Convolution

Implemented in Verilogger and Xilinx

Convolution (Simple Method):

Objective:

We want to design a filter using convolution method.Code in c language is written which satisfies the hardware of Convolution.

Code:

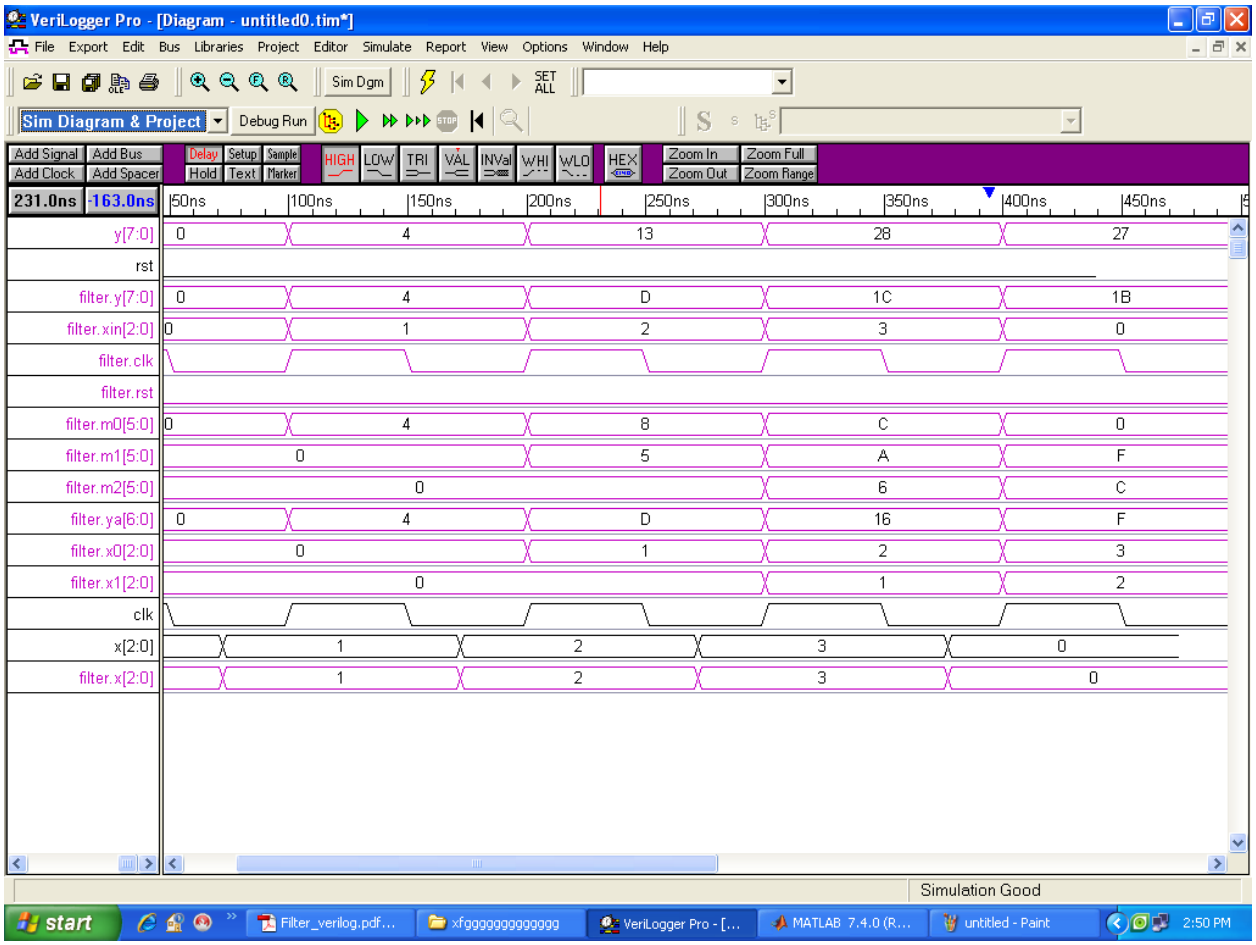
```
module filter(y,x,clk,rst);

parameter a0=3'b100 ;

parameter a1=3'b101 ;

parameter a2=3'b110 ;

output[7:0]y;

input [2:0]x;

input clk,rst;

wire[5:0] m0;

wire[5:0] m1;

wire[5:0] m2;

wire[6:0] ya;

reg[2:0] x0,x1,xin;

always@(posedge clk or posedge rst)

        if(rst)

        begin x0<=0 ; x1<=0;

    end

    else

        begin x0<=xin ; x1<=x0;

     xin<=x;

        end

assign m0=xin*a0;
```

```
    assign m1=x0*a1;

    assign m2=x1*a2;

    assign ya=m0+m1;

    assign y =ya+m2;

endmodule
```

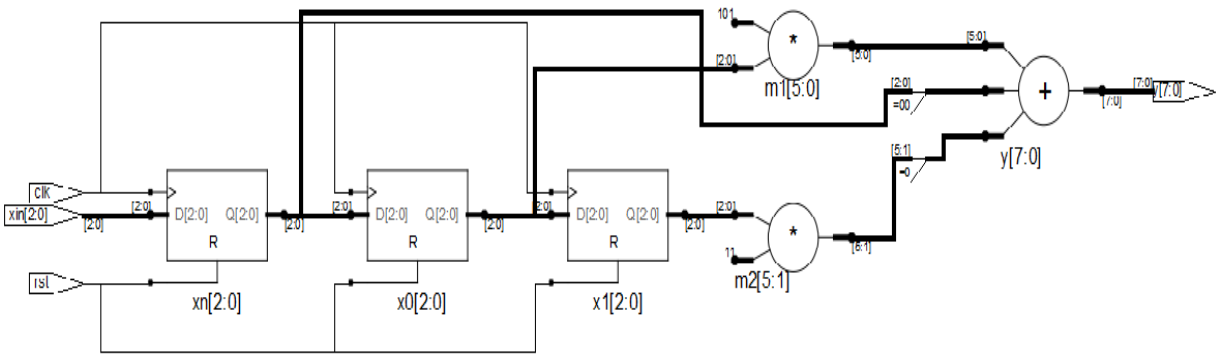
Result

The method,though not much efficient is a useful technique to build a required filter.Given below are timing Diagram and few of the characteristics of this method.

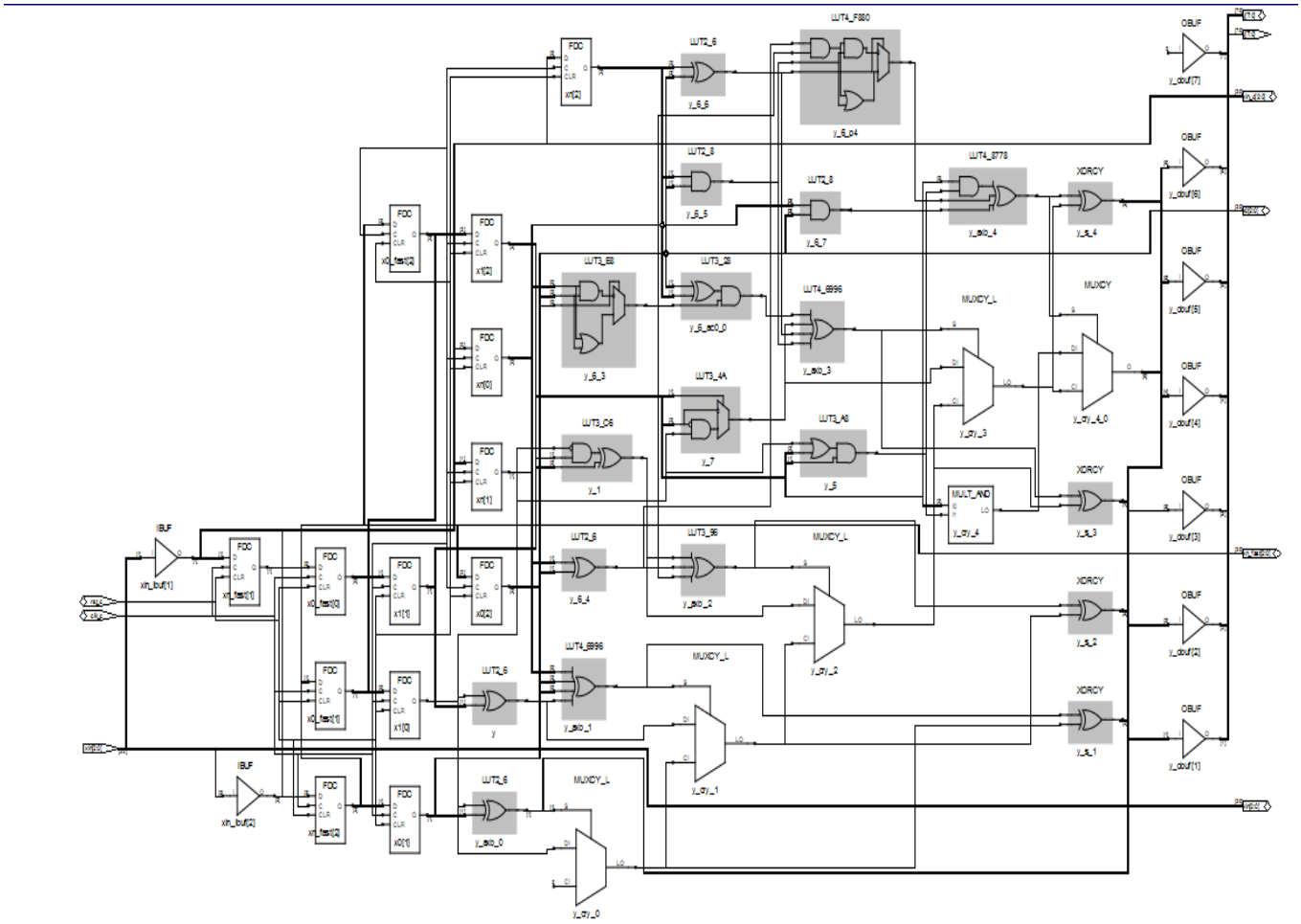
Timing Diagram:



RTL View:



Technology View:



Frequency:

```
Performance Summary
*******************


Worst slack in design: -0.265

                Requested     Estimated     Requested     Estimated               Clock         Clock
Starting Clock  Frequency     Frequency     Period        Period        Slack     Type          Group
------------------------------------------------------------------------------------------------------------------
filter|clk      663.4 MHz     535.6 MHz     1.503         1.768         -0.265    inferred      Autoconstr_clkgroup_0
==================================================================================================================
```

Resources:

```
Clock Relationships
*******************

Clocks                      |    rise  to  rise     |    fall  to  fal
-------------------------------------------------------------------
Starting     Ending         |  constraint  slack    |  constraint  sla
-------------------------------------------------------------------
filter|clk   filter|clk     |  1.503       -0.265   |  No paths    -
===================================================================

---------------------------------------
Resource Usage Report for filter

Mapping to part: xc3s200ft256-4
Cell usage:
FDC                 15 uses
GND                 1 use
MULT_AND            1 use
MUXCY               1 use
MUXCY_L             4 uses
XORCY               4 uses
LUT2                6 uses
LUT3                6 uses
LUT4                4 uses

I/O ports: 13
I/O primitives: 12
IBUF              4 uses
OBUF              8 uses

BUFGP             1 use

I/O Register bits:                      3
Register bits not including I/Os:   12  (0%)

Global Clock Buffers: 1 of 8  (12%)

Total load per clock:
   filter|clk: 15

Mapping Summary:
Total  LUTs: 16  (0%)

Mapper successful!
Process took 0h:00m:01s realtime, 0h:00m:01s cputime
# Wed Mar 14 23:10:11 2012

################################################################]
```