به نام خدا

University of Tehran

Electronic and Computer Engineering

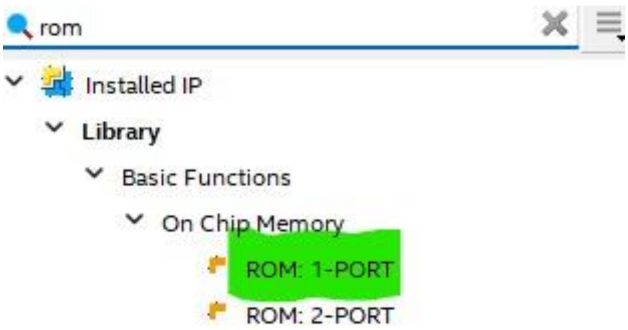
Digital Systems 1

Computer Assignment 6

Amirhossein Yavarikhoo	نام و نام خانوادگی
810199514	شماره دانشجویی
1401/3/23	تاریخ ارسال گزارش

Part A

To create a ROM, we use on chip memory 1-port ROM from basic function in quartus library.



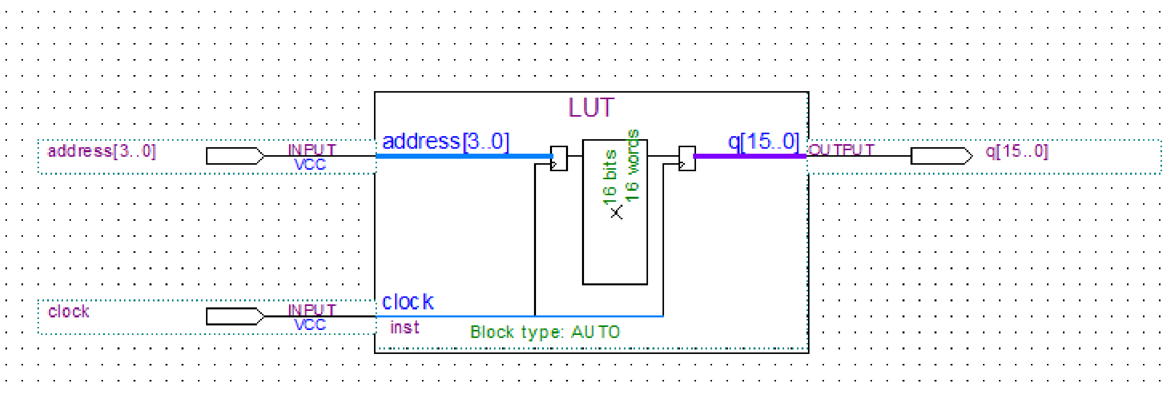
To instantiate it we call it LUT(look up table).

Each word/data that is saved on this memory is a 16 bit number so the width of each word is 16 bits and the memory should have 16 words. This results in address for each memory be 4 bits (address [3:0]). Since this memory is read-only, we have to store initial values manually. To do so, first we open a new memory initialization file(.mif) from quartus. The values for 1/k is less than one so we have to shift them to the left by 16 bits (which equals to multiplying them by 2^{16}) and in the end consider this 16 bits as fractions.

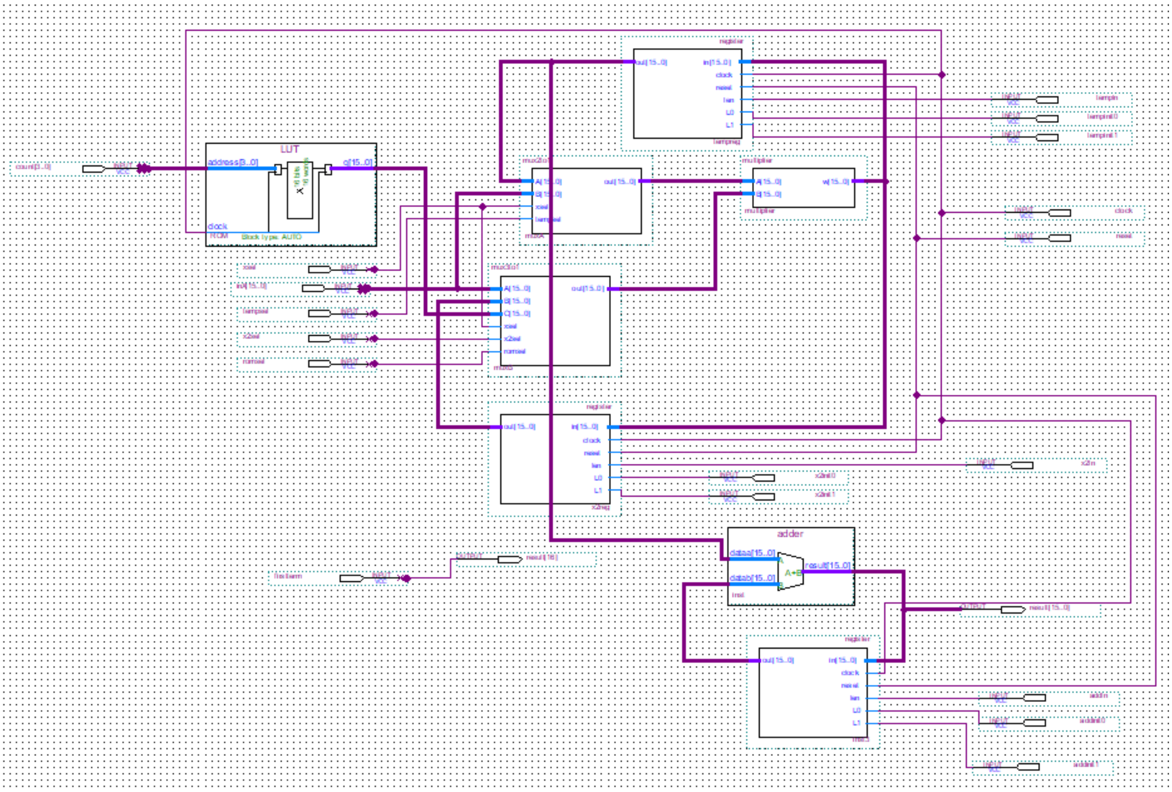
Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	FFFF	FFFF	8000	5555	4000	3333	2AAA	2492	....
8	2000	1C71	1999	1745	1555	1381	1249	0000	....

Hexadecimal values for each 1/k is represented above.

We load this file in ROM and we have the look up table needed for this project.



Part B

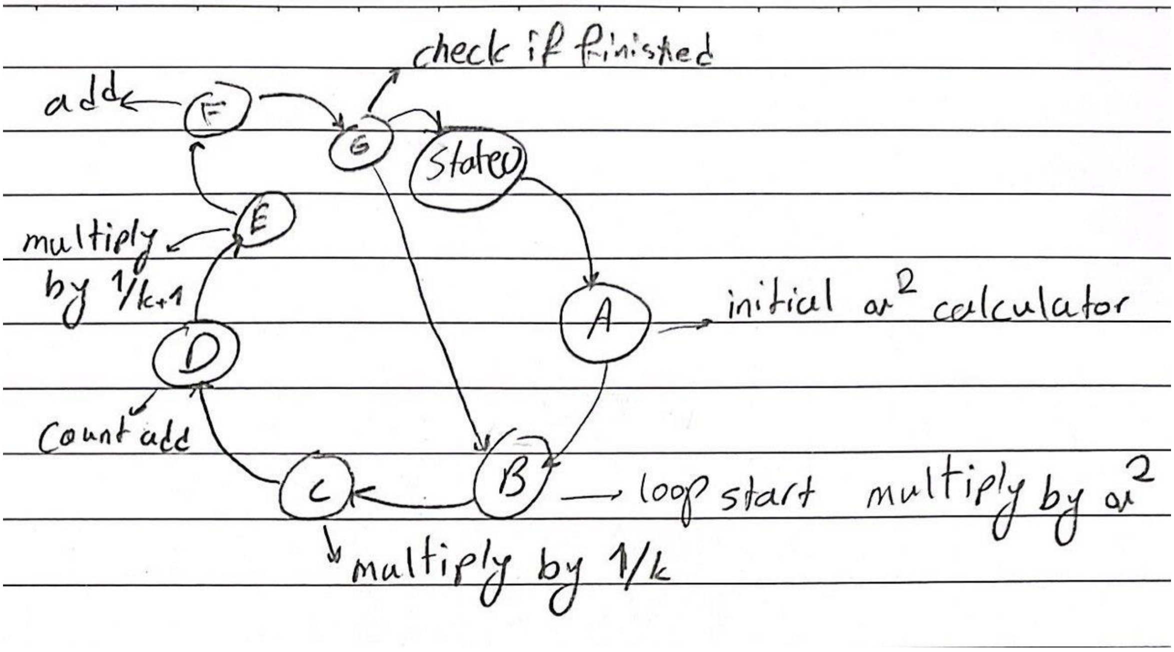


Synthesis report is shown below:

Flow Status	Successful - Tue Jun 21 17:19:20 2022
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	coshcal
Top-level Entity Name	coshcal
Family	Cyclone IV E
Total logic elements	116 / 6,272 (2 %)
Total registers	48
Total pins	53 / 92 (58 %)
Total virtual pins	0
Total memory bits	256 / 276,480 (< 1 %)
Embedded Multiplier 9-bit elements	2 / 30 (7 %)
Total PLLs	0 / 2 (0 %)
Device	EP4CE6E22C6
Timing Models	Final

Part C

State diagram:



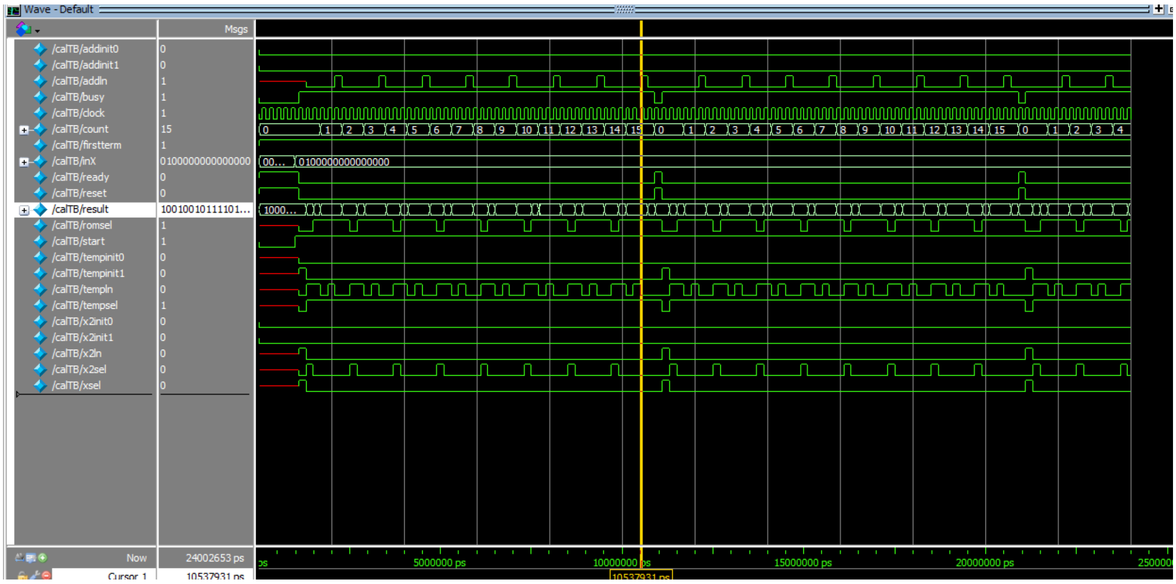
Verilog code for this state diagram is shown below:

```
`timescale 1ns/1ns
module controller (input start,clock,output reg [3:0] count,output reg xsel,tempsel,
                        x2sel,romsel,x2init0,x2init1,x2ln,templn,tempinit0,tempinit1,reset,addinit0,addinit1,addln,busy,ready,output firstterm);
        assign firstterm=1'b1;
        reg [2:0] pstate,nstate;
        parameter [2:0] state0=3'd0,stateA=3'd1,stateB=3'd2,stateC=3'd3,stateD=3'd4,stateE=3'd5,stateF=3'd6,stateG=3'd7;
        assign pstate=state0;
        always @(pstate,start) begin
                case (pstate)
                        state0:begin
                                reset=1;
                                count=4'd0;
                                ready=1;
                                count=0;x2init0=0;x2init1=0;addinit0=0;addinit1=0;busy=0;
                                if (start==1'b1) nstate<=stateA;
                                else nstate<=state0;
                                end
                        stateA:begin
                                reset=0;xsel=1;x2sel=0;romsel=0;templn=0;busy=1;tempsel=0;tempinit0=0;tempinit1=1;
                                x2ln=1;ready=0;
                                nstate<=stateB;
                                end
                        stateB:begin
                                tempinit1=0;xsel=0;tempsel=1;x2sel=1;romsel=0;x2ln=0;addln=0;templn=1;
                                nstate<=stateC;
                                end
                        stateC: begin
                                tempsel=1;x2sel=0;romsel=1;
                                nstate<=stateD;
                                end
                        stateD:begin
                                templn=0;count=count+1;
                                nstate<=stateE;
                                end
                        stateE: begin
                                tempsel=1;x2sel=0;romsel=1;templn=1;
                                nstate<=stateF;
                                end
                        stateF: begin
                                addln=1;templn=0;
                                nstate<=stateG;
                                end
                        stateG:begin
                                addln=0;
                                if (count==15) nstate<=state0;
                                else begin nstate<=stateB;count=count+1;end
                                end
                endcase
        end
        always@(posedge clock) begin
                pstate<=nstate;
                end
endmodule
```

Part D

Testbench for this circuit is shown below:

```verilog
`timescale 1ns/1ns
module calTB();
        reg clock,start;
        reg xsel,tempsel,x2sel,romsel,x2init0,x2init1,x2ln,templn,tempinit0,tempinit1,reset,addinit0,addinit1,addln,busy,ready,firstterm;
        reg [16:0] result;
        reg [3:0] count;
        reg [15:0] inX;
        controller UUT(start,clock,count,xsel,tempsel,x2sel,romsel,x2init0,x2init1,x2ln,templn,tempinit0,tempinit1,reset,addinit0,addinit1,addln,busy,ready,firstterm);
        coshcal UUT1(result,clock,reset,templn,tempinit0,tempinit1,xsel,tempsel,inX,x2sel,romsel,x2ln,x2init0,x2init1,count,addln,addinit0,addinit1,firstterm);
        initial begin
        clock=1'b0;
        repeat (240) #100 clock=~clock;
        end
        initial begin
        start=1'b0;
        inX=16'd100;
        #1000;
        inX=16'd16384;
        start=1'b1;
        end
endmodule
```



This value is almost the same to real cosh(0.5).