Name: Huzaifah Tariq Ahmed HUID: ha07151

## Habib University
## School of Science & Engineering

| | |
|---|---|
| Course: | Digital System Design |
| Semester | Fall 2024 |
| Quiz # | 1 |
| Date, Time : | Feb. 19, 2024, 1 hour |
| Instructor : | Syed Arsalan Jawed |
| Total Marks : | 10 |

**Instructions for Students**

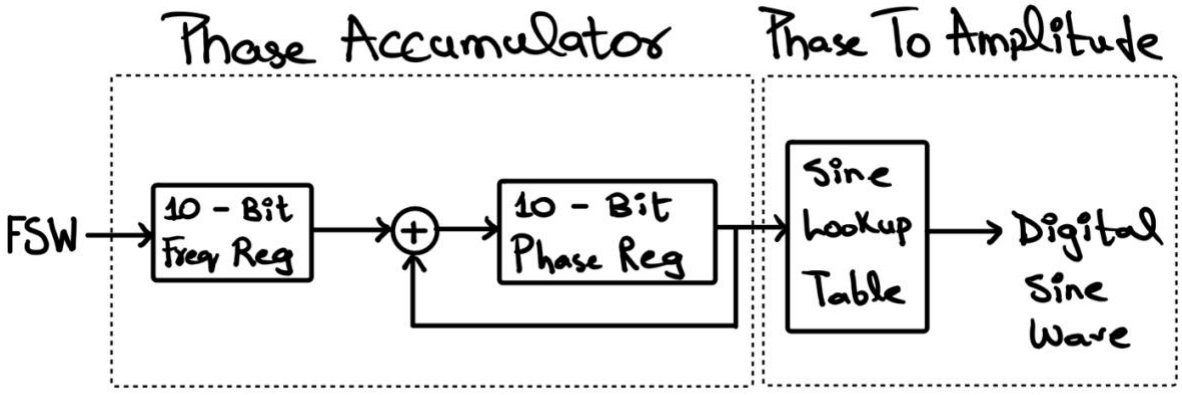
1. Provide your answer within the empty spaces given after each question.
2. Keep your answers concise and to the point. Provide neat and tidy calculations and plots.
3. Make reasonable assumptions related to concepts and associated parameters motivated by your text and reference book.
4. Support your answer with equations and simulation where asked and needed.

**Question # 1 : [CLO-1]** , 2, 4]

We need to generate discrete-time Sine waves of different frequencies with a 10-bit digital representation, the master clock of the system is 1MHz and the sine-wave frequency configurable between 1-10kHz and 125k-250kHz. You may use 8-10 bit registers in your design. You may use Internet or your text book to define an architecture and highlight the functional components. Then write synthesizable Verilog code for it. Simulate your design in Xilinx Vivado and demonstrate the functionality. Draw your architecture below, attach your Verilog code and the generated gatelevel netlist schematic and motivate that it is a true representation of your Verilog code.

[10]

# Proposed Architecture:



## Phase Accumulator:

Code:

```
`timescale 1ns / 1ps

module phase_counter(
        input clk,
        input reset,
        input [9:0] fsw,
        output reg [9:0] counter
        );

        always @ (posedge clk or posedge reset)
        begin
          if(reset)
            counter <= 0;
          else if (counter < 1024)
            counter <= counter + fsw;
          else
            counter <= 0;
        end
endmodule
```

## Phase To Amplitude:

Code:

```
`timescale 1ns/1ps


module phase_to_amplitude(
        input [9:0] counter,
        input reset,
        output reg [9:0] dds_sin);

        always @ (*)
        begin
          if (reset)
            dds_sin = 50;
          else if (counter >= 0 && counter < 11)
            dds_sin <= 50;
```

```
15            else if (counter >= 11 && counter < 21)
16              dds_sin <= 53;
17            else if (counter >= 21 && counter < 31)
18              dds_sin <= 56;
19            else if (counter >= 31 && counter < 41)
20              dds_sin <= 59;
21            else if (counter >= 41 && counter < 52)
22              dds_sin <= 62;
23            else if (counter >= 52 && counter < 63)
24              dds_sin <= 65;
25            else if (counter >= 63 && counter < 73)
26              dds_sin <= 68;
27            else if (counter >= 73 && counter < 83)
28              dds_sin <= 71;
29            else if (counter >= 83 && counter < 93)
30              dds_sin <= 74;
31            else if (counter >= 93 && counter < 104)
32              dds_sin <= 77;
33            else if (counter >= 104 && counter < 115)
34              dds_sin <= 79;
35            else if (counter >= 115 && counter < 125)
36              dds_sin <= 82;
37            else if (counter >= 125 && counter < 135)
38              dds_sin <= 84;
39            else if (counter >= 135 && counter < 145)
40              dds_sin <= 86;
41            else if (counter >= 145 && counter < 156)
42              dds_sin <= 88;
43            else if (counter >= 156 && counter < 166)
44              dds_sin <= 90;
```

```
45            else if (counter >= 166 && counter < 176)
46              dds_sin <= 92;
47            else if (counter >= 176 && counter < 186)
48              dds_sin <= 94;
49            else if (counter >= 186 && counter < 196)
50              dds_sin <= 95;
51            else if (counter >= 196 && counter < 207)
52              dds_sin <= 96;
53            else if (counter >= 207 && counter < 217)
54              dds_sin <= 97;
55            else if (counter >= 217 && counter < 227)
56              dds_sin <= 98;
57            else if (counter >= 227 && counter < 237)
58              dds_sin <= 99;
59            else if (counter >= 237 && counter < 247)
60              dds_sin <= 99;
61            else if (counter >= 247 && counter < 258)
62              dds_sin <= 100;
63            else if (counter >= 258 && counter < 268)
64              dds_sin <= 100;
```

```
65            else if (counter >= 268 && counter < 278)
66              dds_sin <= 100;
67            else if (counter >= 278 && counter < 288)
68              dds_sin <= 99;
69            else if (counter >= 288 && counter < 298)
70              dds_sin <= 99;
71            else if (counter >= 298 && counter < 309)
72              dds_sin <= 98;
73            else if (counter >= 309 && counter < 319)
74              dds_sin <= 97;
75            else if (counter >= 319 && counter < 329)
76              dds_sin <= 96;
77            else if (counter >= 329 && counter < 339)
78              dds_sin <= 95;
79            else if (counter >= 334 && counter < 344)
80              dds_sin <= 94;
81            else if (counter >= 344 && counter < 355)
82              dds_sin <= 92;
83            else if (counter >= 355 && counter < 365)
84              dds_sin <= 90;
85            else if (counter >= 365 && counter < 375)
86              dds_sin <= 88;
87            else if (counter >= 375 && counter < 385)
88              dds_sin <= 86;
```

```
89            else if (counter >= 385 && counter < 395)
90              dds_sin <= 84;
91            else if (counter >= 395 && counter < 407)
92              dds_sin <= 82;
93            else if (counter >= 407 && counter < 418)
94              dds_sin <= 79;
95            else if (counter >= 418 && counter < 428)
96              dds_sin <= 77;
97            else if (counter >= 428 && counter < 439)
98              dds_sin <= 74;
99            else if (counter >= 439 && counter < 449)
100             dds_sin <= 71;
101           else if (counter >= 449 && counter < 460)
102             dds_sin <= 68;
103           else if (counter >= 460 && counter < 470)
104             dds_sin <= 65;
105           else if (counter >= 470 && counter < 480)
106             dds_sin <= 62;
107           else if (counter >= 480 && counter < 490)
108             dds_sin <= 59;
109           else if (counter >= 490 && counter < 500)
110             dds_sin <= 56;
111           else if (counter >= 500 && counter < 512)
112             dds_sin <= 53;
113           else if (counter >= 512 && counter < 522)
114             dds_sin <= 50;
```

```
115             else if (counter >= 522 && counter < 532)
116               dds_sin <= 47;
117             else if (counter >= 432 && counter < 542)
118               dds_sin <= 43;
119             else if (counter >= 542 && counter < 552)
120               dds_sin <= 40;
121             else if (counter >= 552 && counter < 553)
122               dds_sin <= 37;
123             else if (counter >= 553 && counter < 563)
124               dds_sin <= 34;
125             else if (counter >= 563 && counter < 573)
126               dds_sin <= 31;
127             else if (counter >= 573 && counter < 583)
128               dds_sin <= 28;
129             else if (counter >= 583 && counter < 593)
130               dds_sin <= 26;
131             else if (counter >= 593 && counter < 604)
132               dds_sin <= 23;
133             else if (counter >= 604 && counter < 614)
134               dds_sin <= 20;
135             else if (counter >= 614 && counter < 624)
136               dds_sin <= 18;
137             else if (counter >= 624 && counter < 634)
138               dds_sin <= 16;
139             else if (counter >= 634 && counter < 644)
140               dds_sin <= 13;
141             else if (counter >= 644 && counter < 665)
142               dds_sin <= 11;
143             else if (counter >= 665 && counter < 675)
144               dds_sin <= 9;
145             else if (counter >= 675 && counter < 685)
146               dds_sin <= 8;
147             else if (counter >= 685 && counter < 695)
148               dds_sin <= 6;
149             else if (counter >= 695 && counter < 705)
150               dds_sin <= 5;
151             else if (counter >= 705 && counter < 716)
152               dds_sin <= 3;
153             else if (counter >= 716 && counter < 726)
154               dds_sin <= 2;
155             else if (counter >= 726 && counter < 736)
156               dds_sin <= 1;
157             else if (counter >= 736 && counter < 746)
158               dds_sin <= 1;
159             else if (counter >= 746 && counter < 756)
160               dds_sin <= 0;
161             else if (counter >= 756 && counter < 767)
162               dds_sin <= 0;
163             else if (counter >= 767 && counter < 777)
164               dds_sin <= 0;
```

```
165             else if (counter >= 777 && counter < 787)
166               dds_sin <= 0;
167             else if (counter >= 787 && counter < 797)
168               dds_sin <= 0;
169             else if (counter >= 797 && counter < 807)
170               dds_sin <= 1;
171             else if (counter >= 807 && counter < 819)
172               dds_sin <= 1;
173             else if (counter >= 819 && counter < 829)
174               dds_sin <= 2;
175             else if (counter >= 829 && counter < 839)
176               dds_sin <= 3;
177             else if (counter >= 839 && counter < 849)
178               dds_sin <= 5;
179             else if (counter >= 849 && counter < 859)
180               dds_sin <= 6;
181             else if (counter >= 859 && counter < 870)
182               dds_sin <= 8;
183             else if (counter >= 870 && counter < 880)
184               dds_sin <= 9;
185             else if (counter >= 880 && counter < 890)
186               dds_sin <= 11;
187             else if (counter >= 890 && counter < 900)
188               dds_sin <= 13;
189             else if (counter >= 900 && counter < 910)
190               dds_sin <= 16;
191             else if (counter >= 910 && counter < 921)
192               dds_sin <= 18;
193             else if (counter >= 921 && counter < 932)
194               dds_sin <= 20;
195             else if (counter >= 932 && counter < 942)
196               dds_sin <= 23;
197             else if (counter >= 942 && counter < 952)
198               dds_sin <= 26;
199             else if (counter >= 952 && counter < 962)
200               dds_sin <= 28;
201             else if (counter >= 962 && counter < 973)
202               dds_sin <= 31;
203             else if (counter >= 973 && counter < 983)
204               dds_sin <= 34;
205             else if (counter >= 983 && counter < 993)
206               dds_sin <= 37;
207             else if (counter >= 993 && counter < 1003)
208               dds_sin <= 40;
209             else if (counter >= 1003 && counter < 1013)
210               dds_sin <= 43;
211             else if (counter >= 1013 && counter <= 1023)
212               dds_sin <= 47;
213             else
214               dds_sin <= 50;
```

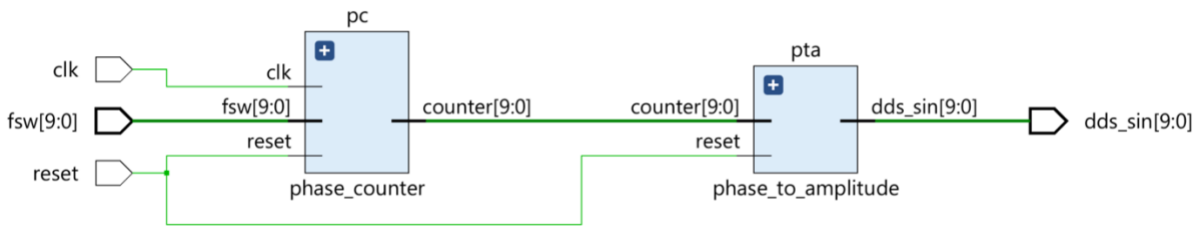
```
215         end
216 endmodule
```

# Top Module:

Code:

```
1  `timescale 1ns / 1ps
2
3  module topmodule(clk,
4                   reset,
5                   fsw,
6                   dds_sin);
7
8          input clk;
9          input reset;
10         input [9:0] fsw;
11
12         output [9:0] dds_sin;
13
14
15         wire [9:0] counter;
16
17         phase_counter pc(clk,reset,fsw,counter);
18
19         phase_to_amplitude pta(counter,reset,dds_sin);
20
21 endmodule
```

Schematic:




As you can see this Vivado Schematic is pretty much same as what was originally proposed. You can see a 10 bit fsw going into phase counter which is our phase accumulator and then the counter value going to phase to amplitude block where it assigns a specific amplitude to that counter value, giving us a digital sine wave as output.

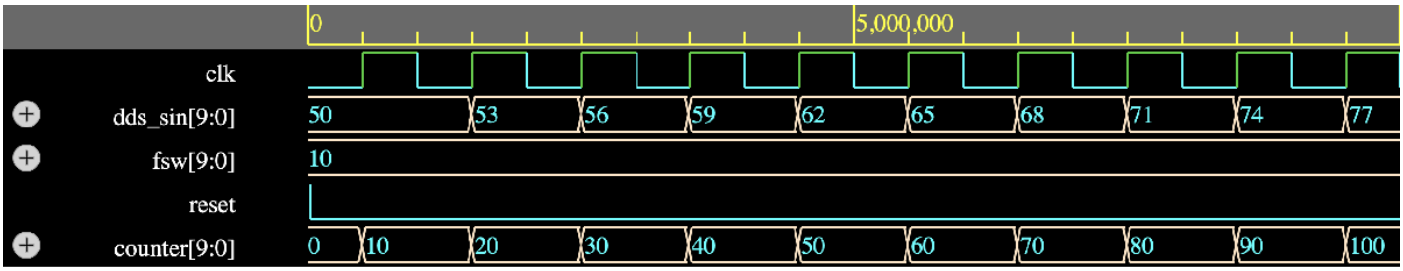
Testbench:

```
`timescale 1ns/1ps

`include "phase_counter.v"
`include "phase_to_amplitude.v"
`include "top_module.v"

module tb_topmodule();

  reg clk;
  reg reset;
  reg [9:0] fsw;
  reg [9:0] dds_sin;

  // Instantiate the top module
  topmodule uut(
    .clk(clk),
    .reset(reset),
    .fsw(fsw),
    .dds_sin(dds_sin)
  );

  // Clock generation
  initial begin
    clk = 0;
    forever #500 clk = ~clk; // 1 MHz clock
  end

  // Initial stimulus
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars;
    reset = 1;
    fsw = 205;
    #10 reset = 0; // Deassert reset after 10 time units
  end

  // Monitor for observing signals
  always @(posedge clk) begin
    $display("Time=%t, Counter=%0d, DDS_Sin=%0d", $time, uut.counter, uut.dds_sin);
  end

  // Stop the simulation after some time
  initial #105000 $finish;

endmodule
```

# Results (Using Matlab)

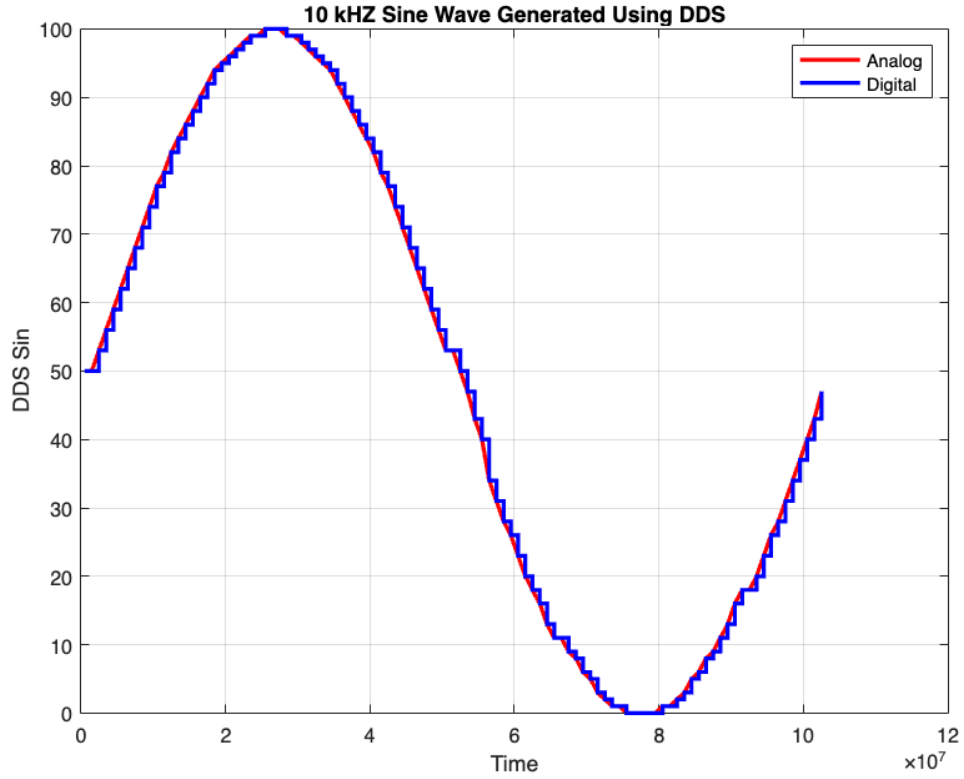
## 10 kHz Sine Wave

### Simulation:



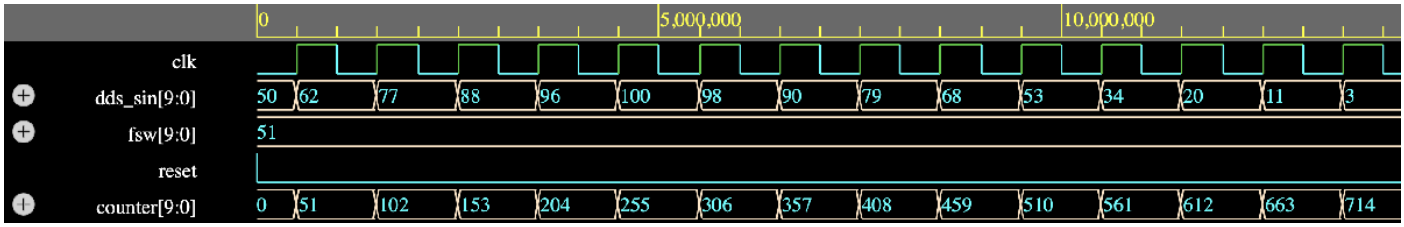

### Graphing (For 102.5 us):

```
% Open the file for reading
fileID = fopen('10kHz.txt', 'r');

% Read the data using fscanf
data = fscanf(fileID, '# KERNEL: Time= %f, Counter= %d, DDS_Sin= %d\n', [3,
Inf]);

% Close the file
fclose(fileID);

% Extract counter, DDS_Sin, and time values
time = data(1, :);
counter = data(2, :);
dds_sin = data(3, :);

% Plot the graphs
plot(time, dds_sin, "r", "LineWidth", 2);
hold on
stairs(time, dds_sin, "b", "LineWidth", 2);
hold off

% Add labels and title
legend('Analog', 'Digital');
xlabel('Time');
ylabel('DDS Sin');
title('10 kHZ Sine Wave Generated Using DDS');

% Show the grid
grid on;
```




## 50kHz Sine Wave

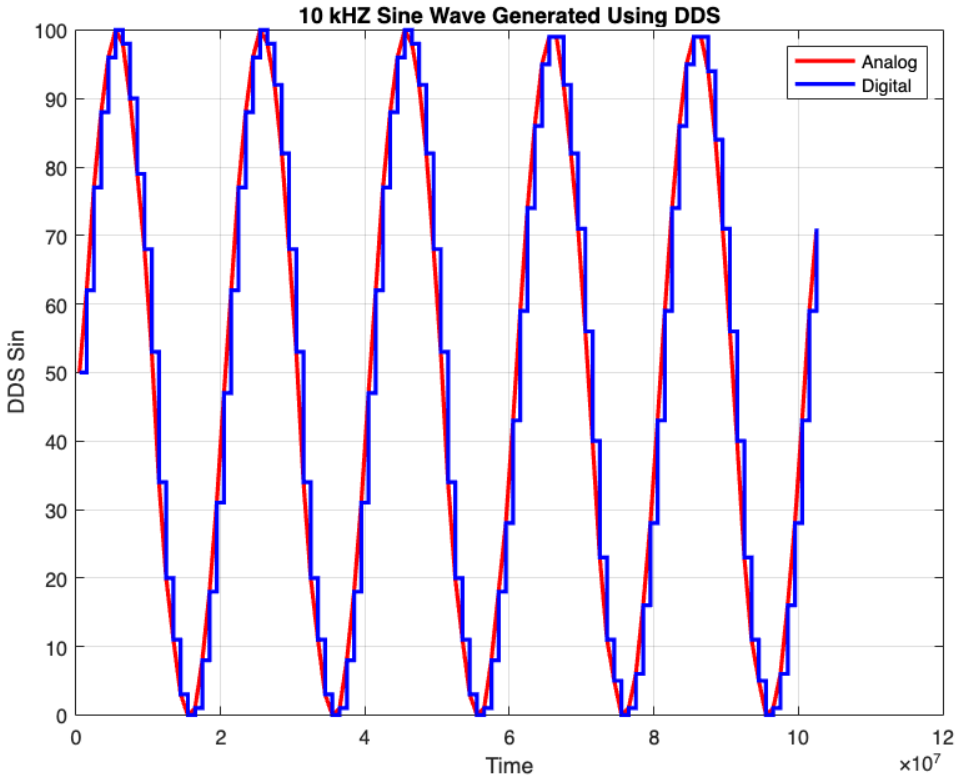
### SImulation:




### Graphing (For 102.5 us):

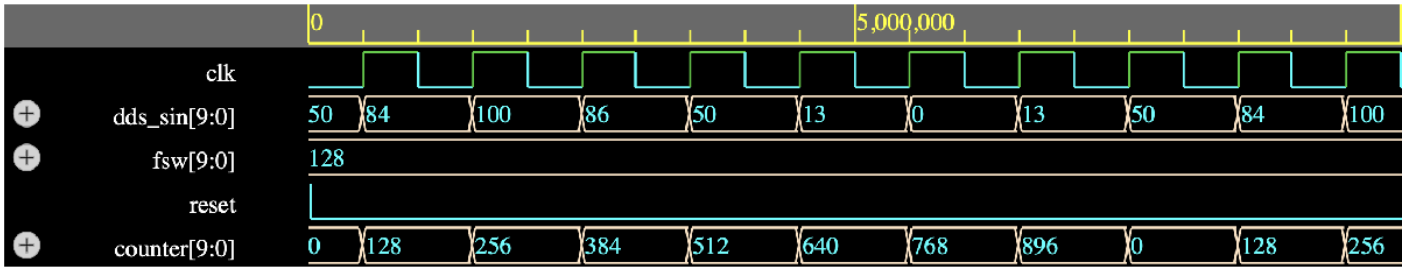
```
% Open the file for reading
fileID = fopen('50kHz.txt', 'r');

% Read the data using fscanf
data = fscanf(fileID, '# KERNEL: Time= %f, Counter= %d, DDS_Sin= %d\n', [3,
Inf]);

% Close the file
fclose(fileID);

% Extract counter, DDS_Sin, and time values
time = data(1, :);
counter = data(2, :);
```

```
dds_sin = data(3, :);

% Plot the graphs
plot(time, dds_sin, "r", "LineWidth", 2);
hold on
stairs(time, dds_sin, "b", "LineWidth", 2);
hold off

% Add labels and title
legend('Analog', 'Digital');
xlabel('Time');
ylabel('DDS Sin');
title('10 kHZ Sine Wave Generated Using DDS');

% Show the grid
grid on;
```



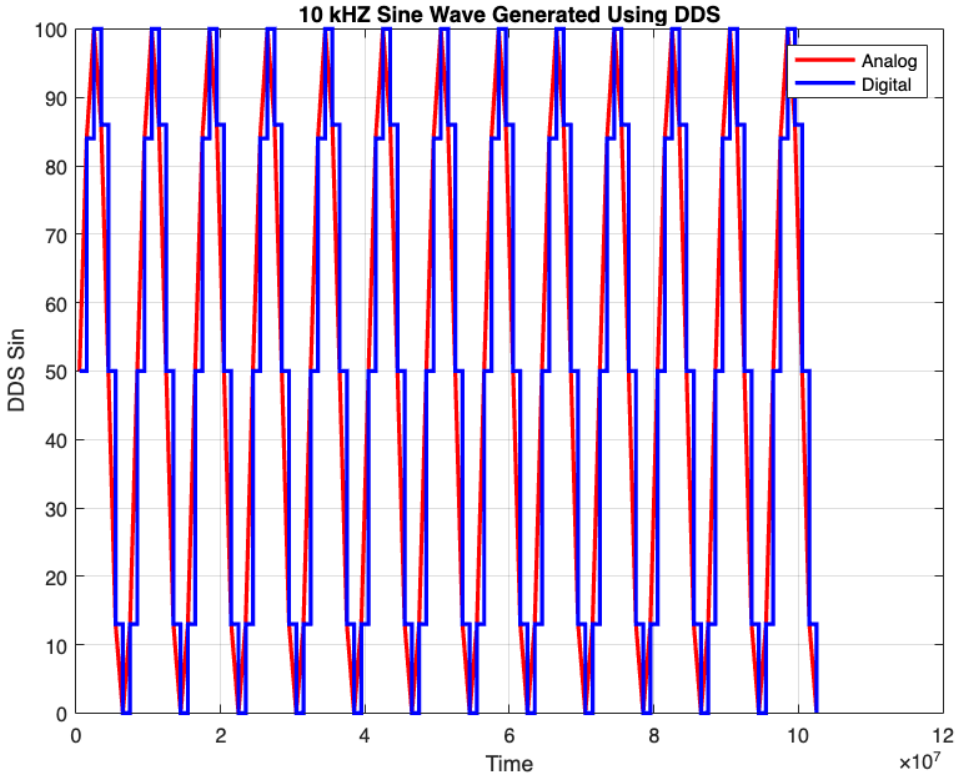

## 125kHz Sine Wave

**Simulation:**



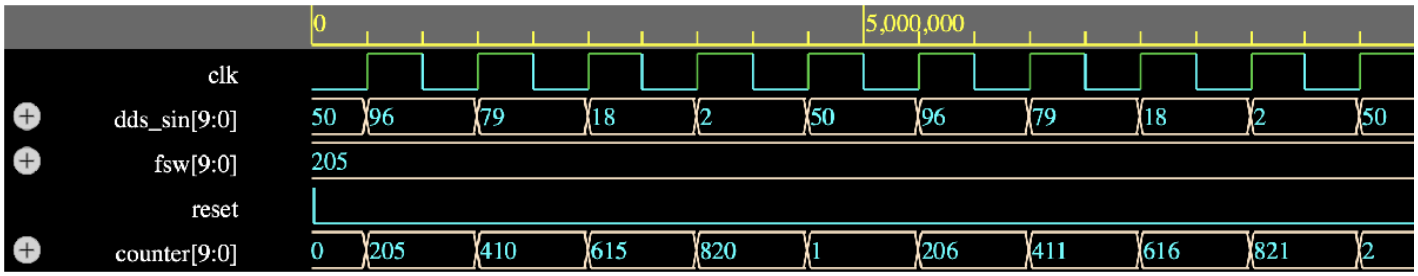

### Graphing (For 102.5 us):

```
% Open the file for reading
fileID = fopen('125kHz.txt', 'r');

% Read the data using fscanf
data = fscanf(fileID, '# KERNEL: Time= %f, Counter= %d, DDS_Sin= %d\n', [3,
Inf]);

% Close the file
fclose(fileID);

% Extract counter, DDS_Sin, and time values
time = data(1, :);
counter = data(2, :);
dds_sin = data(3, :);

% Plot the graphs
plot(time, dds_sin, "r", "LineWidth", 2);
hold on
stairs(time, dds_sin, "b", "LineWidth", 2);
hold off

% Add labels and title
legend('Analog', 'Digital');
xlabel('Time');
ylabel('DDS Sin');
title('10 kHZ Sine Wave Generated Using DDS');

% Show the grid
grid on;
```




## 200kHz Sine Wave

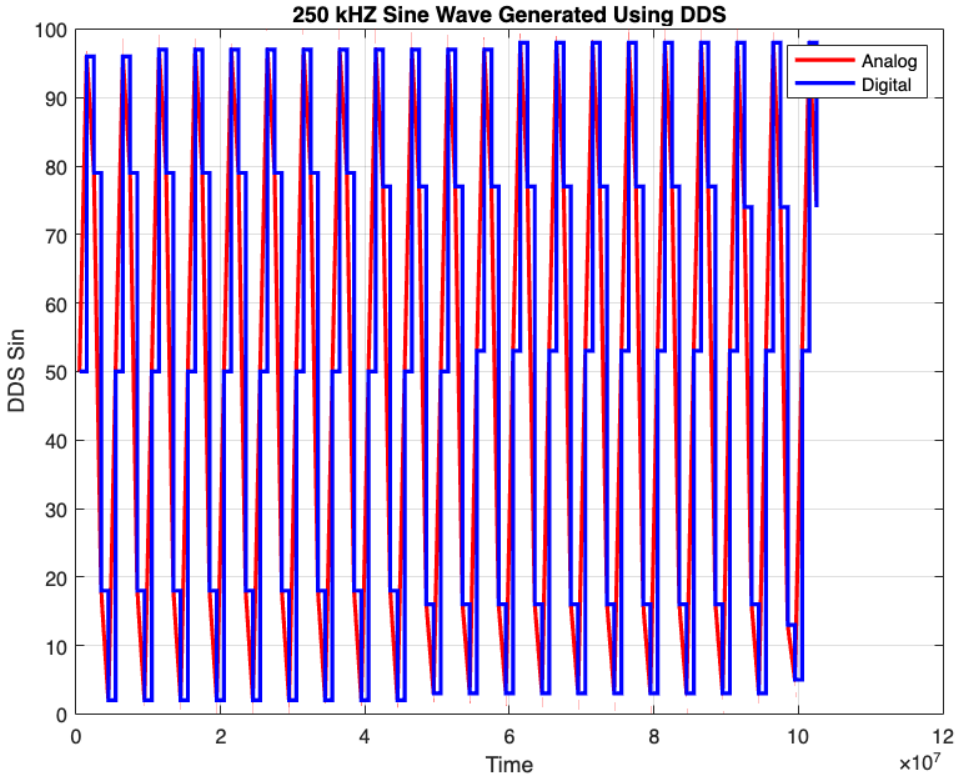
### Simulation:




### Graphing (For 102.5 us):

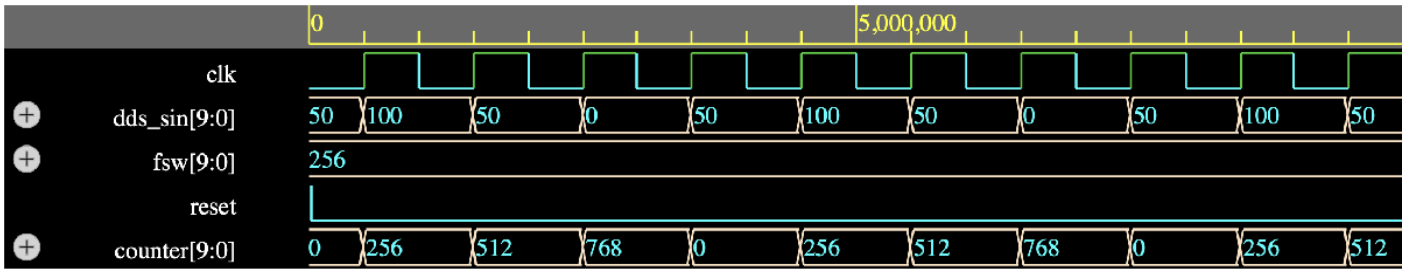
```
% Open the file for reading
fileID = fopen('200kHz.txt', 'r');

% Read the data using fscanf
data = fscanf(fileID, '# KERNEL: Time= %f, Counter= %d, DDS_Sin= %d\n', [3,
Inf]);

% Close the file
fclose(fileID);

% Extract counter, DDS_Sin, and time values
time = data(1, :);
```

```
counter = data(2, :);
dds_sin = data(3, :);

% Plot the graphs
plot(time, dds_sin, "r", "LineWidth", 2);
hold on
stairs(time, dds_sin, "b", "LineWidth", 2);
hold off

% Add labels and title
legend('Analog', 'Digital');
xlabel('Time');
ylabel('DDS Sin');
title('250 kHZ Sine Wave Generated Using DDS');

% Show the grid
grid on;
```



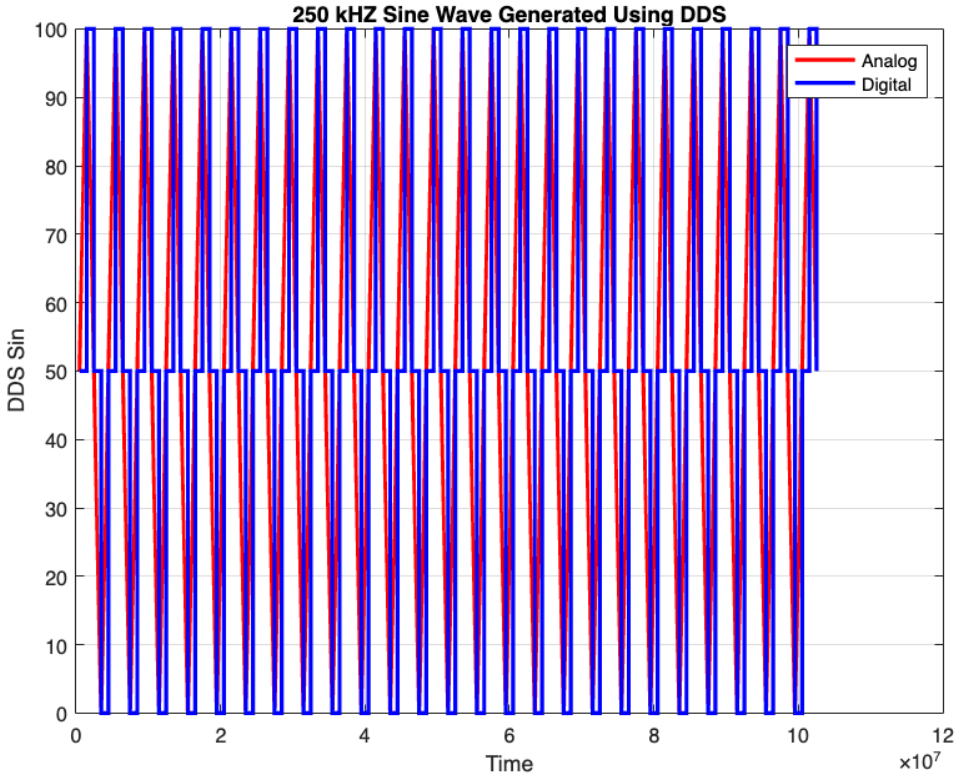

## 250kHz Sine Wave

**Simulation:**




### Graphing (For 102.5 us):

```
% Open the file for reading
fileID = fopen('250kHz.txt', 'r');

% Read the data using fscanf
data = fscanf(fileID, '# KERNEL: Time= %f, Counter= %d, DDS_Sin= %d\n', [3,
Inf]);

% Close the file
fclose(fileID);

% Extract counter, DDS_Sin, and time values
time = data(1, :);
counter = data(2, :);
dds_sin = data(3, :);

% Plot the graphs
plot(time, dds_sin, "r", "LineWidth", 2);
hold on
stairs(time, dds_sin, "b", "LineWidth", 2);
hold off

% Add labels and title
legend('Analog', 'Digital');
xlabel('Time');
ylabel('DDS Sin');
title('250 kHZ Sine Wave Generated Using DDS');

% Show the grid
grid on;
```


250 kHZ Sine Wave Generated Using DDS
Analog
Digital
DDS Sin
Time
×10^7
0
10
20
30
40
50
60
70
80
90
100
0
2
4
6
8
10
12