIN_AH17	3.3-V LVTTL		16mA ...ault)	
KEY1	Input	PIN_AH16	4A	B4A_N0	PIN_AH16	3.3-V LVTTL		16mA ...ault)	
LED	Output	PIN_W15	5A	B5A_N0	PIN_W15	3.3-V LVTTL		16mA ...ault)	1 (default)
result[11]	Output	PIN_Y11	3A	B3A_N0	PIN_Y11	3.3-V LVTTL		16mA ...ault)	1 (default)
result[10]	Output	PIN_AB25	5B	B5B_N0	PIN_AB25	3.3-V LVTTL		16mA ...ault)	1 (default)
result[9]	Output	PIN_Y18	5A	B5A_N0	PIN_Y18	3.3-V LVTTL		16mA ...ault)	1 (default)
result[8]	Output	PIN_AA18	4A	B4A_N0	PIN_AA18	3.3-V LVTTL		16mA ...ault)	1 (default)
result[7]	Output	PIN_AA19	4A	B4A_N0	PIN_AA19	3.3-V LVTTL		16mA ...ault)	1 (default)
result[6]	Output	PIN_Y19	5A	B5A_N0	PIN_Y19	3.3-V LVTTL		16mA ...ault)	1 (default)
result[5]	Output	PIN_AC23	4A	B4A_N0	PIN_AC23	3.3-V LVTTL		16mA ...ault)	1 (default)
result[4]	Output	PIN_C12	8A	B8A_N0	PIN_C12	3.3-V LVTTL		16mA ...ault)	1 (default)
result[3]	Output	PIN_D12	8A	B8A_N0	PIN_D12	3.3-V LVTTL		16mA ...ault)	1 (default)
result[2]	Output	PIN_AD23	4A	B4A_N0	PIN_AD23	3.3-V LVTTL		16mA ...ault)	1 (default)
result[1]	Output	PIN_AE23	4A	B4A_N0	PIN_AE23	3.3-V LVTTL		16mA ...ault)	1 (default)
result[0]	Output	PIN_AD5	3A	B3A_N0	PIN_AD5	3.3-V LVTTL		16mA ...ault)	1 (default)
SQ	Output	PIN_E8	8A	B8A_N0	PIN_E8	3.3-V LVTTL		16mA ...ault)	1 (default)
SW0	Input	PIN_Y24	5B	B5B_N0	PIN_Y24	3.3-V LVTTL		16mA ...ault)	
SW1	Input	PIN_W24	5B	B5B_N0	PIN_W24	3.3-V LVTTL		16mA ...ault)	
SW2	Input	PIN_W21	5B	B5B_N0	PIN_W21	3.3-V LVTTL		16mA ...ault)	

Figure A1. DDS System Pin Assignments



```
create_clock -name "CLOCK_50" -period 20.000ns [get_ports {CLOCK_50}]
derive_pll_clocks
derive_clock_uncertainty
```

Figure A2. DDS System Clock Generation



I/O Name		Physical Component
refclk		CLOCK_50
reset		SW0
CLR		SW0
CF		SW1
INC		KEY0
DEC		KEY1
nCLK		LED0
CLK		GPIO_0[0]
SQ		GPIO_0[1]
sel (MUX1)		SW2
sel (MUX2)		SW0
result	11	GPIO_0[32]
	10	GPIO_0[30]
	9	GPIO_0[28]
	8	GPIO_0[26]
	7	GPIO_0[24]
	6	GPIO_0[22]
	5	GPIO_0[20]
	4	GPIO_0[18]
	3	GPIO_0[16]
	2	GPIO_0[14]
	1	GPIO_0[12]
	0	GPIO_0[10]

Figure A3. DDS Schematic I/O Pin to DE-10 Physical Components

[1] “Sine wave,” *Wikipedia*, 03-Dec-2021. [Online]. Available: https://en.wikipedia.org/wiki/Sine_wave. [Accessed: 13-Dec-2021].

[2] “Triangle Wave,” *Wikipedia*, 10-Dec-2021. [Online]. Available: https://en.wikipedia.org/wiki/Triangle_wave. [Accessed: 13-Dec-2021].