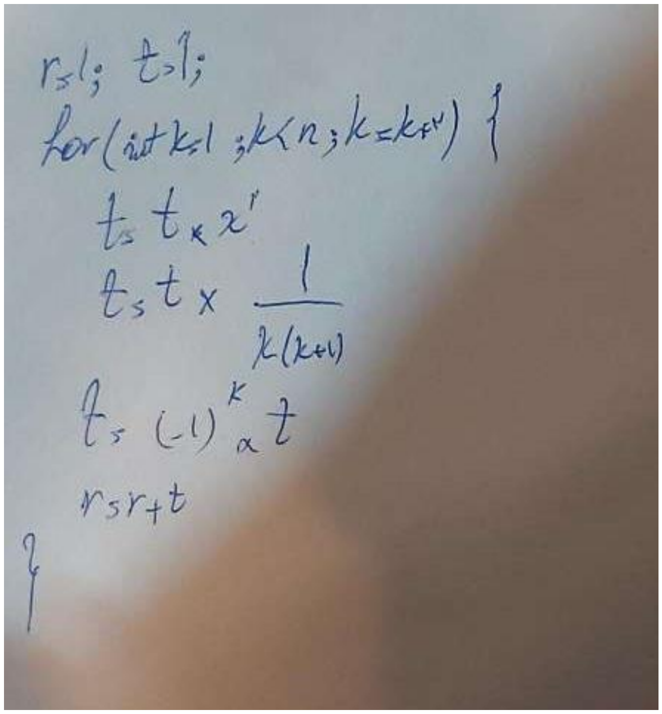
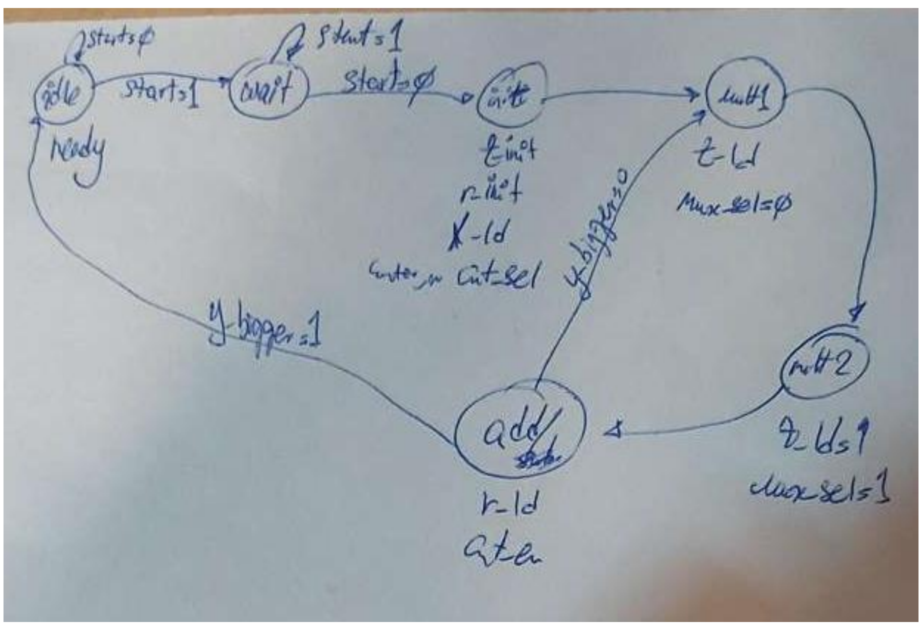
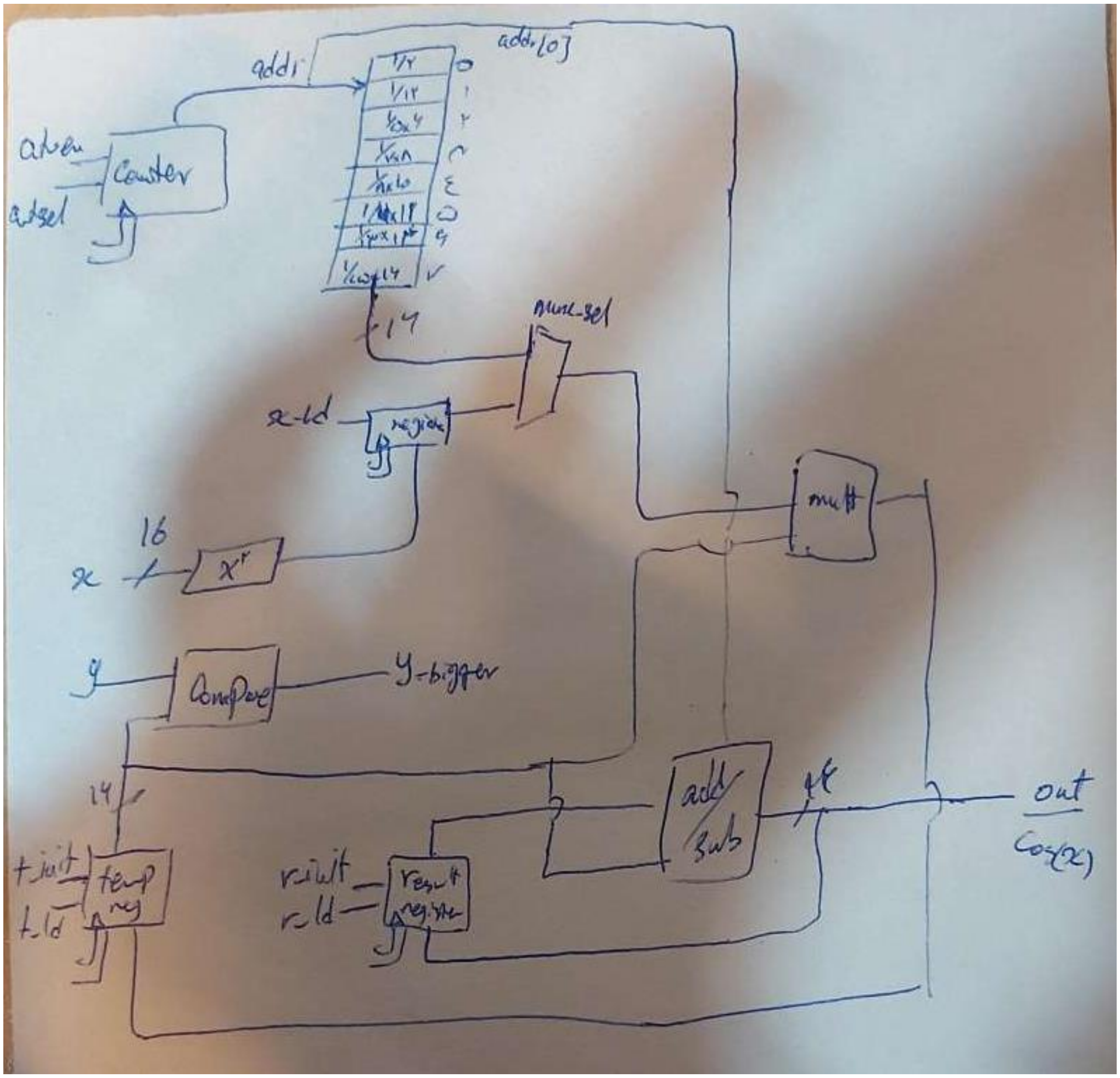

r=1; t=1;
for(int k=1 ;k<n;k=k+1) {
t=t×x²
t=t× 1/k(k+1)
t=(-1)^k×t
r=r+t
}


start=ϕ
idle
start=1
wait
start=1
start=ϕ
init
mult1
ready
t_init
r_init
k_ld
cnt_sel
y_bigger=0
t_ld
Mux_sel=ϕ
y_bigger=1
mult2
add
t_ld=1
Mux_sel=1
r_ld
cnt_en


addr
addr[0]
Counter
mux-sel
register
16
x
mult
Compare
y
y-bigger
add/sub
out
Cos(x)
temp reg
result register

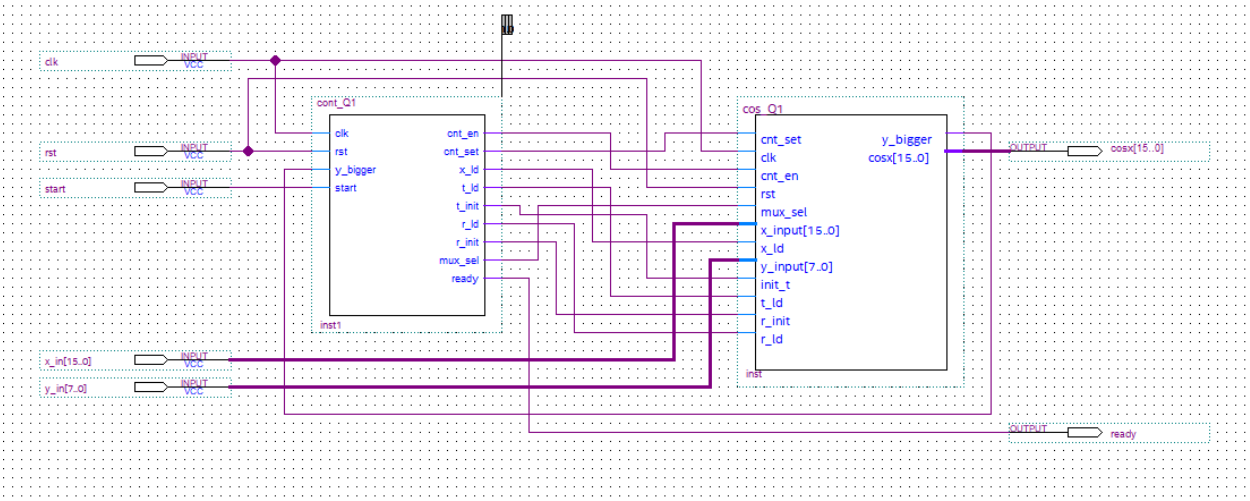
```
1  `timescale 1ns/1ns
2  module cont_Q1 (
3      input clk,rst,y_bigger,start, output reg cnt_en, cnt_set, x_ld, t_ld, t_init, r_ld, r_init, mux_sel,ready);
4      reg[2:0] ns,ps;
5      parameter [2:0]  Idle = 0, Wait = 1,  Init = 2, Mult1 = 3, Mult2 = 4, Add = 5;
6      always @(ps,y_bigger) begin
7          {cnt_en, cnt_set, x_ld, t_ld, t_init , r_ld, r_init, mux_sel, ready} = 9'b0;
8          case(ps)
9              Idle : begin
10                 ns = start ? Wait : Idle;
11                 ready = 1'b1;
12             end
13             Wait : begin
14                 ns = start ? Wait : Init;
15             end
16             Init : begin
17                 ns = Mult1;
18                 t_init = 1'b1;
19                 r_init = 1'b1;
20                 x_ld = 1'b1;
21                 cnt_set  = 1'b1;
22             end
23             Mult1 : begin
24                 ns = Mult2;
25                 mux_sel = 1'b0;
26                 t_ld = 1'b1;
27             end
28             Mult2 : begin
29                 ns = Add;
30                 mux_sel = 1'b1;
31                 t_ld = 1'b1;
32             end
33             Add : begin
34                 ns = y_bigger ? Idle : Mult1;
35                 r_ld = 1'b1;
36                 cnt_en = 1'b1;
37             end
38         endcase
39     end
40     always @(posedge clk,posedge rst) begin
41         if(rst)
42             ps <= Idle;
43         else
44             ps <= ns;
45     end
46 endmodule
```

combinational

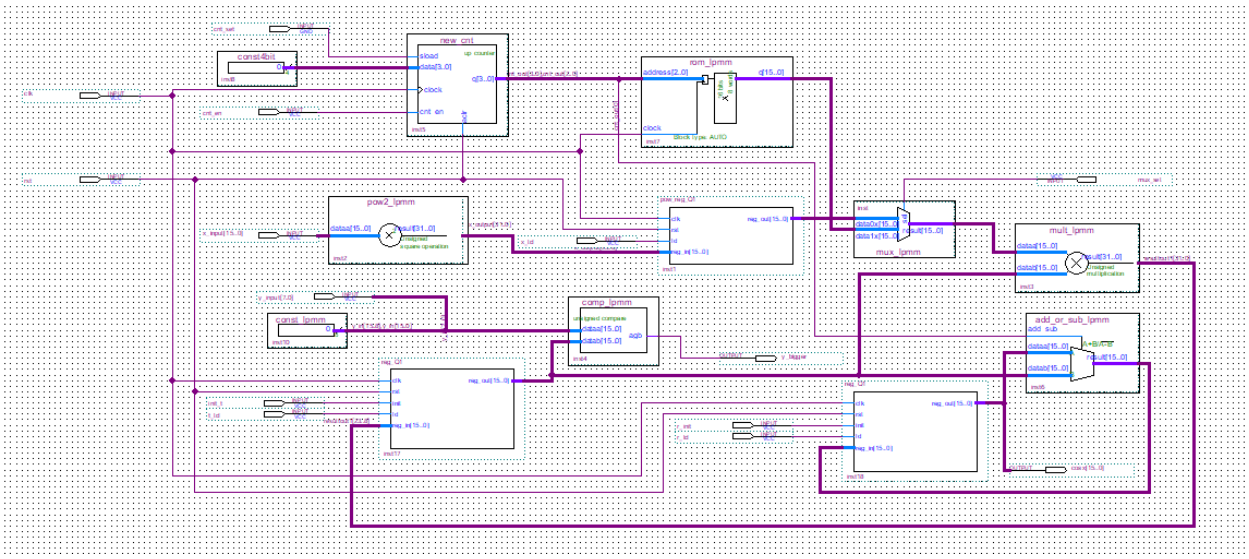
sequential

controller (above):

final cos(x):



DataPath:



```
1   `timescale 1ns/1ns
2   module reg_Q1 (input clk,rst,init,ld, input[15:0] reg_in, output reg [15:0] reg_out );
3       always @(posedge clk, posedge rst) begin
4           if (rst)
5               reg_out <= 16'b0;
6           else begin
7               if (init)
8                   reg_out <= 16'b0000000011111111;
9               else
10                  reg_out <= (ld) ? reg_in : reg_out;
11
12          end
13      end
14  endmodule
```

```
1   `timescale 1ns/1ns
2   module pow_reg_Q1(input clk, rst, ld, input[15:0] reg_in, output reg[15:0] reg_out);
3       always@(posedge clk, posedge rst) begin
4           if(rst)
5               reg_out <= 16'b0;
6           else
7               reg_out <= (ld) ? reg_in : reg_out;
8       end
9   endmodule
```

Cos(x) compilation:

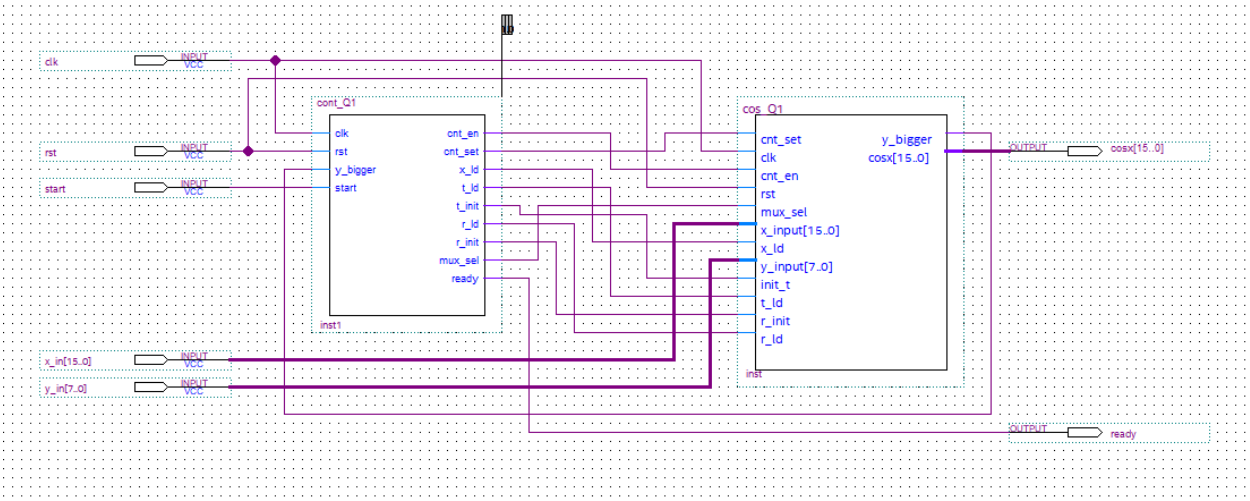
Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Assembler
- Timing Analyzer
- EDA Netlist Writer
- Flow Messages
- Flow Suppressed Messages

Flow Summary

<<Filter>>

Flow Status	Successful - Tue Jan 09 19:36:27 2024
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	final_cosx_Q1
Top-level Entity Name	final_cosx_Q1
Family	Cyclone IV GX
Device	EP4CGX15BF14A7
Timing Models	Final
Total logic elements	585 / 14,400 (4 %)
Total registers	57
Total pins	44 / 81 (54 %)
Total virtual pins	0
Total memory bits	128 / 552,960 (< 1 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)



Testbench:

```
1  `timescale 1ns/1ns
2  module cosx_tb();
3      reg Start=0,clk=0,rst=0;
4      reg [7:0] y=8'b11111111;
5      reg [15:0] x=16'b0000000110000000; //cos(1.5) = 0.0707 -> 0.082
6      wire [15:0] cosx;
7      wire ready;
8      final_cosx_Q1 CUT(ready,clk,rst,x,y,Start,cosx);
9      always begin #41;clk=~clk;end
10     initial begin
11         #82;
12         #82;
13         #82;
14         #20; Start = 1;
15         #82; Start=0;
16         #1640;
17         #82;
18         #82;
19         #1640;
20         #82;
21         #82;
22         #82;#82; y=8'b00000001; x=16'b0000001000000000; //cos(2) = -0.41 -> -0.41
23         #82; Start = 1;
24         #82; Start=0;
25         #1640;
26         #82;
27         #82;
28         #82;
29         #82;
30         $stop;
31     end
32 endmodule
```