Duke

Complex Multiplier for FFT Algorithm

Team Members:

Xuan Bao	Xiaoyan Yin
Kai Wang	Yang Jiang

ECE 261 Project 2008

Index

1 **Brief Introduction**

2 Structure implementation

3 Feature Evaluation

Brief Introduction to the project

- Fast Fourier Transform (FFT) is an efficient algorithm to compute the Discrete Fourier Transform (DFT) and its inverse.
- The algorism is based on a factorization of N as described in Fig.1-1. A 16-point DFT is divided into 4 4-point DFT.

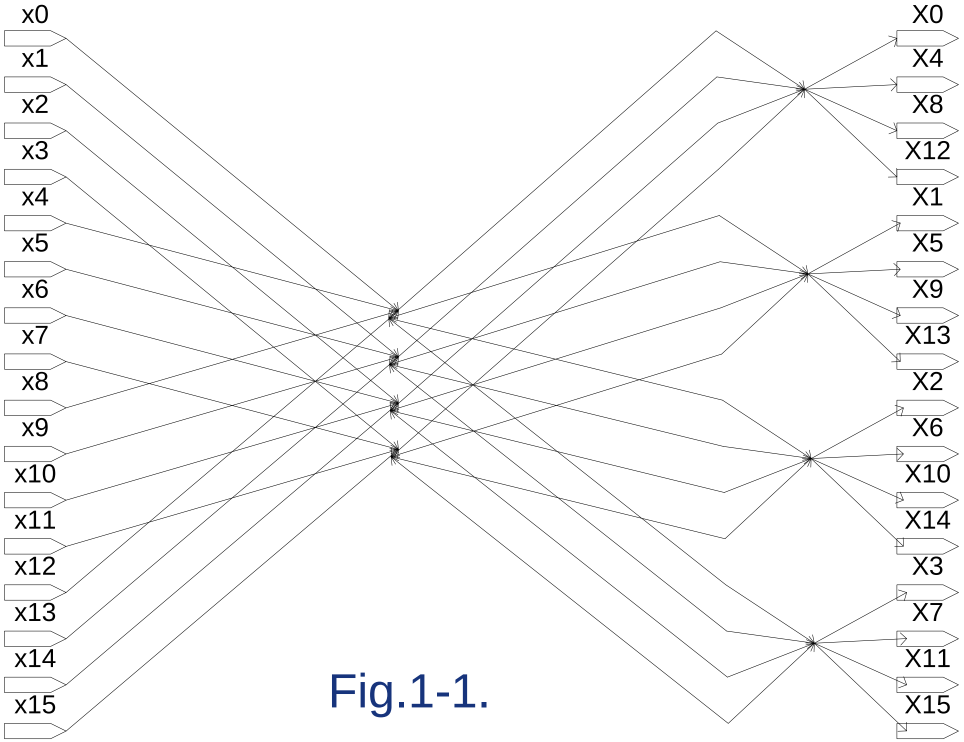
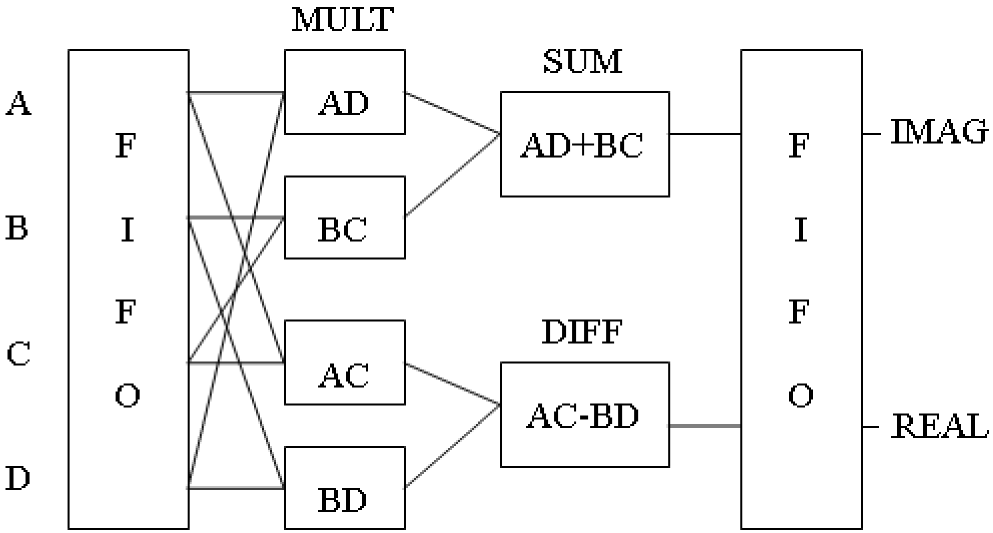


Fig.1-1.

Brief Introduction to the project

Fig.1-2. shows the basic unit used for implementing FFT algorithm, the complex multiplier. The function is

(A+Bj)(C+Dj)=(AC-BD)+j(AD+BC)

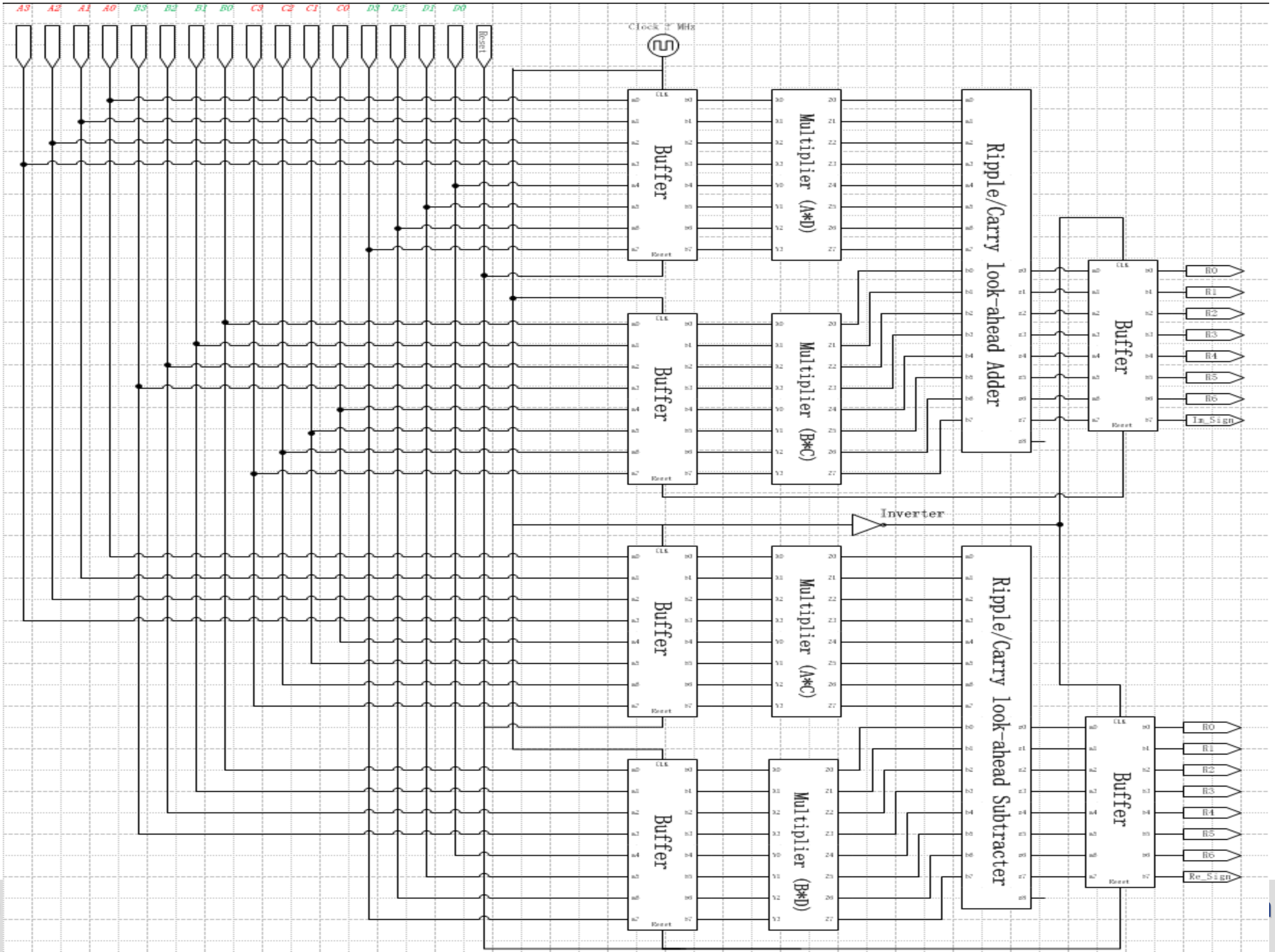


(A+Bj)(C+Dj)=(AC-BD)+j(AD+BC)

Fig.1-2.

Brief Introduction to the project

The top level overview of our design is given as Fig 1-3.



Duke

Index

1. Brief Introduction
2. **Structure Implementation**
3. Feature Evaluation

Structure Implementation

- Structure consists of…

FIFO × 6

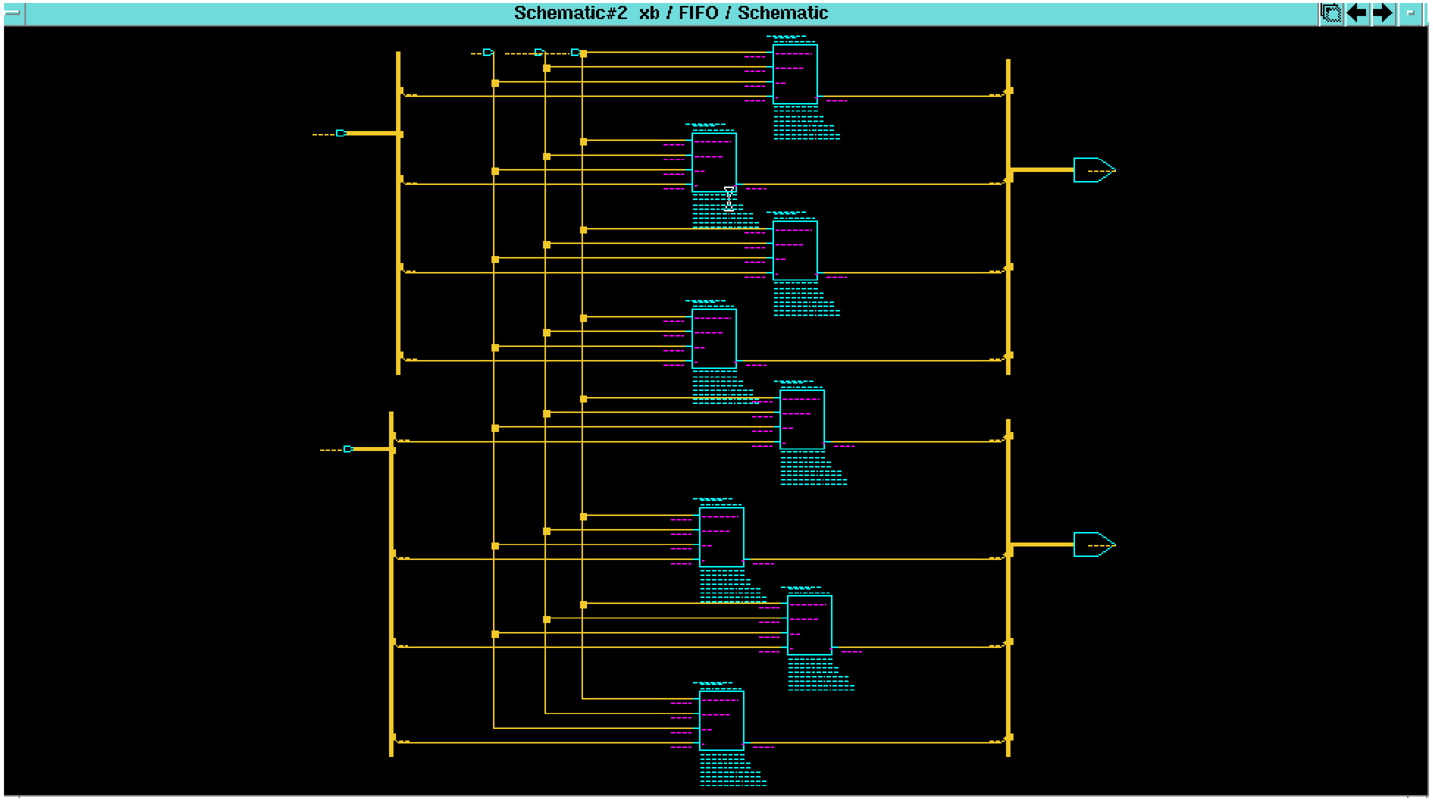
Ripple carry Adder × 1

Ripple carry Subtractor × 1

Signed Multiplier × 4

FIFO Implementation

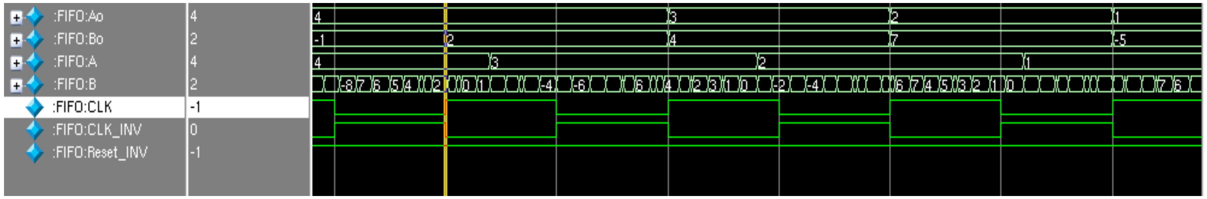
- Schematic

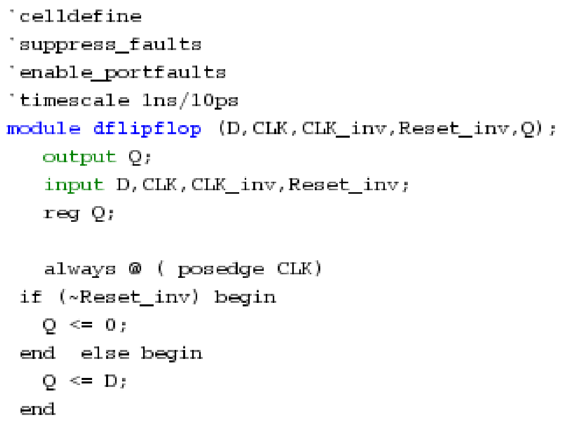


This is the schematic of one FIFO component. It contains 8 Dflipflops with reset and works as a buffer for both input and output.

FIFO Implementation

Digital Simulation





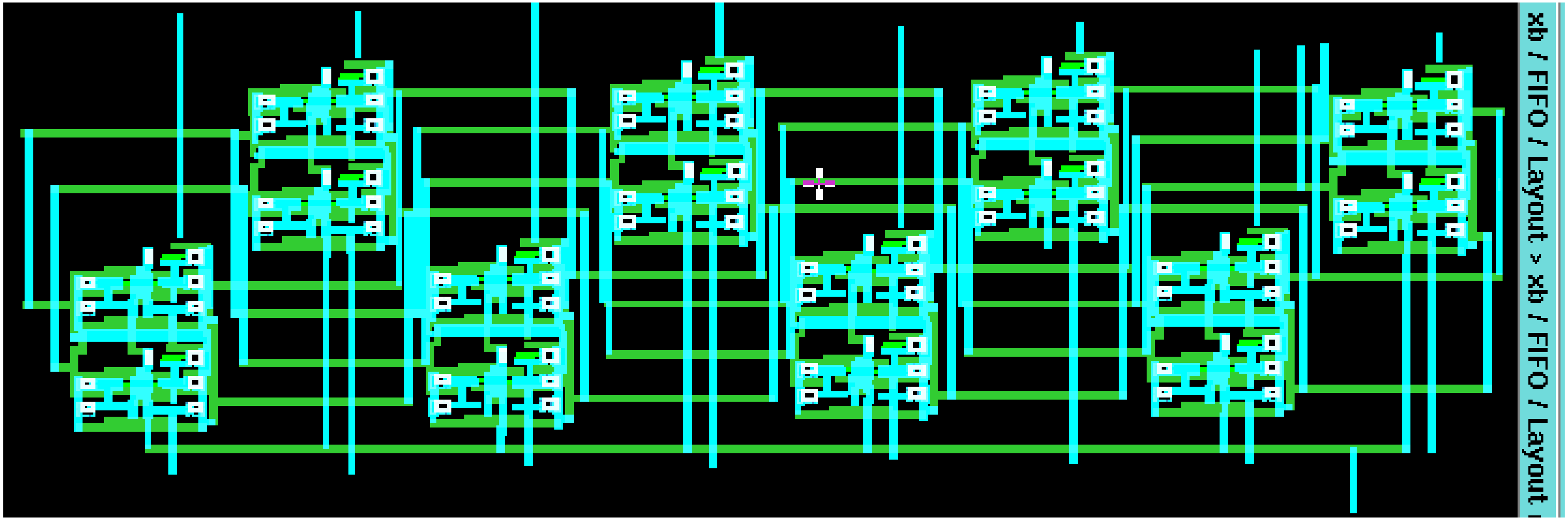
```
`celldefine
`suppress_faults
`enable_portfaults
`timescale 1ns/10ps
module dflipflop (D,CLK,CLK_inv,Reset_inv,Q);
   output Q;
   input D,CLK,CLK_inv,Reset_inv;
   reg Q;

   always @ ( posedge CLK)
 if (~Reset_inv) begin
   Q <= 0;
 end  else begin
   Q <= D;
 end
```

This is the digital simulation result of FIFO component. All possible patterns have been exhaustively tested. To make top level testing easier, we then rewrote the verilog for Dflipflop. The key part of codes is shown above.

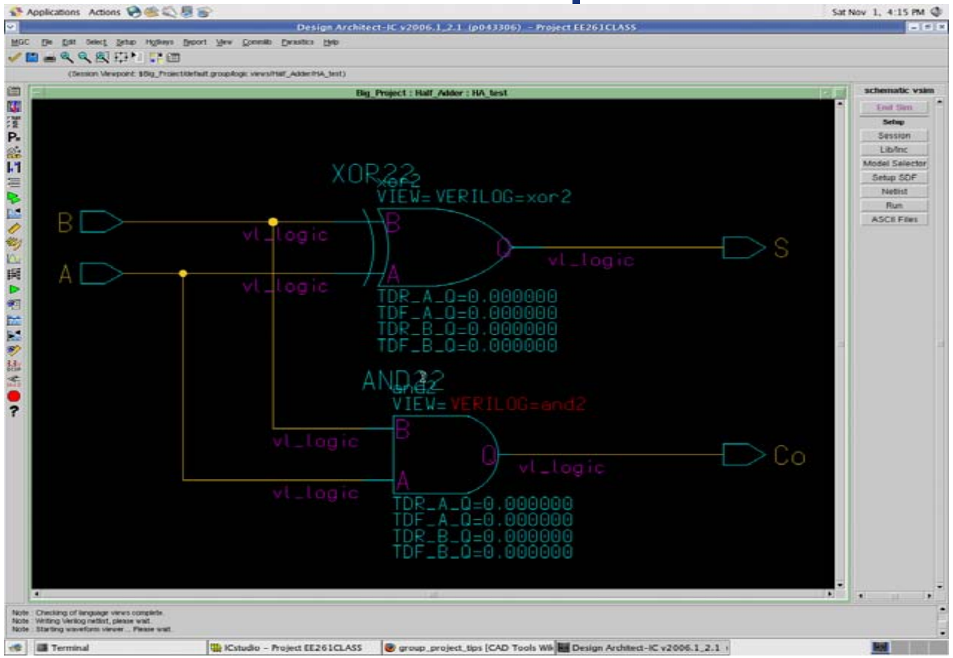
FIFO Implementation

- Layout



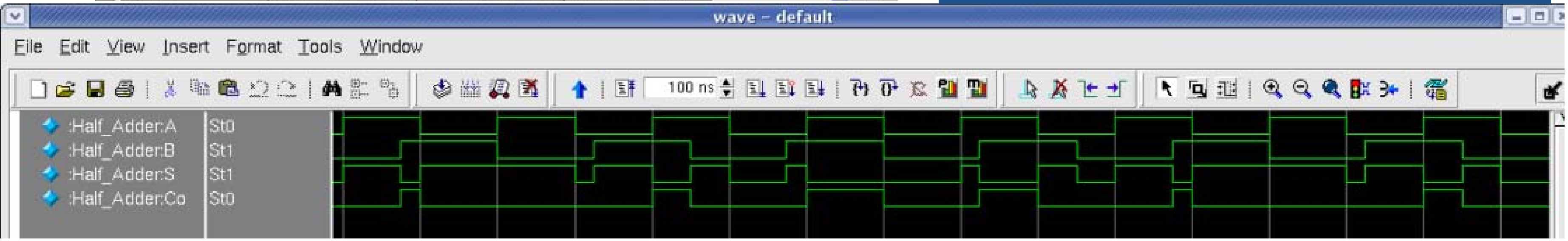
Ripple Carry Adder Implementation

- Halfadder implementation



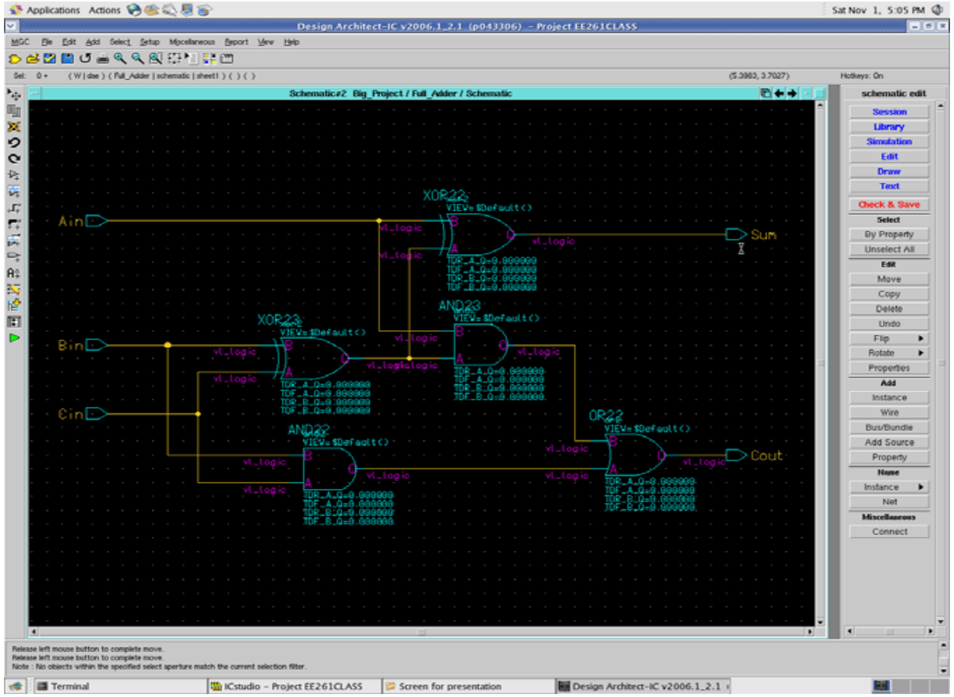
Note : Checking of language views complete.
Note : Writing Verilog netlist, please wait.
Note : Starting waveform viewer ... Please wait.

A	B	S	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



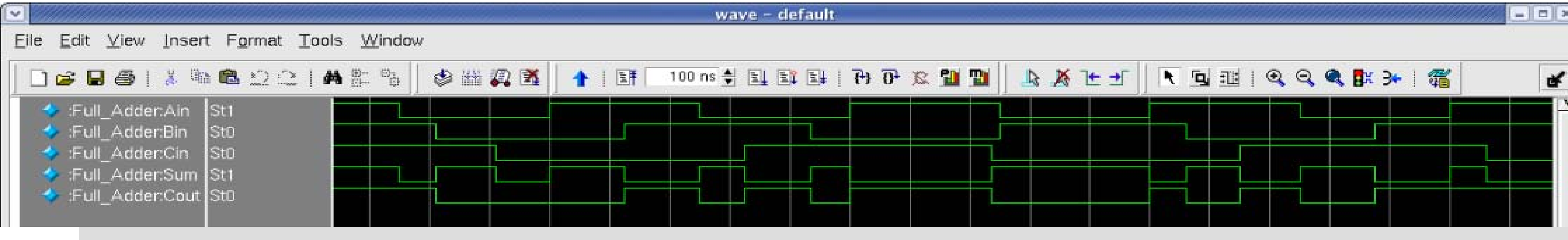
Ripple Carry Adder Implementation

- Fulladder implementation



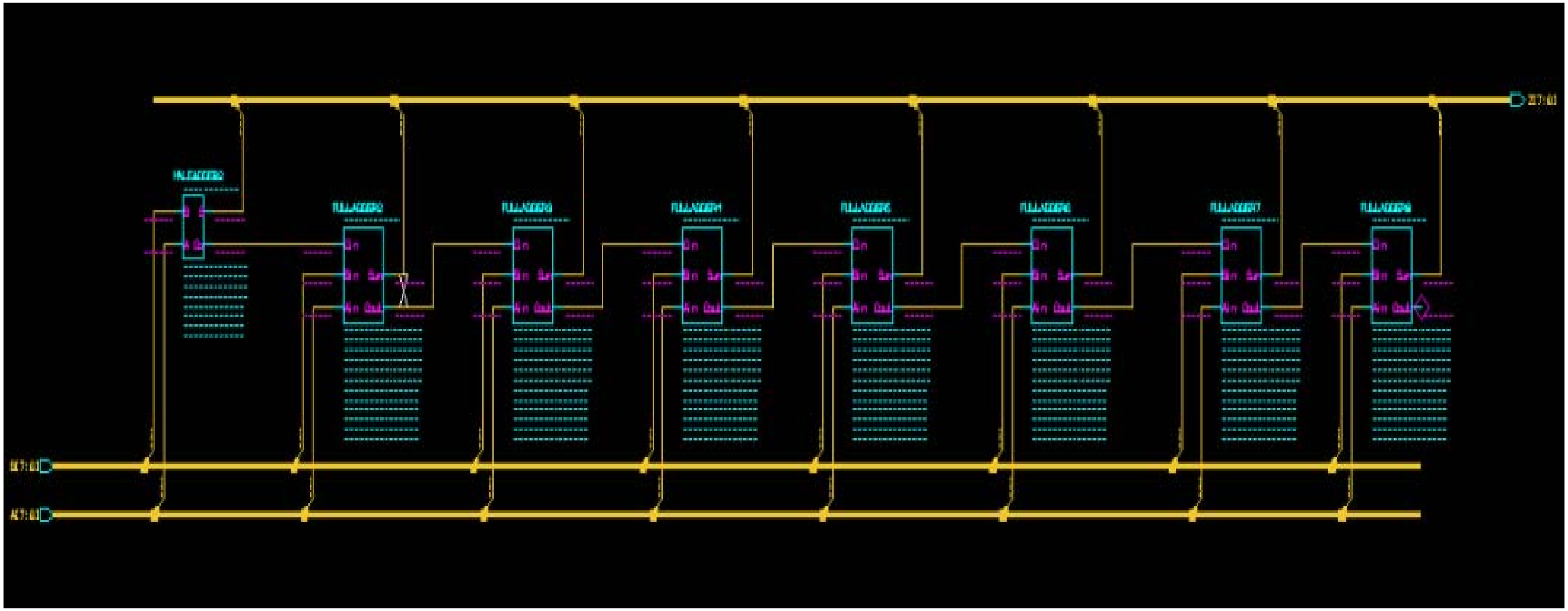
Release left mouse button to complete move.
Release left mouse button to complete move.
Note : No objects within the specified select aperture match the current selection filter.

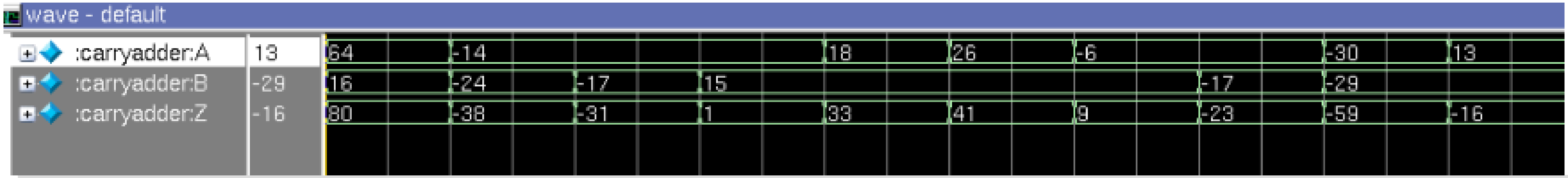
Input			Output	
Ain	Bin	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Ripple Carry Adder Implementation

- 8 bit Ripple Carry Adder implementation

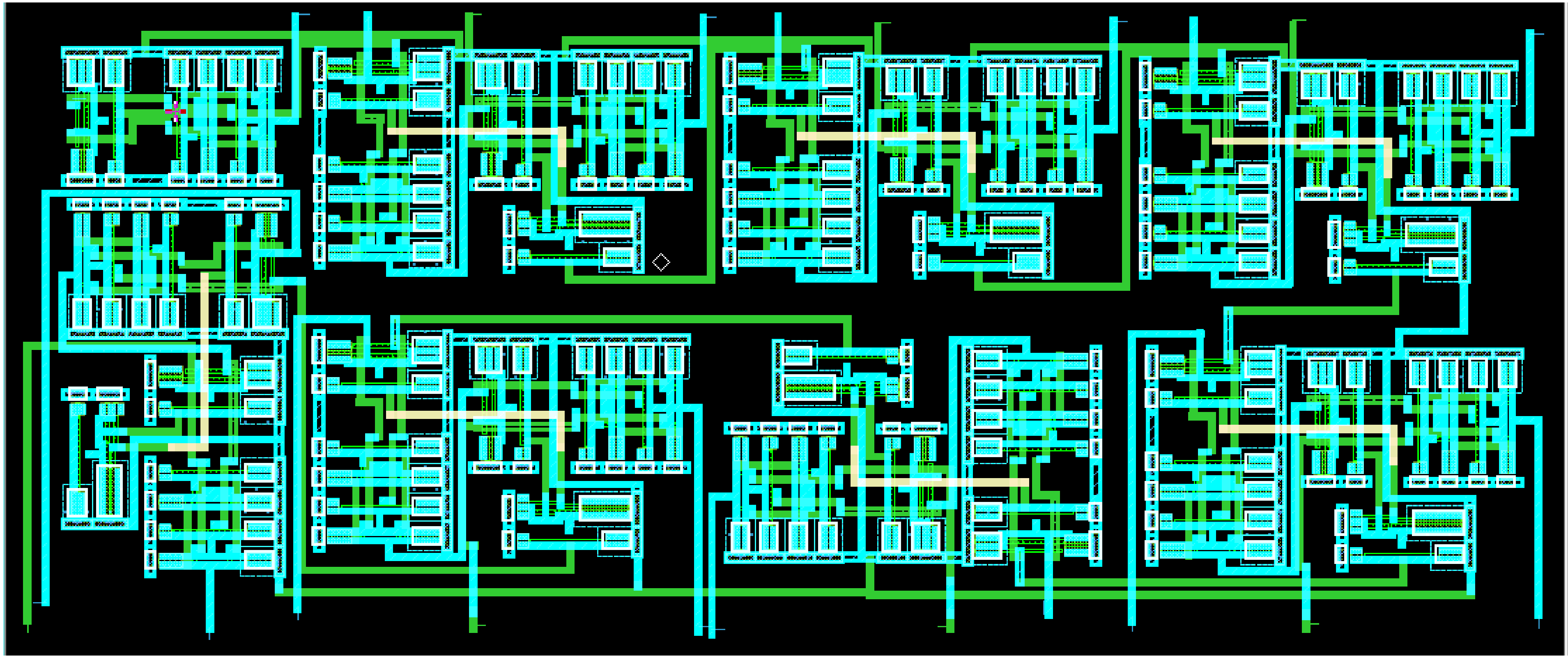




```
:carryadder:A   13    64   -14                       18    26    -6          -30   13
:carryadder:B   -29   16   -24   -17   15                             -17   -29
:carryadder:Z   -16   80   -38   -31   1             33    41    9     -23   -59   -16
```

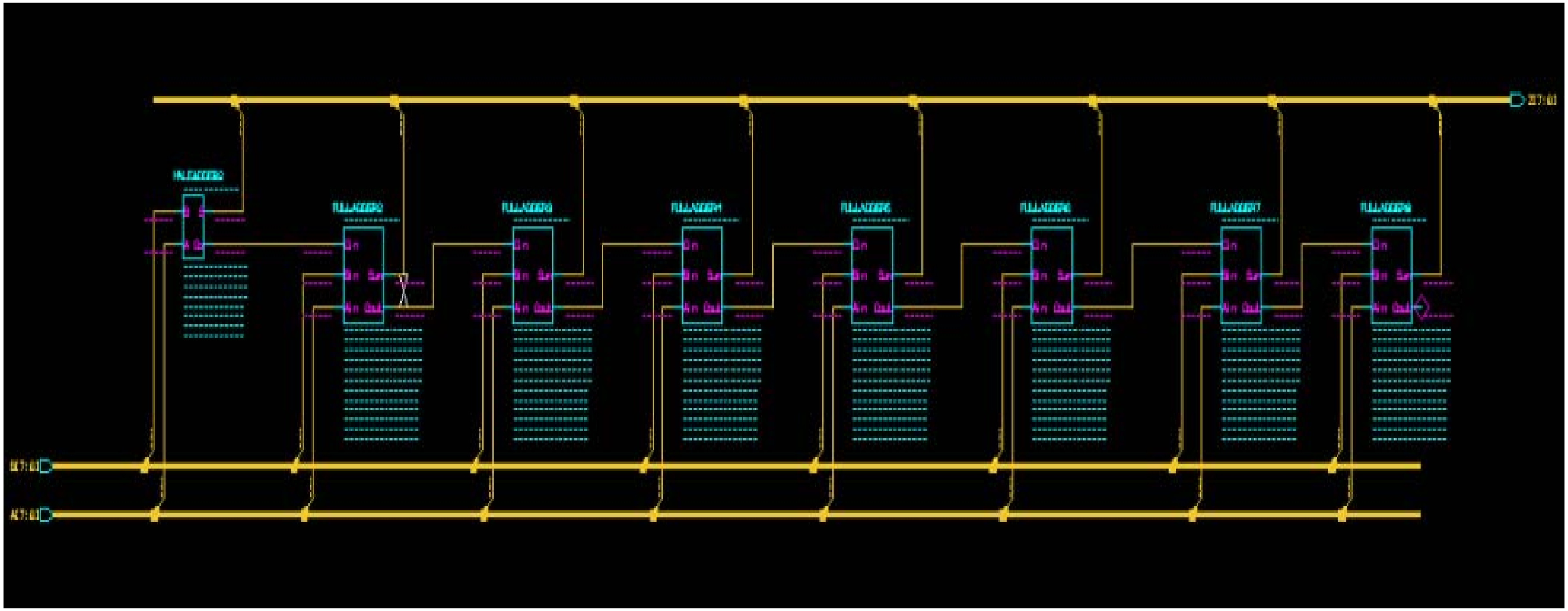
Ripple Carry Adder Implementation

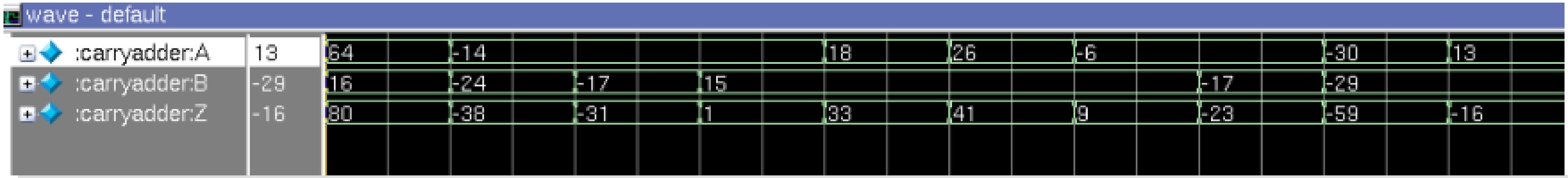
- Layout



Subtractor Implementation

- 8 bit Ripple Carry Subtractor implementation

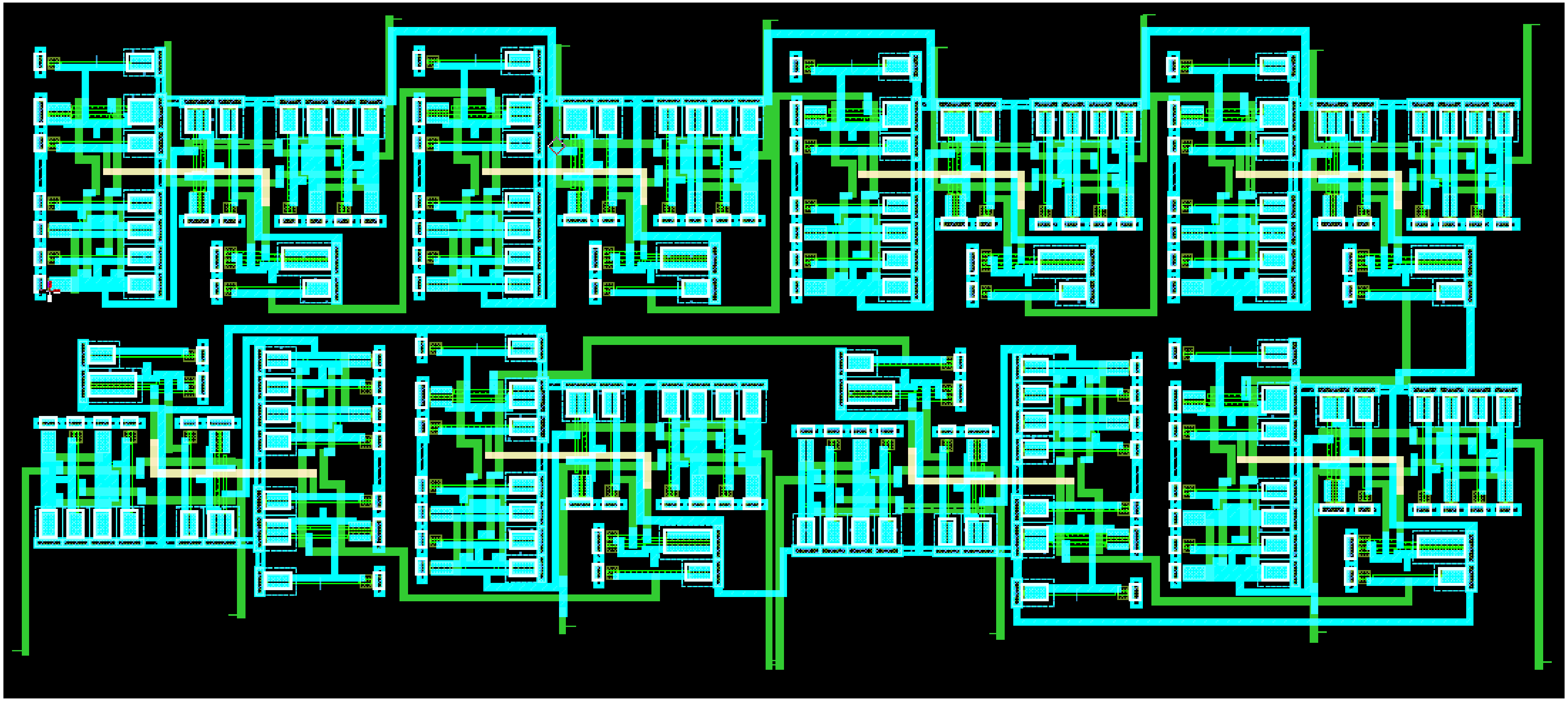




```
:carryadder:A  13   64  -14                       18   26   -6             -30  13
:carryadder:B  -29  16  -24   -17   15                               -17   -29
:carryadder:Z  -16  80  -38   -31   1             33   41   9        -23   -59  -16
```

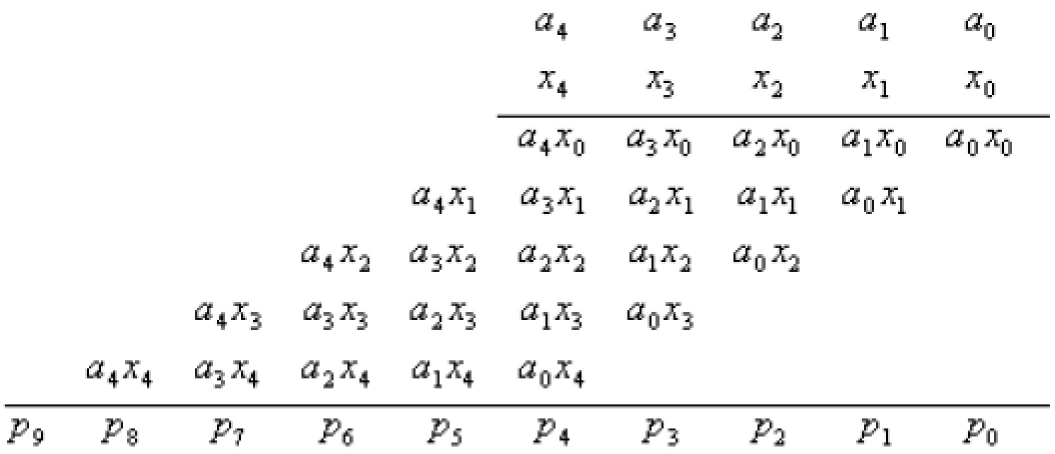
Subtractor Implementation

- Layout



Multiplier Implementation

- Unsigned Multiplication



$$\begin{array}{cccccccccc}
 & & & & & a_4 & a_3 & a_2 & a_1 & a_0 \\
 & & & & & x_4 & x_3 & x_2 & x_1 & x_0 \\
\cline{6-10}
 & & & & & a_4x_0 & a_3x_0 & a_2x_0 & a_1x_0 & a_0x_0 \\
 & & & & a_4x_1 & a_3x_1 & a_2x_1 & a_1x_1 & a_0x_1 & \\
 & & & a_4x_2 & a_3x_2 & a_2x_2 & a_1x_2 & a_0x_2 & & \\
 & & a_4x_3 & a_3x_3 & a_2x_3 & a_1x_3 & a_0x_3 & & & \\
 & a_4x_4 & a_3x_4 & a_2x_4 & a_1x_4 & a_0x_4 & & & & \\
\hline
p_9 & p_8 & p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}$$

Algorithm: shift and addition

Multiplier Implementation

Signed Multiplication

$$
\begin{array}{cccccccccc}
 & & & & & a_4 & a_3 & a_2 & a_1 & a_0 \\
 & & & & & x_4 & x_3 & x_2 & x_1 & x_0 \\
\hline
 & & & & & \overline{a_4x_0} & a_3x_0 & a_2x_0 & a_1x_0 & a_0x_0 \\
 & & & & \overline{a_4x_1} & a_3x_1 & a_2x_1 & a_1x_1 & a_0x_1 & \\
 & & & \overline{a_4x_2} & a_3x_2 & a_2x_2 & a_1x_2 & a_0x_2 & & \\
 & & \overline{a_4x_3} & a_3x_3 & a_2x_3 & a_1x_3 & a_0x_3 & & & \\
 & a_4x_4 & \overline{a_3x_4} & \overline{a_2x_4} & \overline{a_1x_4} & \overline{a_0x_4} & & & & \\
1 & & & & 1 & & & & & \\
\hline
p_9 & p_8 & p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}
$$

How to modify?

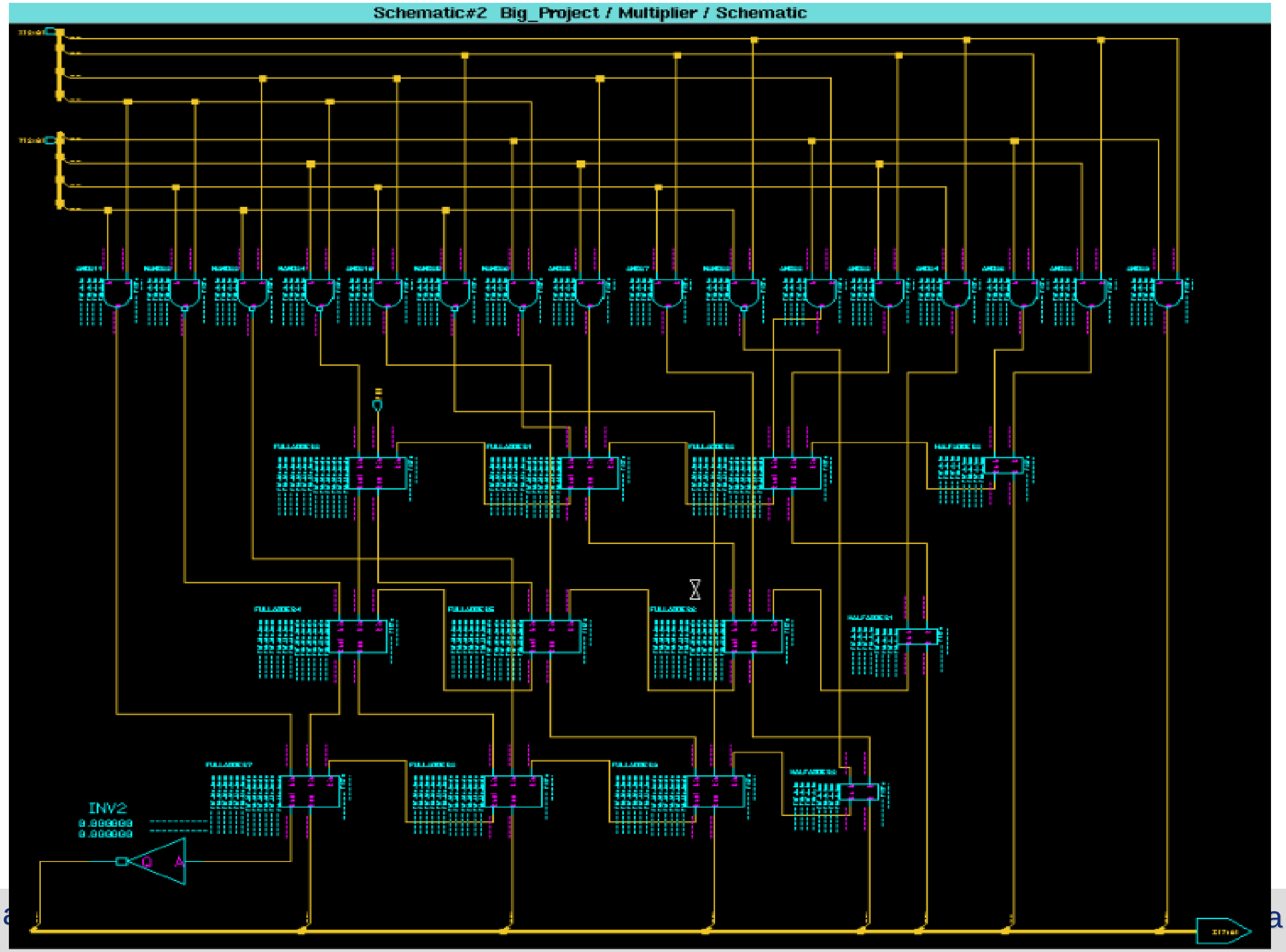
Only need a few more full adders and change some AND2 gates to NAND2 gates.

Algorithm we use: Baugh-Woodley technique

Need some modifications based on unsigned multiplication.

Multiplier Implementation

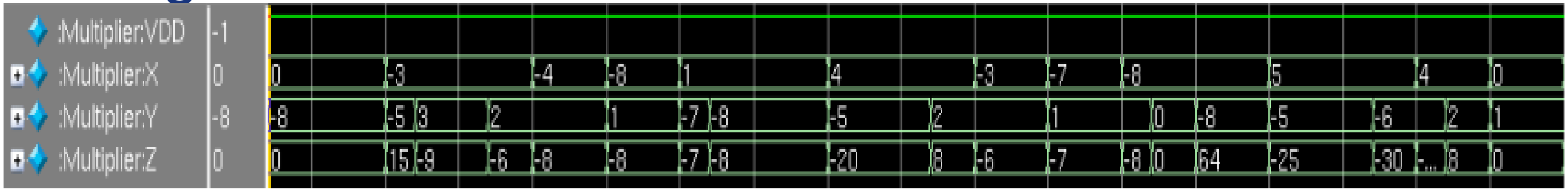
- Schematic



Schematic#2 Big_Project / Multiplier / Schematic

Multiplier Implementation

- Digital simulation



Results: 0*-8=0
-3*-5=15
-3*3=-9
-3*2=-6
-4*2=-8
…

Realizes the multiplication of input X and Y, with the correct outcome Z=XY, which will be sent to Ripple Carry Adder or Ripple Carry Subtractor to calculate:

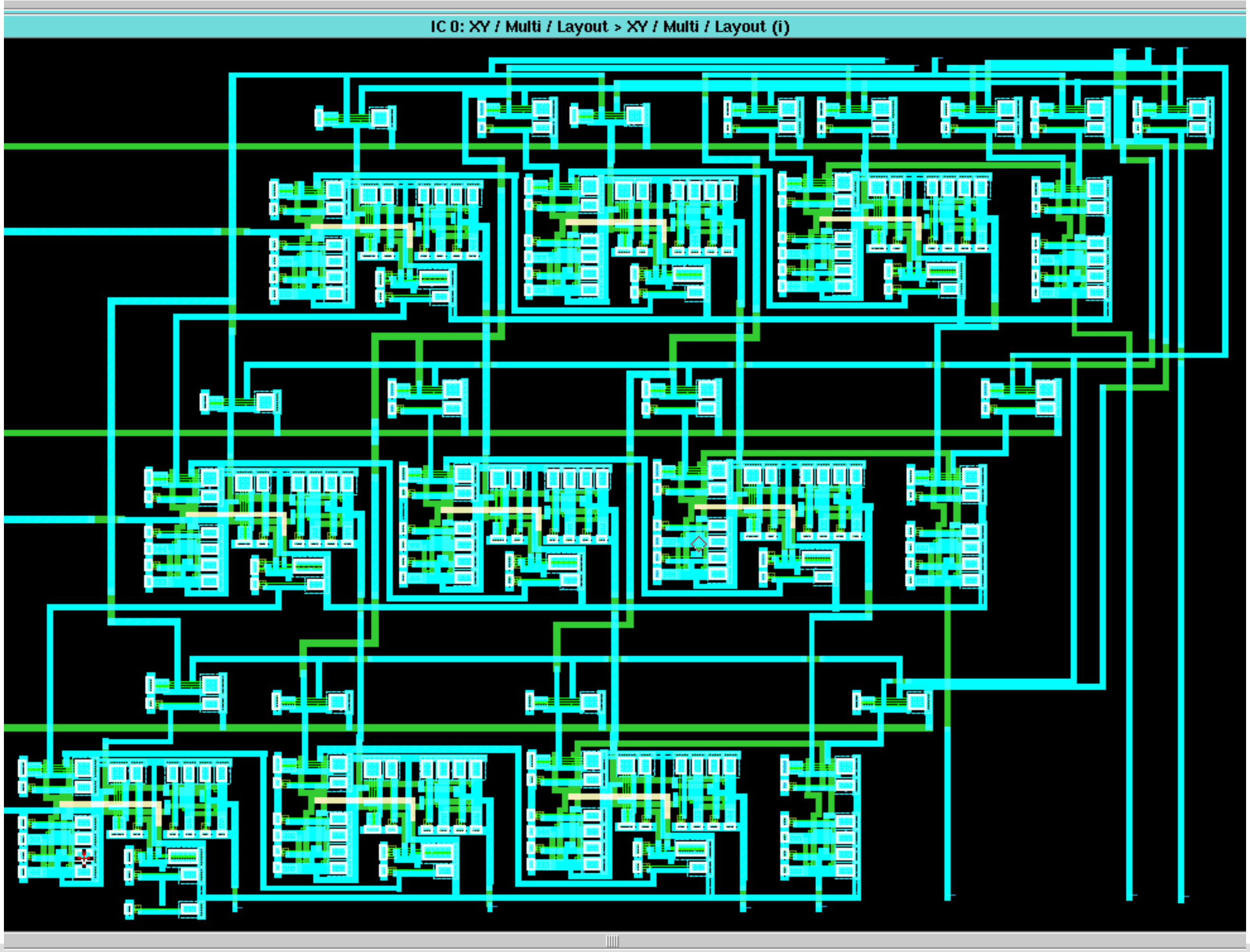
AD+BC & AC-BD

Consists of:

Nand2	× 6
And2	× 10
Full Adders	× 9
Half Adders	× 3
Inv	× 1

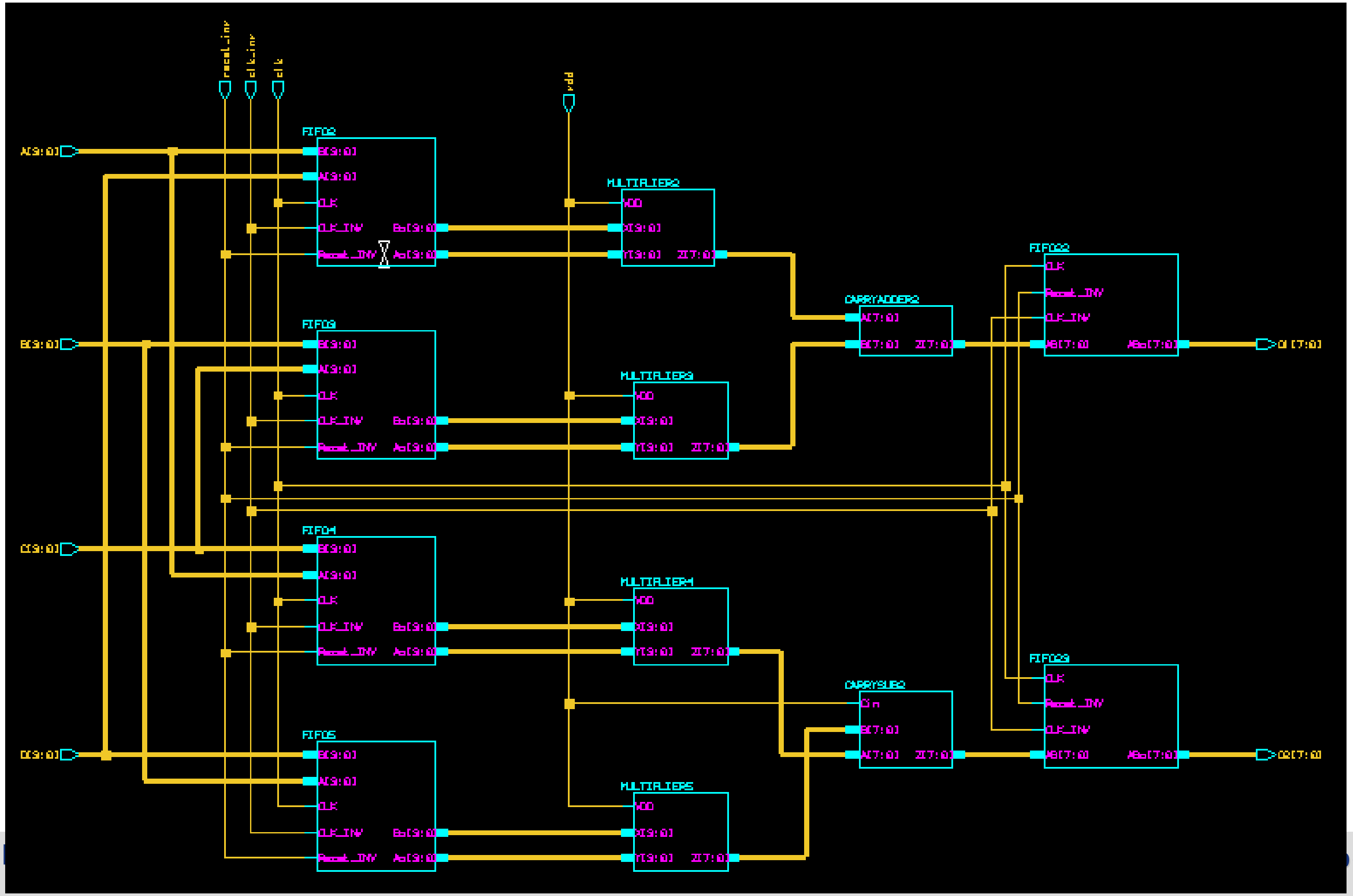
Multiplier Implementation

- Layout



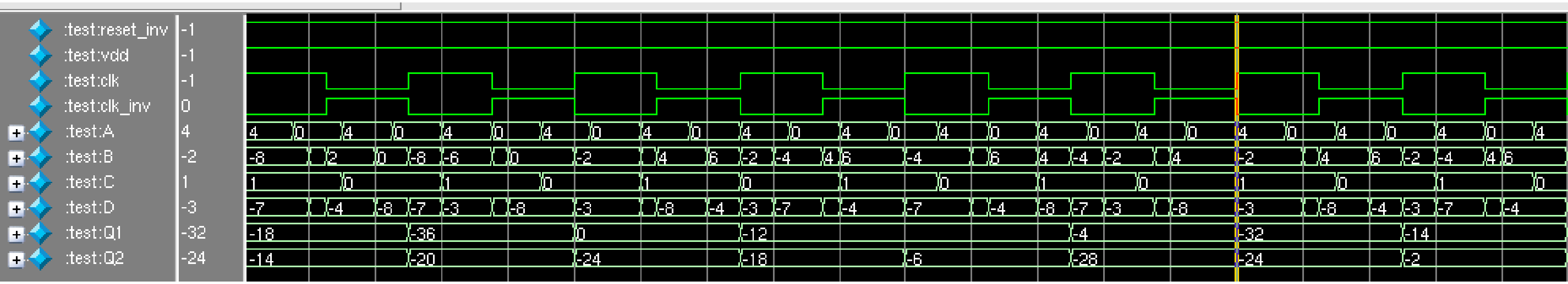
System Implementation

- Schematic

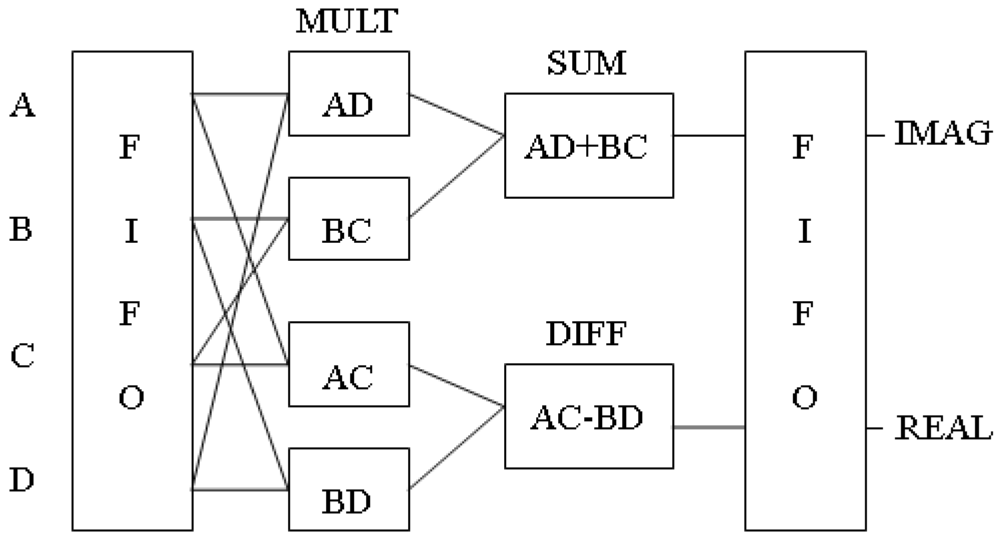


System Implementation

Digital Simulation



:test:reset_inv -1
:test:vdd -1
:test:clk -1
:test:clk_inv 0
:test:A 4
:test:B -2
:test:C 1
:test:D -3
:test:Q1 -32
:test:Q2 -24



(A+Bj)(C+Dj)=(AC-BD)+j(AD+BC)

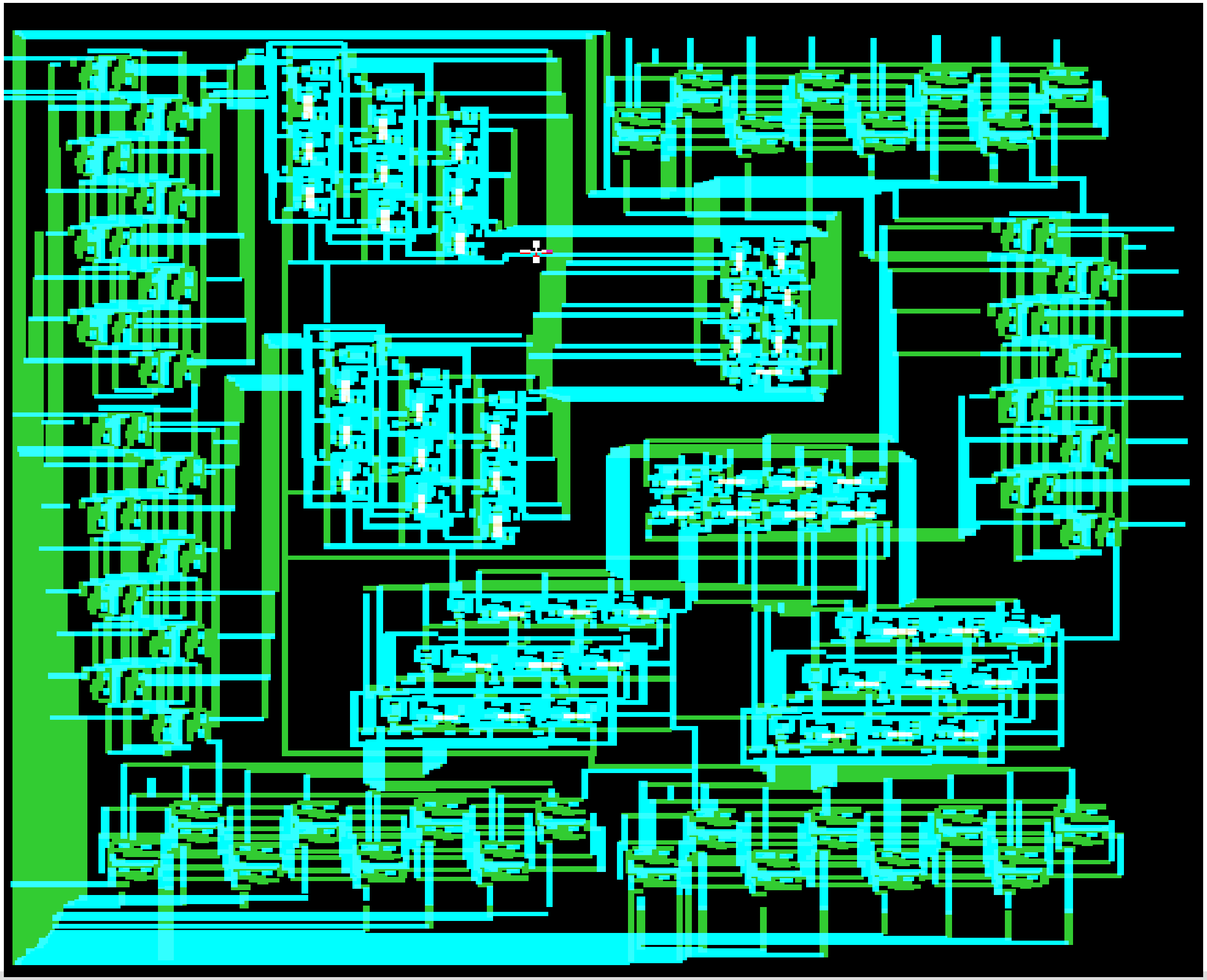
Q1= AD+BC

Q2= AC-BD

The latency between input and output is one clock cycle

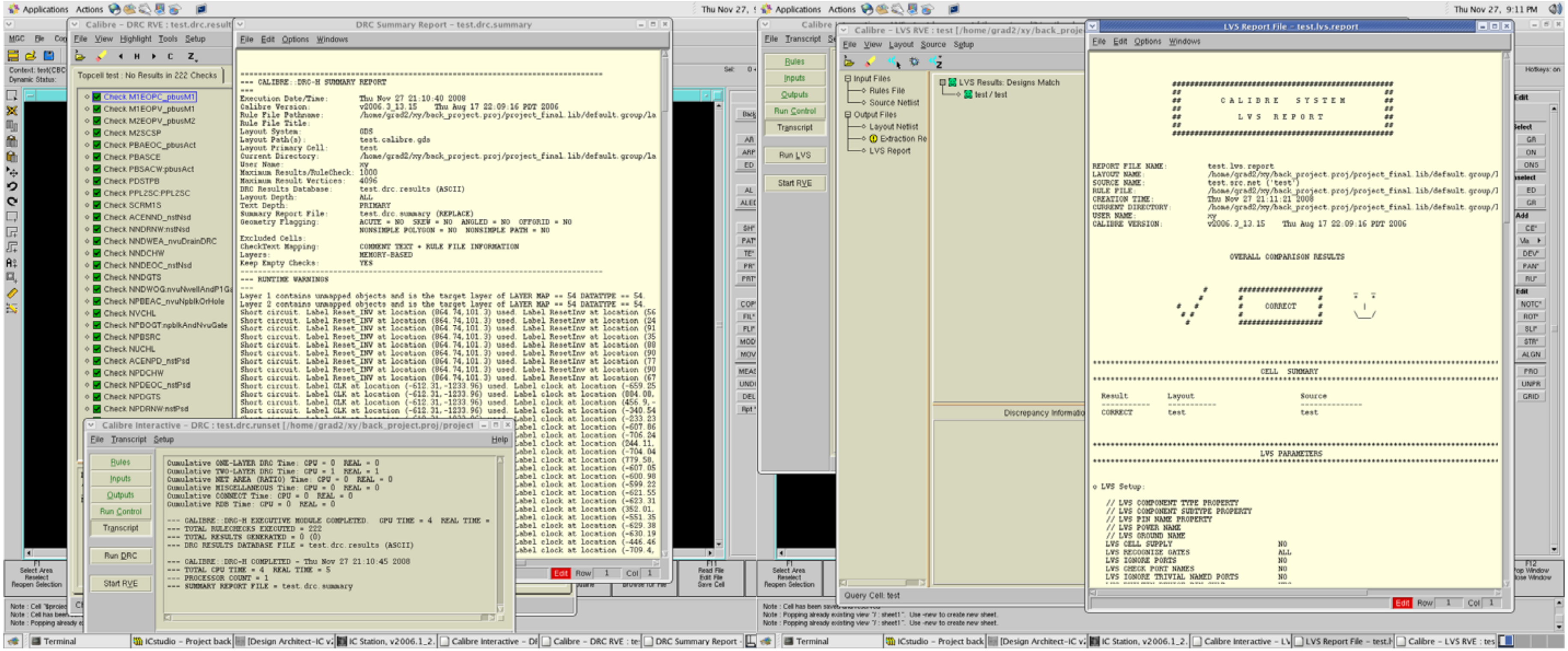
System Implementation

- Layout



System Implementation

- Pass DRC & LVS



```
==================================================================================
--- CALIBRE::DRC-H SUMMARY REPORT
---
Execution Date/Time:        Thu Nov 27 21:10:40 2008
Calibre Version:            v2006.3_13.15    Thu Aug 17 22:09:16 PDT 2006
Rule File Pathname:         /home/grad2/xy/back_project.proj/project_final.lib/default.group/la
Rule File Title:
Layout System:              GDS
Layout Path(s):             test.calibre.gds
Layout Primary Cell:        test
Current Directory:          /home/grad2/xy/back_project.proj/project_final.lib/default.group/la
User Name:                  xy
Maximum Results/RuleCheck:  1000
Maximum Result Vertices:    4096
DRC Results Database:       test.drc.results (ASCII)
Layout Depth:               ALL
Text Depth:                 PRIMARY
Summary Report File:        test.drc.summary (REPLACE)
Geometry Flagging:          ACUTE = NO  SKEW = NO  ANGLED = NO  OFFGRID = NO
                            NONSIMPLE POLYGON = NO  NONSIMPLE PATH = NO
Excluded Cells:
CheckText Mapping:          COMMENT TEXT + RULE FILE INFORMATION
Layers:                     MEMORY-BASED
Keep Empty Checks:          YES
==================================================================================
--- RUNTIME WARNINGS
---
Layer 1 contains unmapped objects and is the target layer of LAYER MAP == 54 DATATYPE == 54.
Layer 2 contains unmapped objects and is the target layer of LAYER MAP == 54 DATATYPE == 54.
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (56
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (24
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (91
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (35
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (88
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (90
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (77
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (90
Short circuit. Label Reset_INV at location (864.74,101.3) used. Label ResetInv at location (67
Short circuit. Label CLK at location (-612.31,-1233.96) used. Label clock at location (-659.25
Short circuit. Label CLK at location (-612.31,-1233.96) used. Label clock at location (884.08,
Short circuit. Label CLK at location (-612.31,-1233.96) used. Label clock at location (456.9,-
Short circuit. Label CLK at location (-612.31,-1233.96) used. Label clock at location (-340.54
                                                                abel clock at location (-233.23
                                                                abel clock at location (-607.86
                                                                abel clock at location (-706.24
                                                                abel clock at location (244.11.
                                                                abel clock at location (-704.04
                                                                abel clock at location (779.58.
                                                                abel clock at location (-607.05
                                                                abel clock at location (-600.98
                                                                abel clock at location (-599.22
                                                                abel clock at location (-621.55
                                                                abel clock at location (-623.31
                                                                abel clock at location (352.01.
                                                                abel clock at location (-551.35
                                                                abel clock at location (-629.38
                                                                abel clock at location (-630.19
                                                                abel clock at location (-446.46
                                                                abel clock at location (-709.4,
```

```
Cumulative ONE-LAYER DRC Time:  CPU = 0  REAL = 0
Cumulative TWO-LAYER DRC Time:  CPU = 1  REAL = 1
Cumulative NET AREA (RATIO) Time:  CPU = 0  REAL = 0
Cumulative MISCELLANEOUS Time:  CPU = 0  REAL = 0
Cumulative CONNECT Time:  CPU = 0  REAL = 0
Cumulative RDB Time:  CPU = 0  REAL = 0

--- CALIBRE::DRC-H EXECUTIVE MODULE COMPLETED.  CPU TIME = 4  REAL TIME =
--- TOTAL RULECHECKS EXECUTED = 222
--- TOTAL RESULTS GENERATED = 0 (0)
--- DRC RESULTS DATABASE FILE = test.drc.results (ASCII)

--- CALIBRE::DRC-H COMPLETED - Thu Nov 27 21:10:45 2008
--- TOTAL CPU TIME = 4  REAL TIME = 5
--- PROCESSOR COUNT = 1
--- SUMMARY REPORT FILE = test.drc.summary
```

```
                  ##################################################################
                  ##                                                              ##
                  ##              C A L I B R E    S Y S T E M                    ##
                  ##                                                              ##
                  ##                  L V S   R E P O R T                         ##
                  ##                                                              ##
                  ##################################################################


REPORT FILE NAME:         test.lvs.report
LAYOUT NAME:              /home/grad2/xy/back_project.proj/project_final.lib/default.group/l
SOURCE NAME:              test.src.net ('test')
RULE FILE:                /home/grad2/xy/back_project.proj/project_final.lib/default.group/l
CREATION TIME:            Thu Nov 27 21:11:21 2008
CURRENT DIRECTORY:        /home/grad2/xy/back_project.proj/project_final.lib/default.group/l
USER NAME:                xy
CALIBRE VERSION:          v2006.3_13.15    Thu Aug 17 22:09:16 PDT 2006


                              OVERALL COMPARISON RESULTS


                        #      ###################       _   _
                       #       #                 #       *   *
                  #   #        #     CORRECT     #         |
                   # #         #                 #       \___/
                    #          ###################


*************************************************************************************
                                   CELL   SUMMARY
*************************************************************************************

  Result          Layout                         Source
  -----------     -----------                    --------------
  CORRECT         test                           test


*************************************************************************************
                                  LVS PARAMETERS
*************************************************************************************


o LVS Setup:

   // LVS COMPONENT TYPE PROPERTY
   // LVS COMPONENT SUBTYPE PROPERTY
   // LVS PIN NAME PROPERTY
   // LVS POWER NAME
   // LVS GROUND NAME
   LVS CELL SUPPLY                          NO
   LVS RECOGNIZE GATES                      ALL
   LVS IGNORE PORTS                         NO
   LVS CHECK PORT NAMES                     NO
   LVS IGNORE TRIVIAL NAMED PORTS           NO
```

Index

1 Brief Introduction

2 Structure Implementation

3 **Feature Evaluation**

Feature Evaluation

- Number of transistors

Components				Total transistors
SUM*1	DIFF*1	MULT*4	FIFO*6	
252	288	434*4	256*6	3812

- Area (mm^2)

2.011*2.416=4.850mm^2

- Power Consumption:

$$
\begin{aligned}
P &= \alpha \cdot C \cdot V^2 \cdot f \\
&= 0.1 \cdot \left[3812 \cdot (12\lambda) \cdot (0.8\mu m / 2\lambda) \cdot (2\, fF / \mu m) \right] \cdot 5^2 \cdot 10^6 \\
&= 0.0915 mW / MHz
\end{aligned}
$$

Feature Evaluation

Power Consumption:

From eldo simulation can get latency in critical path:

FIFO_latency=1.45833ns

Multiplier_latency=4.06742ns

Subtractor_latency=0.933838ns

So, total latency=1.45833+4.06742+0.933838
=6.459588(ns)

Feature Evaluation

Power Consumption:

So, the maximum frequency is:

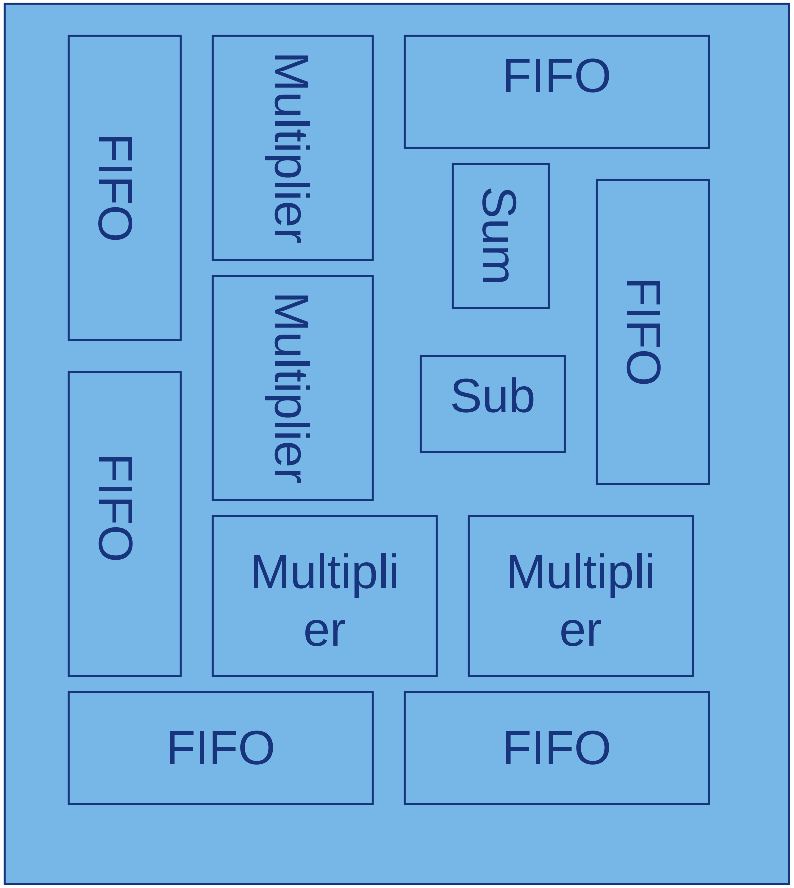
$$f = \frac{1}{total\_latency} = \frac{1}{6.459588 \times 10^{-9}} = 155(MHz)$$

So, power consumption is:

$$\begin{aligned} P &= \alpha \cdot C \cdot V^2 \cdot f \\ &= 0.1 \cdot \left[3812 \cdot \left(12\lambda\right) \cdot \left(0.8\mu m / 2\lambda\right) \cdot \left(2\,fF / \mu m\right)\right] \cdot 5^2 \cdot 10^6 \\ &= 0.0915 mW / MHz \cdot 155 MHz \\ &= 14 mW \end{aligned}$$

Feature Evaluation

- Top-down demonstration of layout. Firstly, the top level. Let's take a look at the floor plan.


FIFO
Multiplier
FIFO
Sum
FIFO
Multiplier
Sub
FIFO
Multiplier
Multiplier
FIFO
FIFO