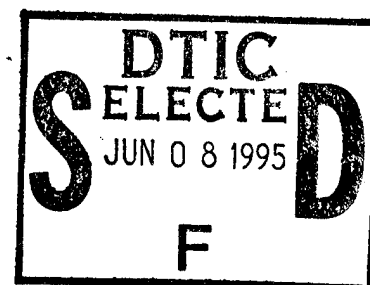


NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



*Original contains color
plates: All DTIC reproductions
will be in black and
white*

THESIS

A CMOS, VLSI, IMPLEMENTATION OF A FFT FOR CYCLIC SPECTRAL ANALYSIS

by

Kevin L. Jackson

March 1995

Thesis Advisor:

Herschel H. Loomis, Jr.

Co-Advisors:

Raymond F. Bernstein, Jr.

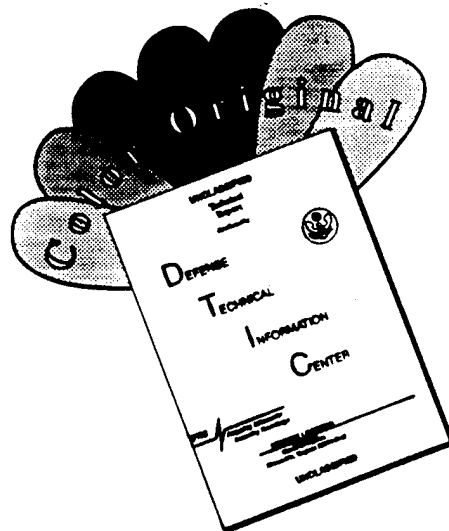
Douglas J. Fouts

Approved for public release; distribution is unlimited

19950607 011

DTIC QUALITY INSPECTED 8

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 1995		3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE A CMOS, VLSI, Implementation of a FFT for Cyclic Spectral Analysis				5. FUNDING NUMBERS	
6. AUTHOR(S) Kevin L. Jackson					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.					
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Cyclic Spectrum Analysis exploits the cyclostationary properties of signals and systems. Military use of this technology is focused on its use in a near real time analytical environment. Such a system requires high speed arithmetic processing in calculating large Fourier transforms quickly. This thesis reviews a previous implementation and then presents a new design using Verilog hardware description language and the Epoch silicon compiler. Using these modern computer aided design tools, the ASIC design was simulated and layout completed using a one micron, two-metal process rule set. The final layout consists of 434,138 transistors on a 11,190 x 15,642 micron die. Simulations indicated that the chip would be capable of operating at a 25 Mhz clock rate while dissipating .8 watts of power. Embedded timing analysis tools displayed all critical timing paths which allowed the identification of specific design improvements. If implemented, these changes could double the clock rate of the processor.					
14. SUBJECT TERMS Digital Arithmetic; FFT Butterfly Design; Cyclic Spectrum Analysis; Verilog; Hardware Description Languages; Epoch Silicon Compiler; ASIC Design; CMOS Design				15. NUMBER OF PAGES 336	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL		

Approved for public release; distribution unlimited.

A CMOS, VLSI, Implementation of a
FFT for Cyclic Spectral Analysis

by

Kevin L. Jackson
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1979

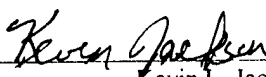
Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

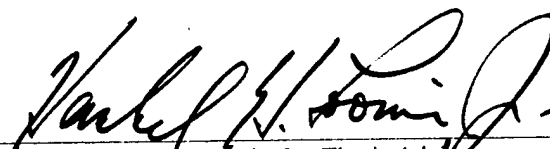
from the

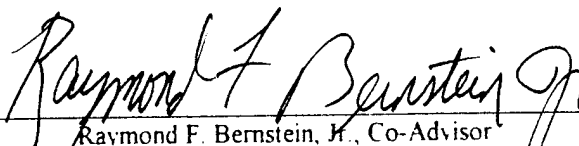
NAVAL POSTGRADUATE SCHOOL
March 1995

Author:


Kevin L. Jackson

Approved by:


Herschel H. Loomis, Jr., Thesis Advisor


Raymond F. Bernstein, Jr., Co-Advisor


Douglas J. Fouts, Co-Advisor


Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

Cyclic Spectrum Analysis exploits the cyclostationary properties of signals and systems. Military use of this technology is focused on its use in a near real time analytical environment. Such a system requires high speed arithmetic processing in calculating large Fourier transforms quickly. This thesis reviews a previous implementation and then presents a new design using the Verilog hardware description language and the Epoch silicon compiler. Using these modern computer aided design tools, the ASIC design was simulated and layout completed using a 1 μ m, two-metal process rule set. The final layout consists of 434,138 transistors on a 11,190 x 15,642 μ m die. Simulations indicated that the chip would be capable of operating at a 25 Mhz clock rate while dissipating .8 watts of power. Embedded timing analysis tools displayed all critical timing paths which allowed the identification of specific design improvements. If implemented, these changes could double the clock rate of the processor.

Accession For		
NTIS	CRA&I	☒
DTIC	TAB	☐
Unannounced		☐
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and/or Special	
A-1		

TABLE OF CONTENTS

I. INTRODUCTION	1
A. CYCLIC SPECTRUM ANALYSIS	1
B. GENERAL BUTTERFLY MACHINE	1
C. THESIS OVERVIEW	2
II. CYCLIC SPECTRAL ANALYZER REQUIREMENTS	4
A. FUNCTIONAL REQUIREMENTS	4
B. ALGORITHM DESCRIPTIONS	5
C. FFT IMPLEMENTATION	9
III. CSA PROCESSOR DESIGN	10
A. TOP LEVEL PROCESSOR DESIGN	10
B. SINGLE PROCESSOR PROGRAMMING	10
C. FFT CHIP - ZIMMER DESIGN	12
D. FFT CHIP - NEW DESIGN	14
IV. CHIP DESIGN AND SIMULATION	15
A. VERILOG HDL	15
B. EPOCH SILICON COMPILER	16
C. VERILOG - EPOCH INTERFACE	17
D. ASIC DESIGN PROCESS	18
E. CHIP HDL DESIGN	18
F. CHIP SIMULATION	32
V. CHIP FLOORPLANNING AND TIMING ANALYSIS	34
A. EPOCH FLOORPLANNING	34
B. TACTIC TIMING ANALYZER	35
C. FINAL CHIP DESIGN	37
D. CHIP TESTING	37
VI. CONCLUSIONS AND RECOMMENDATIONS	38
APPENDIX A SPECIFICATION SUMMARY	40
APPENDIX B VERILOG HDL CODE	48
APPENDIX C EPOCH VERILOG HDL COMPILATION	97
APPENDIX D EPOCH AUTOCOMPILATION	152
APPENDIX E FLOATING POINT ADDER TEST OUTPUT	248
APPENDIX F FLOATING POINT MULTIPLIER TEST OUTPUT	275
APPENDIX G LAYOUT PLATES	300
APPENDIX H PART SPECIFICATIONS (SELECTED)	306
LIST OF REFERENCES	325

INITIAL DISTRIBUTION LIST	326

LIST OF TABLES

TABLE 1 - VERILOG MODEL TYPES	15
TABLE 2 - ROUNDING MODES	22
TABLE 3 - FLOATING POINT MULTIPLIER SHIFT OPTIONS	23

LIST OF FIGURES

Figure 1 - Estimating The Time Smoothed Cyclic Periodogram	5
Figure 2 - Time Smoothed Cyclic Cross Periodogram Implementation	6
Figure 3 - Time Algorithm With FFT Output.....	6
Figure 4 - FAM Tiling Of The Bifrequency Plane	7
Figure 5 - SSCA Tiling Of The Bifrequency Plane.....	7
Figure 6 - Cyclostationary Processing Dataflow	10
Figure 7 - General Butterfly Machine.....	11
Figure 8 - FFT Chip.....	12
Figure 9 - FFT Arithmetic Unit - Zimmer Chip Design.....	13
Figure 10 - Epoch Design Flow.....	17
Figure 11 - Design Hierarchy	19
Figure 12 - Floating Point Adder Stage 1	24
Figure 13 - Floating Point Adder Stage 2.....	25
Figure 14 - Floating Point Adder Stage 3.....	26
Figure 15 - Floating Point Adder Stage 4.....	26
Figure 16 - Floating Point Adder Stage 5.....	27
Figure 17 - Floating Point Adder Stage 6.....	28
Figure 18 - Floating Point Multiplier Stage 1	29
Figure 19 - Floating Point Multiplier Stage 2-4.....	30
Figure 20 - Floating Point Multiplier Stage 5.....	31
Figure 21 - Floating Point Multiplier Stage 6.....	32
Figure 22 - Floating Point Multiplier Stage 7.....	32
Figure 23 - Floorplanning.....	35
Figure 24 - TACTIC Timing Graphical Display.....	35
Figure 25 - Single Timing Path Display	36
Figure 26 - Clock Buffer Tree	36

I. INTRODUCTION

A. CYCLIC SPECTRUM ANALYSIS

Cyclic Spectrum Analysis (CSA) evaluates the cyclostationary properties of signals and systems. This form of analysis addresses statistical variations of a signal or system that is rhythmic, seasonal, or otherwise cyclic in nature. Examples include meteorology time-series data, sinusoidal communications signals and periodic scanning radar emissions. For military applications, near real-time cyclostationary analysis could be used for obtaining operational and tactical intelligence. Although CSA is based on the well-known techniques of conventional spectrum analysis, computational speed requirements for near real time analysis are daunting. The number of computations associated with Fourier transformation, convolutions, and product modulations required in CSA far exceeds the capability of a general purpose computer to accomplish in real time. This fact makes the use of an application specific integrated circuit (ASIC) desirable. This thesis will focus on the design of an ASIC capable of performing complex number multiplication and four point FFT's in a near real time analysis environment.

B. GENERAL BUTTERFLY MACHINE

The central unit in the envisioned CSA processor is to be capable of performing a single operation on an input stream of sampled data. Ideally, this butterfly machine could selectively complete vector multiplies, vector additions, or four point FFT computations. To enable fast processing, the processor is designed as a pipeline, allowing production of one data element each clock cycle after the time for latency had elapsed. The pipelined data consists of digitized signal samples represented by complex floating point values. To ensure accuracy and standardization, an IEEE 32-bit single floating point standard representation is desired.

Previous work resulted in a ASIC design based on the Genesil Silicon Compiler and Logic-Compiler. [Ref. 1] This effort, however, did not conform with IEEE standards for floating point calculations. Ordinarily, even with this shortcoming, previous work is an adequate basis for future designs. Unfortunately, the Genesil Silicon Compiler used is no longer available at NPS nor is it commercially supported. This fact forced a complete redesign of the ASIC. The original architecture however is preserved.

The redesigned integrated circuit will consist of a complex floating point multiplier and adder capable of performing 32-bit, IEEE standard, radix-4 fast Fourier transform butterfly calculations. Verilog hardware design language (HDL) was selected for use in this design because of its prevalence in the commercial ASIC market. Use of a HDL gives future efforts the ability to modify the existing design and port it to other design and analysis tools. The use of a HDL is typical of leading edge VLSI design. Epoch, a graphical design tool and silicon compiler developed by Cascade Automated Design, was used for layout placement, routing, and timing analysis. This tool was chosen because of its compatibility with the Verilog HDL, available pre-defined MSI logic modules, and the software's ability to recompile designs using different rule sets. This last capability allows the easy translation of designs to new CMOS processes. The goal of this thesis is the design of an ASIC capable of performing those operations necessary for conducting CSA. Additionally, data will be generated to allow design of a controller for the chip and fabrication of the design.

C. THESIS OVERVIEW

This thesis presents a top level layout for a Cyclic Spectral Analyzer, discusses a possible design for a General Butterfly Machine (GBM) and present the detailed design for the FFT butterfly unit. Chapter II will define CSA functional requirements and algorithm implementation. Chapter III will present the top level layout and GBM design. It will also detail previous work on the FFT unit and outline the new design. After familiarizing the reader with the Verilog and Epoch tools used, Chapter IV will present the

design details of the new chip. This will include a discussion of the simulations conducted. Chapter V will discuss floorplanning, timing analysis and power calculations. That chapter will also outline a strategy for testing the completed chip after fabrication. Conclusions and recommendations for future research are given in Chapter VI.

II. CYCLIC SPECTRAL ANALYZER REQUIREMENTS

A. FUNCTIONAL REQUIREMENTS

Cyclostationary waveforms have statistical parameters that vary periodically with time. The spectral correlation function (SCF) provides a second order, frequency-domain, statistical description of such a waveform. Solutions to the problem of detecting the presence of cyclostationary signals buried in noise and interference relies on an estimation of the SCF. By interpreting this estimate as a cross spectrum, spectral analysis principles can be applied [Ref. 2]. Factors such as time and frequency resolution and smoothing are critical to effective analysis.

In many military applications, information derived from cycle frequency analysis can lead to operational and tactical intelligence. Since the specific cycle frequencies of interest are not known beforehand, the SCF must be estimated for a range of cycle frequencies. Conventional cross spectral analysis methods provide a frequency resolution of $1/\Delta t$, where Δt is the length of the observation interval. Complete characterization of a spectral observation requires a discrete set of spectral frequencies $f_k = k\Delta f$, where Δf is the frequency resolution of the analyzer and cycle frequencies $\alpha_k = k\Delta\alpha$ over the frequency regions $0 \leq f_k \leq \frac{1}{2}$. The requirement for statistical reliability and cycle frequency spacing lead to $\frac{\Delta t}{4\Delta f}$ individual estimates being required for complete analysis. Since the computational cost of each estimate also increases linearly with Δt , the cost for complete analysis using conventional methods is proportional to Δt^2 . Two proposed algorithms, the FFT accumulation method (FAM) and the strip spectral correlation analyzer (SSCA), reduce this dependency to $\Delta t \log \Delta t$, however, this is still higher than the capability of most general purpose analyzers [Ref 2].

B. ALGORITHM DESCRIPTIONS

Both FAM and SSCA are based on the time smoothed cyclic cross periodogram [Ref. 3]

$$S_{xy_T}^{\alpha}(n, f)_{\Delta t} = \frac{1}{T} \left\langle X_T(n, f + \alpha/2) Y_T^*(n, f - \alpha/2) \right\rangle_{\Delta t} \quad (1)$$

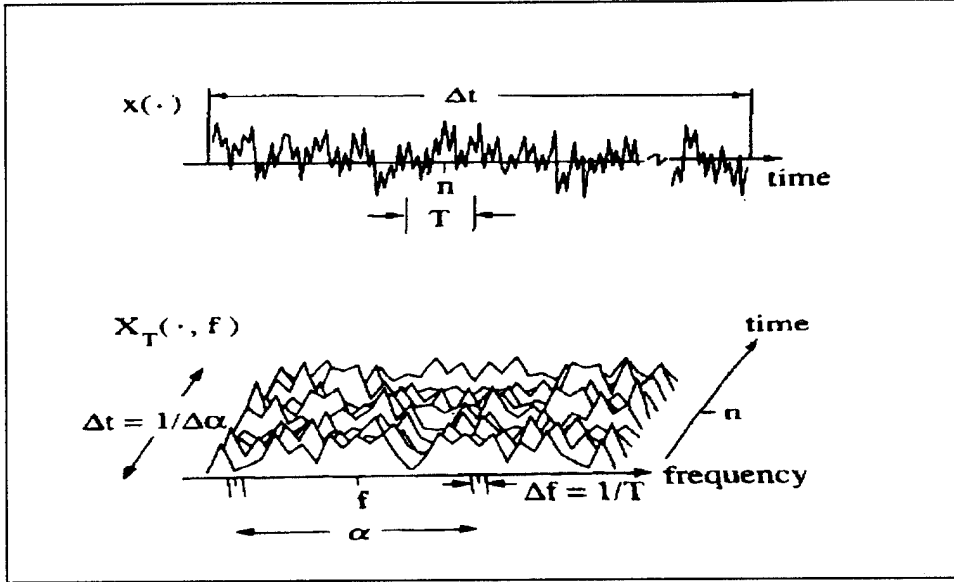


Figure 1 - Estimating the time smoothed cyclic periodogram [Ref. 4]

The periodogram indicates the correlation of spectral components of $x(n)$ with spectral components of $y(n)$ over a time span of Δt seconds. The spectral components $X_T(n, f + \alpha/2)$ and $Y_T(n, f - \alpha/2)$ are the complex envelopes of narrowband, bandpass components of a signal [Ref. 4, p.40]. As Figure 1 illustrates, a data tapering window of length T seconds slides over the data for a time span of Δt seconds. The complex demodulates of the data within the window are computed and used to estimate the cyclic spectrum at the point (f_0, α_0) . An implementation of the time smoothed cyclic cross periodogram is shown in Figure 2. L represents the decimation parameter to be defined later. The data tapering window, $g(n)$, has a width $\Delta t = NT_s$. The correlation operation is expressed as

$$S_{xy_T}^{\alpha_0}(n, f_0)_{\Delta t} = \sum_r X_T(r, f_1) Y_T^*(r, f_2) g(n - r) \quad (2)$$

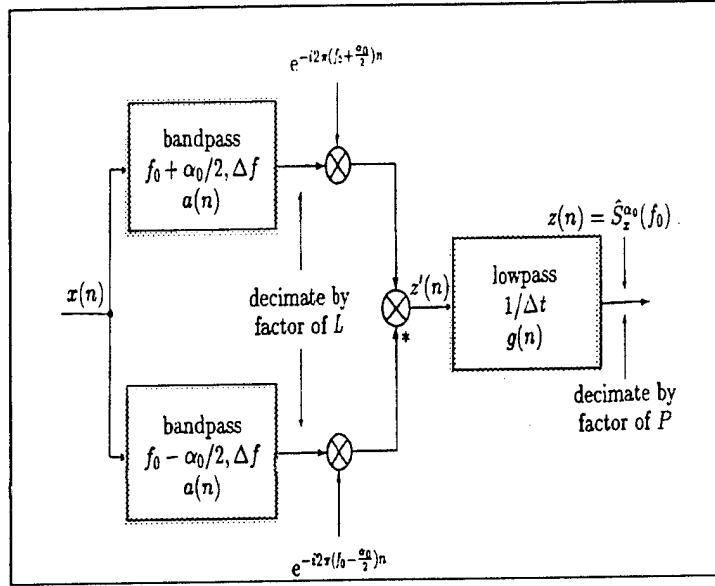


Figure 2 - Time smoothed cyclic cross periodogram implementation [Ref 4, p. 40]

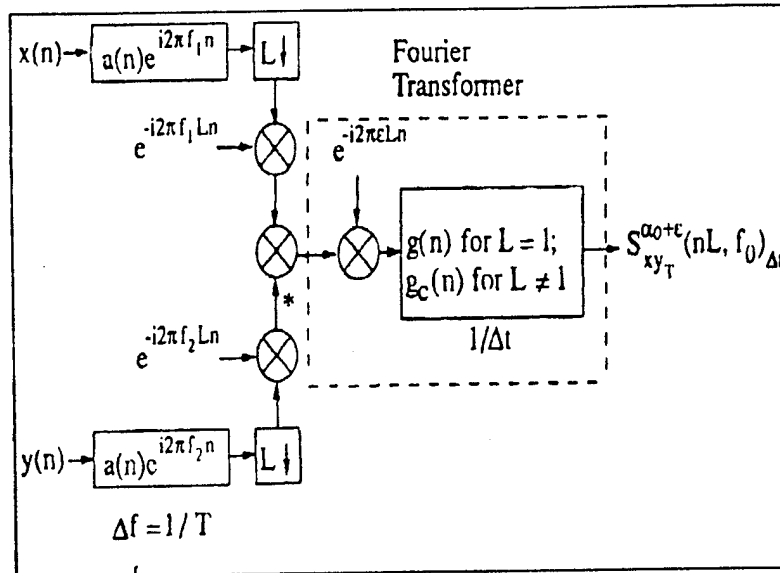


Figure 3 - Time algorithm with FFT output

This implementation [Ref. 4, pp 40-41] leads to definitions for the CSA cell within a bifrequency plane (Figure 2). Computational efficiency of the basic time smoothing algorithm can be improved by decimating the outputs. This can be done by shifting input data into the correlation multiplier and filters in blocks of L samples where $L < N$. This reduces the overall complexity of the algorithm by the factor L .

If the product sequence in Figure 2 is shifted by an amount ϵ from α_0 to $\alpha_0 + \epsilon$, the output sum can be evaluated with an N -point FFT (Figure 3) [Ref. 4, p 44].

FAM is a FFT based time smoothing algorithm. This method channelizes the data with an N' -point FFT that is hopped over the data in blocks of L samples. After the complex demodulates are computed, product sequences are formed and Fourier Transformed with a P -point FFT. The algorithm allows a highly paralleled implementation leading to a fast and efficient microprocessor design. The output can be expressed mathematically as

$$S_{xy_T}^{\alpha_i + q\Delta\alpha}(nL, f_j)_{\Delta t} = \sum_r X_T(rL, f_k) Y_T^*(rL, f_l) g_c(n-r) e^{-i2\pi r q/P} \quad (3)$$

and shown graphically in a tiling of the bifrequency plane (Figure 4). The SSCA directly multiplies the complex demodulates $X_T(n, f_k)$ by $y^*(n)$. This produces point estimates along a frequency-skewed family of lines $\alpha = 2f_k - 2f$ (Figure 5). This process is expressed mathematically by

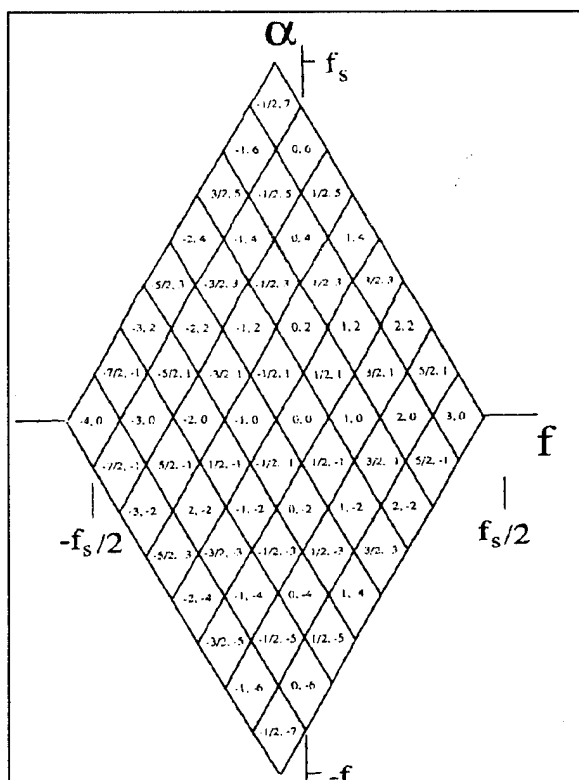


Figure 4 - FAM tiling of the bifrequency plane

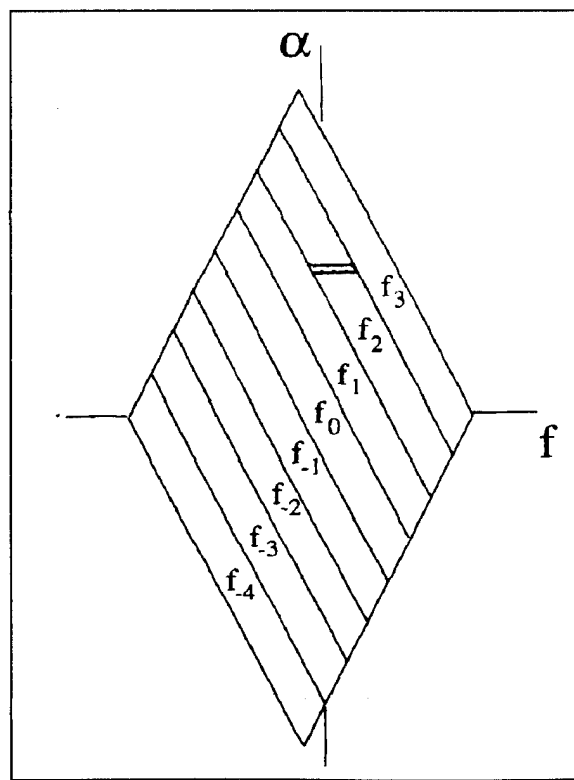


Figure 5 - SSCA tiling of the bifrequency plane

$$S_{xy_T}^{\alpha_i + q\Delta\alpha} \left(n, \frac{f_k}{2} - q \frac{\Delta\alpha}{2} \right)_{\Delta t} = \sum_r X_T(r, f_k) y^*(r) g(n-r) e^{-i2\pi r q / N} \quad (4).$$

The FFT is central to both FAM and SSCA. This makes the design of a near real time FFT architecture vital to CSA development. An additional capability to form the complex vector product of long vectors is required, for the calculation of correlation products.

C. FFT IMPLEMENTATION

Digital implementation choices revolve around decimation in time (DIT) or frequency (DIF), the use of radix-2 or radix-4 butterfly machines, the implementation of the FAM or SSCA algorithm and integer versus floating point number representations. Although more intricate in nature, the use of radix-4 butterflies in a DIT algorithm provides increased performance in VLSI chips.[Ref. 1, p. 60] The efficient design and implementation of a radix-4 FFT is therefore undertaken. For digital implementation, data representing complex floating point numbers is pipelined and streamed through a processor chip [Ref. 5]. Higher radix FFTs can be computed by conducting multiple passes through a radix-4 processor. DIT and DIF computations differ in when the “twiddle factor” multiply is completed. Time decimation executes the multiplies before the butterfly, whereas in frequency decimation it is done subsequent to it. In this thesis, a decimation in time approach is assumed.

Analysis of the relative complexities of performing the FAM and SSCA algorithms has shown that the SSCA is quantitatively less complicated [Ref. 2, p. 719]. In this design, however, the choice of algorithm is immaterial because the butterfly unit is needed for each. We will now discuss specific processor design.

III. CSA PROCESSOR DESIGN

A. TOP LEVEL PROCESSOR DESIGN

In signal analysis applications, analog data is sampled and converted through an analog to digital converter. This data forms the input for the CSA processor (Figure 6). Within the processor, the butterfly machine buffers the data for processing by a three port FFT processor (Figures 7 and 8). The processor controller is programmed to conduct multiple passes through the FFT processor to conduct correlation multiplies and high order FFTs. Multiple butterfly machines, operating in parallel, would give the CSA processor the ability to process incoming data at the high rates required for real-time analysis. The bifrequency plane values produced could be presented to an analyst in a graphical display.

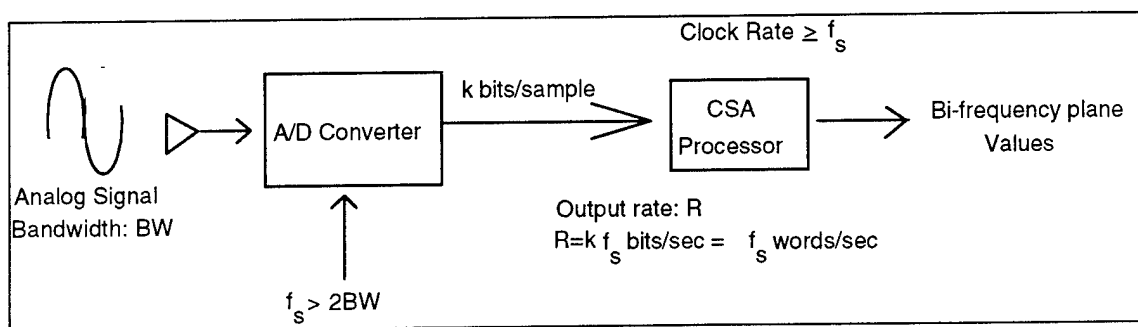


Figure 6 - Cyclostationary Processing Dataflow

B. SINGLE PROCESSOR PROGRAMMING

The FFT processor program consists of blocks and sub-blocks. A block contains input specifications, the output specification, and the list of required passes. The input specification identifies the two input buffers. For a correlation multiply, the inputs would be from the coefficient buffer and from a selected data buffer. For FFTs, the input is from the data buffers. The output specification either returns intermediate results to a buffer for further computations or releases the data to the output bus. Each pass

description identifies the input source, the output destination and an operation code for the FFT processor.

Within the source and destination descriptions are base addressing and address sequencing information

[Ref.5].

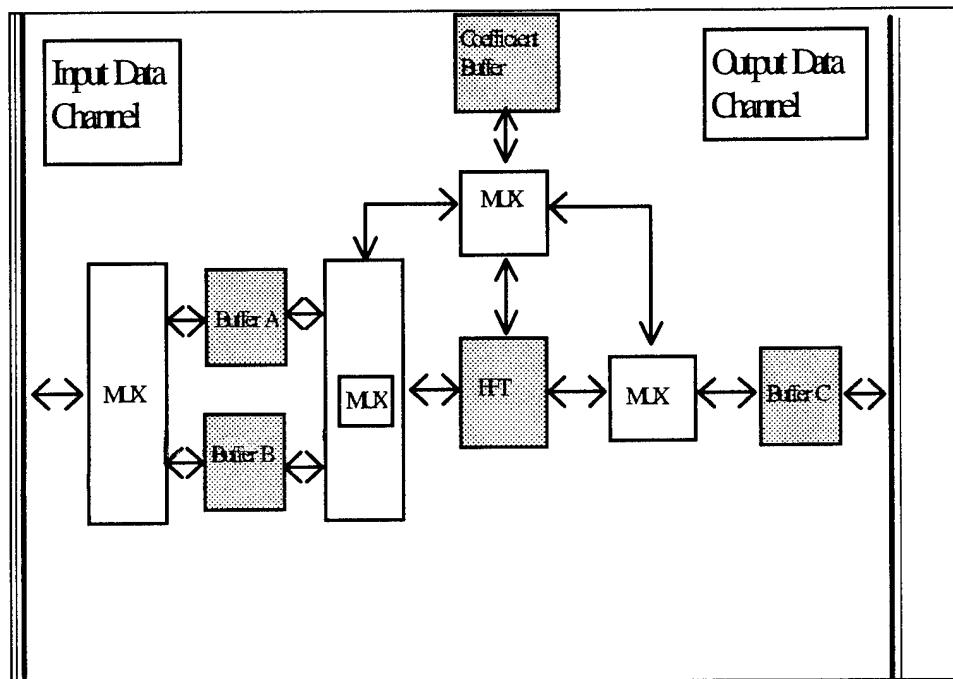


Figure 7 - General Butterfly Machine

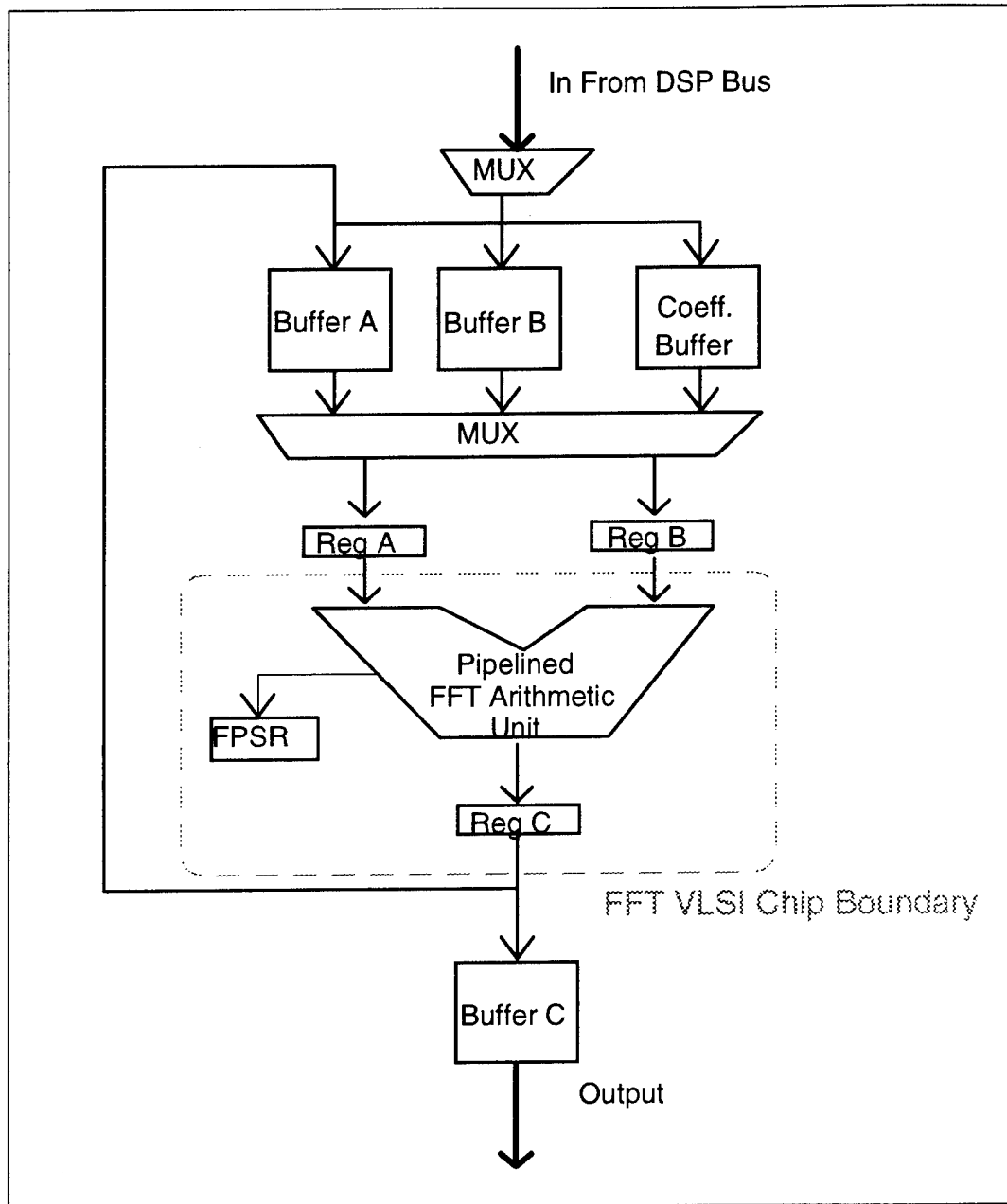


Figure 8 - FFT Chip

C. FFT CHIP - ZIMMER DESIGN

In the Zimmer design, input is through a pipelined complex floating point multiplier. This complex multiplier scaled the inputs by the FFT external twiddle factors. The multiplier output loads four registers that transfer these scaled values into a latch for four clock cycles. The latch output feeds a multiplexer group that selects the appropriate input values for computation of the FFT. By implementing the FFT in

radix-4, internal twiddle factor "multiplication" becomes simple sign changes and output selections [Ref.1, p.85]. The FFT is completed by adding the four multiplexer outputs resulting in the final transform value. The final design used four floating point multipliers, eight floating point adders, two 4-input multiplexers, three data registers, one latch and some assorted logic. Initial layout with the Genesil standard compiler proved infeasible due to size. The design was optimized using Autologic synthesis tools and Logic-Compiler alternative compiler. The final design used 1.0 μ m (two metal, one poly) technology and resulted in a 200,000 sq. mil chip (including pads). Specifications for the Zimmer chip design were as follows:

- 20 bit word size (1 sign, 6 exponent, 13 mantissa)
- Normalized numbers only - IEEE infinity, denormalized numbers and NaNs not recognized.
- True zero recognized by all zeros in the exponent.
- Mantissa in signed magnitude.
- Exponent in excess 2^5 code.
- Smallest magnitude number: $1.000000000000_2 \times 2^{-31} / 4.65661287308_{10} \times 10^{-10}$
- Largest magnitude number: $1.111111111111_2 \times 2^{31} / 4.29470515201_{10} \times 10^9$

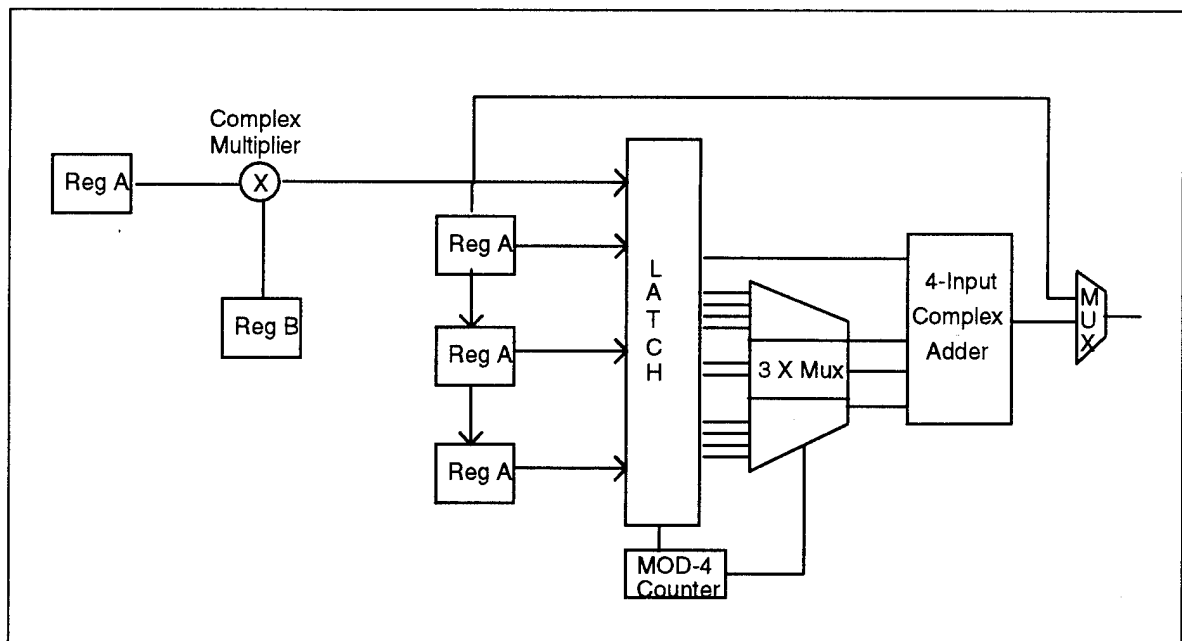


Figure 9 - FFT Arithmetic Unit - Zimmer Chip Design

D. FFT CHIP - NEW DESIGN

Initially, different approaches to the basic chip design were explored. This was done to give the new design the ability to perform a wider range of arithmetic operations. A design using multiplier-accumulators initially showed promise. This approach is similar to the TRW LSI Products Design implementation of the TMC2310.6 The TMC2310 uses multiplier-accumulators to perform forward and reverse DIT Cooley-Tukey FFTs on real or complex data. It uses a radix-2 decomposition on 16-bit data. Further analysis indicates that such an approach requires more hardware than the Zimmer design and leads to more latency in the pipeline. Time restriction prevented further analysis of this design. This resulted in the preservation of the approach demonstrated in the Zimmer chip. Future efforts should explore the use of accumulators in this application.

Specifications for the new chip were set as follows:

- 32 bit word size (1 sign, 8 exponent, 23 mantissa).
- Normalized and denormalized numbers.
- IEEE infinity and signaling NaNs.
- Underflow and Overflow signals.
- IEEE Standard Single Format.
- Mantissa in signed magnitude.
- Exponent in excess +127 code.
- Smallest magnitude number: $0.000000000000000000000001_2 \times 2^{-126} / 1.40129846432_{10} \times 10^{-45}$.
- Largest magnitude number: $1.1111111111111111111111112 \times 2^{127} / 3.4028234663710 \times 10^{38}$.

Upon completion and successful simulation of this new design, fabrication is planned using the MOSIS facility. Compatibility with established procedures restricts process selection to a defined set. For this project .6 μm , .8 μm and 1.2 μm processes were considered. The Epoch compiler can be used to compile for any process for which the rule set is installed. For this design .7 μm (3 metal, 2 poly) and 1.0 μm (2 metal, 1 poly) were applicable. Remaining within these practical restraints resulted in a design decision to compile using 1 μm (2 metal, 1 poly) technology. This will enable the design to be scaled up for a 1.2 μm fabrication with minimal risk. Once a .8 μm rule set is was made available, follow-on designs could recompile and fabricate using the .6 μm or .8 μm MOSIS process. This is desirable since smaller feature sizes can lead to faster operating speeds. An operating clock speed of 45 Mhz was set as the design goal.

IV. CHIP DESIGN AND SIMULATION

A. VERILOG HDL

Verilog Hardware Description Language (HDL) [Ref. 7] was developed by Cadence Design Systems.

It can be used to describe both the functions of a design and the components and connections to components within the design. A Verilog model can be developed in the four different levels of abstraction shown in Table 1.

algorithmic	a model that implements a design algorithm in a high-level language construct
RTL	Register Transfer Language a model that describes the flow of data between registers and how a design processes that data
gate-level	a model that describes the logic gates and the connections between logic gates in the design
switch-level	a model that describes the transistors and storage nodes in a device and the connections between them

Table 1 - Verilog Model Types

Modules are the basic building block in a Verilog design. These modules contain the physical or algorithmic description of the full design or a portion of the design. Modules can incorporate other modules to facilitate top-down and bottom up design. The behavioral language provides the following capabilities:

- structured procedures for sequential or concurrent execution.
- explicit control of the time of procedure activation specified by both delay expressions and by value changes called event expressions.
- explicitly named events to trigger the enabling and disabling of actions in other procedures.
- procedural constructs for conditional, if-else, case, and looping operations.
- procedures called tasks that can have parameters and non-zero time duration.
- procedures called functions that allow the definition of new operators.
- arithmetic, logical, bit-wise, and reduction operators for expressions.

The structural language (for gate- and switch- level models) provides the following capabilities:

- a complete set of combinational primitives.
- primitives for bi-directional pass and resistive devices.
- the ability to model dynamic MOS models with charge sharing and charge decay.

The ability to utilize a single language for all aspects of a design enhances Verilog's ease of use. The language is capable of conducting mixed-level modeling and simulation. Its interactive capabilities allow circuit patching and debugging within a simulation run. Graphical signal displays are also available for timing analysis.

B. EPOCH SILICON COMPILER

Epoch [Ref. 8] is a schematic- (or HDL-) driven physical design tool. By building integrated chip layout expertise into the software, physical design tasks have been reduced significantly. This gives the high-level designer complete control of the design from schematic to silicon fabrication. Typical designs begin with the implementation of a chip architecture using a collection of Epoch-resident cells and/or functional descriptions that are directly synthesizable into such cells. After design input via a schematic or HDL, the design is translated into an Epoch netlist. With functionally descriptions automatic synthesis is performed and implemented into standard cell gates. This netlist is used in design compilation. The steps involved in this process include:

- Creation of standard cell blocks and placement of all blocks and cells to optimize for density and route minimization and optionally for timing
- Performs global and detailed routing on the blocks using routing channels created during placement
- Sizes the buffers based on the capacitive loads of the cells and routes.
- Estimates power consumption at the different power nodes on the chip and assigns appropriate widths to the power rails
- Adjust microplacement of the blocks to accommodate blocks that have become large due the upsizing of buffers.
- Rerouting of the chip with correct power rail sizes.

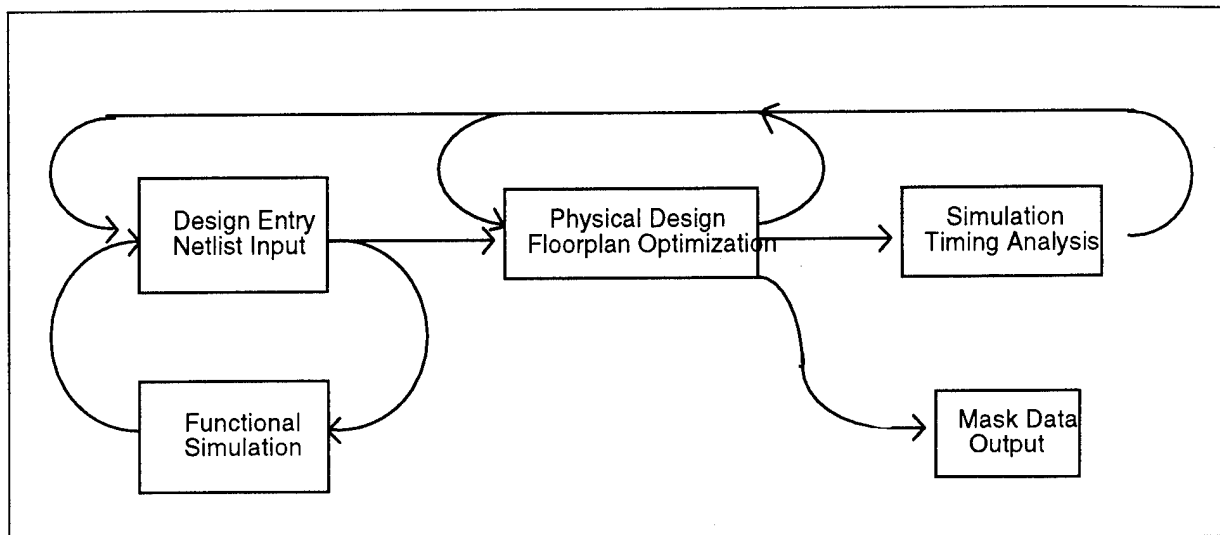


Figure 10 - Epoch Design Flow

This tool features a process-independent silicon compilation. This allows the designer to defer final process selection and provides for design migration and reusability. An open system approach is used and performance driven physical designs are possible. The delay models used within Epoch have been shown to be within 5% of Spice models.

C. VERILOG - EPOCH INTERFACE

Epoch's design automation relies on the use of the Epoch resident parts library or the use of directly synthesizable HDL descriptions. As a result, Epoch Verilog is a subset of the standard Verilog language that supports most, but not all of the Verilog standard constructs. Epoch Verilog constructs include the following:

- Nets - only wire, supply0 and supply1 are supported. Nets may be declared as busses.
- Ports - Terminal nets must be declared as ports, either input, output, or inout.
- Text Substitution - The `define, `undef and `include are supported.
- Compiler Directives are imbedded into the Verilog code using the "//epoch set_attribute" keyword.
- Regs are memoried for simulation purposes.
- Sequential assignment occur in the context of always statements and take the form of if, case and other types of statements.
- Functions are supported to provide a means for allowing subroutines.
- Input don't cares are implemented using CASEX or CASEZ statements.
- Non-memoried outputs must be assigned explicit default states.
- Addition, subtraction, multiplication and division are only legal in constant expressions.
- The forever, repeat, while, for, assign, disable, and wait statements are not allowed within an always block.

Constructs that are not directly support by Epoch Verilog can usually be expressed with alternative syntax that is supported.

D. ASIC DESIGN PROCESS

For this effort, chip design was initially completed using Verilog. Epoch resident parts were used when possible by invoking a command line option in Verilog. This gave the simulation visibility into the Epoch Verilog model subdirectory. After verifying correct operation, the HDL was used as the input to the netlist generation module of Epoch. This produces a "VerilogOUT" file that has backannotated timing information and RC interconnect delays. This new file can then be simulated for timing analysis and final design. Once this simulation output is verified, the design can be compiled and floorplanned. TACTIC, a timing analysis tool, is then used to analyze the effects of placement, routing and buffer sizing on timing. After completing any required manual floorplanning , a GDSII file can be generated form the design for silicon fabrication.

E. CHIP HDL DESIGN

Use of a HDL for this design led to a hierarchical, modular design strategy. The functions, such as a fixed point multiplier or barrel shifter, were chosen as the lowest level module in the overall design. The next level up in the hierarchy consists of the floating point multiplier and floating point adder. The 4-input complex floating point adder and complex floating point multiplier comprise the third level in this bottoms-up structure. The 4-input complex adder consists of six floating point adders, while the complex multiplier was is composed of four floating point multipliers and two floating point adders. Finally, the top layer was defined by one complex multiplier and one 4-input complex adder. At each upper level, additional logic structures were used as required to complete datapath requirements. Figure 11 outlines the HDL hierarchy.

The following sections describe each lower level module in detail. Their use can be seen in the floating point adder and floating point multiplier diagrams Figures 12 through 22. The Verilog HDL code for these modules is contained in Appendix B.

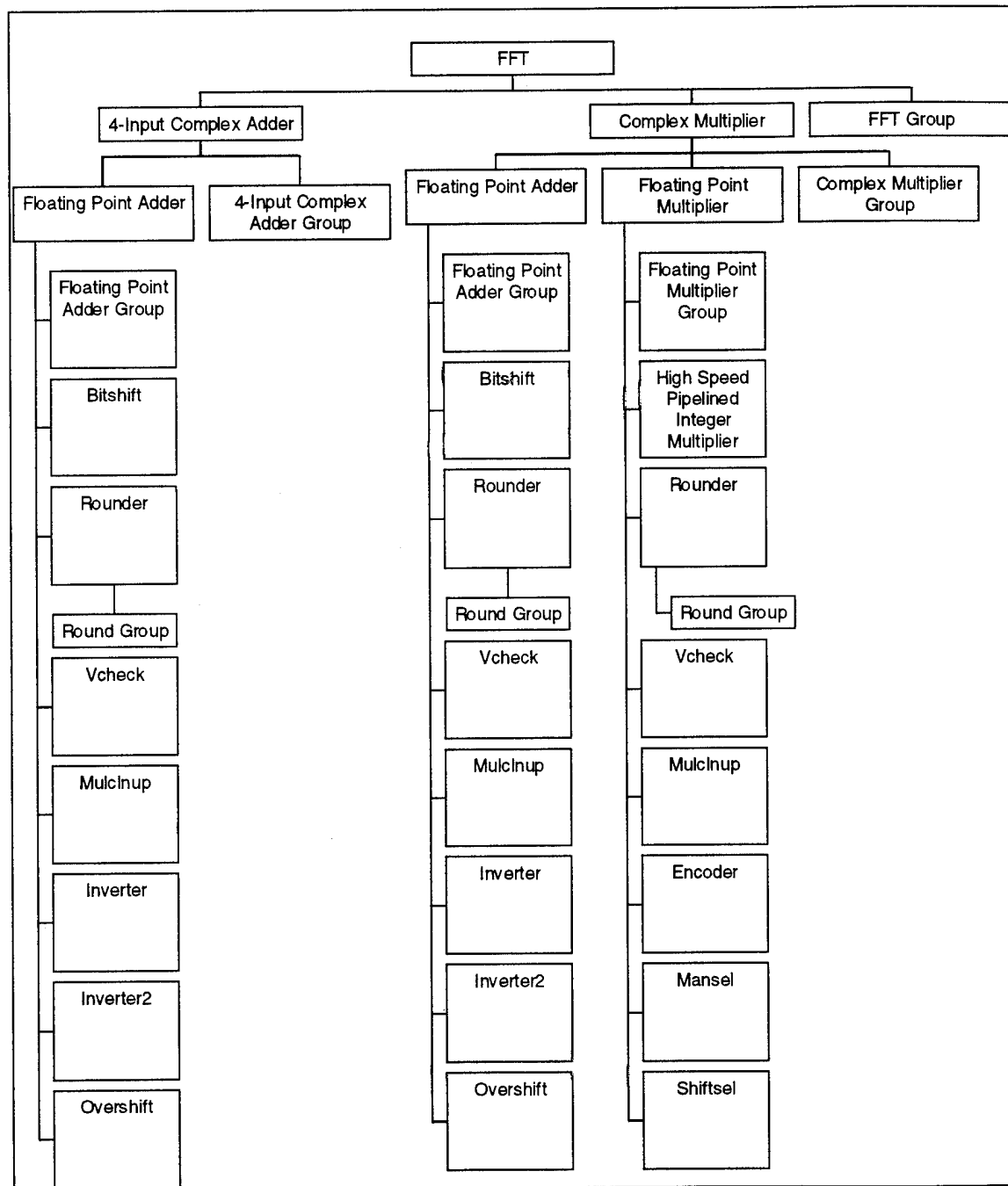
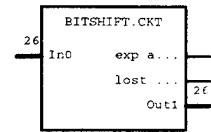


Figure 11 - Design Hierarchy

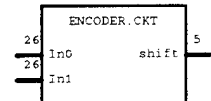
1. Bitshift - Pre-rounding bit shifter

Bitshift takes a 25-bit input and conducts a right shift if the most significant bit (msb) is set. A one in the msb indicates that previous computations produced a carry into the 2^1 valued bit. To maintain normalized values a right shift and increase of the exponential value is required. The exp_add output indicates this requirement. The least significant bit shifted out in this event may create a non-exact result. This is indicated by the setting of the lost_bit output. The lost_bit is later used in floating point adder logic to determine an inexact answer.



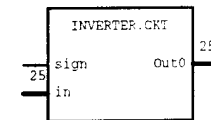
2. Encoder - Priority Encoder

The 48-bit input to the encoder is scanned for the most significant set bit. The location of this bit (the number of zeros to the left of the most significant one plus 1) is sent out for use in the multiplier's barrel shifter.



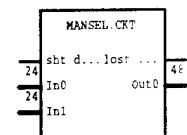
3. Inverter / Inverter2

The Inverter takes a signed-magnitude value and inverts the value bits in preparation for 1s' complement computations. Inverter2 takes a 26-bit 1s' complement number and inverts it is the resultant signed-magnitude value is negative.



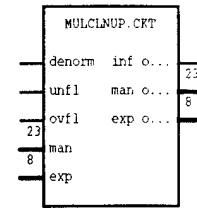
4. Mansel - Mantissa Selector

In the floating-point multiplier, Mansel inputs bit indicating shift direction and the 48-bit product. Similar in function to bitshift, if a right shift is indicated, a zero is prepended to bits 47-1 of the product and the lost_bit output is set. Otherwise, the product is unaltered.



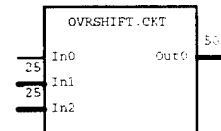
5. Mulclnup - The Cleanup Module

The 23-bit, post-rounded mantissa, 8-bit resultant exponent, and denormalized, underflow, and overflow flags are used in this module to ensure correct final output. If the denormalized flag is set, the final exponent output is set to zero. For an underflow, the exponent and mantissa values are set to zero. An overflow sets the exponent to “1111111” and sets an infinite-result flag.



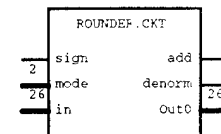
6. Overshift

This module sets the mantissa to zero if a shift greater than 50 is indicated for the product.



7. Rounder

Rounder implements the four IEEE rounding modes through the use of a Motorola 68040 rounding algorithm [Ref. 9]. The denormalized flag and an indicator to increment the exponent is produced. Table 2 describes each rounding mode and required selection values.



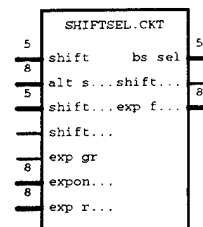
Mode	Name	Result
00	Round to Minus	The result is the value closest to and no greater than the infinitely precise intermediate result (possibly minus infinity)
01	Round to Nearest	The representable value nearest to the infinitely precise intermediate value is the result. If the two nearest representable values are equally near (a tie), then the one with the least significant bit equal to zero (even) is the result. This is sometimes referred to as “round to even”.
10	Round to Plus	The result is the value closest to and no less than the infinitely precise intermediate result (possibly plus infinity)
11	Chop Mode	The result is the value closest to, and no greater in magnitude than, the infinitely precise intermediate result. This effectively clears the bits to the right of the rounding point

Table 2 - Rounding Modes

8. Shiftsel - Shift Value Selector

The multiplier presents four different shift possibilities:

- If the most significant bit is one, the mantissa must be shifted right one place and one added to the exponent
- If the third most significant bit is 1, the mantissa must be shifted left one place and one subtracted from the exponent.
- If the exponent value is greater than the value that indicates the location of the most significant set bit, the mantissa must be shifted left a value equal to the location minus two. The shifted value must also be subtracted from the exponent.
- If the exponent value is less than or equal to the value that indicates the location of the most significant set bit, the mantissa must be shifted left a value equal to the exponent value. The exponent must then be set to zero.



Examples of these options are presented in Table 3. Shiftsel takes all the values required to determine which possibility exists and outputs the correct shift direction, shift amount and exponent value.

Input Value	Encoder Output	Shift Direction	Exponent In	Shift Amount	Exponent Out	Output Value	Remarks
10.000	1	Right	5	1	6	01.000	
01.000	2	N/A	5	0	5	01.000	
00.100	3	Left	5	1	4	01.000	
00.010	4	Left	5	2	3	01.000	Encoder < Exponent
00.00001	7	Left	4	4	0	00.1000	Denormalized output; Encoder > Exponent

Table 3 - Floating Point Multiplier Shift Options

9. Vcheck - Initial data formatter

The vcheck module takes the incoming data and formats it for processing through its host unit. After first checking for an infinite or NaN input, data is characterized as normalized or denormalized. Normalized values have a 1 bit prepended to the mantissa value, while denormalized numbers get a zero attached. The exponential value for denormalized numbers are also set to 1 in accordance with the excess 127 code.

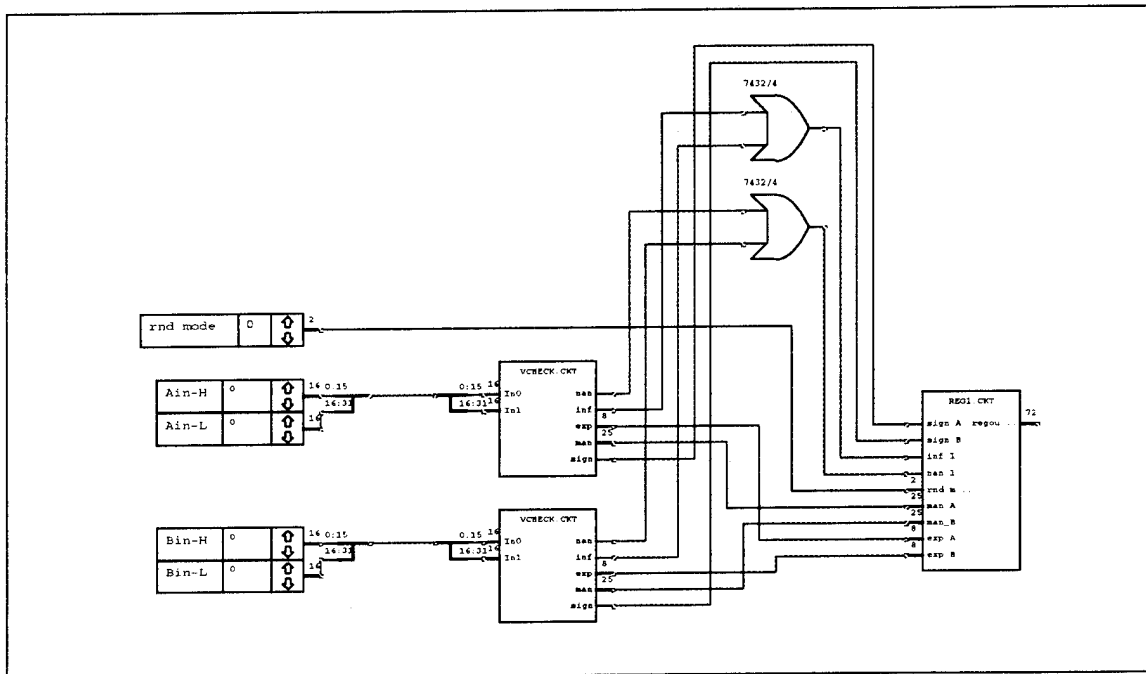
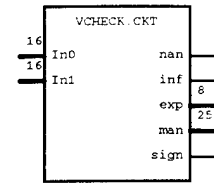


Figure 12 - Floating Point Adder Stage 1

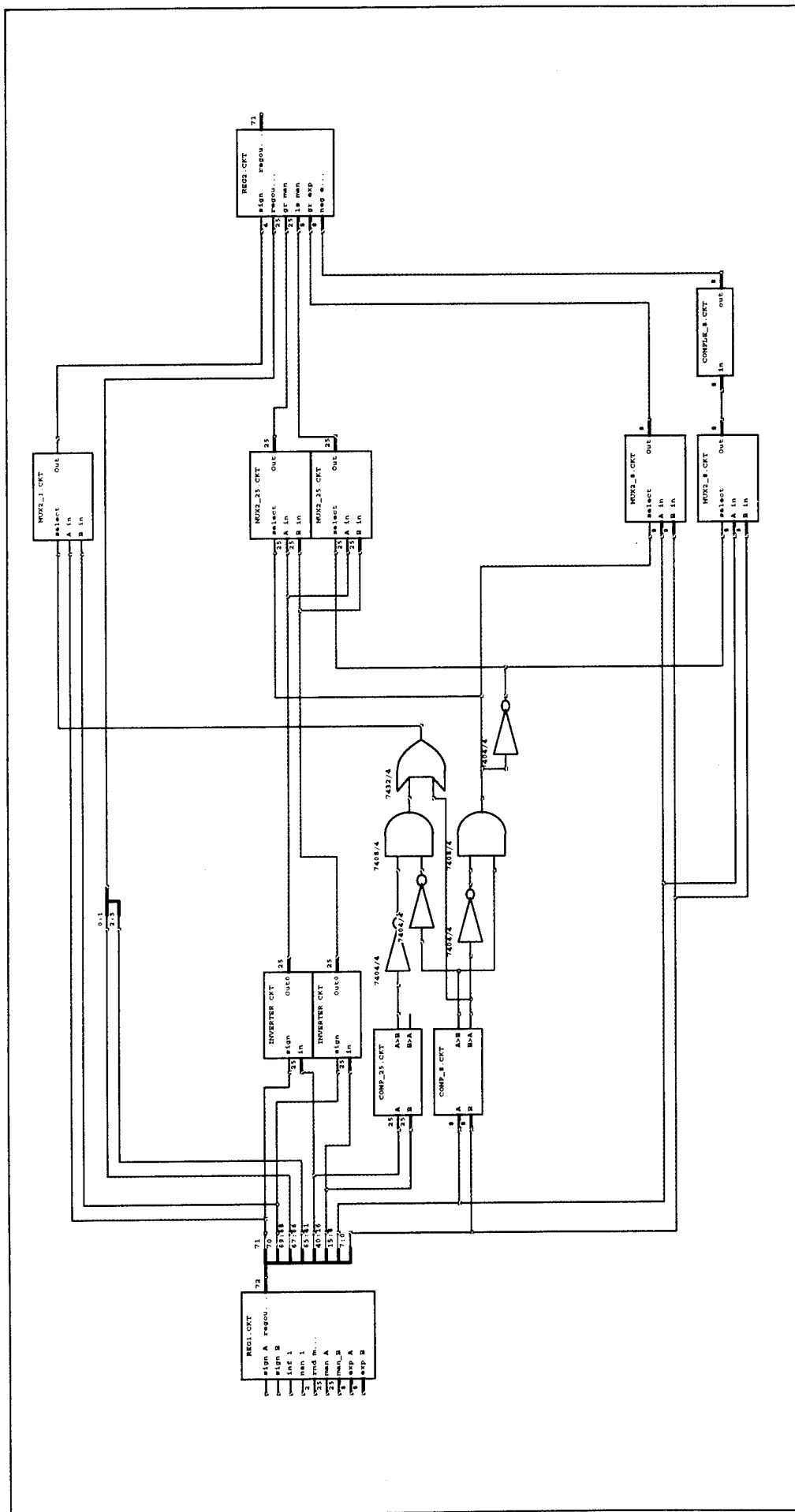


Figure 13- Floating Point Adder Stage 2

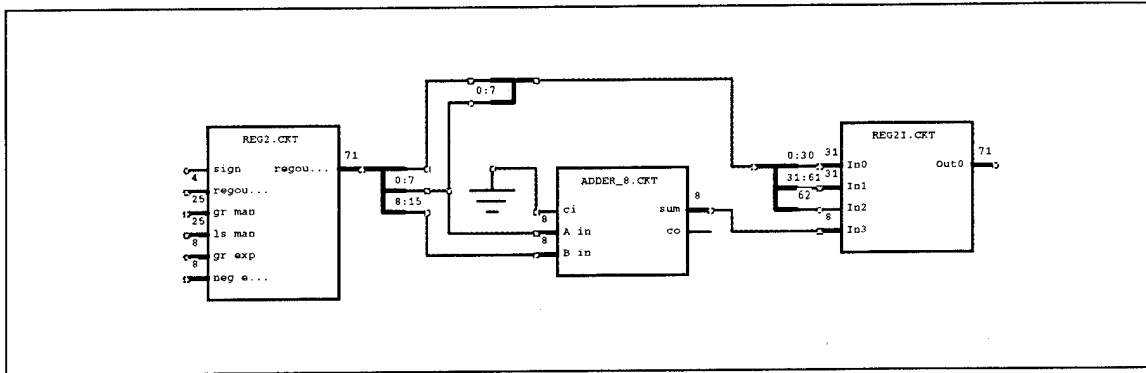


Figure 14 - Floating Point Adder Stage 3

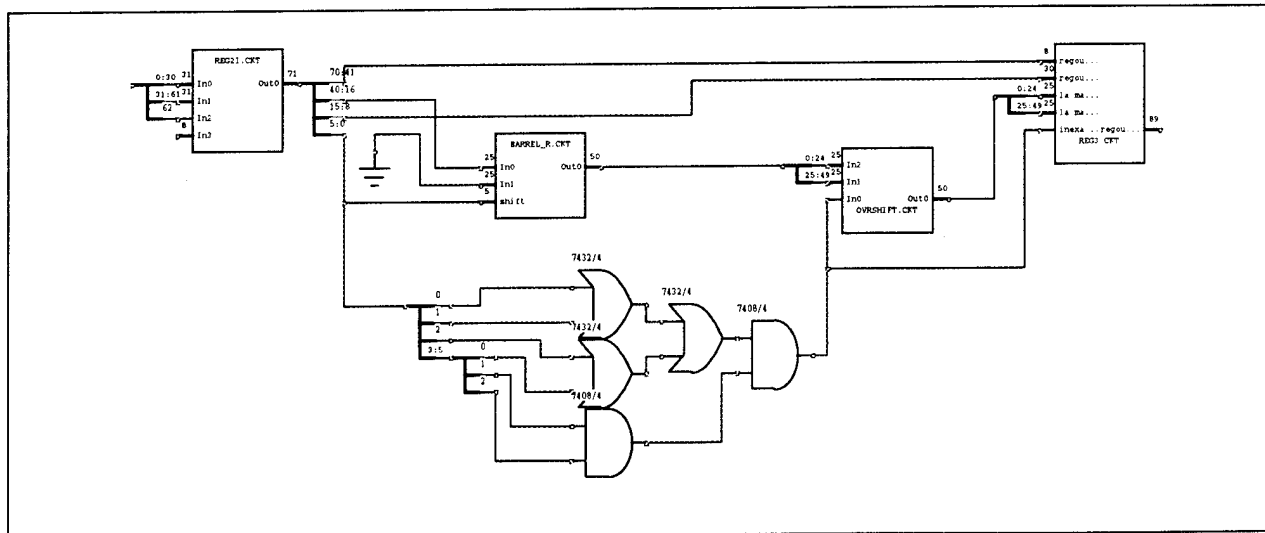


Figure 15 - Floating Point Adder Stage 4

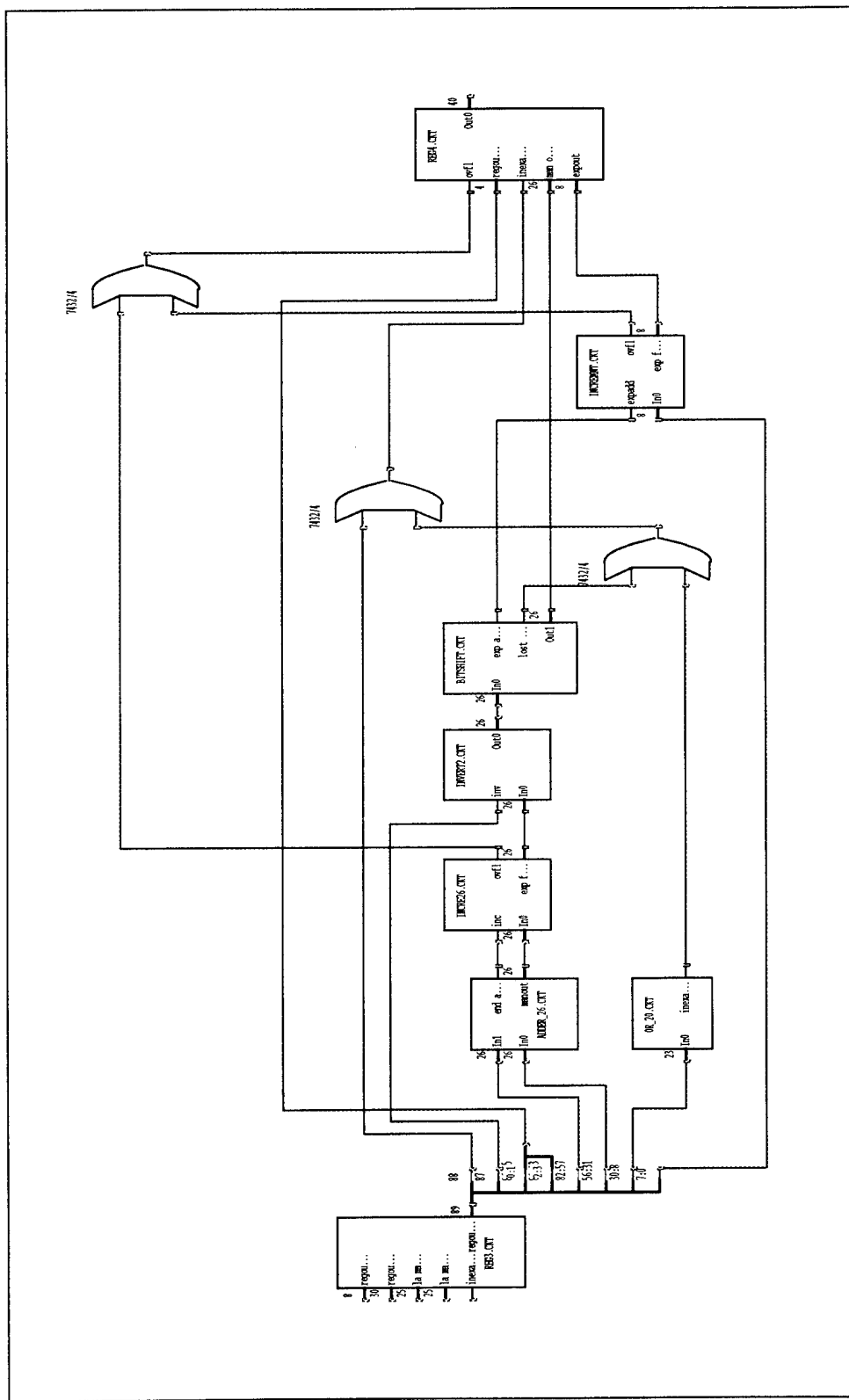


Figure 16 - Floating Point Adder Stage 5

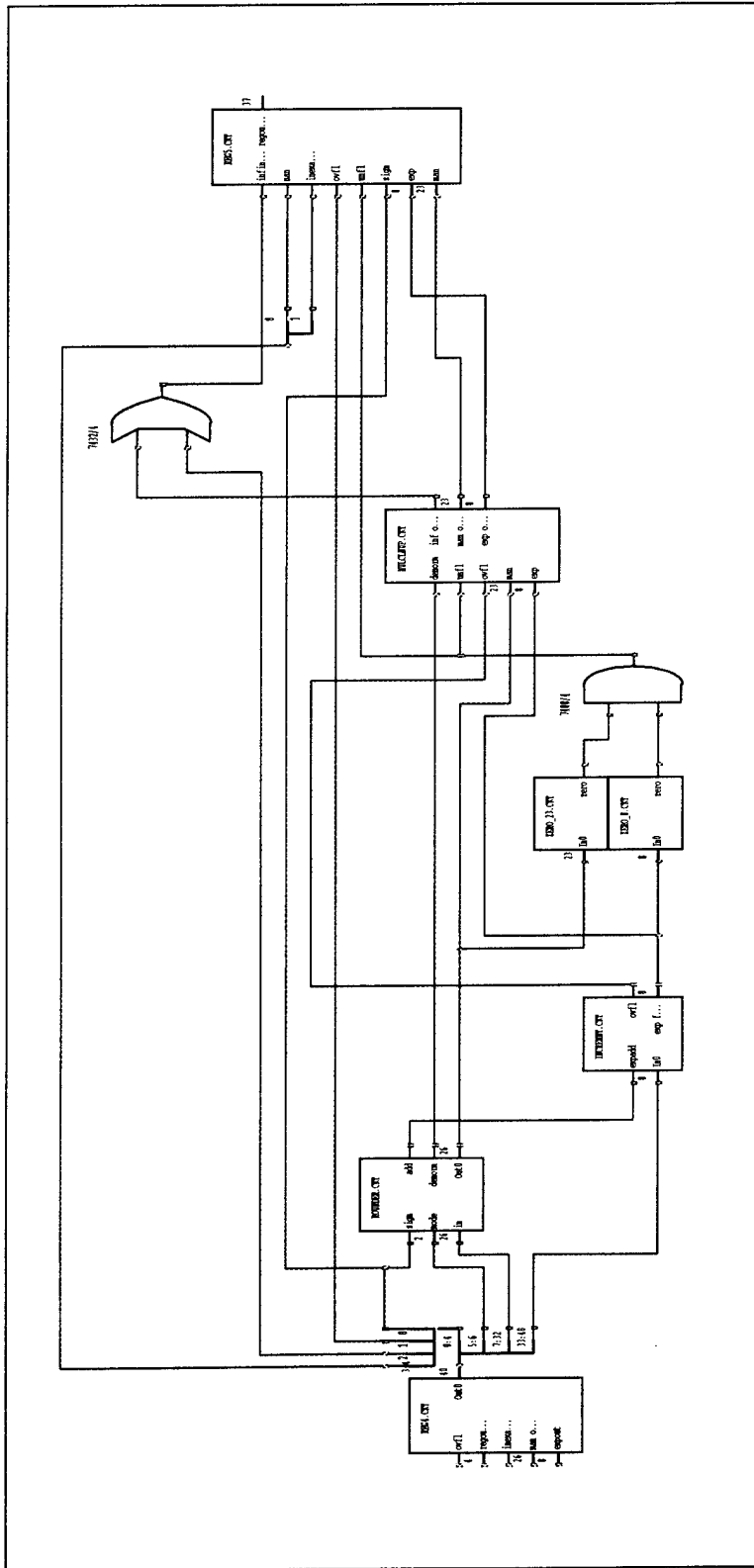


Figure 17 - Floating Point Adder Stage 6

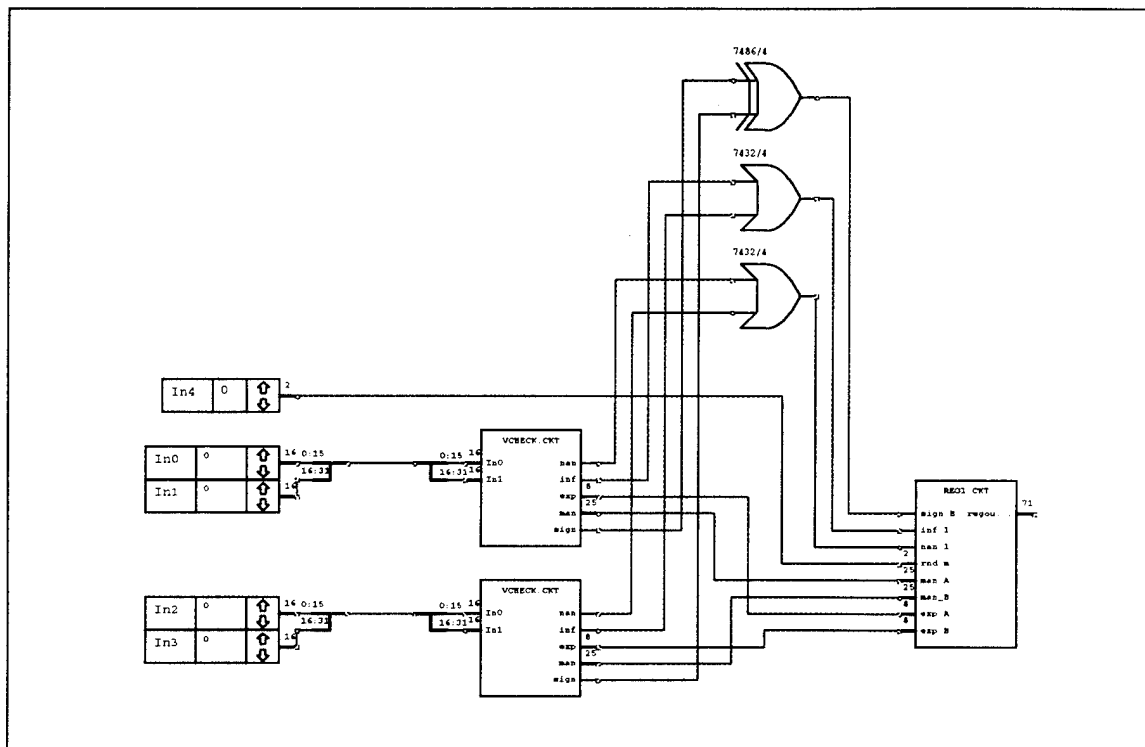


Figure 18 - Floating Point Multiplier Stage 1

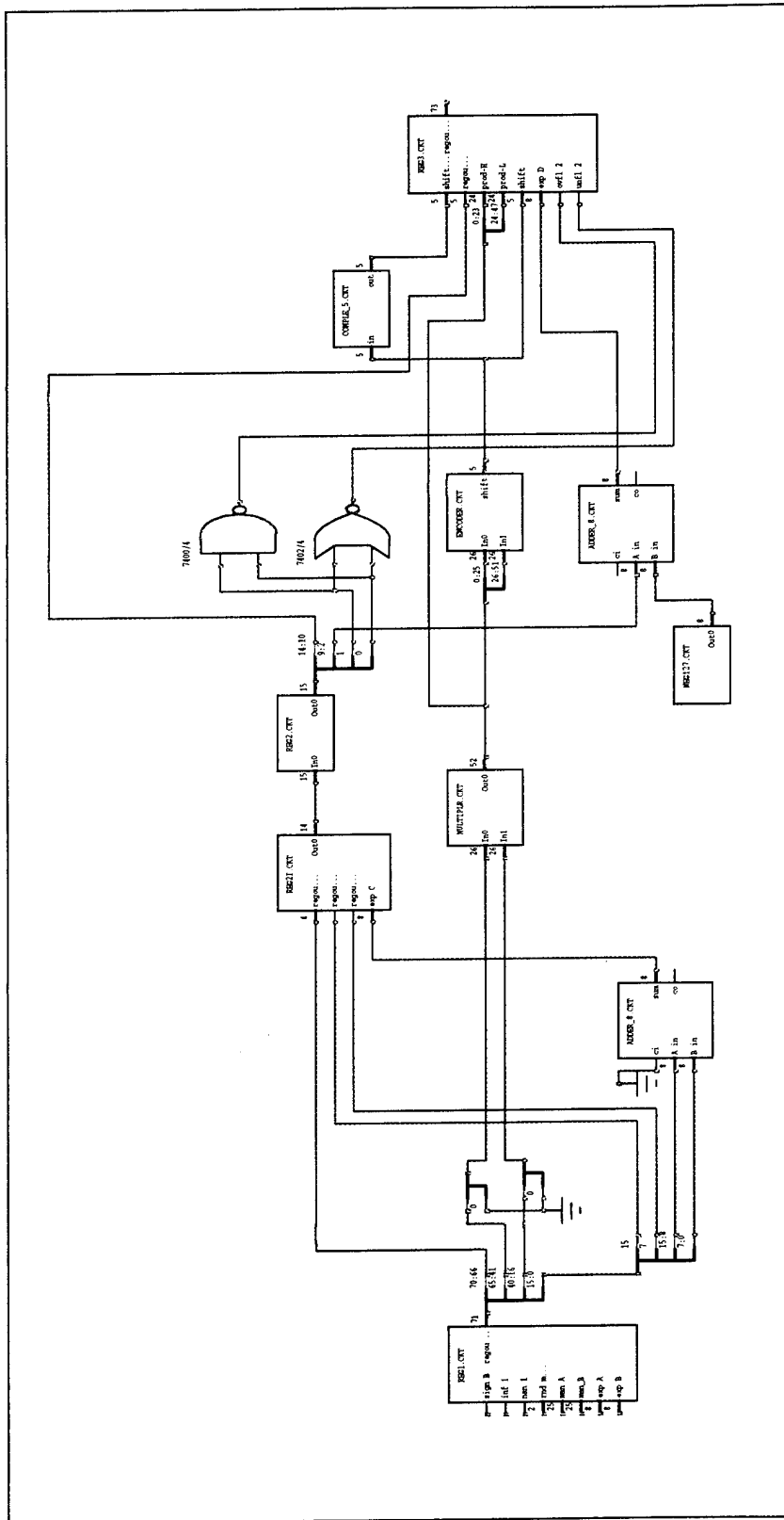


Figure 19- Floating Point Multiplier Stage 2-4

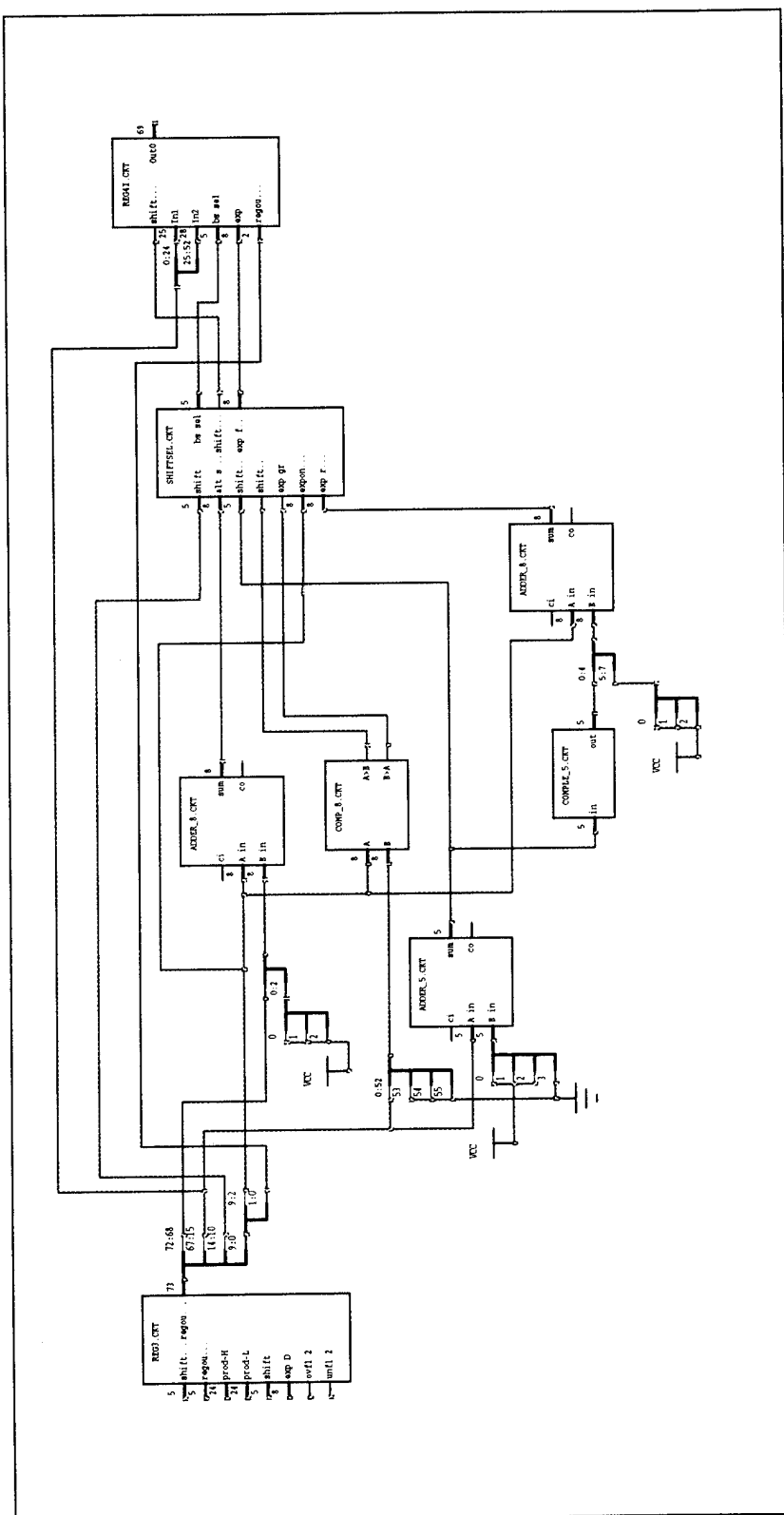


Figure 20 - Floating Point Multiplier Stage 5

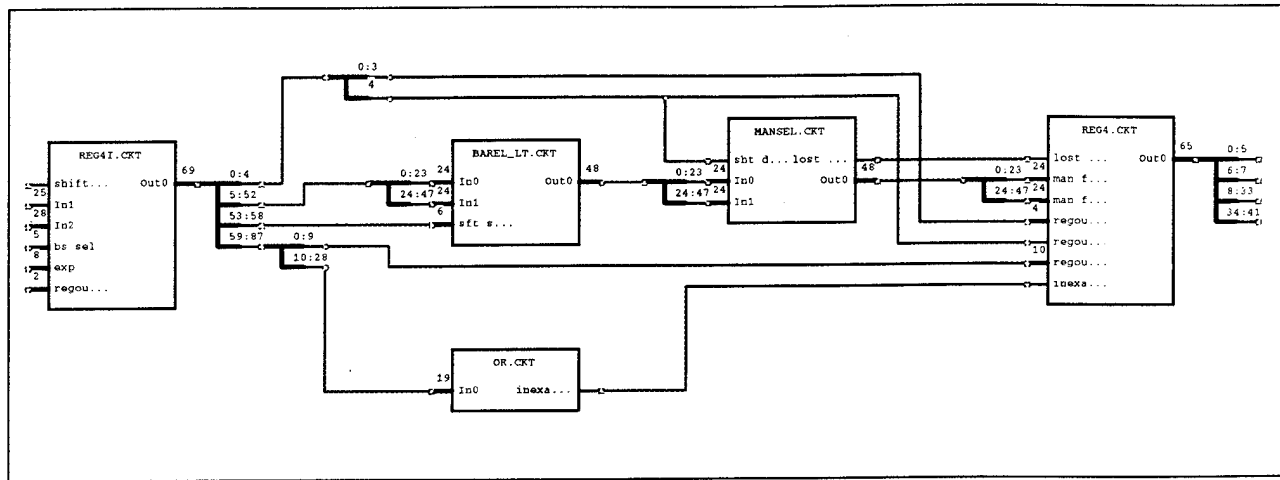


Figure 21 - Floating Point Multiplier Stage 6

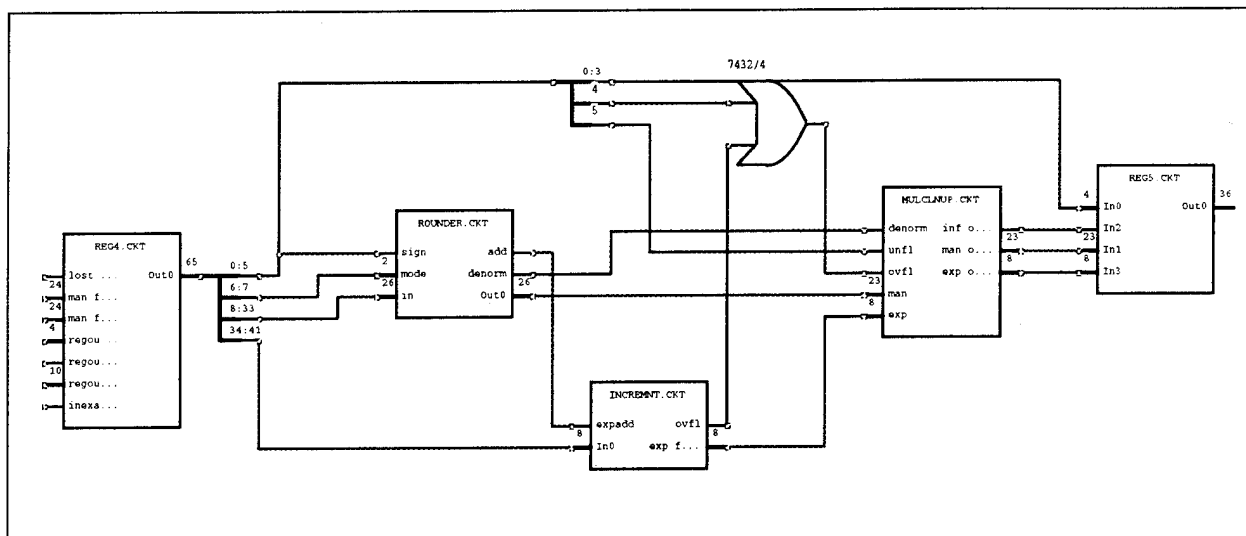


Figure 22 - Floating Point Multiplier Stage 7

F. CHIP SIMULATION

After the HDL code was written, initial debugging and testing was accomplished using Verilog interactively. Testbench programs, "mpltst" and "addtst" were used to verify operation. Verilog command files were written to automate tracking of register values during simulation. The "\$monitor" and "\$gr_waves" commands were used to record input/output transitions and to graphically display the waveforms of up to twenty nets. The test vectors used included normalized and denormalized numbers, zero, infinity, and NaNs. After these initial tests, the "fpadder" and "fpmplier" files were compiled

through Epoch using the "VerilogIN" procedures. This series of steps converts the Verilog structural description into an Epoch netlist. Synthesis on logic descriptions is also performed. The compilation process also checks for mismatched net widths and unconnected inputs or outputs. Timing information is also embedded into the design. The "VerilogOUT" procedure translates the Epoch netlist into a new Verilog HDL file. This file can be run using Verilog to verify operation and timing. An optional Testbench file can also be generated to aid in simulation. VerilogOUT files were generated for the floating point adder and multiplier. Success in the simulation using these files were accepted as a final check on the HDL description.

After compilation of the complex multiplier and 4-input complex adder cells, tests using the VerilogOUT files were tried. After numerous failures due to inadequate memory for the datastructure, Verilog code testing beyond the initial levels was abandoned. Error codes indicated a requirement for over 76 megabytes of memory and consultations with the system administrator failed to provide a practical solution. Epoch provides an option to produce top-level only VerilogOUT files. This may provide an alternative to this type of testing. Time constraints prevented further exploration of this option. Appendix G and H contains the floating point adder and floating point multiplier testbench programs used and examples of simulation results.

V. CHIP FLOORPLANNING AND TIMING ANALYSIS

A. EPOCH FLOORPLANNING

Following the creation of an Epoch netlist, the design was ready for the Epoch design compilation process. This is initiated by using the “Automatic Compile” command. This step completes the following actions:

- Divides standard cells into standard cell groups.
- Generates datapath groups.
- Places all single leaf cells, groups and other blocks for optimal density and route minimization (and optionally for timing constraints). Placement is performed bottom up, from the lowest to the highest level composites.
- Places pads and creates the pad ring.
- Performs global and detailed routing and creates balanced clock trees.
- Sizes buffers based on capacitive loading (and optionally on timing constraints).
- Calculates power dissipation for power rail sizing.
- Regenerates cells with correct buffer sizes.
- Adjusts microplacement of blocks and repeats routing with correct power rail widths [Ref. 8].

The final output is a finished chip design consisting of a core and a surrounding pad ring. Figure 23 shows an example of a non-optimized chip layout (without pads) after automatic compile. In these examples the lower level modules have retained their identity. This is done by including a “FIXEDBLOCK” compiler directive in the Verilog code. If this is done for low level modules, gross floorplanning inefficiencies can occur. In a final design the “FIXEDBLOCK” attribute should be removed, allowing optimal placement of standard cells and cell blocks. After routing and placement, the design can again be simulated using the VerilogOUT process. The next design step is timing analysis.

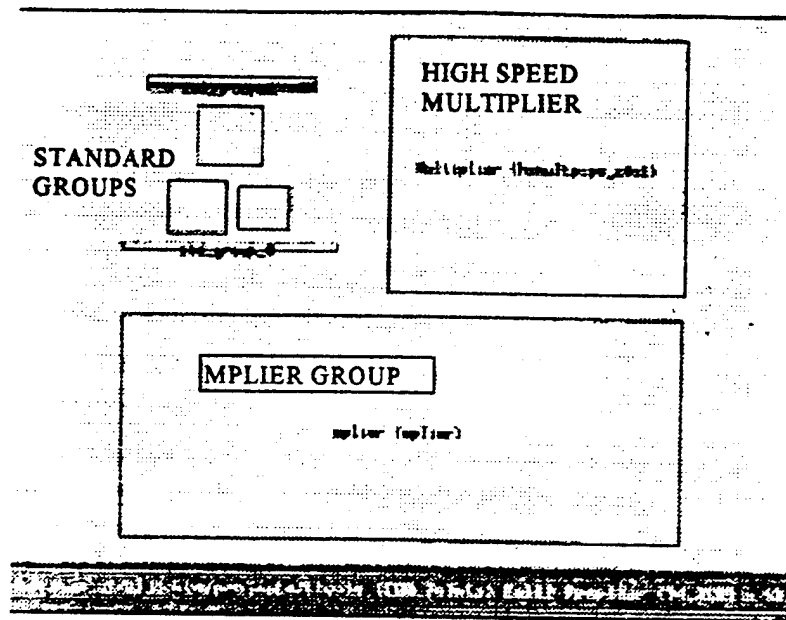


Figure 23- Floorplanning

B. TACTIC TIMING ANALYZER

Timing analysis is done using the TACTIC tool. These algorithms use the placement and timing information to produce a graphical display of all timing paths and timing violations (Figure 24). By "zooming in" on the critical lines in this full display, node labels can be seen and the critical timing paths identified (Figure 25). These names relate directly to the Verilog HDL modules. The time values along the bottom of the display indicate the node transition within one clock period. This allows timing delays to be quickly pinpointed. At this point design iteration is used to improve timing through design code modifications.

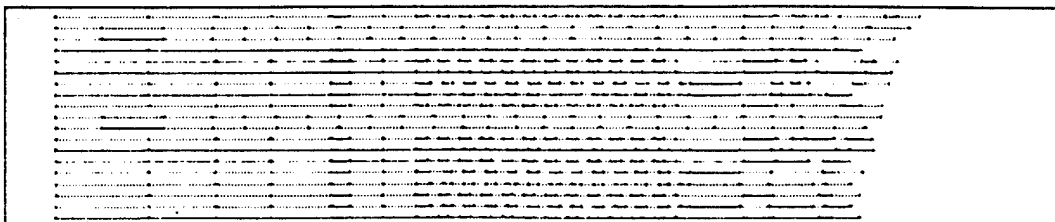


Figure 24 - TACTIC timing graphical display

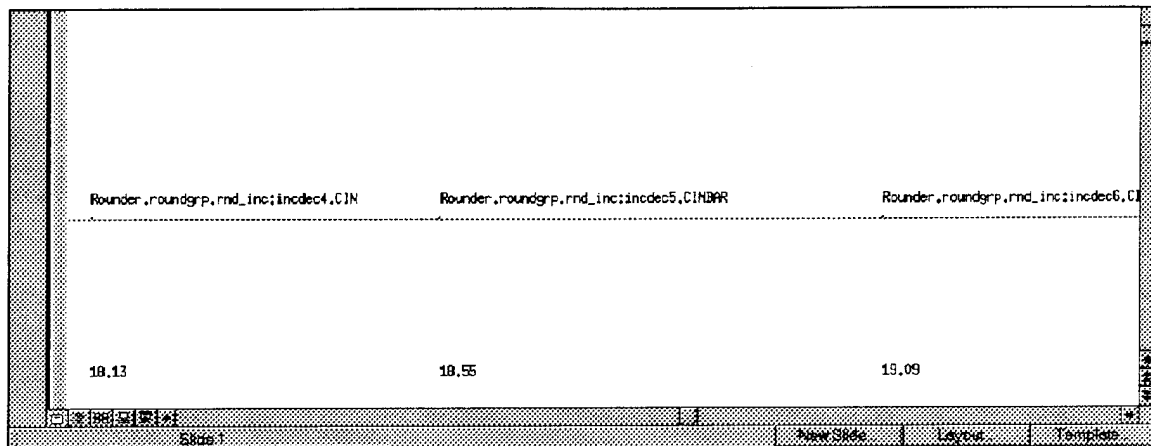


Figure 25- Single timing path display

The two most critical timing paths in this design are in stage four of the floating point multiplier and stage six of the floating point adder. Excessive delays occur in these paths due to the serial use of adders and incrementers. Simulation indicates a maximum clock speed of 25 Mhz. However, placing additional registers along these two paths will increase operating speed to 45 Mhz. Additional timing delays were caused by excessive fan-out of the clock signal. The clock routing algorithms embedded in Epoch do not balance skew between clock nets [Ref. 8]. Clock buffer trees as shown in Figure 26 are therefore required.

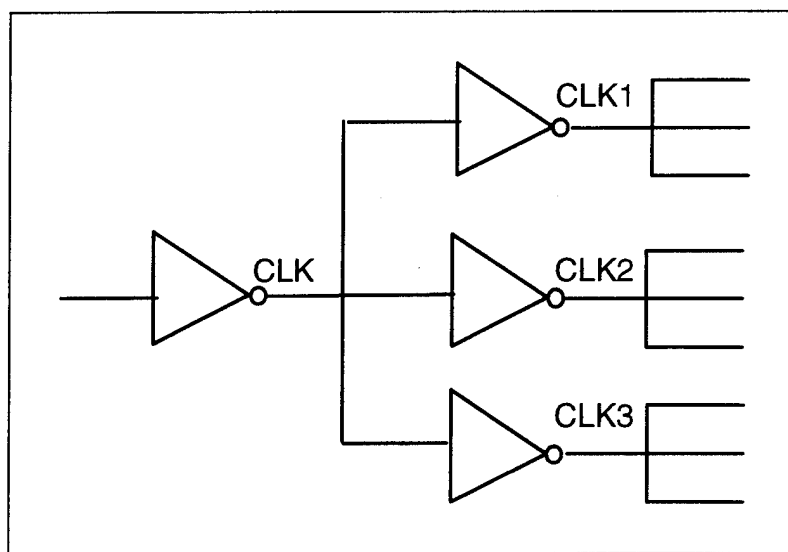


Figure 26 - Clock buffer tree

C. FINAL CHIP DESIGN

The final radix-4 complex floating point butterfly design contained over 434,000 transistors on a 271,000 square mil chip. Each floating point adder requires 10,000 sq. mils and 24,312 transitions. The floating point multipliers each cover 20,507 sq. mils and use over 55,000 transistors. Time restraints prevented simulations of the complete design, however the floating point adder and floating point multiplier portions were tested extensively. Appendix A contains the specifications for the major design components. Appendix C and D contain the Verilog HDL compilation and Epoch autocompilation reports. Appendix G contain color plates of the chip layout and floorplan. Appendix H contains Epoch generated specifications for a selected set of parts.

D. CHIP TESTING

Memory constraint prevented extensive testing of the Verilog HDL code for this design. Time limitations also contributed to an inability complete testing. Follow-on work should focus on either obtaining the memory resources required for VerilogOUT code testing or using the top-level only output option. Improvement efforts should focus on increasing the number of registers in the both the floating point adder and multiplier. The two most complicated modules , Rounder and Shiftsel, are also candidates for additional testing.

VI. CONCLUSIONS AND RECOMMENDATIONS

This project has resulted in the design and successful simulation of an IEEE standard, 32-bit, FFT Transformer. The final design uses 434,138 transistors in a 204 pin, 440 x 615 mil package. The 12-stage complex adder and 13-stage complex multiplier will operate at a 25 Mhz clock speed and dissipate .8 watts of power. This work has also shown the utility of the structured design environment afforded by the use of Verilog and Epoch. This software gives the individual complete control through all phases of a design. The fact that the bulk of this work was completed in 60 days attests to this software's capability.

Design improvement efforts should focus on increasing the number of registers in both the floating point adder and multiplier. Timing delays caused by the serial use of adders and incrementers has resulted in a clock speed slower than the 45 Mhz design goal. Dividing the Rounder function into two modules and including an additional pipeline stage will increase chip speed. Design changes aimed at avoiding the serial use of adders and incrementers in floating point multiplier stage five and floating point adder stage six should also be completed. A reduction in chip size will be realized once a .8 mμ rule set is made available for Epoch. A recompilation of the design with the Epoch "FIXEDBLOCK" attributes removed would result in a higher transistor density. This would result in further size reductions.

Although Verilog contains powerful constructs for the design engineer, its use is hampered by the lack of a graphical interface. Such interfaces have been developed by commercial vendors. A graphical interface would greatly improve efficiency when creating a large design.

APPENDIX A SPECIFICATION SUMMARY

GEOSTAT version 2.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading file fftchip ...

Print file root statistics.

Bounding Box: -559500, -782110, 559505, 782110.

11190.050 x 15642.200 microns, 175036999.240 square microns.

440.553 x 615.835 mils, 271307.906 square mils.

Number of Pins = 204.

Number of unique cells = 6.

Number of Blocks = 420

Number of Sub-Glues = 1

Total Number of Instances = 421

Total number of nets = 408.

Total metal1 layer route length = 1896987.84 microns.

Total metal2 layer route length = 397789.68 microns.

Total metal3 layer route length = 0.00 microns.

Total route length = 2294777.52 microns.

Total number of vias = 3918.

Total number of segments = 0.

Reading transistor view ...

Total number of 434138 transistors.

0.625 Square mils per Transistor.

1.600 Transistors per square mil.

Power Dissipation = 762555 micro-watts.

GEOSTAT version 2.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading file fft ...

Print file root statistics.

Bounding Box: -6520, -6750, 946430, 1427090.

9529.500 x 14338.400 microns, 136637786.522 square microns.

375.177 x 564.504 mils, 211789.000 square mils.

Number of Pins = 814.

Number of unique cells = 24.

Number of Standard cells = 684

Number of Sub-Glues = 2

Total Number of Instances = 686

Total number of nets = 958.

Total metal1 layer route length = 2019410.72 microns.

Total metal2 layer route length = 803834.56 microns.

Total metal3 layer route length = 0.00 microns.

Total route length = 2823245.28 microns.

Total number of vias = 25628.

Total number of segments = 0.

Reading transistor view ...

Total number of 433648 transistors.

0.488 Square mils per Transistor.

2.048 Transistors per square mil.

Power Dissipation = 708729 micro-watts.

GEOSTAT version 2.1.
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 Reading file complex_adder_4_input ...
 Print file root statistics.
 Bounding Box: -1540, -10630, 577745, 721510.
 5792.850 x 7321.400 microns, 42411772.139 square microns.
 228.065 x 288.244 mils, 65738.375 square mils.

 Number of Pins = 457.
 Number of unique cells = 3.
 Number of Standard cells = 10
 Number of Sub-Glues = 6
 Total Number of Instances = 16

 Total number of nets = 494.
 Total metal1 layer route length = 912923.76 microns.
 Total metal2 layer route length = 218564.56 microns.
 Total metal3 layer route length = 0.00 microns.
 Total route length = 1131488.32 microns.
 Total number of vias = 5936.
 Total number of segments = 0.
 Reading transistor view ...
 Total number of 145962 transistors.
 0.450 Square mils per Transistor.
 2.220 Transistors per square mil.
 Power Dissipation = 215625 micro-watts.

GEOSTAT version 2.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading file complex_mplier ...

Print file root statistics.

Bounding Box: -600, 0, 792530, 946430.

7931.300 x 9464.300 microns, 75064199.192 square microns.

312.256 x 372.610 mils, 116349.742 square mils.

Number of Pins = 797.

Number of unique cells = 4.

Number of Standard cells = 10

Number of Sub-Glues = 6

Total Number of Instances = 16

Total number of nets = 366.

Total metal1 layer route length = 944627.12 microns.

Total metal2 layer route length = 470926.20 microns.

Total metal3 layer route length = 0.00 microns.

Total route length = 1415553.32 microns.

Total number of vias = 7380.

Total number of segments = 0.

Reading transistor view ...

Total number of 269670 transistors.

0.431 Square mils per Transistor.

2.318 Transistors per square mil.

Power Dissipation = 434280 micro-watts.

GEOSTAT version 2.1.
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 Reading file fpadder ...
 Print file root statistics.
 Bounding Box: 0, 0, 230800, 279500.
 2308.000 x 2795.000 microns, 6450860.000 square microns.
 90.866 x 110.039 mils, 9998.853 square mils.

 Number of Pins = 146.
 Number of unique cells = 55.
 Number of Standard cells = 1300
 Number of Datapaths = 1
 Total Number of Instances = 1301

 Total number of nets = 1469.
 Total metal1 layer route length = 495568.28 microns.
 Total metal2 layer route length = 334620.30 microns.
 Total metal3 layer route length = 0.00 microns.
 Total route length = 830188.58 microns.
 Total number of vias = 28375.
 Total number of segments = 0.
 Reading transistor view ...
 Total number of 24312 transistors.
 0.411 Square mils per Transistor.
 2.431 Transistors per square mil.
 Power Dissipation = 35636 micro-watts.

GEOSTAT version 2.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kjackson/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading file fpmplier ...

Print file root statistics.

Bounding Box: -9240, -3040, 378445, 338725.

3876.850 x 3417.650 microns, 13249716.358 square microns.

152.632 x 134.553 mils, 20537.102 square mils.

Number of Pins = 262.

Number of unique cells = 66.

Number of Standard cells = 1077

Number of Datapaths = 2

Total Number of Instances = 1079

Total number of nets = 1272.

Total metal1 layer route length = 769149.68 microns.

Total metal2 layer route length = 368610.44 microns.

Total metal3 layer route length = 0.00 microns.

Total route length = 1137760.12 microns.

Total number of vias = 29776.

Total number of segments = 0.

Reading transistor view ...

Total number of 55239 transistors.

0.372 Square mils per Transistor.

2.690 Transistors per square mil.

Power Dissipation = 91287 micro-watts.

GEOSTAT version 2.1.
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 Reading file rounder ...
 Print file root statistics.
 Bounding Box: -1160, -490, 2560, 4165.
 37.200 x 46.550 microns, 1731.660 square microns.
 1.465 x 1.833 mils, 2.684 square mils.

 Number of Pins = 57.
 Number of unique cells = 8.
 Number of Standard cells = 58
 Total Number of Instances = 58

 Total number of nets = 114.
 Total metal1 layer route length = 0.00 microns.
 Total metal2 layer route length = 0.00 microns.
 Total metal3 layer route length = 0.00 microns.
 Total route length = 0.00 microns.
 Total number of vias = 0.
 Total number of segments = 0.
 Reading transistor view ...
 Total number of 701 transistors.
 0.004 Square mils per Transistor.
 261.170 Transistors per square mil.
 No power dissipation value found.

APPENDIX B VERILOG HDL CODE

```
module addertst;

// epoch set_attribute FIXEDBLOCK =1

    reg[1:0]          rnd_mode;    //rounding mode
    reg[31:0]        Ain,Bin;
    reg              clock,clear;

`include "header.v"

    integer i,j; parameter cycle=100;

    fpadder INSTAN (clear,clock,Ain,Bin,rnd_mode,regout5);

    initial clock=0 ;

    always #(cycle/2) clock=~clock; //generator

    initial begin
        $display("\n ***** 2x2=4
***** \n");

        #(cycle) Ain = 32'h40000000; Bin =32'h40000000; rnd_mode =0; clear =
1;

        $display("\n ***** Positive X Positive (Normalized) /pos + pos exp
***** \n");

        #(cycle) Ain = 32'h71800000; Bin =32'h40000000; rnd_mode =0; clear =
1;

        #(cycle) Ain = 32'h73c00000; Bin =32'h40000000; rnd_mode =0; clear =
1;

        $display("\n ***** Pos x Pos / neg +neg exp
***** \n");
```

```

#(cycle) Ain = 32'h03800000; Bin =32'h03800000; rnd_mode =0; clear =
1;

$display("\n *****Positive (Norm) X Positive (Unnorm)
***** \n");

#(cycle) Ain = 32'h00800000; Bin =32'h00400000; rnd_mode =0; clear =
1;

$display("\n *****Positive X Positive (Unnorm)
***** \n");

#(cycle) Ain = 32'h00400000; Bin =32'h00400000; rnd_mode =0; clear =
1;

$display("\n ***** Negative X Positive
***** \n");

#(cycle) Ain = 32'h81811100; Bin =32'h01800000; rnd_mode =0; clear =
1;

$display("\n ***** Negative X Negative (underflow)
***** \n");

#(cycle) Ain = 32'h81800111; Bin =32'h81811100; rnd_mode =0; clear =
1;

$display("\n ***** Negative x Positive
***** \n");

#(cycle) Ain = 32'h44001100; Bin =32'h83800000; rnd_mode =0; clear =
1;

$display("\n ***** 2x2=4
***** \n");

#(cycle) Ain = 32'h44000000; Bin =32'hde000000; rnd_mode =0; clear =
1;

$display("\n ***** Negative x Negative
***** \n");

#(cycle) Ain = 32'hde000000; Bin =32'hde000000; rnd_mode =0; clear = 1;

```

```

$display("\n ***** Denor x Denorm
***** \n");

#(cycle) Ain = 32'h00100000; Bin =32'h00100000; rnd_mode =0; clear =
1;

$display("\n ***** Inf x pos ***** \n");

#(cycle) Ain = 32'h7f800010; Bin =32'h40000010; rnd_mode =0; clear = 1;

#(cycle) Ain = 32'h7fc11111; Bin =32'h40011111; rnd_mode =0; clear = 1;
#(cycle) Ain =32'h00000000; Bin = 32'h00000000; rnd_mode=0;
#(4*cycle);
#cycle $finish;

end

initial begin
    $monitor ("%0d ", $time,, "clock: %b \n\n A sign: %b \n A mantissa: %b \n A
exponent: %b", clock,Ain[31],Ain[22:0],Ain[30:23],
    "\n\n B sign: %b \n B mantissa: %b \n B exponent:
%b",Bin[31],Bin[22:0],Bin[30:23],
    "\n\n Res sign: %b \n Res mantissa: %b \n Res exponent:
%b",regout5[31],regout5[22:0],regout5[30:23],

    "\n\n Rounding Mode: %b Clear: %b ",rnd_mode,clear,
    "\n\n NAN: %b INF: %b OVFL: %b UNFL: %b INEXACT: %b
\n\n",regout5[35],regout5[36],regout5[33],regout5[32],regout5[34]);

end

endmodule

```

```
module bitshift (in,out,lost_bit,exp_add);
```

```
//epoch pre_compiled bitshift
```

```
input [25:0] in; //input operand
```

```
output [25:0] out; //result operand  
output lost_bit,exp_add;
```

```
reg [25:0] out;  
reg lost_bit,exp_add;
```

```
always @ (in) begin
```

```
    out=shift(in);
```

```
    if (in[25]==1) begin  
        lost_bit=in[0];  
        exp_add=1;  
    end
```

```
    else begin  
        lost_bit=0;  
        exp_add=0;  
    end
```

```
end
```

```
// shift function
```

```
function [25:0] shift;
```

```
    input [25:0] in;
```

```
    if (in[25]==1)  
        shift={1'b0,in[25:1]};
```

```
    else  
        shift=in;
```

```
endfunction
```

```
endmodule
```



```
// Parameter Declaration
```

```
parameter    NUMBER_SIZE = 32;
parameter    CYCLE= 10;
parameter    EXP_SIZE = 8;
parameter    MANTISSA_SIZE = 23;
parameter    COMPLEX_NUMBER_SIZE=(2*NUMBER_SIZE);
parameter    INSTRUCTION_SIZE=8;
parameter    STATUS_REG_SIZE=8;
```

```
//Register declarations
```

```
reg    [COMPLEX_NUMBER_SIZE-1:0] Reg_A, Reg_B,
                                             Pre_1, Pre_2, Pre_3,
                                             Latch_0,Latch_1,Latch_2,Latch_3,
                                             Result;
```

```
reg    [INSTRUCTION_SIZE-1:0]    ir;
reg    [STATUS_REG_SIZE-1:0]      fpsr;
reg                                         reset;
```

```
//Instruction Fields
```

```
`define OPCODE    ir[3:0];
`define    VALID          ir[4];
`define PROCESS    ir[5];
```

```
// Define opcodes for each instruction
```

```
`define NOP    4'b0000
`define STALL    4'b0001
`define FLUSH    4'b0010
`define CADD 4'b0011
`define CMUL 4'b0100
`define FFT4  4'b0101
```

```

module complex_mplier (clear,clock, A_Real,A_Imag, B_Real,B_Imag,
rnd_mode,Prod_Real,Prod_Imag,FPSR);

//epoch set_attribute FIXEDBLOCK=1
//epoch pre_compiled complex_mplier

parameter    NUMBER_SIZE=32;

input  clear,clock;
input  [NUMBER_SIZE-1:0]  A_Real,A_Imag,B_Real,B_Imag;
input  [1:0]              rnd_mode;

output [NUMBER_SIZE-1:0]   Prod_Real,Prod_Imag;
output [4:0]              FPSR;

wire   [36:0] FP1_out,FP2_out,FP3_out,FP4_out,Result_Imag,Result_Real;

wire   M_ovf,M_unf,M_inexact,M_nan,M_inf,ovfl,unfl,inexact,nan,inf;

reg    [31:0] Prod_Real,Prod_Imag;
reg    [4:0]  FPSR;

//epoch set_attribute FIXEDBLOCK=1

fpmplier FP_Multiplier1 (clear,clock,A_Real,B_Real,rnd_mode,FP1_out),
          FP_Multiplier2 (clear,clock,A_Real,B_Imag,rnd_mode,FP2_out),
          FP_Multiplier3 (clear,clock,A_Imag,B_Real,rnd_mode,FP3_out),
          FP_Multiplier4 (clear,clock,A_Imag,B_Imag,rnd_mode,FP4_out);

fpadder   FP_Adder1 (clear, clock,
FP2_out[31:0],FP3_out[31:0],rnd_mode,Result_Imag),
          FP_Adder2 (clear, clock, FP1_out[31:0], FP4_out[31:0],rnd_mode,Result_Real);

stdor4 or1(FP1_out[32],FP2_out[32],FP3_out[32],FP4_out[32],M_ovf),
        or2(FP1_out[33],FP2_out[33],FP3_out[33],FP4_out[33],M_unf),
        or3(FP1_out[34],FP2_out[34],FP3_out[34],FP4_out[34],M_inexact),
        or4(FP1_out[35],FP2_out[35],FP3_out[35],FP4_out[35],M_nan),
        or5(FP1_out[36],FP2_out[36],FP3_out[36],FP4_out[36],M_inf);

stdor3 or6(Result_Imag[32],Result_Real[32],M_ovf,ovfl),

```

```

    or7(Result_Imag[33],Result_Real[33],M_unf,unfl),
    or8(Result_Imag[34],Result_Real[34],M_inexact,inexact),
    or9(Result_Imag[35],Result_Real[35],M_nan,nan),
    or10(Result_Imag[36],Result_Real[36],M_inf,inf);

always @ (Result_Imag or Result_Real) begin
    Prod_Real=Result_Real[31:0];
    Prod_Imag=Result_Imag[31:0];
    FPSR={inf,nan,inexact,unfl,ovfl};
end

endmodule

```

```

module
complex_adder_4_input(clear,clock,rnd_mode,A_Real,A_Imag,B_Real,B_Imag,C_Real,C
_Imag,D_Real,D_Imag,Result_Real,Result_Imag,FPSR);

//epoch set_attribute FIXEDBLOCK=1
// epoch pre_compiled complex_adder_4_input

`define      Result1      Result1[36:32]
`define      Result2      Result2[36:32]
`define      Result3      Result3[36:32]
`define      Result4      Result4[36:32]
`define      Result5      Result5[36:32]
`define      Result6      Result5[36:32]

parameter    NUMBER_SIZE=32;

input  clear,clock;
input  [NUMBER_SIZE-1:0]  A_Real,A_Imag,B_Real,B_Imag,
                        C_Real,C_Imag,D_Real,D_Imag;

input  [1:0]      rnd_mode;

output [NUMBER_SIZE-1:0]  Result_Real,Result_Imag;
output [4:0]              FPSR;

reg  [NUMBER_SIZE-1:0]  Result_Real,Result_Imag;
wire [4:0]              FPSR;

wire [36:0] Result1,Result2,Result3,Result4,Result5,Result6;

wire [4:0]  Partial_FPSR;

wire  M_ovf,M_unf,M_inexact,M_nan,M_inf,ovfl,unfl,inexact,nan,inf;

reg  [31:0] Prod_Real,Prod_Imag;

always @ (clock) begin
    Result_Real= Result5[31:0];
    Result_Imag= Result6[31:0];
end

//epoch set_attribute FIXEDBLOCK=1

```

```

fpadder      FP_Adder1 (clear, clock, A_Real, B_Real, rnd_mode, Result1),
             FP_Adder2 (clear, clock, A_Imag, B_Imag, rnd_mode, Result2),
             FP_Adder3 (clear, clock, C_Real, D_Real, rnd_mode, Result3),
             FP_Adder4 (clear, clock, C_Imag, D_Imag, rnd_mode, Result4),
             FP_Adder5 (clear, clock, Result1[31:0], Result3[31:0], rnd_mode, Result5),
             FP_Adder6 (clear, clock, Result2[31:0], Result4[31:0], rnd_mode, Result6);

stdor4 or1(Result1[32], Result2[32], Result3[32], Result4[32], Partial_FPSR[0]),
      or2(Result1[33], Result2[33], Result3[33], Result4[33], Partial_FPSR[1]),
      or3(Result1[34], Result2[34], Result3[34], Result4[34], Partial_FPSR[2]),
      or4(Result1[35], Result2[35], Result3[35], Result4[35], Partial_FPSR[3]),
      or5(Result1[36], Result2[36], Result3[36], Result4[36], Partial_FPSR[4]);

stdor3 or6(Result5[32], Result6[32], Partial_FPSR[0], FPSR[0]),
      or7(Result5[33], Result6[33], Partial_FPSR[1], FPSR[1]),
      or8(Result5[34], Result6[34], Partial_FPSR[2], FPSR[2]),
      or9(Result5[35], Result6[35], Partial_FPSR[3], FPSR[3]),
      or10(Result5[36], Result6[36], Partial_FPSR[4], FPSR[4]);

endmodule

```

```

module encoder (in,out);

//epoch pre_compiled encoder

input [47:0]  in;
output [4:0]  out;

reg  [4:0]  out,a;
reg  [47:0] h;

always @(in or h or a) begin

    out=encode(in);
end

function [4:0] encode;

    input [47:0]  in;

    begin
        h[47]=in[47];

        h[46]=in[46] & ~in[47];

        h[45]=in[45] & ~in[46] & ~in[47];

        h[44]=in[44] & ~in[45] & ~in[46] & ~in[47];

        h[43]=in[43] & ~in[44] & ~in[45] & ~in[46] & ~in[47];

        h[42]=in[42] & ~in[43] & ~in[44] & ~in[45] & ~in[46] & ~in[47];

        h[41]=in[41] & ~in[42] & ~in[43] & ~in[44] & ~in[45] & ~in[46]
        & ~in[47];

        h[40]=in[40] & ~in[41] & ~in[42] & ~in[43]
        & ~in[44] & ~in[45] & ~in[46] & ~in[47] ;

        h[39]=in[39] & ~in[40] & ~in[41] & ~in[42] & ~in[43]
        & ~in[44] & ~in[45] & ~in[46] & ~in[47] ;

        h[38]=in[38] & ~in[39] & ~in[40] & ~in[41] & ~in[42] & ~in[43]

```

[illegible]


```
h[12]=in[12] & ~in[13] & ~in[14] & ~in[15] & ~in[16] & ~in[17]
& ~in[18] & ~in[19] & ~in[20] & ~in[21] & ~in[22] & ~in[23]
& ~in[24] & ~in[25] & ~in[26] & ~in[27] & ~in[28] & ~in[29]
& ~in[30] & ~in[31] & ~in[32] & ~in[33] & ~in[34] & ~in[35]
& ~in[36] & ~in[37] & ~in[38] & ~in[39] & ~in[40] & ~in[41]
& ~in[42] & ~in[43] & ~in[44] & ~in[45] & ~in[46] & ~in[47]
;
```

```
h[10]=in[10] & ~in[11] & ~in[12] & ~in[13] & ~in[14] & ~in[15]
& ~in[16] & ~in[17] & ~in[18] & ~in[19] & ~in[20] & ~in[21]
& ~in[22] & ~in[23] & ~in[24] & ~in[25] & ~in[26] & ~in[27]
& ~in[28] & ~in[29] & ~in[30] & ~in[31] & ~in[32] & ~in[33]
& ~in[34] & ~in[35] & ~in[36] & ~in[37] & ~in[38] & ~in[39]
& ~in[40] & ~in[41] & ~in[42] & ~in[43] & ~in[44] & ~in[45]
& ~in[46] & ~in[47];
```

```
h[8]=in[8] & ~in[9] & ~in[10] & ~in[11] & ~in[12] & ~in[13]
& ~in[14] & ~in[15] & ~in[16] & ~in[17] & ~in[18] & ~in[19]
& ~in[20] & ~in[21] & ~in[22] & ~in[23] & ~in[24] & ~in[25]
& ~in[26] & ~in[27] & ~in[28] & ~in[29] & ~in[30] & ~in[31]
& ~in[32] & ~in[33] & ~in[34] & ~in[35] & ~in[36] & ~in[37]
& ~in[38] & ~in[39] & ~in[40] & ~in[41] & ~in[42] & ~in[43]
& ~in[44] & ~in[45] & ~in[46] & ~in[47];
```

61


```

& ~in[26] & ~in[27] & ~in[28] & ~in[29] & ~in[30] & ~in[31] &
~in[32] & ~in[33] & ~in[34] & ~in[35] & ~in[36] & ~in[37]
& ~in[38] & ~in[39] & ~in[40] & ~in[41] & ~in[42] & ~in[43]
& ~in[44] & ~in[45] & ~in[46] & ~in[47] ;

h[1]=in[1] & ~in[2] & ~in[3] & ~in[4] & ~in[5] & ~in[6]
& ~in[7] & ~in[8] & ~in[9] & ~in[10] & ~in[11] & ~in[12]
& ~in[13] & ~in[14] & ~in[15] & ~in[16] & ~in[17] & ~in[18]
& ~in[19] & ~in[20] & ~in[21] & ~in[22] & ~in[23] & ~in[24]
& ~in[25] & ~in[26] & ~in[27] & ~in[28] & ~in[29] & ~in[30] & ~in[31]
&
~in[32] & ~in[33] & ~in[34] &
~in[35] & ~in[36] &
~in[37] & ~in[38] & ~in[39] & ~in[40] & ~in[41] &
~in[42] & ~in[43] & ~in[44] & ~in[45] & ~in[46] &
~in[47] ;

h[0]=in[0] & ~in[1] & ~in[2] & ~in[3] & ~in[4] & ~in[5] &
~in[6] & ~in[7] & ~in[8] & ~in[9] & ~in[10] & ~in[11] &
~in[12] & ~in[13] & ~in[14] & ~in[15] & ~in[16] & ~in[17] &
~in[18] & ~in[19] & ~in[20] & ~in[21] & ~in[22] & ~in[23] &
~in[24] & ~in[25] & ~in[26] & ~in[27] & ~in[28] & ~in[29] &
~in[30] & ~in[31] & ~in[32] & ~in[33] & ~in[34] & ~in[35] &
~in[36] & ~in[37] & ~in[38] & ~in[39] & ~in[40] & ~in[41] &
~in[42] & ~in[43] & ~in[44] & ~in[45] & ~in[46] & ~in[47] ;

a[0]=h[1]lh[3]lh[5]lh[7]lh[9]lh[11]lh[13]lh[15]lh[17]lh[19]l
h[21]lh[23]lh[25]lh[27]lh[29]lh[31]lh[33]lh[35]lh[37]l
h[39]lh[41]lh[43]lh[45]lh[47];

a[1]=h[46]lh[45]lh[42]lh[41]lh[38]lh[37]lh[34]lh[33]l
h[30]lh[29]lh[26]lh[25]lh[22]lh[21]lh[18]lh[17]lh[14]
lh[13]lh[10]lh[9]lh[6]lh[5]lh[2]lh[1];

a[2]=h[44]lh[43]lh[42]lh[41]lh[36]lh[35]lh[34]lh[33]lh[28]lh[27]lh[26]l
h[25]lh[20]lh[19]lh[18]lh[16]lh[12]lh[11]lh[10]lh[9]lh[4]lh[3]lh[2]lh[1];

a[3]=h[40]lh[39]lh[38]lh[37]lh[36]lh[35]lh[33]lh[32]lh[23]lh[22]lh[21]l
h[20]lh[19]lh[18]lh[17]lh[16]lh[7]lh[6]lh[5]lh[4]lh[3]lh[2]lh[1]lh[0];

a[4]=h[32]lh[31]lh[30]lh[29]lh[28]lh[27]lh[26]lh[25]lh[24]lh[23]lh[22]l

```

```
h[21]lh[20]lh[19]lh[18]lh[17];
```

```
encode=a;
```

```
end
```

```
endfunction
```

```
endmodule
```

```

module fft (clear,clock_in,ready,load,rnd_mode,Ain,Bin,Cout,FPSR);

// epoch set_attribute FIXEDBLOCK =1
//epoch pre_compiled fft

parameter    NUMBER_SIZE=64;

input                clear,clock_in,ready,load;
input  [1:0]          rnd_mode;
input  [NUMBER_SIZE-1:0]    Ain,Bin;

output [NUMBER_SIZE-1:0]    Cout;
output [4:0]                FPSR;

wire [NUMBER_SIZE-1:0]  in,regout1,regout2,regout3,regout4,
                        A,B,C,D,negB,negC,negD,jB,jD,
                        negjB,negjD;
wire [NUMBER_SIZE-1:0]  S2,S3,S4;
wire                    strobe;
wire  [1:0]             mux_sel;
wire  [127:0]           latch_out_real,latch_out_imag;
wire  [4:0]             Mply_FPSR,Adder_FPSR,FPSR1,FPSR2,FPSR3,FPSR4,FPSR5;

`define      group "fft_top"
`define      imptype 0
`define      A_real latch_out_real[127:96]
`define      A_imag      latch_out_imag [127:96]
`define      B_real latch_out_real [95:64]
`define      B_imag      latch_out_imag [95:64]
`define      C_real latch_out_real [64:33]
`define      C_imag      latch_out_imag [64:33]
`define      D_real latch_out_real [32:0]
`define      D_imag      latch_out_imag [32:0]
`define      A      {latch_out_real[127:96],latch_out_imag [127:96]}
`define      B      {latch_out_real [95:64],latch_out_imag [95:64]}
`define      C      {latch_out_real [64:33],latch_out_imag [64:33]}
`define      D      {latch_out_real [32:0],latch_out_imag [32:0]}
//define      negA
      {~latch_out_real[127],latch_out_real[126:97],~latch_out_imag[127],latch_out_im
ag[126:97]}

```

```

`define      negB
    {~latch_out_real[96],latch_out_real[95:65],~latch_out_imag[96],latch_out_imag[
95:65]
`define      negC
    {~latch_out_real[64],latch_out_real[63:33],~latch_out_imag[64],latch_out_imag[
63:33]
`define      negD
    {~latch_out_real[32],latch_out_real[31:0],~latch_out_imag[32],latch_out_imag[3
1:0]
// define      jA
    {~latch_out_imag[127],latch_out_imag[126:97],latch_out_real[127:96]}
`define      jB      {~latch_out_imag[96],latch_out_imag[95:65],latch_out_real
[95:64]}
// define      jC      {~latch_out_imag[64],latch_out_imag[63:33],latch_out_real
[64:33]}
`define      jD      {~latch_out_imag[32],latch_out_imag[31:0],latch_out_real [32:0]}
// define      negjA  {latch_out_imag
[127:96],~latch_out_real[127],latch_out_real[126:97]}
`define      negjB  {latch_out_imag
[95:64],~latch_out_real[96],latch_out_real[95:65]}
// define      negjC  {latch_out_imag
[64:33],~latch_out_real[64],latch_out_real[63:33]}
`define      nejD   {latch_out_imag [32:0],~latch_out_real[32],latch_out_real[31:0]}

//epoch set_attribute FIXEDBLOCK=1

complex_mplier
Complex_Multiplier(clear,clock,Ain[63:32],Ain[31:0],Bin[63:32],Bin[31:0],rnd_mode,in[
63:32],in[31:0],Mply_FPSR);
complex_adder_4_input
Complex_Adder_4_Input(clear,clock,rnd_mode,A[63:32],A[31:0],S2[63:32],S2[31:0],S3
[63:32],S3[31:0],S4[63:32],S4[31:0],Cout[63:32],Cout[31:0],Adder_FPSR);

stdand2      and1 (clock_in,ready,clock);

dff_c # (64, `imptype, `group)
    reg1 (clock, clear, in,regout1),
    reg2 (clock, clear, regout1,regout2),
    reg3 (clock, clear, regout2,regout3);

latch_c #(128, `imptype, `group)
    Latch1(clear,{in[63:32],regout1[63:32],regout2[63:32],regout3[63:32]},strobe,lat
ch_out_real),

```

```
Latch2(clear,{in[31:0],regout1[31:0],regout2[31:0],regout3[31:0]},strobe,latch_o  
ut_imag);
```

```
sctrl # (2,`imptype,`group)  
Counter (clock, 2'b00,1'b1,clear,strobe,mux_sel);
```

```
mux4 # (64,`imptype,`group)  
Mux1 (B,negjB,negB,jB,mux_sel[0],mux_sel[1],S2),  
Mux2 (C,negC,C,negC,mux_sel[0],mux_sel[1],S3),  
Mux3 (D,jD,negD,negjD,mux_sel[0],mux_sel[1],S4);
```

```
dff_c # (5,`imptype,`group)  
FPSReg1 (clock,clear,Mply_FPSR,FPSR1),  
FPSReg2 (clock,clear,FPSR1,FPSR2),  
FPSReg3 (clock,clear,FPSR2,FPSR3),  
FPSReg4 (clock,clear,FPSR3,FPSR4),  
FPSReg5 (clock,clear,FPSR4,FPSR5);
```

```
stdor2 or1 (FPSR5[0],Adder_FPSR[0],FPSR[0]),  
or2 (FPSR5[1],Adder_FPSR[1],FPSR[1]),  
or3 (FPSR5[2],Adder_FPSR[2],FPSR[2]),  
or4 (FPSR5[3],Adder_FPSR[3],FPSR[3]),  
or5 (FPSR5[4],Adder_FPSR[4],FPSR[4]);
```

```
endmodule
```



```

/* FFT Chip structural description
** fftchip.v
*/

`define slim 0
`define pitch "4MA"
`define drive "4MA"
`define numbits 64
`define powertype "REGULAR"

module fftchip (gndpad, vddpad, clearpad, clockpad, readypad, loadpad, rnd_modepad,
Ainpad, Binpad, Coutpad, FPSRpad);

input          clearpad, clockpad, readypad, loadpad;
input [1:0]     rnd_modepad;
input [numbits-1:0] Ainpad, Binpad;
output [numbits-1:0] Coutpad;
output [4:0]    FPSRpad;

wire  clear, clock, ready, load;
wire [1:0] rnd_mode;
wire [numbits-1:0] Ain, Bin, Cout;
wire [4:0] FPSR;

supply1      Vdd;
supply0      GND;

padin_unbuf #(1,64, `slim, `pitch)
    inpad0 (Ainpad, Ain);
padin_unbuf #(66, 129, `slim, `pitch)
    inpad1 (Binpad, Bin);
padin_unbuf #(130, 131, `slim, `pitch)
    inpad2 (rnd_modepad, rnd_mode);
padin_unbuf #(132,132, `slim, `pitch)
    inpad3 (loadpad, load);
padin_unbuf #(133, 133, `slim, `pitch)
    inpad4 (readypad, ready);
padin_unbuf #(134, 134, `slim, `pitch)
    inpad5 (clockpad, clock);
padin_unbuf #(135, 135, `slim, `pitch)
    inpad6 (clearpad, clear);
padout #(137, 200, `slim, `drive)
    outpad1 (Cout, Coutpad);
padout #(201, 205, `slim, `drive)
    outpad2 (FPSR, FPSRpad);

```

```
padgnd #(65, `slim, `powertype, `pitch)
    pwrpad1 (gndpad);
padvdd #(136, `slim, `powertype, `pitch)
    pwrpad2 (vddpad);

fft
    chip(clear, clock, ready, load, rnd_mode, Ain, Bin, Cout, FPSR);

endmodule
```

```

module fpadder (clear,clock,Ain,Bin,rnd_mode,regout5);
//epoch set_attribute FIXEDBLOCK =1
//epoch pre_compiled fpadder

//parameters

parameter    NUMBER_SIZE=32;
parameter    EXP_SIZE=8;
parameter    MANTISSA_SIZE=23;

`define      group    "fpadder_group"
`define      imptype  0

input  clear,clock;
input  [NUMBER_SIZE-1:0] Ain,Bin;
input  [1:0]          rnd_mode;

output [36:0] regout5;

wire  sign,sign1,sign2;
wire  sign_A, sign_B, nan_A,nan_B;
wire  inf_A, inf_B,inf_fin;
wire  ovfl2,unfl1,unfl2,unfl3,rndadd,norout,exp_gr,exp_add,
sft_gr,equal,sft_dir,unfl,inf1,nan1,inexact,ovfl,A_man_gr,B_man_gr,
A_exp_gr,B_exp_gr,inexact_sft,lo_shift,hi_shift,inexact_shift;

wire  [71:0] regout1;
wire  [70:0] regout2;
wire  [70:0] regout2i;
wire  [88:0] regout3;
wire  [44:0] regout4;
wire  [36:0] regout5;

wire [EXP_SIZE-1:0] exp_A, exp_B,exp_fin,exp_fin2,exp,
exp_out,gr_exp,ls_exp,neg_exp,shift;
wire [MANTISSA_SIZE+1:0] man_A,man_B,man_A2,man_B2,gr_man,ls_man;

wire [MANTISSA_SIZE-1:0] man_fin,man_fin2;

wire [25:0] man_out,man_out2,man_inv,man_out3,man_out_comp;
wire [49:0] ls_man_out,ls_man_out2;

```

```

wire nc; // not connected
wire [7:0]    neg127=8'b10000001; //negative 127

supply1 vdd;
supply0 gnd;


vcheck Value_Test_A (Ain,nan_A,inf_A,exp_A,man_A,sign_A),
        Value_Test_B (Bin,nan_B,inf_B,exp_B,man_B,sign_B);


stdor2 or1 (inf_A, inf_B,inf1),
        or2 ( nan_A, nan_B,nan1);


inverter man_invA (regout1[65:41],regout1[71],man_A2),
        man_invB (regout1[40:16],regout1[70],man_B2);


mux2 # (8,`imptype,`group)
        Exp_mux1 (regout1[15:8],regout1[7:0],~man_sel,gr_exp),
        Exp_mux2 (regout1[15:8],regout1[7:0],man_sel,ls_exp);


mux2 # (25,`imptype,`group)
        Man_mux1 (man_A2,man_B2,~man_sel,gr_man),
        Man_mux2 (man_A2,man_B2,man_sel,ls_man);


mux2 # (1,`imptype,`group)
        Sign_mux (regout1[71],regout1[70],sign2,sign);


stdand2 and1 (A_exp_gr,~B_exp_gr,man_sel),
        and2 (~A_man_gr,~A_exp_gr,sign1),
        and4 (regout2i[4],regout2i[5],hi_shift),
        and5 (hi_shift,lo_shift,inexact_shift);


mcomp # (8,`imptype,`group)
        E_Comparator (regout1[15:8],regout1[7:0],A_exp_gr,B_exp_gr);


mcomp # (25,`imptype,`group)
        M_Comparator (regout1[66:42],regout1[41:17],A_man_gr,B_man_gr);

```

```

compl #(8,`imptype,`group)
    exp_compl (1'b0,ls_exp,nc,nc,neg_exp);

addhs # (8,1,`group)
    Adder1(regout2[7:0], regout2[15:8],gnd,nc,shift);

barrelright # (50,6,`imptype,`group)
    shifter ({regout2i[40:16],25'b0},regout2i[5:0], ls_man_out);

overshift overshift (ls_man_out,inexact_sft,ls_man_out2);

addhs # (26,1,`group)
    Adder2(regout3[82:57], regout3[57:32],gnd,end_arnd_add,man_out);

stdor2 //or12 (regout3[30],regout3[29],inex1),

    or11 (regout4[42],ovfl4,ovfl),
    or14 (sign1,B_exp_gr,sign2),
    or17 (regout4[40],inf_out,inf_fin);

stdor3 or15 (inexact_shift,regout2i[6],regout2i[7],inexact_sft),
    or9 (inexact1,lost_bit1,regout3[88],inexact2),
    or7 (regout3[8],inex2,inex3,inex7);

stdor4 or13 (regout3[28],regout3[27],regout3[26],regout3[25],inex2),
    or3 (regout3[24],regout3[23],regout3[22],regout3[21],inex3),
    or4 (regout3[20],regout3[19],regout3[18],regout3[17],inex4),
    or5 (regout3[16],regout3[15],regout3[14],regout3[13],inex5),
    or6 (regout3[12],regout3[11],regout3[10],regout3[9],inex6),
    or8 (inex4,inex5,inex6,inex7,inexact1),
    or16 (regout2i[0],regout2i[1],regout2i[2],regout2i[3],lo_shift);

inc # (26,`imptype,`group)
    Incrementer1(end_arnd_add,man_out,nc,nc,man_out2);

bitshift Bitshift1 (man_inv,man_out3,lost_bit1,exp_add);

inc # (8,`imptype,`group)
    Incrementer2(exp_add,regout3[7:0],ovfl2,nc,exp_out);

inverter2 man_out_inv (man_out2,regout3[87],man_inv);

```

```

rounder Rounder
({regout4[32:8],regout4[43:42]},man_fin,regout4[34:33],regout4[40],rndadd,denorm);

inc # (8,`imptype,`group)
    Incrementer3(rndadd,regout4[7:0],ovfl4,nc,exp_fin);

zero # (23,`imptype,`group)
    Zero_Dect1(man_fin,zero1);

zero # (8,`imptype,`group)
    Zero_Dect2(exp_fin,zero2);

stdand3 and3(zero1,zero2,regout4[35],unfl);

dff_c # (72,`imptype,`group)
    reg1
(clock,clear,{sign_A,sign_B,inf1,nan1,rnd_mode,man_A,man_B,exp_A,exp_B},
    regout1);

dff_c # (71,`imptype,`group)
    reg2 (clock,clear,{sign,regout1[69:66],gr_man,ls_man,gr_exp,neg_exp},
        regout2),
    reg2i(clock, clear, {regout2[70:8],shift},regout2i);

dff_c # (89,`imptype,`group)
    reg3 (clock,clear,{inexact_sft,regout2i[70:41],ls_man_out2,regout2i[15:8]},
        regout3);

dff_c # (45,`imptype,`group)
    reg4
(clock,clear,{lost_bit1,regout3[31],ovfl2,regout3[87:83],inexact2,regout3[67:66]
    ,man_out3,exp_out},regout4);

dff_c # (37,`imptype,`group)
    reg5 (clock,clear,{inf_fin,regout4[39],regout4[35],ovfl,unfl,regout4[41],
        exp_fin2,man_fin2},regout5);

mulclnup cleanup (man_fin,exp_fin,denorm,unfl,ovfl,man_fin2,exp_fin2,inf_out);

endmodule

```

```

module fpmplier (clear,clock,Ain,Bin,rnd_mode,regout5);

//epoch set_attribute FIXEDBLOCK=1
//epoch pre_compiled fpmplier

parameter    NUMBER_SIZE=32;
parameter    EXP_SIZE=8;
parameter    MANTISSA_SIZE=23;

input  clear,clock;
input  [NUMBER_SIZE-1:0]  Ain,Bin;
input  [1:0]              rnd_mode;

output [36:0] regout5;

`define      sign_A Ain[NUMBER_SIZE-1]
`define      sign_B Bin[NUMBER_SIZE-1]
`define      exp_A  Ain[NUMBER_SIZE-2:NUMBER_SIZE-2-EXP_SIZE]
`define      exp_B  Bin[NUMBER_SIZE-2:NUMBER_SIZE-2-EXP_SIZE]
`define      man_A  Ain[NUMBER_SIZE-1-EXP_SIZE:0]
`define      man_B  Bin[NUMBER_SIZE-1-EXP_SIZE:0]
//parameters

wire  sign,sign1;
wire  sign_A, sign_B, nan_A,nan_B;
wire  inf_A, inf_B,inf_fin;
wire  ovfl1,ovfl2,ovfl3,unfl1,unfl2,unfl3,rndadd,norout,exp_gr,
      sft_gr,equal,sft_dir,unfl,inf1,nan1,inexact,ovfl,ovfl4,ovfl_fin;

wire  [70:0] regout1;
wire  [14:0] regout2i;
wire  [14:0] regout2;
wire  [72:0] regout3;
wire  [68:0] regout4i;
wire  [64:0] regout4;
wire  [36:0] regout5;

wire [EXP_SIZE-1:0] exp_A, exp_B, exp_C, exp_D,exp_fin,exp,
                  alt_shift_sel,exp_out,exp_out2,exp_reduced;
wire [MANTISSA_SIZE+1:0]  man_A,man_B;

```

```

wire [((MANTISSA_SIZE+3)*2)-1:0]      prod;
wire [((MANTISSA_SIZE+1)*2)-1:0]      man_out,man_fin;
wire [MANTISSA_SIZE-1:0]              man_rnd,man;
wire [4:0]      shift,bs_sel,shift_neg,shift_m_2,shift_m_2_neg;

wire nc; // not connected
wire [7:0]      neg127=8'b10000001; //negative 127

supply1 vdd;
supply0 gnd;

vcheck Value_Test_A (Ain,nan_A,inf_A,exp_A,man_A,sign_A),
      Value_Test_B (Bin,nan_B,inf_B,exp_B,man_B,sign_B);

hsmult2piped_c #(MANTISSA_SIZE+3,MANTISSA_SIZE+3,52)
      Multiplier({ 1'b0,regout1[65:41]},{ 1'b0,regout1[40:16]},
      clock,clear,prod);

addhs # (8,1,"mplier")
      Adder1(regout1[15:8], regout1[7:0],gnd,nc,exp_C),
      Adder2(regout2[9:2],neg127,gnd,nc,exp_D),
      Adder3({ 3'b111,regout3[72:68]}, regout3[9:2],gnd,nc,
      alt_shift_sel),
      Adder5({ 3'b111,shift_m_2_neg},regout3[9:2],gnd,nc,exp_reduced);

addrip # (5,0,"mplier")
      Adder4(regout3[14:10],5'b11110,gnd,nc,shift_m_2);

rounder Rounder (regout4[62:36],man_rnd,regout4[12:11],
      regout4[13],rndadd,denorm);

inc # (8,0, "mplier")
      Incrementer(regout4[10],regout4[9:2],ovfl3,nc,exp_out),
      Incrementer2(rndadd,exp_out,ovfl4,nc,exp_out2);

```



```

stdxor2      xor1 (sign_A,sign_B,sign1);

stdnand2 nand1 (regout2[0],regout2[1],ovfl1); //indicate overflow
stdnor2      nor1 (ovfl1,exp_D[7],ovfl2);

stdnor2      nor2 (regout2[0],regout2[1],unfl1); //indicate underflow
stdand2      and2 (unfl1,exp_D[7],unfl2);

stdor2 or2 (ovfl3,regout4[1],ovfl),

            or4 (inf_A, inf_B,inf1),
            or5 ( nan_A, nan_B,nan1);

stdor2 or13 (regout4[10],inf2,inf_fin),
            or14 (ovfl,ovfl4,ovfl_fin);

stdor4 or7 (regout4[33],regout4[32],regout4[31],regout4[30],inex2),
        or8 (regout4[29],regout4[28],regout4[27],regout4[26],inex3),
        or9 (regout4[25],regout4[24],regout4[23],regout4[22],inex4),
        or10 (regout4[21],regout4[20],regout4[19],regout4[18],inex5),
        or11 (regout4[17],regout4[16],inex1,inex2,inex6),
        or12 (inex3,inex4,inex5,inex6,inexact),
        or6 (regout4[64],regout4[36],regout4[35],regout4[34],inex1);

barrelleft # (48,6,0,"mplier")
            shifter (regout4i[62:15],{1'b0,regout4i[14:10]}, man_out);

encoder encoder (prod[47:0],shift);

dff_c # (71,0,"mplier")
        reg1 (clock,clear,{inf1,nan1,sign1,rnd_mode,man_A,man_B,exp_A
            ,exp_B},regout1);

dff_c # (15,0,"mplier")
        reg2i (clock, clear, {regout1[70:66],exp_C,regout1[15],regout1[7]}
            ,regout2i),
        reg2 (clock, clear, regout2i,regout2);

```

```

dff_c # (73,0,"mplier")
    reg3 (clock, clear, {shift_neg,regout2[14:10],prod[47:0],shift,
        exp_D,ovfl2,unfl2},regout3);

dff_c # (69, 0, "mplier")
    reg4i (clock, clear, {sft_dir,regout3[67:15],bs_sel[4:0],exp_fin,
        regout3[1:0]},regout4i);

dff_c # (65,0, "mplier")
    reg4 (clock, clear, {lost_bit,man_fin,regout4i[66:62],regout4i[68],regout4i[9:0]},
regout4);

dff_c # (37, 0, "mplier")
    reg5 (clock, clear, {inf_fin,regout4[15],inexact,ovfl_fin,regout4[0]
        ,regout4[14],exp,man},regout5);

compl # (5,0,"mplier")
    Complementor (1'b0,shift,nc,nc,shift_neg),
    Complementor2 (1'b0,shift_m_2,nc,nc,shift_m_2_neg);

mcomp # (8,0,"mplier")
    Comparator ({3'b0,regout3[13:9]},regout3[8:1],sft_gr,exp_gr);

shiftsel shiftsel (regout3[14:10],alt_shift_sel,shift_m_2,sft_gr,exp_gr,
    regout3[9:2],exp_reduced,bs_sel,exp_fin,sft_dir);

mansel mansel (man_out,regout4i[68],man_fin,lost_bit);

mulclnup cleanup (man_rnd,exp_out2,denorm,regout4[0],ovfl,man,exp,inf2);

endmodule

```

```
module inverter (in,sign,out);

// epoch pre_compiled inverter

input sign;
input [24:0] in;

output [24:0] out;
reg [24:0] out;
always @ (in or sign) begin
    if (sign == 1)
        out=~in;
    else
        out=in;
    end
endmodule
```

```
module inverter2 (in,sign,out);

//epoch pre_compiled inverter2

input sign;
input [25:0] in;

output [25:0] out;
reg [25:0] out;
always @ (in or sign) begin
    if (sign == 1)
        out=~in;
    else
        out=in;
    end
endmodule
```

```

module mansel (manin,sft_dir,manout,lost_bit);

//epoch pre_compiled mansel

input [47:0]  manin;
input        sft_dir;
output [47:0] manout;
output       lost_bit;

reg  [47:0] manout;
reg       lost_bit;

always @(manin or sft_dir)
    begin
        if (sft_dir == 0)begin
            manout=manin;
            lost_bit=0;
        end
        else begin
            manout={ 1'b0,manin[47:1]};
            lost_bit=1;
        end
    end
endmodule

```

```
module mulcnp (man_rnd, exp, denorm, unfl, ovfl, man_out, exp_out, inf_out);
```

```
//epoch pre_compiled mulcnp
```

```
input [22:0]  man_rnd;    //rounded mantissa
```

```
input [7:0]   exp;        //exponent input
```

```
input         unfl,ovfl,denorm;    // underflow & overflow & denorm
```

```
output [7:0]  exp_out;    //output exponent
```

```
output [22:0] man_out;    //output mantissa
```

```
output       inf_out;     //infinity flag
```

```
reg [7:0]     exp_out;
```

```
reg [22:0]    man_out;
```

```
reg          inf_out;
```

```
always @ (man_rnd or ovfl or unfl or exp or denorm) begin
```

```
    man_out=man_check(man_rnd,ovfl,unfl);
```

```
    exp_out=exp_check(exp,ovfl,unfl);
```

```
    inf_out=inf_check(man_rnd,exp);
```

```
end
```

```
function [7:0] exp_check;
```

```
    input [7:0] exp;
```

```
    input     ovfl,unfl;
```

```
    begin
```

```
        if (unfl) begin
```

```
            exp_check=8'b00000000;
```

```
        end
```

```
        else if (ovfl) begin
```

```
            exp_check=8'b11111111;
```

```
        end
```

```
        else if (denorm) begin
```

```
            exp_check=8'b00000000;
```

```
        end
```

```

        else
            exp_check=exp;
        end
    endfunction

function [22:0] man_check;

    input [22:0] man_rnd;
    input      ovfl,unfl;
    begin
        if (unfl)
            man_check=23'b0;
        else if (ovfl)
            man_check=23'b0;
        else
            man_check=man_rnd;
        end
    endfunction

function inf_check;

    input [22:0] man_rnd;
    input [7:0]  exp;

    begin
        if (man_rnd==23'b0) begin
            inf_check=0;

            if (exp==8'b11111111)
                inf_check=1;
            end
        end
    endfunction

endmodule

```

```
module mplertst;
```

```
// epoch set_attribute FIXEDBLOCK =1
```

```
    reg[1:0]          rnd_mode;    //rounding mode
    reg[31:0]         Ain,Bin;
    reg               clock,clear;
```

```
`define      group "mplier"
`define      imptype 1
`define      sign_A Ain[NUMBER_SIZE-1]
`define      sign_B Bin[NUMBER_SIZE-1]
`define      exp_A Ain[NUMBER_SIZE-2:NUMBER_SIZE-2-EXP_SIZE]
`define      exp_B Bin[NUMBER_SIZE-2:NUMBER_SIZE-2-EXP_SIZE]
`define      man_A Ain[NUMBER_SIZE-1-EXP_SIZE:0]
`define      man_B Bin[NUMBER_SIZE-1-EXP_SIZE:0]
```

```
//parameters
```

```
parameter    NUMBER_SIZE=32;
parameter    EXP_SIZE=8;
parameter    MANTISSA_SIZE=23;
```

```
wire  sign,sign1;
wire  sign_A, sign_B, nan_A,nan_B;
wire  inf_A, inf_B,inf_fin;
wire
    ovfl1,ovfl2,ovfl3,unfl1,unfl2,unfl3,rndadd,norout,exp_gr,sft_gr,equal,sft_dir,unfl,in
    fl,nan1,inexact,ovfl;
```

```
wire  [70:0] regout1;
wire  [14:0] regout2;
wire  [72:0] regout3;
wire  [68:0] regout4;
wire  [36:0] regout5;
```

```
wire [EXP_SIZE-1:0] exp_A, exp_B, exp_C, exp_D,exp_fin,exp,alt_shift_sel,exp_out;
wire [MANTISSA_SIZE+1:0] man_A,man_B;
wire [((MANTISSA_SIZE+3)*2)-1:0] prod;
```



```

wire [((MANTISSA_SIZE+1)*2)-1:0]      man_out,man_fin;
wire [MANTISSA_SIZE-1:0]              man_rnd,man;
wire [4:0]      shift,bs_sel,shift_neg,shift_m_2;

integer i,j; parameter cycle=100;

fpmplier INSTAN (clear,clock,Ain,Bin,rnd_mode,regout5);

initial clock=0 ;

always #(cycle/2) clock=~clock; //generator

initial begin
    $display("\n ***** 2x2=4
***** \n");

    #(cycle) Ain = 32'h40000000; Bin =32'h40000000; rnd_mode =0; clear =
1;

    $display("\n ***** Positive X Positive (Normalized) /pos + pos exp
***** \n");

    #(cycle) Ain = 32'h71800000; Bin =32'h40000000; rnd_mode =0; clear =
1;

    #(cycle) Ain = 32'h73c00000; Bin =32'h40000000; rnd_mode =0; clear =
1;

    $display("\n ***** Pos x Pos / neg +neg exp
***** \n");

    #(cycle) Ain = 32'h03800000; Bin =32'h03800000; rnd_mode =0; clear =
1;

    $display("\n *****Positive (Norm) X Positive (Unnorm)
***** \n");

    #(cycle) Ain = 32'h40800000; Bin =32'h01400000; rnd_mode =0; clear =
1;

    $display("\n *****Positive X Positive (Unnorm)
***** \n");

```

```

1;

#(cycle) Ain = 32'h01400000; Bin =32'h40400000; rnd_mode =0; clear =

$display("\n ***** Negative X Positive
***** \n");

#(cycle) Ain = 32'h81811100; Bin =32'h40800000; rnd_mode =0; clear =

1;

$display("\n ***** Negative X Negative (underflow)
***** \n");

#(cycle) Ain = 32'hA1800111; Bin =32'hA1811100; rnd_mode =0; clear =

1;

$display("\n ***** Negative x Positive
***** \n");

#(cycle) Ain = 32'h44001100; Bin =32'h83800000; rnd_mode =0; clear =

1;

$display("\n ***** 2x2=4
***** \n");

#(cycle) Ain = 32'h44000000; Bin =32'hde000000; rnd_mode =0; clear =

1;

$display("\n ***** Negative x Negative
***** \n");

#(cycle) Ain = 32'hde000000; Bin =32'hde000000; rnd_mode =0; clear = 1;

$display("\n ***** Denor x Denorm
***** \n");

#(cycle) Ain = 32'h00100000; Bin =32'h44000000; rnd_mode =0; clear =

1;

$display("\n ***** Inf x pos ***** \n");

#(cycle) Ain = 32'h7f800010; Bin =32'h40000010; rnd_mode =0; clear = 1;

#(cycle) Ain = 32'h7ec11111; Bin =32'h40011111; rnd_mode =0; clear = 1;

```

```

        #(cycle) Ain =32'h00000000; Bin = 32'h00000000; rnd_mode=0;
        #(4*cycle);
        #cycle $finish;
    end

initial begin
    $monitor ("%0d ", $time,, "clock: %b \n\n A sign: %b \n A mantissa: %b \n A
exponent: %b", clock,Ain[31],Ain[22:0],Ain[30:23],
        "\n\n B sign: %b \n B mantissa: %b \n B exponent:
%b",Bin[31],Bin[22:0],Bin[30:23],
        "\n\n Res sign: %b \n Res mantissa: %b \n Res exponent:
%b",regout5[31],regout5[22:0],regout5[30:23],

        "\n\n Rounding Mode: %b Clear: %b ",rnd_mode,clear,
        "\n\n NAN: %b INF: %b OVFL: %b UNFL: %b INEXACT: %b
\n\n",regout5[35],regout5[36],regout5[33],regout5[32],regout5[34]);

    end

endmodule

```

```

module rounder (in,out,mode,sign,rndadd,denorm);

parameter //epoch enum stat

        RM=2'd0,    // round to minus
        RN=2'd1,    //round to nearest
        RP=2'd2,    //round to plus
        RZ=2'd3;    //chop mode

//

input [26:0]  in;          // mantissa in
input [1:0]   mode;        // rounding mode
input        sign;        // resultant sign
output [22:0] out;        // rounded mantissa
output       denorm,rndadd;

reg          rndadd,denorm,sel,addlsb;

reg  [1:0]   /* epoch enum stat */ state;

wire        nc; // not connected
wire [22:0]  out;
wire [24:0]  iout;

`define lsb_bit      in[3]
`define guard_bit   in[2]
`define round_bit   in[1]
`define sticky_bit  in[0]

`define zero  1'b0
`define one   1'b1
`define imptype 1

supply0      gnd;

inc # (25,1,"rnder")
    Incrementer(addlsb,{1'b0,in[26:3]},nc,nc,iout);

mux2 # (23,1,"rnder")

```

```
MUX (iout[22:0], iout[23:1],sel,out);
```

```
always @(mode or state) begin
```

```
    state=RZ;
```

```
    if (mode==2'd0)
```

```
        state=RM;
```

```
    else if (mode==2'd1)
```

```
        state=RN;
```

```
    else if (mode==2'd2)
```

```
        state=RP;
```

```
    else if (mode==2'd3)
```

```
        state=RZ;
```

```
end
```

```
always @ (in[3:0] or sign or state)
```

```
begin
```

```
    addlsb=1'b0;
```

```
    if (!(`guard_bit==1 & `round_bit ==0 & `sticky_bit==0))
```

```
        addlsb=0;
```

```
    else if(`guard_bit==1 & `round_bit ==0 & `sticky_bit==0)
```

```
begin
```

```
    addlsb=1'b0;
```

```
    case(state)
```

```
        RM: begin
```

```
            if (sign)
```

```
                addlsb= 1'b1 ;
```

```
            else addlsb=1'b0;
```

```
        end
```

```
        RP: begin
```

```
            if(!sign)
```

```
                addlsb= 1'b1;
```

```
            else addlsb=1'b0;
```

```
        end
```

```
        RN: begin
```

```
            if (`guard_bit==1'b1 & `round_bit ==1'b0 &
```

```
`sticky_bit==1'b0)
```

```

        begin
            if (^lsb_bit==1'b1)
                addlsb=1'b1;
            else addlsb=1'b0;
            end
            else if (^guard_bit==1)
                addlsb=1'b1;
            else addlsb=1'b0;
            end
        RZ: begin
            addlsb=1'b0;
        end

        default:addlsb=1'b0;

    endcase

end

end

always @(iout[24:23]) begin

    if(iout[24]==1) begin
        sel=1'b1;
        rndadd=1;
        denorm=0;
    end
    else if(iout[23]==0) begin
        denorm=1;
        rndadd=0;
        sel=1'b0;
    end
    else begin
        sel=1'b0;
        rndadd=0;
        denorm=0;
    end

end

end

```

endmodule

```

module shiftsel
(sft_amt,alt_shift_sel,shift_m_2,sft_gr,exp_gr,exp,exp_reduced,sft_sel,exp_fin,sft_dir);

```

```

// epoch pre_compiled shiftsel

```

```

input [4:0]  sft_amt,shift_m_2;
input  sft_gr,exp_gr;
input [7:0]  exp,alt_shift_sel,exp_reduced;
output [4:0] sft_sel;
output [7:0] exp_fin;
output      sft_dir;
reg [6:0]    car;
reg [7:0]    exp_fin,tin,in;
reg [4:0]    sft_sel;
reg          sft_dir;

```

```

always @(sft_amt or exp or exp_reduced or sft_gr or exp_gr or alt_shift_sel or sft_sel or
sft_dir or exp_fin or tin or car or shift_m_2) begin

```

```

    if (sft_amt == 5'b00010) begin
        exp_fin=exp;
        sft_sel=5'b00000;
        sft_dir=0;
    end

```

```

    else if (sft_amt == 5'b00001) begin
        exp_fin=inc(exp);
        sft_sel=5'b00000;
        sft_dir=1;
    end

```

```

    else begin

```

```

        if (exp_gr == 1) begin
            exp_fin=exp_reduced;
            sft_sel=shift_m_2;
            sft_dir=0;

```

```

        end

```

```

        else if (sft_gr == 1) begin
            exp_fin=8'b00000000;
            sft_sel=exp[4:0];
            sft_dir=0;

```



```

        end
        else begin
            exp_fin=inc(alt_shift_sel);
            sft_sel=shift_m_2;
            sft_dir=0;
        end
    end
end

function [7:0] inc;

    input [7:0] in;

    begin
        tin[0]=~in[0];
        car[0]=in[0];

        tin[1]=in[1]^car[0];
        car[1]=in[1]&car[0];

        tin[2]=in[2]^car[1];
        car[2]=in[2]&car[1];

        tin[3]=in[3]^car[2];
        car[3]=in[3]&car[2];

        tin[4]=in[4]^car[3];
        car[4]=in[4]&car[3];

        tin[5]=in[5]^car[4];
        car[5]=in[5]&car[4];

        tin[6]=in[6]^car[5];
        car[6]=in[6]&car[5];

        tin[7]=in[7]^car[6];

        inc=tin;
    end

endfunction

endmodule

```

```

module vcheck (in, nan, inf, exp,man,sign);
parameter    NUMBER_SIZE=32;
parameter    EXP_SIZE=8;
parameter    MANTISSA_SIZE=23;

```

```

// epoch pre_compiled vcheck

```

```

input [NUMBER_SIZE-1:0] in;
output      nan, inf, sign;
output [EXP_SIZE-1:0]      exp;
output [MANTISSA_SIZE+1:0]  man;

```

```

reg      nan, inf, sign;
reg [1:0]      state;
reg [EXP_SIZE-1:0]      exp;
reg [MANTISSA_SIZE+1:0]  man;

```

```

`define      exp_in in[30:23]
`define      man_in in[22:0]
`define      zero 1'b0
`define      one 1'b1

```

```

always @(in)
begin
    if (`exp_in==8'b11111111)

        state=2'b00;

    else if(`exp_in==8'b0)

        state=2'b10;

    else

        state=2'b01;
end

```

```

always @ (in or state)

```

```

begin

```

```

    /*state machine*/

```

```

case(state)

    2'b00:begin
        if(`man_in == 23'b0) // Infinity
            begin
                exp = 255;
                man = {`zero,`one,`man_in};
                nan =0;
                inf=1;

                sign=in[31];
            end
        else //NaN
            begin
                exp = 255;
                man = {`zero,`one,`man_in};
                nan =1;
                inf =0;

                sign=in[31];
            end
        end
    end

    2'b01:begin
        exp=`exp_in;
        man={`zero,`one,`man_in};
        nan=0;
        inf=0;

        sign=in[31];
    end

    2'b10:begin
        exp = 8'b1;
        man={`zero,`zero,`man_in};
        nan =0;
        inf=0;

        sign=in[31];
    end
end

```

/* default - Normalized number*/

```
        default:begin

            exp=`exp_in;
            man={`zero,`one,`man_in};
            nan=0;
            inf=0;

            sign=in[31];
        end

    endcase
end
endmodule
```


APPENDIX C EPOCH VERILOG HDL COMPILATION

```
/local/epoch/bin/verilogcomp -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/verilogcomp.log -top fftchip
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/verilog/fftchip.v -y
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/verilog +libext+.v+ -phase -testability
+checkwidth +checkinputs +checkoutputs +subremake -syngate
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
```

Starting Synthesis of logic block: [rounder, p1]

```
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p1.fin
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/f#p1.mac
```

Starting Parsing
Parsing Runtime: 0.1 sec

Starting Language Synthesis
Language Synthesis Runtime: 0.0 sec

Starting Symbolic Minimization
Symbolic Minimization Runtime: 0.0 sec

Starting Symbolic Encoding
Symbolic Encoding Runtime: 0.0 sec

Starting Boolean Minimization
Boolean Minimization Runtime: 0.0 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.0 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:
Primary Inputs : 7
Primary Outputs : 1
Intermediate Vars : 0
Literals : 18
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:
Primary Inputs : 7
Primary Outputs : 1
Intermediate Vars : 2
Literals : 13
Maximum Levels : 3
Maximum Fanout : 1

Weak Division Runtime: 0.1 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdnor3	54	1	54
stdnand2	39	1	39
stdmux2	50	2	100
stdnor2	39	1	39
stdinv	22	1	22

TOTAL GATE COST : 254
TOTAL GATE COUNT : 6

MAXIMUM FANOUT : 1
MAXIMUM INT_FAN : 1
MAXIMUM INBAR_FAN: 1
MAXIMUM LEVEL : 5
Technology Mapping Runtime: 0.2 sec

Structure Composition Runtime: 0.3 sec

Starting Netlist Generation
Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p1.net
Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p1.net

Finished the translation of netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p1.net
Synthesis of logic block: [rounder, p1]
Runtime: 0.7 sec

Starting Synthesis of logic block: [rounder, p2]

Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p2.fin
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/f#p2.mac

Starting Parsing
Parsing Runtime: 0.0 sec

Starting Language Synthesis
Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization
Boolean Minimization Runtime: 0.0 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.0 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 2
Primary Outputs : 1
Intermediate Vars : 0
Literals : 2
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 2
Primary Outputs : 1
Intermediate Vars : 0
Literals : 2
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.0 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdnor2	39	1	39

TOTAL GATE COST : 39
TOTAL GATE COUNT : 1
MAXIMUM FANOUT : 1
MAXIMUM INT_FAN : 1
MAXIMUM INBAR_FAN: 0
MAXIMUM LEVEL : 1
Technology Mapping Runtime: 0.1 sec

Structure Composition Runtime: 0.2 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p2.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p2.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p2.net

Synthesis of logic block: [rounder, p2]

Runtime: 0.4 sec

Starting Synthesis of logic block: [rounder, p0]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p0.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/f#p0.mac

Starting Parsing

Parsing Runtime: 0.0 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 0.0 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic

Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.0 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 2
Primary Outputs : 2
Intermediate Vars : 0
Literals : 2
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 2
Primary Outputs : 2
Intermediate Vars : 0
Literals : 2
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.0 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdinv	22	2	44

TOTAL GATE COST : 44

TOTAL GATE COUNT : 2

MAXIMUM FANOUT : 1

MAXIMUM INT_FAN : 0

MAXIMUM INBAR_FAN: 1

MAXIMUM LEVEL : 1

Technology Mapping Runtime: 0.1 sec

Structure Composition Runtime: 0.2 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p0.net
Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p0.net

Finished the translation of netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/rounder/p0.net
Synthesis of logic block: [rounder, p0]
Runtime: 0.4 sec

Starting flattening module rounder

Flattening through instance: p1 of cell: p1

Flattening through instance: p2 of cell: p2

Flattening through instance: p0 of cell: p0

The design has been flattened.

flattening module rounder
Runtime: 0.0 sec

Starting Synthesis of logic block: [overshift, p3]

Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/overshift/p3.fin
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/overshift/f#p3.mac

Starting Parsing
Parsing Runtime: 0.1 sec

Starting Language Synthesis
Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization
Boolean Minimization Runtime: 0.1 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.1 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:
Primary Inputs : 51
Primary Outputs : 50
Intermediate Vars : 0
Literals : 100
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:
Primary Inputs : 51
Primary Outputs : 50
Intermediate Vars : 0
Literals : 100
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.1 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdnor2	39	50	1950
stdinv	22	50	1100

TOTAL GATE COST : 3050

TOTAL GATE COUNT : 100

MAXIMUM FANOUT : 1

MAXIMUM INT_FAN : 1

MAXIMUM INBAR_FAN: 1

MAXIMUM LEVEL : 2

Technology Mapping Runtime: 0.6 sec

Structure Composition Runtime: 0.8 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/overshift/p3.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/overshift/p3.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/overshift/p3.net

Synthesis of logic block: [overshift, p3]

Runtime: 1.7 sec

Starting flattening module overshift

Flattening through instance: p3 of cell: p3

The design has been flattened.

flattening module overshift

Runtime: 0.1 sec

Starting Synthesis of logic block: [inverter2, p4]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter2/p4.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter2/f#p4.mac

Starting Parsing
Parsing Runtime: 0.1 sec

Starting Language Synthesis
Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization
Boolean Minimization Runtime: 0.1 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.1 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:
Primary Inputs : 27
Primary Outputs : 26
Intermediate Vars : 0
Literals : 104
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:
Primary Inputs : 27
Primary Outputs : 26
Intermediate Vars : 0
Literals : 104
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.1 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdxnor2	50	26	1300
stdinv	22	3	66

TOTAL GATE COST : 1366

TOTAL GATE COUNT : 29

MAXIMUM FANOUT : 9

MAXIMUM INT_FAN : 1

MAXIMUM INBAR_FAN: 9

MAXIMUM LEVEL : 2

Technology Mapping Runtime: 0.8 sec

Structure Composition Runtime: 1.0 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter2/p4.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter2/p4.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter2/p4.net

Synthesis of logic block: [inverter2, p4]

Runtime: 1.6 sec

Starting flattening module inverter2

Flattening through instance: p4 of cell: p4

The design has been flattened.

flattening module inverter2

Runtime: 0.1 sec

Starting Synthesis of logic block: [vcheck, p6]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p6.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/f#p6.mac

Starting Parsing

Parsing Runtime: 0.2 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.2 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 1.1 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic

Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.1 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 33

Primary Outputs : 11

Intermediate Vars : 0

Literals : 126

Maximum Levels : 1

Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 33

Primary Outputs : 11

Intermediate Vars : 2

Literals : 54
Maximum Levels : 2
Maximum Fanout : 9

Weak Division Runtime: 0.2 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdnor2	39	2	78
stdand2	44	1	44
stdand3	59	1	59
stdinv	22	2	44
stdnand2	39	1	39
stdao21	63	7	441
stdor2	43	1	43
stdand4	69	1	69
stdnor4	69	5	345
stdnor3	54	1	54

TOTAL GATE COST : 1216
TOTAL GATE COUNT : 22
MAXIMUM FANOUT : 9
MAXIMUM INT_FAN : 9
MAXIMUM INBAR_FAN: 2
MAXIMUM LEVEL : 4
Technology Mapping Runtime: 0.5 sec

Structure Composition Runtime: 0.7 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p6.net

NOTE: OUTPUT zman_24__ IS GND.

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p6.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p6.net

Synthesis of logic block: [vcheck, p6]

Runtime: 3.2 sec

Starting Synthesis of logic block: [vcheck, p5]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p5.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/f#p5.mac

Starting Parsing

Parsing Runtime: 0.0 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 0.3 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic

Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.0 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 8

Primary Outputs : 2

Intermediate Vars : 0

Literals : 24

Maximum Levels : 1

Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 8
Primary Outputs : 2
Intermediate Vars : 0
Literals : 24
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.0 sec

Starting Technology Mapping

GATE TYPE GATE COST TIMES USED TOTAL COST

stdor4	74	1	74
stdmux2	50	4	200
stdand2	44	1	44
stdinv	22	4	88
stdnor4	69	2	138

TOTAL GATE COST : 544

TOTAL GATE COUNT : 12

MAXIMUM FANOUT : 1

MAXIMUM INT_FAN : 1

MAXIMUM INBAR_FAN: 1

MAXIMUM LEVEL : 3

Technology Mapping Runtime: 0.4 sec

Structure Composition Runtime: 0.4 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p5.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p5.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/vcheck/p5.net

Synthesis of logic block: [vcheck, p5]

Runtime: 1.0 sec

Starting flattening module vcheck

Flattening through instance: p6 of cell: p6

Flattening through instance: p5 of cell: p5

The design has been flattened.

flattening module vcheck
Runtime: 0.1 sec

Starting Synthesis of logic block: [inverter, p7]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter/p7.fin
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter/f#p7.mac

Starting Parsing
Parsing Runtime: 0.1 sec

Starting Language Synthesis
Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization
Boolean Minimization Runtime: 0.0 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.1 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 26
Primary Outputs : 25
Intermediate Vars : 0
Literals : 100
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 26
Primary Outputs : 25
Intermediate Vars : 0
Literals : 100
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.1 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdxnor2	50	25	1250
stdinv	22	3	66

TOTAL GATE COST : 1316

TOTAL GATE COUNT : 28

MAXIMUM FANOUT : 9

MAXIMUM INT_FAN : 1

MAXIMUM INBAR_FAN: 9

MAXIMUM LEVEL : 2

Technology Mapping Runtime: 0.8 sec

Structure Composition Runtime: 0.9 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter/p7.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter/p7.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/inverter/p7.net

Synthesis of logic block: [inverter, p7]

Runtime: 1.5 sec

Starting flattening module inverter

Flattening through instance: p7 of cell: p7

The design has been flattened.

flattening module inverter

Runtime: 0.0 sec

Starting Synthesis of logic block: [bitshift, p8]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/bitshift/p8.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/bitshift/f#p8.mac

Starting Parsing

Parsing Runtime: 0.2 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.1 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 0.1 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.1 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 26
Primary Outputs : 26
Intermediate Vars : 0
Literals : 100
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 26
Primary Outputs : 26
Intermediate Vars : 0
Literals : 100
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.1 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdmux2	50	24	1200
stdor2	43	1	43
stdinv	22	3	66
stdand2	44	1	44

TOTAL GATE COST : 1353
TOTAL GATE COUNT : 29
MAXIMUM FANOUT : 9
MAXIMUM INT_FAN : 1
MAXIMUM INBAR_FAN: 9
MAXIMUM LEVEL : 2

Technology Mapping Runtime: 0.8 sec

Structure Composition Runtime: 0.9 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/bitshift/p8.net

NOTE: OUTPUT zout_25__ IS GND.

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/bitshift/p8.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/bitshift/p8.net

Synthesis of logic block: [bitshift, p8]

Runtime: 1.7 sec

Starting flattening module bitshift

Flattening through instance: p8 of cell: p8

The design has been flattened.

flattening module bitshift

Runtime: 0.0 sec

Starting Synthesis of logic block: [mulcnp, p9]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mulcnp/p9.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mulcnp/f#p9.mac

Starting Parsing

Parsing Runtime: 0.3 sec

Starting Language Synthesis
Language Synthesis Runtime: 0.1 sec

Starting Boolean Minimization
Boolean Minimization Runtime: 0.1 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.1 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:
Primary Inputs : 34
Primary Outputs : 32
Intermediate Vars : 0
Literals : 140
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:
Primary Inputs : 34
Primary Outputs : 32
Intermediate Vars : 5
Literals : 111
Maximum Levels : 2
Maximum Fanout : 10

Weak Division Runtime: 0.2 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdand4	69	1	69

stdnor4	69	4	276
stdoi21	54	8	432
stdnor2	39	23	897
stdor2	43	5	215
stdnand4	69	5	345
stdinv	22	32	704

TOTAL GATE COST : 2938

TOTAL GATE COUNT : 78

MAXIMUM FANOUT : 10

MAXIMUM INT_FAN : 10

MAXIMUM INBAR_FAN: 2

MAXIMUM LEVEL : 4

Technology Mapping Runtime: 0.9 sec

Structure Composition Runtime: 1.1 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mulcnp/p9.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mulcnp/p9.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mulcnp/p9.net

Synthesis of logic block: [mulcnp, p9]

Runtime: 2.4 sec

Starting flattening module mulcnp

Flattening through instance: p9 of cell: p9

The design has been flattened.

flattening module mulcnp

Runtime: 0.1 sec

Starting Synthesis of logic block: [fpadder, s0]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fpadder/s0.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fpadder/f#s0.mac

Starting Parsing

Parsing Runtime: 0.0 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 0.0 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic

Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.0 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 4

Primary Outputs : 4

Intermediate Vars : 0

Literals : 4

Maximum Levels : 1

Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 4

Primary Outputs : 4

Intermediate Vars : 0
Literals : 4
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.0 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdinv	22	4	88

TOTAL GATE COST : 88
TOTAL GATE COUNT : 4
MAXIMUM FANOUT : 1
MAXIMUM INT_FAN : 0
MAXIMUM INBAR_FAN: 1
MAXIMUM LEVEL : 1
Technology Mapping Runtime: 0.1 sec

Structure Composition Runtime: 0.2 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fpadder/s0.net
Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fpadder/s0.net

Finished the translation of netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fpadder/s0.net
Synthesis of logic block: [fpadder, s0]
Runtime: 0.4 sec

Starting flattening module fpadder

Flattening through instance: s0 of cell: s0

The design has been flattened.

flattening module fpadder

Runtime: 0.2 sec

Starting Synthesis of logic block: [encoder, p10]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/encoder/p10.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/encoder/f#p10.mac

Starting Parsing

Parsing Runtime: 1.1 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.3 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 0.4 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic

Boolean Minimization for Multi-level Logic Runtime: 0.2 sec

Structure Optimization Runtime: 0.3 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 48

Primary Outputs : 6

Intermediate Vars : 0

Literals : 1647

Maximum Levels : 1

Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 48
Primary Outputs : 6
Intermediate Vars : 62
Literals : 247
Maximum Levels : 18
Maximum Fanout : 4

Weak Division Runtime: 8.3 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdnor2	39	11	429
stdnor3	54	18	972
stdao211	79	1	79
stdaoi21	54	13	702
stdoi32	85	1	85
stdoi21	54	11	594
stdnor4	69	12	828
stdoi31	70	2	140
stdnand2	39	6	234
stdinv	22	28	616
stdor4	74	8	592
stdor3	59	2	118
stdor2	43	3	129

TOTAL GATE COST : 5518

TOTAL GATE COUNT : 116

MAXIMUM FANOUT : 4

MAXIMUM INT_FAN : 4

MAXIMUM INBAR_FAN: 2

MAXIMUM LEVEL : 22

Technology Mapping Runtime: 2.2 sec

Structure Composition Runtime: 10.6 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/encoder/p10.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/encoder/p10.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/encoder/p10.net

Synthesis of logic block: [encoder, p10]

Runtime: 14.2 sec

Starting flattening module encoder

Flattening through instance: p10 of cell: p10

The design has been flattened.

flattening module encoder

Runtime: 0.1 sec

Starting Synthesis of logic block: [shiftsel, p11]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/shiftsel/p11.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/shiftsel/f#p11.mac

Starting Parsing

Parsing Runtime: 0.2 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.4 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 4.8 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic

Boolean Minimization for Multi-level Logic Runtime: 1.0 sec

Structure Optimization Runtime: 2.0 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 36

Primary Outputs : 14

Intermediate Vars : 0

Literals : 1813

Maximum Levels : 1

Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 36

Primary Outputs : 14

Intermediate Vars : 46

Literals : 218

Maximum Levels : 9

Maximum Fanout : 10

Weak Division Runtime: 2.8 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdoi222	100	8	800
stdxnor2	50	7	350
stdoi22	69	8	552
stdand2	44	12	528
stdxor2	49	7	343
stdnor2	39	8	312
stdinv	22	18	396
stdmux2	50	5	250
stdnor3	54	3	162
stdoi211	79	1	79

stdor3 59 2 118

TOTAL GATE COST : 3890
TOTAL GATE COUNT : 79
MAXIMUM FANOUT : 10
MAXIMUM INT_FAN : 10
MAXIMUM INBAR_FAN: 3
MAXIMUM LEVEL : 9
Technology Mapping Runtime: 1.7 sec

Structure Composition Runtime: 4.5 sec

Starting Netlist Generation
Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/shiftsel/p11.net
Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/shiftsel/p11.net

Finished the translation of netlist file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/shiftsel/p11.net
Synthesis of logic block: [shiftsel, p11]
Runtime: 13.1 sec

Starting flattening module shiftsel

Flattening through instance: p11 of cell: p11

The design has been flattened.

flattening module shiftsel
Runtime: 0.1 sec

Starting Synthesis of logic block: [mansel, p12]

Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p

Technology: cmos
/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mansel/p12.fin
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mansel/f#p12.mac

Starting Parsing
Parsing Runtime: 0.2 sec

Starting Language Synthesis
Language Synthesis Runtime: 0.1 sec

Starting Boolean Minimization
Boolean Minimization Runtime: 0.3 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic
Boolean Minimization for Multi-level Logic Runtime: 0.1 sec

Structure Optimization Runtime: 0.3 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:
Primary Inputs : 49
Primary Outputs : 48
Intermediate Vars : 0
Literals : 190
Maximum Levels : 1
Maximum Fanout : 1

Output logic statistics:
Primary Inputs : 49
Primary Outputs : 48
Intermediate Vars : 0
Literals : 190
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.3 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdoai22	68	46	3128
stdand2	44	1	44
stdmux2	50	1	50
stdinv	22	53	1166

TOTAL GATE COST : 4388

TOTAL GATE COUNT : 101

MAXIMUM FANOUT : 10

MAXIMUM INT_FAN : 1

MAXIMUM INBAR_FAN: 10

MAXIMUM LEVEL : 2

Technology Mapping Runtime: 1.5 sec

Structure Composition Runtime: 1.9 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mansel/p12.net

Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mansel/p12.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/mansel/p12.net

Synthesis of logic block: [mansel, p12]

Runtime: 3.5 sec

Starting flattening module mansel

Flattening through instance: p12 of cell: p12

The design has been flattened.

flattening module mansel

Runtime: 0.1 sec

Starting Synthesis of logic block: [fft, s1]

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

/local/epoch/bin/cpp /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fft/s1.fin

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fft/f#s1.mac

Starting Parsing

Parsing Runtime: 0.0 sec

Starting Language Synthesis

Language Synthesis Runtime: 0.0 sec

Starting Boolean Minimization

Boolean Minimization Runtime: 0.0 sec

Starting Structure Optimization

Starting Boolean Minimization for Multi-level Logic

Boolean Minimization for Multi-level Logic Runtime: 0.0 sec

Structure Optimization Runtime: 0.0 sec

Starting Structure Composition

Starting Weak Division

Input logic statistics:

Primary Inputs : 6

Primary Outputs : 6

Intermediate Vars : 0

Literals : 6

Maximum Levels : 1

Maximum Fanout : 1

Output logic statistics:

Primary Inputs : 6
Primary Outputs : 6
Intermediate Vars : 0
Literals : 6
Maximum Levels : 1
Maximum Fanout : 1

Weak Division Runtime: 0.0 sec

Starting Technology Mapping

GATE TYPE	GATE COST	TIMES USED	TOTAL COST
stdinv	22	6	132

TOTAL GATE COST : 132
TOTAL GATE COUNT : 6
MAXIMUM FANOUT : 1
MAXIMUM INT_FAN : 0
MAXIMUM INBAR_FAN: 1
MAXIMUM LEVEL : 1

Technology Mapping Runtime: 0.1 sec

Structure Composition Runtime: 0.2 sec

Starting Netlist Generation

Writing net file /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fft/s1.net
Netlist Generation Runtime: 0.0 sec

No errors. No warnings.

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fft/s1.net

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fft/s1.net
Synthesis of logic block: [fft, s1]
Runtime: 0.4 sec

Starting flattening module fft

Flattening through instance: s1 of cell: s1

The design has been flattened.

flattening module fft

Runtime: 1.6 sec

Writing /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fftchip/fftchip.net
Writing circuit bitshift.
Writing circuit rounder.
Writing circuit vcheck.
Writing circuit mulcnp.
Writing circuit inverter.
Writing circuit inverter2.
Writing circuit overshift.
Writing circuit fpadder.
Writing circuit complex_adder_4_input.
Writing circuit encoder.
Writing circuit mansel.
Writing circuit shiftsel.
Writing circuit fpmplier.
Writing circuit complex_mplier.
Writing circuit fft.
Writing circuit fftchip.
File /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fftchip/fftchip.def is up to date
Writing /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fftchip/fftchip.vcmd
Writing /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fftchip/fftchip.ref
/local/epoch/bin/netinput -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/netinput.log fftchip
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos

----- NET2NP3 -----

Beginning to translate netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fftchip/fftchip.net
net2np3(WARNING) (line: 266)can not find net: in__F_1(0) on cell: bitshift
net2np3(WARNING) (line: 1445)can not find net: exp__F_3(0) on cell: mulcnp
net2np3(WARNING) (line: 1446)can not find net: exp__F_4(0) on cell: mulcnp

```

net2np3(WARNING) (line: 1505 )can not find net: man_rnd__F_2(0) on cell: mulcnpup
net2np3(WARNING) (line: 1506 )can not find net: man_rnd__F_4(0) on cell: mulcnpup
net2np3(WARNING) (line: 6949 )can not find net: a(0) on cell: encoder
net2np3(WARNING) (line: 6950 )can not find net: h(0) on cell: encoder
net2np3(WARNING) (line: 7001 )can not find net: in__F_5(0) on cell: encoder
net2np3(WARNING) (line: 8293 )can not find net: car(0) on cell: shiftsel
net2np3(WARNING) (line: 8322 )can not find net: in(0) on cell: shiftsel
net2np3(WARNING) (line: 8323 )can not find net: in__F_6(0) on cell: shiftsel
net2np3(WARNING) (line: 8324 )can not find net: in__F_7(0) on cell: shiftsel
net2np3(WARNING) (line: 8345 )can not find net: tin(0) on cell: shiftsel

```

Finished the translation of netlist file:

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fftchip/fftchip.net

----- BUILD LEAF CELL -----

```

Reading geometry cell addhs#o0 ...
Reading geometry cell dpfadder_bit#o3 ...
Reading geometry cell dpfadder_bit#o6 ...
Reading geometry cell dpfadder_bit#o9 ...
Reading geometry cell dpfadder_bit#o5 ...
Reading geometry cell dpfadder_bit#o11 ...
Reading geometry cell dpfadder_bit#o7 ...
Reading geometry cell dpfadder_bit#o12 ...
Reading geometry cell addhs#o1 ...
Reading geometry cell dpfadder_bit#o8 ...
Reading geometry cell dpfadder_bit#o10 ...
Reading geometry cell dpfadder_bit#o4 ...
Reading geometry cell stdinv#o0 ...
Reading geometry cell stdand2#o0 ...
Reading geometry cell stdmux2#o0 ...
Reading geometry cell stdor2#o0 ...
Reading geometry cell mcomp#o0 ...
Reading geometry cell gtgate#o2 ...
Reading geometry cell gtgate#o0 ...
Reading geometry cell gtgate#o3 ...
Reading geometry cell gtgate#o1 ...
Reading geometry cell mux2#o1 ...
Reading geometry cell gtmux#o0 ...
Reading geometry cell inc#o1 ...
Reading geometry cell gtadd#o0 ...
Reading geometry cell inc#o2 ...
Reading geometry cell mcomp#o1 ...
Reading geometry cell mux2#o2 ...
Reading geometry cell inc#o3 ...

```


Reading geometry cell mux2#o4 ...
 Reading geometry cell stdnand2#o0 ...
 Reading geometry cell stdnor2#o0 ...
 Reading geometry cell stdnor3#o0 ...
 Reading geometry cell mux2#o3 ...
 Reading geometry cell stdnor4#o0 ...
 Reading geometry cell stdor4#o0 ...
 Reading geometry cell stdand4#o0 ...
 Reading geometry cell stdand3#o0 ...
 Reading geometry cell stdao21#o0 ...
 Reading geometry cell zero#o0 ...
 Reading geometry cell gtgate#o4 ...
 Reading geometry cell gtgate#o6 ...
 Reading geometry cell gtgate#o5 ...
 Reading geometry cell zero#o1 ...
 Reading geometry cell stdnand4#o0 ...
 Reading geometry cell stdoi21#o0 ...
 Reading geometry cell compl#o0 ...
 Reading geometry cell dpcompl#o0 ...
 Reading geometry cell dpcompl#o1 ...
 Reading geometry cell stdxnor2#o0 ...
 Reading geometry cell stdor3#o0 ...
 Reading geometry cell dff_c#o0 ...
 Reading geometry cell gtdff#o0 ...
 Reading geometry cell dff_c#o1 ...
 Reading geometry cell dff_c#o2 ...
 Reading geometry cell dff_c#o3 ...
 Reading geometry cell dff_c#o4 ...
 Reading geometry cell barrelright#o0 ...
 Reading geometry cell gtmux#o1 ...

----- NPFLATTEN -----

Flattening through instance: Adder1 of cell: addhs#o0

Flattening through instance: Adder2 of cell: addhs#o1

Flattening through instance: Bitshift1 of cell: bitshift

Flattening through instance: E_Comparator of cell: mcomp#o0

Flattening through instance: Exp_mux1 of cell: mux2#o1

Flattening through instance: Exp_mux2 of cell: mux2#o1

Flattening through instance: Incrementer1 of cell: inc#o1

Flattening through instance: Incrementer2 of cell: inc#o2

Flattening through instance: Incrementer3 of cell: inc#o2

Flattening through instance: M_Comparator of cell: mcomp#o1

Flattening through instance: Man_mux1 of cell: mux2#o2

Flattening through instance: Man_mux2 of cell: mux2#o2

Flattening through instance: Incrementer of cell: inc#o3

Flattening through instance: MUX of cell: mux2#o4

Flattening through instance: Rounder of cell: rounder

Flattening through instance: Sign_mux of cell: mux2#o3

Flattening through instance: Value_Test_A of cell: vcheck

Flattening through instance: Value_Test_B of cell: vcheck

Flattening through instance: Zero_Dect1 of cell: zero#o0

Flattening through instance: Zero_Dect2 of cell: zero#o1

Flattening through instance: cleanup of cell: mulclnup

Flattening through instance: exp_compl of cell: compl#o0

Flattening through instance: man_invA of cell: inverter

Flattening through instance: man_invB of cell: inverter

Flattening through instance: man_out_inv of cell: inverter2

Flattening through instance: overshift of cell: overshift

Flattening through instance: reg1 of cell: dff_c#o0

Flattening through instance: reg2 of cell: dff_c#o1

Flattening through instance: reg2i of cell: dff_c#o1

Flattening through instance: reg3 of cell: dff_c#o2

Flattening through instance: reg4 of cell: dff_c#o3

Flattening through instance: reg5 of cell: dff_c#o4

Flattening through instance: shifter of cell: barrelright#o0

The design has been flattened.

----- NETCHECK -----

Deleted single terminal net: regout3(29) on cell: fpadder.

Deleted single terminal net: regout3(30) on cell: fpadder.

Deleted single terminal net: regout4(36) on cell: fpadder.

Deleted single terminal net: regout4(37) on cell: fpadder.

Deleted single terminal net: regout4(38) on cell: fpadder.

Deleted single terminal net: regout4(44) on cell: fpadder.

netcheck(WARNING) The cell gtdff#o0, instance reg3:reg29 is an unnecessary cell, none of its outputs are being used.

netcheck(WARNING) The cell gtdff#o0, instance reg3:reg30 is an unnecessary cell, none of its outputs are being used.

netcheck(WARNING) The cell gtdff#o0, instance reg4:reg36 is an unnecessary cell, none of its outputs are being used.

netcheck(WARNING) The cell gtdff#o0, instance reg4:reg37 is an unnecessary cell, none of its outputs are being used.

netcheck(WARNING) The cell gtdff#o0, instance reg4:reg38 is an unnecessary cell, none of its outputs are being used.

netcheck(WARNING) The cell gtdff#o0, instance reg4:reg44 is an unnecessary cell, none of its outputs are being used.

netcheck(ERROR) net: Rounder:nc on cell: fpadder has Multiple Output nodes connected together. (To turn off this error message use net attribute:

IGNORE_MULT_DRIVER)

netcheck(ERROR) net: nc on cell: fpadder has Multiple Output nodes connected together. (To turn off this error message use net attribute: IGNORE_MULT_DRIVER)

Finished netcheck process. Please check for error messages.

----- SEPARATOR -----

New cell: fpadding_group is created.

Finished separator process. Please check for error messages.

----- WRITING CELL TO DISK -----

Writing cell fpadding_group to disk
Dump circuit fpadding_group.
Writing cell fpadding to disk
Dump circuit fpadding.

----- WRITING CELL FILE -----

Dump cell fpadding_group.
Dump cell fpadding.

----- BUILD LEAF CELL -----

Reading geometry cell fpadding ...

----- NPFLATTEN -----

cell: complex_adder_4_input has no hierarchy, nothing to flatten

----- NETCHECK -----

Finished netcheck process. Please check for error messages.

----- SEPARATOR -----

Nothing to be separated.

Finished separator process. Please check for error messages.

----- WRITING CELL TO DISK -----

Writing cell complex_adder_4_input to disk
Dump circuit complex_adder_4_input.

----- WRITING CELL FILE -----

Dump cell complex_adder_4_input.

----- BUILD LEAF CELL -----

Reading geometry cell addrip#o0 ...
Reading geometry cell gtadd#o1 ...
Reading geometry cell compl#o1 ...
Reading geometry cell hsmult2piped_c_0#o0 ...
Reading geometry cell stdoi32#o0 ...
Reading geometry cell stdoi31#o0 ...
Reading geometry cell stdoi21#o0 ...
Reading geometry cell stdoi21#o0 ...
Reading geometry cell stdoi22#o0 ...
Reading geometry cell dff_c#o5 ...
Reading geometry cell dff_c#o6 ...
Reading geometry cell dff_c#o7 ...
Reading geometry cell dff_c#o8 ...
Reading geometry cell barrelleft#o0 ...
Reading geometry cell stdxor2#o0 ...
Reading geometry cell stdoi22#o0 ...
Reading geometry cell stdoi222#o0 ...

----- NPFLATTEN -----

Flattening through instance: Adder1 of cell: addhs#o0

Flattening through instance: Adder2 of cell: addhs#o0

Flattening through instance: Adder3 of cell: addhs#o0

Flattening through instance: Adder4 of cell: addrip#o0

Flattening through instance: Adder5 of cell: addhs#o0

Flattening through instance: Comparator of cell: mcomp#o0

Flattening through instance: Complementor of cell: compl#o1

Flattening through instance: Complementor2 of cell: compl#o1

Flattening through instance: Incrementer of cell: inc#o2

Flattening through instance: Incrementer2 of cell: inc#o2

Flattening through instance: Incrementer of cell: inc#o3

Flattening through instance: MUX of cell: mux2#o4

Flattening through instance: Rounder of cell: rounder

Flattening through instance: Value_Test_A of cell: vcheck

Flattening through instance: Value_Test_B of cell: vcheck

Flattening through instance: cleanup of cell: mulclnup

Flattening through instance: encoder of cell: encoder

Flattening through instance: mansel of cell: mansel

Flattening through instance: reg1 of cell: dff_c#o1

Flattening through instance: reg2 of cell: dff_c#o5

Flattening through instance: reg2i of cell: dff_c#o5

Flattening through instance: reg3 of cell: dff_c#o6

Flattening through instance: reg4 of cell: dff_c#o7

Flattening through instance: reg4i of cell: dff_c#o8

Flattening through instance: reg5 of cell: dff_c#o4

Flattening through instance: shifter of cell: barrelleft#o0

Flattening through instance: shiftsel of cell: shiftsel

The design has been flattened.

----- NETCHECK -----

Deleted single terminal net: prod(48) on cell: fpmplier.

Deleted single terminal net: prod(49) on cell: fpmplier.

Deleted single terminal net: prod(50) on cell: fpmplier.

Deleted single terminal net: prod(51) on cell: fpmplier.

Deleted single terminal net: regout4(63) on cell: fpmplier.

Deleted single terminal net: regout4i(67) on cell: fpmplier.

netcheck(WARNING) The cell stdnor4#o0, instance encoder:p10_Izh_8__ is an unnecessary cell, none of its outputs are being used.
 netcheck(WARNING) The cell gtdff#o0, instance reg4:reg63 is an unnecessary cell, none of its outputs are being used.
 netcheck(WARNING) The cell gtdff#o0, instance reg4i:reg67 is an unnecessary cell, none of its outputs are being used.
 netcheck(ERROR) net: Rounder:nc on cell: fpmplier has Multiple Output nodes connected together. (To turn off this error message use net attribute: IGNORE_MULT_DRIVER)
 netcheck(ERROR) net: nc on cell: fpmplier has Multiple Output nodes connected together. (To turn off this error message use net attribute: IGNORE_MULT_DRIVER)

Finished netcheck process. Please check for error messages.

----- SEPARATOR -----

New cell: mplier is created.

Finished separator process. Please check for error messages.

----- WRITING CELL TO DISK -----

Writing cell mplier to disk
 Dump circuit mplier.
 Writing cell fpmplier to disk
 Dump circuit fpmplier.

----- WRITING CELL FILE -----

Dump cell hsmult2piped_c_0#o0.
 Dump cell mplier.
 Dump cell fpmplier.

----- BUILD LEAF CELL -----

Reading geometry cell fpmplier ...

----- NPFLATTEN -----

cell: complex_mplier has no hierarchy, nothing to flatten

----- NETCHECK -----

Finished netcheck process. Please check for error messages.

----- SEPARATOR -----

Nothing to be separated.

Finished separator process. Please check for error messages.

----- WRITING CELL TO DISK -----

Writing cell complex_mplier to disk
Dump circuit complex_mplier.

----- WRITING CELL FILE -----

Dump cell complex_mplier.

----- BUILD LEAF CELL -----

Reading geometry cell complex_adder_4_input ...
Reading geometry cell complex_mplier ...
Reading geometry cell scntrl#o0 ...
Reading geometry cell inc_0#o0 ...
Reading geometry cell gtdff#o1 ...
Reading geometry cell dff_c#o9 ...
Reading geometry cell latch_c#o0 ...
Reading geometry cell gt latch#o0 ...
Reading geometry cell mux4#o0 ...
Reading geometry cell dff_c#o10 ...

----- NPFLATTEN -----

Flattening through instance: inc0 of cell: inc_0#o0

Flattening through instance: Counter of cell: scntrl#o0

Flattening through instance: FPSReg1 of cell: dff_c#o9

Flattening through instance: FPSReg2 of cell: dff_c#o9

Flattening through instance: FPSReg3 of cell: dff_c#o9

Flattening through instance: FPSReg4 of cell: dff_c#o9

Flattening through instance: FPSReg5 of cell: dff_c#o9

Flattening through instance: Latch1 of cell: latch_c#o0

Flattening through instance: Latch2 of cell: latch_c#o0

Flattening through instance: Mux1 of cell: mux4#o0

Flattening through instance: Mux2 of cell: mux4#o0

Flattening through instance: Mux3 of cell: mux4#o0

Flattening through instance: reg1 of cell: dff_c#o10

Flattening through instance: reg2 of cell: dff_c#o10

Flattening through instance: reg3 of cell: dff_c#o10

The design has been flattened.

----- NETCHECK -----

netcheck(WARNING) The cell gagate#o0, instance Counter:inc0:tc_inv is an unnecessary cell, none of its outputs are being used.

Finished netcheck process. Please check for error messages.

----- SEPARATOR -----

Nothing to be separated.

Finished separator process. Please check for error messages.

----- WRITING CELL TO DISK -----

Writing cell fft to disk
Dump circuit fft.

----- WRITING CELL FILE -----

Dump cell fft.

----- BUILD LEAF CELL -----

Writing specification cell padin_unbuf#o7, compiler padinput
Writing specification cell padin_unbuf#o8, compiler padinput
Writing specification cell padin_unbuf#o9, compiler padinput
Writing specification cell padin_unbuf#o10, compiler padinput
Writing specification cell padin_unbuf#o11, compiler padinput
Writing specification cell padin_unbuf#o12, compiler padinput
Writing specification cell padout#o2, compiler padoutput
Writing specification cell padout#o3, compiler padoutput
Writing specification cell padgnd#o1, compiler padgndvdd
Writing specification cell padvdd#o1, compiler padgndvdd
Writing specification cell padgnd#o2, compiler padgndvdd
Writing specification cell padvdd#o2, compiler padgndvdd
Writing specification cell padgnd#o3, compiler padgndvdd
Writing specification cell padvdd#o3, compiler padgndvdd
Writing specification cell padvdd#o4, compiler padgndvdd
Writing specification cell padgnd#o4, compiler padgndvdd

The following cells will be generated:

padin_unbuf#o7
padin_unbuf#o8
padin_unbuf#o9
padin_unbuf#o10
padin_unbuf#o11
padin_unbuf#o12
padout#o2
padout#o3
padgnd#o1
padvdd#o1
padgnd#o2
padvdd#o2
padgnd#o3
padvdd#o3
padvdd#o4
padgnd#o4

/local/epoch/bin/padcells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/padcells.log +geo +sym +sim
+trans padin_unbuf#o7 padin_unbuf#o8 padin_unbuf#o9 padin_unbuf#o10
padin_unbuf#o11 padin_unbuf#o12 padout#o2 padout#o3 padgnd#o1 padvdd#o1
padgnd#o2 padvdd#o2 padgnd#o3 padvdd#o3 padvdd#o4 padgnd#o4

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Calling compiler for cell padin_unbuf#o7...

=====> Generating Layout.

=====> End Layout.

=====> Generating Simulation Model.

=====> End Simulation Model.

=====> Generating Transistor Model.

=====> End Transistor Model.

=====> Generating Layout.

=====> End Layout.

=====> Generating Simulation Model.

=====> End Simulation Model.

=====> Generating Transistor Model.

=====> End Transistor Model.

=====> Generating Layout.

=====> End Layout.

=====> Generating Simulation Model.

=====> End Simulation Model.

=====> Generating Netlist.

Writing cell padin_unbuf#g7 to disk

=====> End Netlist.

Calling compiler for cell padin_unbuf#o8...

=====> Generating Layout.

=====> End Layout.

=====> Generating Simulation Model.

=====> End Simulation Model.

=====> Generating Transistor Model.

=====> End Transistor Model.

=====> Generating Layout.

=====> End Layout.

=====> Generating Simulation Model.

=====> End Simulation Model.

=====> Generating Transistor Model.

=====> End Transistor Model.

=====> Generating Layout.

=====> End Layout.

=====> Generating Simulation Model.

=====> End Simulation Model.

=====> Generating Netlist.

Writing cell padin_unbuf#g8 to disk

=====> End Netlist.

```

Calling compiler for cell padin_unbuf#o9...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Netlist.
Writing cell padin_unbuf#g9 to disk
=====> End Netlist.
Calling compiler for cell padin_unbuf#o10...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Netlist.
Writing cell padin_unbuf#g10 to disk
=====> End Netlist.
Calling compiler for cell padin_unbuf#o11...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.

```

```

=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Netlist.
Writing cell padin_unbuf#g11 to disk
=====> End Netlist.
Calling compiler for cell padin_unbuf#o12...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Netlist.
Writing cell padin_unbuf#g12 to disk
=====> End Netlist.
Calling compiler for cell padout#o2...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.

```

```

=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Netlist.
Writing cell padout#g2 to disk
=====> End Netlist.
Calling compiler for cell padout#o3...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Netlist.
Writing cell padout#g3 to disk
=====> End Netlist.
Calling compiler for cell padgnd#o1...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padgnd#g1 to disk

```

```

=====> End Netlist.
Calling compiler for cell padvdd#o1...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padvdd#g1 to disk
=====> End Netlist.
Calling compiler for cell padgnd#o2...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padgnd#g2 to disk
=====> End Netlist.
Calling compiler for cell padvdd#o2...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.

```

```

=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padvdd#g2 to disk
=====> End Netlist.
Calling compiler for cell padgnd#o3...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padgnd#g3 to disk
=====> End Netlist.
Calling compiler for cell padvdd#o3...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padvdd#g3 to disk
=====> End Netlist.
Calling compiler for cell padvdd#o4...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.

```



```

=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padvdd#g4 to disk
=====> End Netlist.
Calling compiler for cell padgnd#o4...
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Layout.
=====> End Layout.
=====> Generating Simulation Model.
=====> End Simulation Model.
=====> Generating Transistor Model.
=====> End Transistor Model.
=====> Generating Netlist.
Writing cell padgnd#g4 to disk
=====> End Netlist.
No errors. No warnings.
Reading geometry cell fft ...
Reading geometry cell padin_unbuf#o7 ...
Reading geometry cell pdinput#o0 ...
Reading geometry cell padin_unbuf#o8 ...
Reading geometry cell padin_unbuf#o9 ...
Reading geometry cell padin_unbuf#o10 ...
Reading geometry cell padin_unbuf#o11 ...
Reading geometry cell padin_unbuf#o12 ...
Reading geometry cell padout#o2 ...
Reading geometry cell pad_driver#o1 ...
Reading geometry cell pdoutput#o0 ...
Reading geometry cell padout#o3 ...
Reading geometry cell padgnd#o1 ...
Reading geometry cell pdgndvdd#o0 ...
Reading geometry cell padvdd#o1 ...
Reading geometry cell pdgndvdd#o1 ...
Reading geometry cell padgnd#o2 ...
Reading geometry cell padvdd#o2 ...
Reading geometry cell padgnd#o3 ...
Reading geometry cell padvdd#o3 ...
Reading geometry cell padvdd#o4 ...

```

Reading geometry cell padgnd#o4 ...

----- NPFLATTEN -----

Flattening through instance: inpad0 of cell: padin_unbuf#o7

Flattening through instance: inpad1 of cell: padin_unbuf#o8

Flattening through instance: inpad2 of cell: padin_unbuf#o9

Flattening through instance: inpad4 of cell: padin_unbuf#o10

Flattening through instance: inpad5 of cell: padin_unbuf#o11

Flattening through instance: inpad6 of cell: padin_unbuf#o12

Flattening through instance: outpad1 of cell: padout#o2

Flattening through instance: outpad2 of cell: padout#o3

Flattening through instance: pwrpad1 of cell: padgnd#o1

Flattening through instance: pwrpad2 of cell: padvdd#o1

Flattening through instance: pwrpad3 of cell: padgnd#o2

Flattening through instance: pwrpad4 of cell: padvdd#o2

Flattening through instance: pwrpad5 of cell: padgnd#o3

Flattening through instance: pwrpad6 of cell: padvdd#o3

Flattening through instance: pwrpad7 of cell: padvdd#o4

Flattening through instance: pwrpad8 of cell: padgnd#o4

The design has been flattened.

----- NETCHECK -----

netcheck(WARNING) net: gndpad has multi pad connections.

netcheck(WARNING) net: vddpad has multi pad connections.

Finished netcheck process. Please check for error messages.

----- SEPARATOR -----

New cell: fftchip_core is created.

Finished separator process. Please check for error messages.

----- CHECKPLOP -----

(WARNING) CHECKPLOP: No package has been selected,
pin number without PINNUM, SIDE and PADIND will be ignored,
auto pad positioning the unassigned pads ...

PACKAGER: Proceeding with automatic pad order assignment.

Finished with package information check.

----- Current Limitation Check -----

No package has been selected. Using the automatic pad assignment.

(NOTE) There is no Regular or CoreOnly Vdd pad on the East side of the chip.

(NOTE) There is no Regular or CoreOnly GND pad on the East side of the chip.

(NOTE) There is no Regular or CoreOnly GND pad on the South side of the chip.

(NOTE) There is no Regular or CoreOnly Vdd pad on the West side of the chip.

POWERCHECK(WARNING) Vdd and GND pads are not on every side of the design.

The design "fftchip" is OK to proceed further.

Finished with Current Limitation Check.

----- WRITING CELL TO DISK -----

Writing cell fftchip_core to disk

Dump circuit fftchip_core.

Writing cell fftchip to disk

Dump circuit fftchip.

----- WRITING CELL FILE -----

Dump cell fftchip_core.

Dump cell fftchip.

APPENDIX D EPOCH AUTOCOMPILATION

/local/epoch/bin/autocompile -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/autocompile.log -Placement -
Route -BufferSize fftchip
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos

Reading cell file: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fftchip/fftchip.cell

Reading cell file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/layout_parts/fftchip_core/fftchip_core.cell
.....

Reading cell file: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fft/fft.cell

Reading cell file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/layout_parts/complex_mplier/complex_m
plier.cell

Reading cell file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/layout_parts/fpmplier/fpmplier.cell

Reading cell file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/layout_parts/mplier/mplier.cell

Reading cell file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/layout_parts/hsmult2pipd_c_0#o0/hsmult
2pipd_c_0#o0.cell

Reading cell file: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/fpadder/fpadder.cell

Reading cell file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/layout_parts/fpadder_group/fpadder_grou
p.cell

Reading cell file:
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/complex_adder_4_input/complex_adder_
4_input.cell

/local/epoch/bin/dpath5 -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/dpath5.log fpadding_group
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
DATAPATH: Reading datapath geometry.
DATAPATH: Reading footprints.
DATAPATH: Placing columns.
DATAPATH: Initial bit cells placement.
DATAPATH: Routing datapath.

DATAPATH 3.0

size: 26 X 3

Horizontal channels
Contour Router V1.1

channel #26
channel #25
channel #24
channel #23
channel #22
channel #21
channel #20
channel #19
channel #18
channel #17
channel #16
channel #15
channel #14
channel #13
channel #12
channel #11
channel #10
channel #9
channel #8
channel #7
channel #6
channel #5
channel #4
channel #3

channel #2
channel #1
channel #0
Vertical channels
channel #3
channel #2
channel #1
channel #0

DATAPATH: Creating geometry.
Writing cell fpadder_group to disk

/local/epoch/bin/geo2glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/geo2glue.log -geoout -tries 30
fpadder

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading geometry "fpadder"

No. user defined groups: 1

Grouping standard cells

Grouping nets into buses

374 bussed nets eliminated.

Placement data:

1301 instances (1 blocks, 1300 standard cells in 9 groups)

52 nets (local 1041, bussed 32)

107 pins

Running block placement

Block Placement: 10.26 percent done

Block Placement: 20.51 percent done

Block Placement: 30.77 percent done

Block Placement: 41.03 percent done

Block Placement: 51.28 percent done

Block Placement: 61.54 percent done

Block Placement: 73.08 percent done

Block Placement: 83.33 percent done

Block Placement: 93.59 percent done

Block Placement: 100.00 percent done.

Merging groups of 126 and 119 standard cells into 6 rows

Merging groups of 212 and 245 standard cells into 12 rows

Merging groups of 212 and 457 standard cells into 12 rows

Merging groups of 78 and 123 standard cells into 3 rows

Running standard cell placement

Optimizing 5 vertical rows of 49 cells
Optimizing 3 vertical rows of 201 cells
Optimizing 8 vertical rows of 212 cells
Optimizing 12 rows of 669 cells
Optimizing 10 rows of 169 cells
Writing geometry "fpadder"
Writing cell fpadder to disk
Geo2glue completed on "fpadder": Real time: 00:10:02, CPU time: 558.71 seconds.

```
/local/epoch/bin/gluep -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/gluep.log fpadder
GLUE ROUTER version 6.1.
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Glue: reading in geometry file fpadder .....
Initialize power connections.
Read in placement expression.
Position export pins.
Adding north port for GND
Adding east port for GND
Adding west port for GND
Adding south port for VDD
Adding east port for VDD
Adding west port for VDD
Route power nets. (minimum_size_rails)
-- GND
-- VDD
Read in footprint information.
Over the cell routing gates.
Over the cell routing blocks.
Overcell Router V2.0
Generating global router input.
Mixed-Mode Global Router V1.1
    Routing group [0]
    Routing group [1]
    Routing group [2]
    Routing group [3]
    Routing group [4]
Assemble Standard cell groups
Assemble Group std_group_0
Contour Router V1.1
```


Channel no. 0
Channel no. 1
Assemble Group std_group_1
Channel no. 2
Channel no. 3
Channel no. 4
Channel no. 5
Channel no. 6
Channel no. 7
Channel no. 8
Assemble Group rnder
Channel no. 9
Channel no. 10
Channel no. 11
Channel no. 12
Assemble Group std_group_3
Channel no. 13
Channel no. 14
Channel no. 15
Channel no. 16
Channel no. 17
Channel no. 18
Channel no. 19
Channel no. 20
Channel no. 21
Assemble Group std_group_2
Channel no. 22
Channel no. 23
Channel no. 24
Channel no. 25
Channel no. 26
Channel no. 27
Channel no. 28
Channel no. 29
Channel no. 30
Channel no. 31
Channel no. 32
Assemble Blocks
Channel no. 33
Channel no. 34
Channel no. 35
Channel no. 36
Channel no. 37
Route perimeter channels.
Ripping External Channels

Channel no. 38
Channel no. 39
Channel no. 40
Channel no. 41
Generating Geometry.
Writing cell fpadder to disk
Glue: fpadder completed, Real time: 00:05:45, CPU time: 290.25 seconds.

/local/epoch/bin/geo2glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/geo2glue.log -geoout -tries 30
complex_adder_4_input

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading geometry "complex_adder_4_input"

No. user defined groups: 0

Grouping standard cells

Grouping nets into buses

380 bussed nets eliminated.

Placement data:

16 instances (6 blocks, 10 standard cells in 1 groups)

107 nets (local 5, bussed 79)

331 pins

Running block placement

Block Placement: 9.30 percent done

Block Placement: 18.60 percent done

Block Placement: 27.91 percent done

Block Placement: 37.21 percent done

Block Placement: 46.51 percent done

Block Placement: 55.81 percent done

Block Placement: 65.12 percent done

Block Placement: 74.42 percent done

Block Placement: 83.72 percent done

Block Placement: 94.19 percent done

Block Placement: 100.00 percent done.

Running standard cell placement

Optimizing 1 vertical row of 10 cells

Writing geometry "complex_adder_4_input"

Writing cell complex_adder_4_input to disk

Geo2glue completed on "complex_adder_4_input": Real time: 00:00:23, CPU time:
16.92 seconds.

```

/local/epoch/bin/glueup -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/glueup.log
complex_adder_4_input
GLUE ROUTER version 6.1.
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Glue: reading in geometry file complex_adder_4_input .....
Initialize power connections.
Read in placement expression.
Position export pins.
Adding north port for GND
Adding east port for GND
Adding west port for GND
Adding south port for VDD
Adding east port for VDD
Adding west port for VDD
Route power nets. (minimum_size_rails)
-- GND
-- VDD
Read in footprint information.
Over the cell routing gates.
Over the cell routing blocks.
Overcell Router V2.0
OCR: All standard cell design.
Generating global router input.
Mixed-Mode Global Router V1.1
    Routing group [0]
Assemble Standard cell groups
Assemble Group std_group_0
Assemble Blocks
Contour Router V1.1
Channel no. 0
Channel no. 1
Channel no. 2
Channel no. 3
Channel no. 4
Channel no. 5
Route perimeter channels.
Ripping External Channels
Channel no. 6
Channel no. 7

```

Channel no. 8

Channel no. 9

Generating Geometry.

Writing cell complex_adder_4_input to disk

Glue: complex_adder_4_input completed, Real time: 00:03:35, CPU time: 196.62 seconds.

Cell: hsmult2pipd_c_0#o0 has already been placed and routed.

/local/epoch/bin/dpath5 -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/dpath5.log mplier

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

DATAPATH: Reading datapath geometry.

DATAPATH: Reading footprints.

DATAPATH: Placing columns.

DATAPATH: Initial bit cells placement.

DATAPATH: Routing datapath.

DATAPATH 3.0

size: 8 X 6

Horizontal channels

Contour Router V1.1

channel #8

channel #7

channel #6

channel #5

channel #4

channel #3

channel #2

channel #1

channel #0

Vertical channels

channel #6

channel #5

channel #4

channel #3

channel #2

channel #1

channel #0

DATAPATH: Creating geometry.

Writing cell mplier to disk

/local/epoch/bin/geo2glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r

CDA1u2m1p +msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/geo2glue.log -geoout -tries 30
fpmplier

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading geometry "fpmplier"

No. user defined groups: 1

Grouping standard cells

Grouping nets into buses

403 bussed nets eliminated.

Placement data:

1079 instances (2 blocks, 1077 standard cells in 8 groups)

47 nets (local 820, bussed 28)

107 pins

Running block placement

Block Placement: 9.76 percent done

Block Placement: 19.51 percent done

Block Placement: 29.27 percent done

Block Placement: 39.02 percent done

Block Placement: 48.78 percent done

Block Placement: 58.54 percent done

Block Placement: 68.29 percent done

Block Placement: 78.05 percent done

Block Placement: 89.02 percent done

Block Placement: 98.78 percent done

Block Placement: 100.00 percent done.

Merging groups of 162 and 194 standard cells into 9 rows

Merging groups of 76 and 207 standard cells into 10 rows

Merging groups of 283 and 119 standard cells into 12 rows

Running standard cell placement

Optimizing 2 rows of 49 cells

Optimizing 9 vertical rows of 356 cells

Optimizing 1 row of 91 cells

Optimizing 6 vertical rows of 179 cells

Optimizing 12 rows of 402 cells

Writing geometry "fpmplier"

Writing cell fpmplier to disk

Geo2glue completed on "fpmplier": Real time: 00:06:53, CPU time: 377.57 seconds.

/local/epoch/bin/gluep -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p

+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/gluep.log fpmplier

GLUE ROUTER version 6.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Glue: reading in geometry file fpmplier

Initialize power connections.

Read in placement expression.

Position export pins.

Adding north port for GND

Adding east port for GND

Adding west port for GND

Adding south port for VDD

Adding east port for VDD

Adding west port for VDD

Route power nets. (minimum_size_rails)

-- GND

-- VDD

Read in footprint information.

Over the cell routing gates.

Over the cell routing blocks.

Overcell Router V2.0

Generating global router input.

Mixed-Mode Global Router V1.1

Routing group [0]

Routing group [1]

Routing group [2]

Routing group [3]

Routing group [4]

Assemble Standard cell groups

Assemble Group std_group_1

Merging multiple external standard cell connections for net clear

Merging multiple external standard cell connections for net clock

Merging multiple external standard cell connections for net lost_bit

Assemble Group std_group_3

Contour Router V1.1

Channel no. 0

Channel no. 1

Channel no. 2
Channel no. 3
Channel no. 4
Channel no. 5
Channel no. 6
Channel no. 7
Channel no. 8
Channel no. 9
Channel no. 10
Assemble Group rnder
Channel no. 11
Assemble Group std_group_2
Channel no. 12
Channel no. 13
Channel no. 14
Channel no. 15
Channel no. 16
Assemble Group std_group_0
Channel no. 17
Channel no. 18
Channel no. 19
Channel no. 20
Channel no. 21
Channel no. 22
Channel no. 23
Channel no. 24
Assemble Blocks
Channel no. 25
Channel no. 26
Channel no. 27
Channel no. 28
Channel no. 29
Channel no. 30
Route perimeter channels.
Ripping External Channels
Channel no. 31
Channel no. 32
Channel no. 33
Channel no. 34
Generating Geometry.
Writing cell fpmplier to disk
Glue: fpmplier completed, Real time: 00:08:26, CPU time: 446.35 seconds.

/local/epoch/bin/geo2glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW

```

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/geo2glue.log -geoout -tries 30
complex_mplier
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Reading geometry "complex_mplier"
No. user defined groups: 0
Grouping standard cells
Grouping nets into buses
237 bussed nets eliminated.
Placement data:
    16 instances (6 blocks, 10 standard cells in 1 groups)
    122 nets (local 5, bussed 106)
    203 pins
Running block placement
Block Placement: 8.70 percent done
Block Placement: 17.39 percent done
Block Placement: 26.09 percent done
Block Placement: 34.78 percent done
Block Placement: 43.48 percent done
Block Placement: 52.17 percent done
Block Placement: 60.87 percent done
Block Placement: 69.57 percent done
Block Placement: 78.26 percent done
Block Placement: 86.96 percent done
Block Placement: 95.65 percent done
Block Placement: 100.00 percent done.
Running standard cell placement
Optimizing 1 row of 10 cells
Writing geometry "complex_mplier"
Writing cell complex_mplier to disk
Geo2glue completed on "complex_mplier": Real time: 00:00:24, CPU time: 18.56
seconds.

```

```

/local/epoch/bin/glueup -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/glueup.log complex_mplier
GLUE ROUTER version 6.1.
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos

```


Glue: reading in geometry file complex_mplier

Initialize power connections.

Read in placement expression.

Position export pins.

Adding north port for GND

Adding east port for GND

Adding west port for GND

Adding south port for VDD

Adding east port for VDD

Adding west port for VDD

Route power nets. (minimum_size_rails)

-- GND

-- VDD

Read in footprint information.

Over the cell routing gates.

Over the cell routing blocks.

Overcell Router V2.0

OCR: All standard cell design.

Generating global router input.

Mixed-Mode Global Router V1.1

Routing group [0]

Assemble Standard cell groups

Assemble Group std_group_0

Assemble Blocks

Contour Router V1.1

Channel no. 0

Channel no. 1

Channel no. 2

Channel no. 3

Channel no. 4

Channel no. 5

Route perimeter channels.

Ripping External Channels

Channel no. 6

Channel no. 7

Channel no. 8

Channel no. 9

Generating Geometry.

Writing cell complex_mplier to disk

Glue: complex_mplier completed, Real time: 00:02:59, CPU time: 157.31 seconds.

/local/epoch/bin/geo2glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r

CDA1u2m1p +msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/geo2glue.log -geoout -tries 30
fft

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading geometry "fft"

No. user defined groups: 1

Grouping standard cells

Grouping nets into buses

526 bussed nets eliminated.

Placement data:

686 instances (2 blocks, 684 standard cells in 2 groups)

24 nets (local 406, bussed 17)

204 pins

Running block placement

Block Placement: 8.79 percent done

Block Placement: 17.58 percent done

Block Placement: 26.37 percent done

Block Placement: 35.16 percent done

Block Placement: 43.96 percent done

Block Placement: 52.75 percent done

Block Placement: 61.54 percent done

Block Placement: 70.33 percent done

Block Placement: 79.12 percent done

Block Placement: 87.91 percent done

Block Placement: 96.70 percent done

Block Placement: 100.00 percent done.

Running standard cell placement

Optimizing 14 vertical rows of 672 cells

Optimizing 1 vertical row of 12 cells

Writing geometry "fft"

Writing cell fft to disk

Geo2glue completed on "fft": Real time: 00:02:27, CPU time: 133.76 seconds.

/local/epoch/bin/gluep -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p

+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/gluep.log fft

GLUE ROUTER version 6.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Glue: reading in geometry file fft

Initialize power connections.

Read in placement expression.
 Position export pins.
 Adding north port for GND
 Adding east port for GND
 Adding west port for GND
 Adding south port for VDD
 Adding east port for VDD
 Adding west port for VDD
 Route power nets. (minimum_size_rails)
 -- GND
 -- VDD
 Read in footprint information.
 Over the cell routing gates.
 Over the cell routing blocks.
 Overcell Router V2.0
 OCR: All standard cell design.
 Generating global router input.
 Mixed-Mode Global Router V1.1
 Routing group [0]
 Routing group [1]
 Assemble Standard cell groups
 Assemble Group fft_top
 Contour Router V1.1
 Channel no. 0
 Channel no. 1
 Channel no. 2
 Channel no. 3
 Channel no. 4
 Channel no. 5
 Channel no. 6
 Channel no. 7
 Channel no. 8
 Channel no. 9
 Channel no. 10
 Channel no. 11
 Channel no. 12
 Assemble Group std_group_0
 Assemble Blocks
 Channel no. 13
 Channel no. 14
 Channel no. 15
 Route perimeter channels.
 Ripping External Channels
 Channel no. 16
 Channel no. 17

Channel no. 18

Channel no. 19

Generating Geometry.

Writing cell fft to disk

Glue: fft completed, Real time: 00:12:11, CPU time: 672.89 seconds.

/local/epoch/bin/geo2glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r

CDA1u2m1p +msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/geo2glue.log -geoout -tries 30

fftchip_core

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Reading geometry "fftchip_core"

No. user defined groups: 0

Grouping standard cells

Grouping nets into buses

249 bussed nets eliminated.

Placement data:

70 instances (1 blocks, 69 standard cells in 2 groups)

22 nets (local 0, bussed 19)

204 pins

Running block placement

Block Placement: 9.09 percent done

Block Placement: 18.18 percent done

Block Placement: 27.27 percent done

Block Placement: 36.36 percent done

Block Placement: 45.45 percent done

Block Placement: 54.55 percent done

Block Placement: 63.64 percent done

Block Placement: 72.73 percent done

Block Placement: 100.00 percent done.

Merging groups of 36 and 33 standard cells into 1 rows

Running standard cell placement

Optimizing 1 vertical row of 69 cells

Writing geometry "fftchip_core"

Writing cell fftchip_core to disk

Geo2glue completed on "fftchip_core": Real time: 00:00:11, CPU time: 8.95 seconds.

/local/epoch/bin/glueup -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p

+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/glueup.log fftchip_core

GLUE ROUTER version 6.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Glue: reading in geometry file fftchip_core

Initialize power connections.

Read in placement expression.

Position export pins.

Route power nets. (minimum_size_rails)

-- GND

-- VDD

Read in footprint information.

Over the cell routing gates.

Over the cell routing blocks.

Overcell Router V2.0

OCR: All standard cell design.

Generating global router input.

Mixed-Mode Global Router V1.1

Routing group [0]

Assemble Standard cell groups

Assemble Group std_group_0

Assemble Blocks

Contour Router V1.1

Channel no. 0

Route perimeter channels.

Ripping External Channels

Channel no. 1

Channel no. 2

Channel no. 3

Channel no. 4

Generating Geometry.

Writing cell fftchip_core to disk

Glue: fftchip_core completed, Real time: 00:03:32, CPU time: 197.70 seconds.

/local/epoch/bin/plop5 -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/plop5.log fftchip

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

PACKAGER: Reading top level chip data.

PACKAGER: No package has been specified, automatically selecting package.

```

-----> packcell->name = "cda100p433".
-----> packcell->name = "cda100p540".
-----> packcell->name = "cda120p433".
-----> packcell->name = "cda124p335".
-----> packcell->name = "cda132p400".
-----> packcell->name = "cda132p410".
-----> packcell->name = "cda132p540".
-----> packcell->name = "cda16p120".
-----> packcell->name = "cda16p315".
-----> packcell->name = "cda180p480".
-----> packcell->name = "cda20p160".
-----> packcell->name = "cda22p140".
-----> packcell->name = "cda24p350".
-----> packcell->name = "cda28p250".
-----> packcell->name = "cda28p310".
-----> packcell->name = "cda40p250".
-----> packcell->name = "cda40p300".
-----> packcell->name = "cda48p250".
-----> packcell->name = "cda48p310".
-----> packcell->name = "cda48p350".
-----> packcell->name = "cda64p400".
-----> packcell->name = "cda68p400".
-----> packcell->name = "cda72p340".
-----> packcell->name = "cda84p400".
-----> packcell->name = "cda84p450".
-----> packcell->name = "cda84p470".

```

----->Can't find any appropriate package, place pads without package.

PACKAGER: Placing pads.

MOAT ROUTER: Routing East channel.

Contour Router V1.1

MOAT ROUTER: Routing West channel.

MOAT ROUTER: Routing North channel.

MOAT ROUTER: Routing South channel.

MOAT ROUTER: Done.

Writing cell fftchip to disk

Die size : 409 x 576 (mils) Scribe spacing : 0 (mils)

Die size : 10391 x 14625 (um) Scribe spacing : 1 (um)

Module size : 375 x 543 (mils) NONE

Module size : 9520 x 13795 (um)

Number of nets : 408
Max. routing space : 6 (mils) Min. routing space : 4 (mils)
Max. routing space : 165 (um) Min. routing space : 99 (um)
Number of pads : 210

Max. pad height : 6 (mils) Min. pad height : 6 (mils)
Max. pad height : 139 (um) Min. pad height : 139 (um)

/local/epoch/bin/pdabs -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/pdabs.log -B -A fftchip
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
PDABS:

*** Power Dissipation and Buffer Sizing ***

Buffer Size Method: Automatic
PDABS: Clock Frequency is 10 MHz.
Default Duty Cycle is 50 %.
Supply Voltage is 5 Volts.
Temperature is 25 degrees C
PDABS: Reading database ...
PDABS: Calculating loads.
PDABS: Checking buffers and rails.
PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 38X buffer is recommended for instance Counter:reg1
Output with maximum load is Q.
The maximum size (7X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 40X buffer is recommended for instance Counter:reg0
Output with maximum load is Q.
The maximum size (7X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 7X buffer is recommended for instance Latch1:reg98
Output with maximum load is Q.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 141X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 7X buffer is recommended for instance Mux2:mux34
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 7X buffer is recommended for instance Mux1:mux62
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 7X buffer is recommended for instance Mux1:mux37
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 7X buffer is recommended for instance Mux1:mux36
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 7X buffer is recommended for instance Mux1:mux35
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 7X buffer is recommended for instance Mux1:mux34
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 11X buffer is recommended for instance or5
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 12X buffer is recommended for instance or4
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 12X buffer is recommended for instance or3
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 12X buffer is recommended for instance or2
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 12X buffer is recommended for instance or1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path: fftchip/fftchip_core/chip
An external 28X buffer is recommended for instance Counter:inc0:incdec1
Output is COUT.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2
An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2

An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2
An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder2
An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Adder1
An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier4
An external 19X buffer is recommended for instance reg4i:reg14
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier4
An external 20X buffer is recommended for instance reg4i:reg13
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier4

An external 19X buffer is recommended for instance reg4i:reg12
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier4
An external 20X buffer is recommended for instance reg4i:reg11
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier4
An external 19X buffer is recommended for instance reg4i:reg10
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier4
An external 14X buffer is recommended for instance reg4i:reg68
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier3
An external 19X buffer is recommended for instance reg4i:reg14
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier3
An external 20X buffer is recommended for instance reg4i:reg13
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier3
An external 19X buffer is recommended for instance reg4i:reg12
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier3
An external 20X buffer is recommended for instance reg4i:reg11
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier3
An external 19X buffer is recommended for instance reg4i:reg10
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier3
An external 14X buffer is recommended for instance reg4i:reg68
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier2
An external 19X buffer is recommended for instance reg4i:reg14
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier2
An external 20X buffer is recommended for instance reg4i:reg13
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier2
An external 19X buffer is recommended for instance reg4i:reg12
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier2
An external 20X buffer is recommended for instance reg4i:reg11
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier2
An external 19X buffer is recommended for instance reg4i:reg10
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier2

An external 14X buffer is recommended for instance reg4i:reg68
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier1
An external 19X buffer is recommended for instance reg4i:reg14
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier1
An external 20X buffer is recommended for instance reg4i:reg13
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier1
An external 19X buffer is recommended for instance reg4i:reg12
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier1
An external 20X buffer is recommended for instance reg4i:reg11
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier1
An external 19X buffer is recommended for instance reg4i:reg10
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Multiplier/FP_Multiplier1
An external 14X buffer is recommended for instance reg4i:reg68
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder6
An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5

An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5
An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5
An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder5
An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4
An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4
An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4
An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder4

An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder3
An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2
An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2

An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder2
An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 21X buffer is recommended for instance reg2i:reg5
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg4
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg3
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg2
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg1
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 20X buffer is recommended for instance reg2i:reg0
Output with maximum load is Q.
The maximum size (9X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 7X buffer is recommended for instance and1
Output with maximum load is Y.
The maximum size (6X) will be used.

PDABS:(WARNING) Parent instance path:
fftchip/fftchip_core/chip/Complex_Adder_4_Input/FP_Adder1
An external 10X buffer is recommended for instance or15
Output with maximum load is Y.
The maximum size (6X) will be used.

Creating new modules for composite cell: fftchip_core
Instance: outpad2:iocntr210 (BUFFER_SIZE = 5)
Writing specification cell pad_driver#02, compiler pad_driver
Instance: outpad2:iocntr209 (BUFFER_SIZE = 5)
Instance: outpad2:iocntr208 (BUFFER_SIZE = 5)
Instance: outpad2:iocntr207 (BUFFER_SIZE = 5)
Instance: outpad2:iocntr206 (BUFFER_SIZE = 5)
Instance: outpad1:iocntr205 (BUFFER_SIZE = 5)
Instance: outpad1:iocntr204 (BUFFER_SIZE = 5)
Instance: outpad1:iocntr197 (BUFFER_SIZE = 3)
Writing specification cell pad_driver#03, compiler pad_driver
Instance: outpad1:iocntr195 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr194 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr193 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr192 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr191 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr190 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr189 (BUFFER_SIZE = 2)
Writing specification cell pad_driver#04, compiler pad_driver
Instance: outpad1:iocntr188 (BUFFER_SIZE = 2)
Instance: outpad1:iocntr187 (BUFFER_SIZE = 2)
Instance: outpad1:iocntr186 (BUFFER_SIZE = 2)
Instance: outpad1:iocntr185 (BUFFER_SIZE = 2)
Instance: outpad1:iocntr184 (BUFFER_SIZE = 2)
Instance: outpad1:iocntr183 (BUFFER_SIZE = 2)
Instance: outpad1:iocntr182 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr181 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr180 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr179 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr178 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr177 (BUFFER_SIZE = 3)
Instance: outpad1:iocntr168 (BUFFER_SIZE = 5)

Instance: outpad1:iocntr165 (BUFFER_SIZE = 5)
 Instance: outpad1:iocntr164 (BUFFER_SIZE = 5)
 Instance: outpad1:iocntr163 (BUFFER_SIZE = 5)
 Instance: outpad1:iocntr162 (BUFFER_SIZE = 5)
 Instance: outpad1:iocntr161 (BUFFER_SIZE = 5)
 Instance: outpad1:iocntr160 (BUFFER_SIZE = 10)
 Writing specification cell pad_driver#o5, compiler pad_driver
 Instance: outpad1:iocntr159 (BUFFER_SIZE = 10)
 Instance: outpad1:iocntr158 (BUFFER_SIZE = 10)
 Instance: outpad1:iocntr157 (BUFFER_SIZE = 10)
 Instance: outpad1:iocntr156 (BUFFER_SIZE = 10)
 Instance: outpad1:iocntr155 (BUFFER_SIZE = 9)
 Writing specification cell pad_driver#o6, compiler pad_driver
 Instance: outpad1:iocntr154 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr153 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr152 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr151 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr150 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr149 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr148 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr147 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr146 (BUFFER_SIZE = 9)
 Instance: outpad1:iocntr145 (BUFFER_SIZE = 8)
 Writing specification cell pad_driver#o7, compiler pad_driver
 Instance: outpad1:iocntr144 (BUFFER_SIZE = 8)
 Instance: outpad1:iocntr143 (BUFFER_SIZE = 8)
 Instance: outpad1:iocntr142 (BUFFER_SIZE = 8)
 Creating new modules for composite cell: fft
 Instance: reg2:reg61 (BUFFER_SIZE = 2)
 Writing specification cell gtdff#o2, compiler gtdff
 Instance: reg2:reg1 (BUFFER_SIZE = 2)
 Instance: reg1:reg61 (BUFFER_SIZE = 2)
 Instance: reg1:reg50 (BUFFER_SIZE = 2)
 Instance: reg1:reg42 (BUFFER_SIZE = 2)
 Instance: reg1:reg29 (BUFFER_SIZE = 2)
 Instance: reg1:reg28 (BUFFER_SIZE = 2)
 Instance: reg1:reg26 (BUFFER_SIZE = 2)
 Instance: reg1:reg24 (BUFFER_SIZE = 2)
 Instance: reg1:reg20 (BUFFER_SIZE = 2)
 Instance: reg1:reg19 (BUFFER_SIZE = 2)
 Instance: reg1:reg11 (BUFFER_SIZE = 2)
 Instance: reg1:reg9 (BUFFER_SIZE = 2)
 Instance: s1_Izneg_sign_D_real (BUFFER_SIZE = 2)
 Writing specification cell stdinv#o1, compiler gtgate
 Instance: s1_Izneg_sign_D_imag (BUFFER_SIZE = 2)

Instance: s1_Izneg_sign_C_real (BUFFER_SIZE = 2)
 Instance: s1_Izneg_sign_C_imag (BUFFER_SIZE = 2)
 Instance: s1_Izneg_sign_B_real (BUFFER_SIZE = 2)
 Instance: s1_Izneg_sign_B_imag (BUFFER_SIZE = 2)
 Instance: Counter:reg1 (BUFFER_SIZE = 7)
 Writing specification cell gtdff#o3, compiler gtdff
 Instance: Counter:reg0 (BUFFER_SIZE = 7)
 Instance: Latch1:reg127 (BUFFER_SIZE = 6)
 Writing specification cell gtlatch#o1, compiler gtlatch
 Instance: Latch1:reg126 (BUFFER_SIZE = 6)
 Instance: Latch1:reg125 (BUFFER_SIZE = 6)
 Instance: Latch1:reg124 (BUFFER_SIZE = 5)
 Writing specification cell gtlatch#o2, compiler gtlatch
 Instance: Latch1:reg123 (BUFFER_SIZE = 6)
 Instance: Latch1:reg122 (BUFFER_SIZE = 5)
 Instance: Latch1:reg121 (BUFFER_SIZE = 6)
 Instance: Latch1:reg120 (BUFFER_SIZE = 6)
 Instance: Latch1:reg119 (BUFFER_SIZE = 6)
 Instance: Latch1:reg118 (BUFFER_SIZE = 6)
 Instance: Latch1:reg117 (BUFFER_SIZE = 5)
 Instance: Latch1:reg116 (BUFFER_SIZE = 5)
 Instance: Latch1:reg115 (BUFFER_SIZE = 5)
 Instance: Latch1:reg114 (BUFFER_SIZE = 6)
 Instance: Latch1:reg113 (BUFFER_SIZE = 5)
 Instance: Latch1:reg112 (BUFFER_SIZE = 5)
 Instance: Latch1:reg111 (BUFFER_SIZE = 6)
 Instance: Latch1:reg110 (BUFFER_SIZE = 5)
 Instance: Latch1:reg109 (BUFFER_SIZE = 5)
 Instance: Latch1:reg108 (BUFFER_SIZE = 5)
 Instance: Latch1:reg107 (BUFFER_SIZE = 5)
 Instance: Latch1:reg106 (BUFFER_SIZE = 5)
 Instance: Latch1:reg105 (BUFFER_SIZE = 5)
 Instance: Latch1:reg104 (BUFFER_SIZE = 5)
 Instance: Latch1:reg103 (BUFFER_SIZE = 5)
 Instance: Latch1:reg102 (BUFFER_SIZE = 6)
 Instance: Latch1:reg101 (BUFFER_SIZE = 6)
 Instance: Latch1:reg100 (BUFFER_SIZE = 6)
 Instance: Latch1:reg99 (BUFFER_SIZE = 6)
 Instance: Latch1:reg98 (BUFFER_SIZE = 6)
 Instance: Latch1:reg97 (BUFFER_SIZE = 5)
 Instance: Latch1:reg96 (BUFFER_SIZE = 5)
 Instance: Latch1:reg95 (BUFFER_SIZE = 2)
 Writing specification cell gtlatch#o3, compiler gtlatch
 Instance: Latch1:reg94 (BUFFER_SIZE = 2)
 Instance: Latch1:reg93 (BUFFER_SIZE = 2)

Instance: Latch1:reg92	(BUFFER_SIZE = 2)
Instance: Latch1:reg91	(BUFFER_SIZE = 2)
Instance: Latch1:reg90	(BUFFER_SIZE = 2)
Instance: Latch1:reg89	(BUFFER_SIZE = 2)
Instance: Latch1:reg88	(BUFFER_SIZE = 2)
Instance: Latch1:reg87	(BUFFER_SIZE = 2)
Instance: Latch1:reg86	(BUFFER_SIZE = 2)
Instance: Latch1:reg85	(BUFFER_SIZE = 2)
Instance: Latch1:reg84	(BUFFER_SIZE = 2)
Instance: Latch1:reg83	(BUFFER_SIZE = 2)
Instance: Latch1:reg82	(BUFFER_SIZE = 2)
Instance: Latch1:reg81	(BUFFER_SIZE = 2)
Instance: Latch1:reg80	(BUFFER_SIZE = 2)
Instance: Latch1:reg79	(BUFFER_SIZE = 2)
Instance: Latch1:reg78	(BUFFER_SIZE = 2)
Instance: Latch1:reg77	(BUFFER_SIZE = 2)
Instance: Latch1:reg76	(BUFFER_SIZE = 2)
Instance: Latch1:reg75	(BUFFER_SIZE = 2)
Instance: Latch1:reg74	(BUFFER_SIZE = 2)
Instance: Latch1:reg73	(BUFFER_SIZE = 2)
Instance: Latch1:reg72	(BUFFER_SIZE = 2)
Instance: Latch1:reg71	(BUFFER_SIZE = 2)
Instance: Latch1:reg70	(BUFFER_SIZE = 2)
Instance: Latch1:reg69	(BUFFER_SIZE = 2)
Instance: Latch1:reg68	(BUFFER_SIZE = 2)
Instance: Latch1:reg67	(BUFFER_SIZE = 2)
Instance: Latch1:reg66	(BUFFER_SIZE = 2)
Instance: Latch1:reg65	(BUFFER_SIZE = 2)
Instance: Latch1:reg64	(BUFFER_SIZE = 2)
Instance: Latch1:reg63	(BUFFER_SIZE = 2)
Instance: Latch1:reg31	(BUFFER_SIZE = 2)
Instance: Latch1:reg30	(BUFFER_SIZE = 2)
Instance: Latch1:reg29	(BUFFER_SIZE = 2)
Instance: Latch1:reg28	(BUFFER_SIZE = 2)
Instance: Latch1:reg27	(BUFFER_SIZE = 2)
Instance: Latch1:reg26	(BUFFER_SIZE = 2)
Instance: Latch1:reg25	(BUFFER_SIZE = 2)
Instance: Latch1:reg24	(BUFFER_SIZE = 2)
Instance: Latch1:reg23	(BUFFER_SIZE = 2)
Instance: Latch1:reg22	(BUFFER_SIZE = 2)
Instance: Latch1:reg21	(BUFFER_SIZE = 2)
Instance: Latch1:reg20	(BUFFER_SIZE = 2)
Instance: Latch1:reg19	(BUFFER_SIZE = 2)
Instance: Latch1:reg18	(BUFFER_SIZE = 2)
Instance: Latch1:reg17	(BUFFER_SIZE = 2)

Instance: Latch1:reg16 (BUFFER_SIZE = 2)
 Instance: Latch1:reg15 (BUFFER_SIZE = 2)
 Instance: Latch1:reg14 (BUFFER_SIZE = 2)
 Instance: Latch1:reg13 (BUFFER_SIZE = 2)
 Instance: Latch1:reg12 (BUFFER_SIZE = 2)
 Instance: Latch1:reg11 (BUFFER_SIZE = 2)
 Instance: Latch1:reg10 (BUFFER_SIZE = 2)
 Instance: Latch1:reg9 (BUFFER_SIZE = 2)
 Instance: Latch1:reg8 (BUFFER_SIZE = 2)
 Instance: Latch1:reg7 (BUFFER_SIZE = 2)
 Instance: Latch1:reg6 (BUFFER_SIZE = 2)
 Instance: Latch1:reg5 (BUFFER_SIZE = 2)
 Instance: Latch1:reg4 (BUFFER_SIZE = 2)
 Instance: Latch1:reg3 (BUFFER_SIZE = 2)
 Instance: Latch1:reg2 (BUFFER_SIZE = 2)
 Instance: Latch1:reg1 (BUFFER_SIZE = 2)
 Instance: Latch1:reg0 (BUFFER_SIZE = 2)
 Instance: Latch2:reg127 (BUFFER_SIZE = 2)
 Instance: Latch2:reg126 (BUFFER_SIZE = 2)
 Instance: Latch2:reg125 (BUFFER_SIZE = 2)
 Instance: Latch2:reg124 (BUFFER_SIZE = 2)
 Instance: Latch2:reg123 (BUFFER_SIZE = 2)
 Instance: Latch2:reg122 (BUFFER_SIZE = 2)
 Instance: Latch2:reg121 (BUFFER_SIZE = 2)
 Instance: Latch2:reg120 (BUFFER_SIZE = 2)
 Instance: Latch2:reg119 (BUFFER_SIZE = 2)
 Instance: Latch2:reg118 (BUFFER_SIZE = 2)
 Instance: Latch2:reg116 (BUFFER_SIZE = 2)
 Instance: Latch2:reg113 (BUFFER_SIZE = 2)
 Instance: Latch2:reg112 (BUFFER_SIZE = 2)
 Instance: Latch2:reg111 (BUFFER_SIZE = 2)
 Instance: Latch2:reg110 (BUFFER_SIZE = 2)
 Instance: Latch2:reg106 (BUFFER_SIZE = 2)
 Instance: Latch2:reg105 (BUFFER_SIZE = 2)
 Instance: Latch2:reg104 (BUFFER_SIZE = 2)
 Instance: Latch2:reg103 (BUFFER_SIZE = 2)
 Instance: Latch2:reg102 (BUFFER_SIZE = 2)
 Instance: Latch2:reg101 (BUFFER_SIZE = 3)
 Writing specification cell gtlatch#o4, compiler gtlatch
 Instance: Latch2:reg100 (BUFFER_SIZE = 3)
 Instance: Latch2:reg99 (BUFFER_SIZE = 3)
 Instance: Latch2:reg98 (BUFFER_SIZE = 4)
 Writing specification cell gtlatch#o5, compiler gtlatch
 Instance: Latch2:reg97 (BUFFER_SIZE = 2)
 Instance: Latch2:reg96 (BUFFER_SIZE = 2)

Instance: Latch2:reg95	(BUFFER_SIZE = 2)
Instance: Latch2:reg94	(BUFFER_SIZE = 2)
Instance: Latch2:reg93	(BUFFER_SIZE = 2)
Instance: Latch2:reg92	(BUFFER_SIZE = 2)
Instance: Latch2:reg91	(BUFFER_SIZE = 2)
Instance: Latch2:reg90	(BUFFER_SIZE = 2)
Instance: Latch2:reg89	(BUFFER_SIZE = 2)
Instance: Latch2:reg88	(BUFFER_SIZE = 2)
Instance: Latch2:reg87	(BUFFER_SIZE = 2)
Instance: Latch2:reg86	(BUFFER_SIZE = 2)
Instance: Latch2:reg85	(BUFFER_SIZE = 2)
Instance: Latch2:reg84	(BUFFER_SIZE = 2)
Instance: Latch2:reg83	(BUFFER_SIZE = 2)
Instance: Latch2:reg82	(BUFFER_SIZE = 2)
Instance: Latch2:reg81	(BUFFER_SIZE = 2)
Instance: Latch2:reg80	(BUFFER_SIZE = 2)
Instance: Latch2:reg79	(BUFFER_SIZE = 2)
Instance: Latch2:reg78	(BUFFER_SIZE = 2)
Instance: Latch2:reg77	(BUFFER_SIZE = 2)
Instance: Latch2:reg76	(BUFFER_SIZE = 2)
Instance: Latch2:reg75	(BUFFER_SIZE = 2)
Instance: Latch2:reg74	(BUFFER_SIZE = 2)
Instance: Latch2:reg73	(BUFFER_SIZE = 2)
Instance: Latch2:reg72	(BUFFER_SIZE = 2)
Instance: Latch2:reg71	(BUFFER_SIZE = 2)
Instance: Latch2:reg70	(BUFFER_SIZE = 2)
Instance: Latch2:reg69	(BUFFER_SIZE = 2)
Instance: Latch2:reg68	(BUFFER_SIZE = 2)
Instance: Latch2:reg67	(BUFFER_SIZE = 2)
Instance: Latch2:reg66	(BUFFER_SIZE = 2)
Instance: Latch2:reg65	(BUFFER_SIZE = 2)
Instance: Latch2:reg64	(BUFFER_SIZE = 2)
Instance: Latch2:reg63	(BUFFER_SIZE = 2)
Instance: Latch2:reg59	(BUFFER_SIZE = 2)
Instance: Latch2:reg42	(BUFFER_SIZE = 2)
Instance: Latch2:reg31	(BUFFER_SIZE = 2)
Instance: Latch2:reg30	(BUFFER_SIZE = 2)
Instance: Latch2:reg29	(BUFFER_SIZE = 2)
Instance: Latch2:reg28	(BUFFER_SIZE = 2)
Instance: Latch2:reg27	(BUFFER_SIZE = 2)
Instance: Latch2:reg26	(BUFFER_SIZE = 2)
Instance: Latch2:reg25	(BUFFER_SIZE = 2)
Instance: Latch2:reg24	(BUFFER_SIZE = 2)
Instance: Latch2:reg23	(BUFFER_SIZE = 2)
Instance: Latch2:reg22	(BUFFER_SIZE = 2)

Instance: Latch2:reg21 (BUFFER_SIZE = 2)
 Instance: Latch2:reg20 (BUFFER_SIZE = 2)
 Instance: Latch2:reg19 (BUFFER_SIZE = 2)
 Instance: Latch2:reg18 (BUFFER_SIZE = 2)
 Instance: Latch2:reg17 (BUFFER_SIZE = 2)
 Instance: Latch2:reg16 (BUFFER_SIZE = 2)
 Instance: Latch2:reg15 (BUFFER_SIZE = 2)
 Instance: Latch2:reg14 (BUFFER_SIZE = 2)
 Instance: Latch2:reg13 (BUFFER_SIZE = 2)
 Instance: Latch2:reg12 (BUFFER_SIZE = 2)
 Instance: Latch2:reg11 (BUFFER_SIZE = 2)
 Instance: Latch2:reg10 (BUFFER_SIZE = 2)
 Instance: Latch2:reg9 (BUFFER_SIZE = 2)
 Instance: Latch2:reg8 (BUFFER_SIZE = 2)
 Instance: Latch2:reg7 (BUFFER_SIZE = 2)
 Instance: Latch2:reg6 (BUFFER_SIZE = 2)
 Instance: Latch2:reg5 (BUFFER_SIZE = 2)
 Instance: Latch2:reg4 (BUFFER_SIZE = 2)
 Instance: Latch2:reg3 (BUFFER_SIZE = 2)
 Instance: Latch2:reg2 (BUFFER_SIZE = 2)
 Instance: Latch2:reg1 (BUFFER_SIZE = 2)
 Instance: Latch2:reg0 (BUFFER_SIZE = 2)
 Instance: and1 (BUFFER_SIZE = 6)
 Writing specification cell stdand2#o1, compiler gtgate
 Instance: Mux3:mux63 (BUFFER_SIZE = 5)
 Writing specification cell gtmux#o2, compiler gtmux
 Instance: Mux3:mux62 (BUFFER_SIZE = 5)
 Instance: Mux3:mux61 (BUFFER_SIZE = 5)
 Instance: Mux3:mux60 (BUFFER_SIZE = 4)
 Writing specification cell gtmux#o3, compiler gtmux
 Instance: Mux3:mux59 (BUFFER_SIZE = 5)
 Instance: Mux3:mux58 (BUFFER_SIZE = 4)
 Instance: Mux3:mux57 (BUFFER_SIZE = 5)
 Instance: Mux3:mux56 (BUFFER_SIZE = 5)
 Instance: Mux3:mux55 (BUFFER_SIZE = 4)
 Instance: Mux3:mux54 (BUFFER_SIZE = 5)
 Instance: Mux3:mux53 (BUFFER_SIZE = 5)
 Instance: Mux3:mux52 (BUFFER_SIZE = 4)
 Instance: Mux3:mux51 (BUFFER_SIZE = 4)
 Instance: Mux3:mux50 (BUFFER_SIZE = 5)
 Instance: Mux3:mux49 (BUFFER_SIZE = 4)
 Instance: Mux3:mux48 (BUFFER_SIZE = 5)
 Instance: Mux3:mux47 (BUFFER_SIZE = 4)
 Instance: Mux3:mux46 (BUFFER_SIZE = 4)
 Instance: Mux3:mux45 (BUFFER_SIZE = 4)

Instance: Mux3:mux44 (BUFFER_SIZE = 4)
 Instance: Mux3:mux43 (BUFFER_SIZE = 4)
 Instance: Mux3:mux42 (BUFFER_SIZE = 4)
 Instance: Mux3:mux41 (BUFFER_SIZE = 5)
 Instance: Mux3:mux40 (BUFFER_SIZE = 4)
 Instance: Mux3:mux39 (BUFFER_SIZE = 4)
 Instance: Mux3:mux38 (BUFFER_SIZE = 5)
 Instance: Mux3:mux37 (BUFFER_SIZE = 6)
 Writing specification cell gtmux#o4, compiler gtmux
 Instance: Mux3:mux36 (BUFFER_SIZE = 6)
 Instance: Mux3:mux35 (BUFFER_SIZE = 6)
 Instance: Mux3:mux34 (BUFFER_SIZE = 6)
 Instance: Mux3:mux33 (BUFFER_SIZE = 5)
 Instance: Mux3:mux32 (BUFFER_SIZE = 5)
 Instance: Mux3:mux31 (BUFFER_SIZE = 2)
 Writing specification cell gtmux#o5, compiler gtmux
 Instance: Mux3:mux30 (BUFFER_SIZE = 3)
 Writing specification cell gtmux#o6, compiler gtmux
 Instance: Mux3:mux29 (BUFFER_SIZE = 2)
 Instance: Mux3:mux28 (BUFFER_SIZE = 2)
 Instance: Mux3:mux27 (BUFFER_SIZE = 3)
 Instance: Mux3:mux26 (BUFFER_SIZE = 2)
 Instance: Mux3:mux25 (BUFFER_SIZE = 3)
 Instance: Mux3:mux24 (BUFFER_SIZE = 3)
 Instance: Mux3:mux23 (BUFFER_SIZE = 2)
 Instance: Mux3:mux22 (BUFFER_SIZE = 3)
 Instance: Mux3:mux21 (BUFFER_SIZE = 2)
 Instance: Mux3:mux20 (BUFFER_SIZE = 2)
 Instance: Mux3:mux19 (BUFFER_SIZE = 2)
 Instance: Mux3:mux18 (BUFFER_SIZE = 3)
 Instance: Mux3:mux17 (BUFFER_SIZE = 3)
 Instance: Mux3:mux16 (BUFFER_SIZE = 3)
 Instance: Mux3:mux15 (BUFFER_SIZE = 3)
 Instance: Mux3:mux14 (BUFFER_SIZE = 3)
 Instance: Mux3:mux13 (BUFFER_SIZE = 2)
 Instance: Mux3:mux12 (BUFFER_SIZE = 2)
 Instance: Mux3:mux11 (BUFFER_SIZE = 2)
 Instance: Mux3:mux10 (BUFFER_SIZE = 2)
 Instance: Mux3:mux9 (BUFFER_SIZE = 3)
 Instance: Mux3:mux8 (BUFFER_SIZE = 3)
 Instance: Mux3:mux7 (BUFFER_SIZE = 3)
 Instance: Mux3:mux6 (BUFFER_SIZE = 3)
 Instance: Mux3:mux5 (BUFFER_SIZE = 4)
 Instance: Mux3:mux4 (BUFFER_SIZE = 4)
 Instance: Mux3:mux3 (BUFFER_SIZE = 4)

Instance: Mux3:mux2	(BUFFER_SIZE = 5)
Instance: Mux3:mux1	(BUFFER_SIZE = 3)
Instance: Mux3:mux0	(BUFFER_SIZE = 3)
Instance: Mux2:mux63	(BUFFER_SIZE = 5)
Instance: Mux2:mux62	(BUFFER_SIZE = 5)
Instance: Mux2:mux61	(BUFFER_SIZE = 5)
Instance: Mux2:mux60	(BUFFER_SIZE = 4)
Instance: Mux2:mux59	(BUFFER_SIZE = 5)
Instance: Mux2:mux58	(BUFFER_SIZE = 5)
Instance: Mux2:mux57	(BUFFER_SIZE = 5)
Instance: Mux2:mux56	(BUFFER_SIZE = 5)
Instance: Mux2:mux55	(BUFFER_SIZE = 5)
Instance: Mux2:mux54	(BUFFER_SIZE = 5)
Instance: Mux2:mux53	(BUFFER_SIZE = 5)
Instance: Mux2:mux52	(BUFFER_SIZE = 4)
Instance: Mux2:mux51	(BUFFER_SIZE = 4)
Instance: Mux2:mux50	(BUFFER_SIZE = 5)
Instance: Mux2:mux49	(BUFFER_SIZE = 5)
Instance: Mux2:mux48	(BUFFER_SIZE = 5)
Instance: Mux2:mux47	(BUFFER_SIZE = 5)
Instance: Mux2:mux46	(BUFFER_SIZE = 5)
Instance: Mux2:mux45	(BUFFER_SIZE = 4)
Instance: Mux2:mux44	(BUFFER_SIZE = 4)
Instance: Mux2:mux43	(BUFFER_SIZE = 4)
Instance: Mux2:mux42	(BUFFER_SIZE = 4)
Instance: Mux2:mux41	(BUFFER_SIZE = 4)
Instance: Mux2:mux40	(BUFFER_SIZE = 5)
Instance: Mux2:mux39	(BUFFER_SIZE = 4)
Instance: Mux2:mux38	(BUFFER_SIZE = 5)
Instance: Mux2:mux37	(BUFFER_SIZE = 6)
Instance: Mux2:mux36	(BUFFER_SIZE = 6)
Instance: Mux2:mux35	(BUFFER_SIZE = 6)
Instance: Mux2:mux34	(BUFFER_SIZE = 6)
Instance: Mux2:mux33	(BUFFER_SIZE = 5)
Instance: Mux2:mux32	(BUFFER_SIZE = 4)
Instance: Mux2:mux31	(BUFFER_SIZE = 4)
Instance: Mux2:mux30	(BUFFER_SIZE = 3)
Instance: Mux2:mux29	(BUFFER_SIZE = 3)
Instance: Mux2:mux28	(BUFFER_SIZE = 2)
Instance: Mux2:mux27	(BUFFER_SIZE = 3)
Instance: Mux2:mux26	(BUFFER_SIZE = 2)
Instance: Mux2:mux25	(BUFFER_SIZE = 4)
Instance: Mux2:mux24	(BUFFER_SIZE = 3)
Instance: Mux2:mux23	(BUFFER_SIZE = 4)
Instance: Mux2:mux22	(BUFFER_SIZE = 4)

Instance: Mux2:mux21	(BUFFER_SIZE = 3)
Instance: Mux2:mux20	(BUFFER_SIZE = 3)
Instance: Mux2:mux19	(BUFFER_SIZE = 2)
Instance: Mux2:mux18	(BUFFER_SIZE = 3)
Instance: Mux2:mux17	(BUFFER_SIZE = 3)
Instance: Mux2:mux16	(BUFFER_SIZE = 3)
Instance: Mux2:mux15	(BUFFER_SIZE = 4)
Instance: Mux2:mux14	(BUFFER_SIZE = 3)
Instance: Mux2:mux13	(BUFFER_SIZE = 3)
Instance: Mux2:mux12	(BUFFER_SIZE = 2)
Instance: Mux2:mux11	(BUFFER_SIZE = 2)
Instance: Mux2:mux10	(BUFFER_SIZE = 3)
Instance: Mux2:mux9	(BUFFER_SIZE = 3)
Instance: Mux2:mux8	(BUFFER_SIZE = 3)
Instance: Mux2:mux7	(BUFFER_SIZE = 3)
Instance: Mux2:mux6	(BUFFER_SIZE = 3)
Instance: Mux2:mux5	(BUFFER_SIZE = 4)
Instance: Mux2:mux4	(BUFFER_SIZE = 4)
Instance: Mux2:mux3	(BUFFER_SIZE = 4)
Instance: Mux2:mux2	(BUFFER_SIZE = 5)
Instance: Mux2:mux1	(BUFFER_SIZE = 3)
Instance: Mux2:mux0	(BUFFER_SIZE = 3)
Instance: Mux1:mux63	(BUFFER_SIZE = 6)
Instance: Mux1:mux62	(BUFFER_SIZE = 6)
Instance: Mux1:mux61	(BUFFER_SIZE = 6)
Instance: Mux1:mux60	(BUFFER_SIZE = 5)
Instance: Mux1:mux59	(BUFFER_SIZE = 6)
Instance: Mux1:mux58	(BUFFER_SIZE = 6)
Instance: Mux1:mux57	(BUFFER_SIZE = 6)
Instance: Mux1:mux56	(BUFFER_SIZE = 6)
Instance: Mux1:mux55	(BUFFER_SIZE = 6)
Instance: Mux1:mux54	(BUFFER_SIZE = 6)
Instance: Mux1:mux53	(BUFFER_SIZE = 6)
Instance: Mux1:mux52	(BUFFER_SIZE = 6)
Instance: Mux1:mux51	(BUFFER_SIZE = 5)
Instance: Mux1:mux50	(BUFFER_SIZE = 6)
Instance: Mux1:mux49	(BUFFER_SIZE = 5)
Instance: Mux1:mux48	(BUFFER_SIZE = 6)
Instance: Mux1:mux47	(BUFFER_SIZE = 6)
Instance: Mux1:mux46	(BUFFER_SIZE = 6)
Instance: Mux1:mux45	(BUFFER_SIZE = 6)
Instance: Mux1:mux44	(BUFFER_SIZE = 5)
Instance: Mux1:mux43	(BUFFER_SIZE = 6)
Instance: Mux1:mux42	(BUFFER_SIZE = 6)
Instance: Mux1:mux41	(BUFFER_SIZE = 6)

Instance: Mux1:mux40 (BUFFER_SIZE = 6)
 Instance: Mux1:mux39 (BUFFER_SIZE = 6)
 Instance: Mux1:mux38 (BUFFER_SIZE = 6)
 Instance: Mux1:mux37 (BUFFER_SIZE = 6)
 Instance: Mux1:mux36 (BUFFER_SIZE = 6)
 Instance: Mux1:mux35 (BUFFER_SIZE = 6)
 Instance: Mux1:mux34 (BUFFER_SIZE = 6)
 Instance: Mux1:mux33 (BUFFER_SIZE = 6)
 Instance: Mux1:mux32 (BUFFER_SIZE = 6)
 Instance: Mux1:mux30 (BUFFER_SIZE = 2)
 Instance: Mux1:mux29 (BUFFER_SIZE = 2)
 Instance: Mux1:mux28 (BUFFER_SIZE = 2)
 Instance: Mux1:mux27 (BUFFER_SIZE = 2)
 Instance: Mux1:mux26 (BUFFER_SIZE = 3)
 Instance: Mux1:mux25 (BUFFER_SIZE = 2)
 Instance: Mux1:mux24 (BUFFER_SIZE = 2)
 Instance: Mux1:mux23 (BUFFER_SIZE = 2)
 Instance: Mux1:mux22 (BUFFER_SIZE = 2)
 Instance: Mux1:mux21 (BUFFER_SIZE = 2)
 Instance: Mux1:mux20 (BUFFER_SIZE = 2)
 Instance: Mux1:mux19 (BUFFER_SIZE = 2)
 Instance: Mux1:mux18 (BUFFER_SIZE = 2)
 Instance: Mux1:mux17 (BUFFER_SIZE = 2)
 Instance: Mux1:mux16 (BUFFER_SIZE = 2)
 Instance: Mux1:mux15 (BUFFER_SIZE = 2)
 Instance: Mux1:mux14 (BUFFER_SIZE = 2)
 Instance: Mux1:mux13 (BUFFER_SIZE = 2)
 Instance: Mux1:mux12 (BUFFER_SIZE = 2)
 Instance: Mux1:mux11 (BUFFER_SIZE = 2)
 Instance: Mux1:mux10 (BUFFER_SIZE = 2)
 Instance: Mux1:mux9 (BUFFER_SIZE = 2)
 Instance: Mux1:mux8 (BUFFER_SIZE = 2)
 Instance: Mux1:mux7 (BUFFER_SIZE = 2)
 Instance: Mux1:mux6 (BUFFER_SIZE = 2)
 Instance: Mux1:mux5 (BUFFER_SIZE = 3)
 Instance: Mux1:mux4 (BUFFER_SIZE = 3)
 Instance: Mux1:mux3 (BUFFER_SIZE = 3)
 Instance: Mux1:mux2 (BUFFER_SIZE = 3)
 Instance: Mux1:mux1 (BUFFER_SIZE = 3)
 Instance: Mux1:mux0 (BUFFER_SIZE = 2)
 Instance: FPSReg5:reg3 (BUFFER_SIZE = 2)
 Instance: FPSReg5:reg2 (BUFFER_SIZE = 2)
 Instance: FPSReg5:reg1 (BUFFER_SIZE = 2)
 Instance: or5 (BUFFER_SIZE = 6)
 Writing specification cell stdor2#o1, compiler gtgate

Instance: or4 (BUFFER_SIZE = 6)
 Instance: or3 (BUFFER_SIZE = 6)
 Instance: or2 (BUFFER_SIZE = 6)
 Instance: or1 (BUFFER_SIZE = 6)
 Instance: Counter:inc0:incdec1 (BUFFER_SIZE_COUT = 6)
 Writing specification cell gtadd#o2, compiler gtadd
 Creating new modules for composite cell: complex_mplier
 Instance: or7 (BUFFER_SIZE = 3)
 Writing specification cell stdor3#o1, compiler gtgate
 Instance: or6 (BUFFER_SIZE = 3)
 Instance: or9 (BUFFER_SIZE = 3)
 Instance: or10 (BUFFER_SIZE = 3)
 Instance: or8 (BUFFER_SIZE = 3)
 Instance: or2 (BUFFER_SIZE = 2)
 Writing specification cell stdor4#o1, compiler gtgate
 Instance: or1 (BUFFER_SIZE = 2)
 Instance: or4 (BUFFER_SIZE = 2)
 Instance: or5 (BUFFER_SIZE = 2)
 Instance: or3 (BUFFER_SIZE = 2)
 Creating new modules for composite cell: fpadder
 Instance: and3 (BUFFER_SIZE = 2)
 Writing specification cell stdand3#o1, compiler gtgate
 Instance: shifter:bs_mux491 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux481 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux471 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux461 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux451 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux441 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux431 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux421 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux401 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux391 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux381 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux371 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux351 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux341 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux321 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux311 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux231 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux490 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux480 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux470 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux460 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux450 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux440 (BUFFER_SIZE = 2)

Instance: shifter:bs_mux430 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux420 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux410 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux400 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux390 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux380 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux370 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux360 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux350 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux340 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux330 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux320 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux310 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux300 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux290 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux280 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux270 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux260 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux250 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux240 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux230 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux220 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux210 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux200 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux190 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux180 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux170 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux160 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux150 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux140 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux130 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux120 (BUFFER_SIZE = 2)
 Instance: reg5:reg36 (BUFFER_SIZE = 5)
 Writing specification cell gtdff#o4, compiler gtdff
 Instance: reg5:reg35 (BUFFER_SIZE = 4)
 Writing specification cell gtdff#o5, compiler gtdff
 Instance: reg5:reg34 (BUFFER_SIZE = 5)
 Instance: reg5:reg33 (BUFFER_SIZE = 5)
 Instance: reg5:reg32 (BUFFER_SIZE = 5)
 Instance: reg5:reg31 (BUFFER_SIZE = 2)
 Instance: reg5:reg30 (BUFFER_SIZE = 4)
 Instance: reg5:reg29 (BUFFER_SIZE = 4)
 Instance: reg5:reg28 (BUFFER_SIZE = 4)
 Instance: reg5:reg27 (BUFFER_SIZE = 4)
 Instance: reg5:reg26 (BUFFER_SIZE = 4)

Instance: reg5:reg25	(BUFFER_SIZE = 4)
Instance: reg5:reg24	(BUFFER_SIZE = 4)
Instance: reg5:reg23	(BUFFER_SIZE = 4)
Instance: reg5:reg22	(BUFFER_SIZE = 4)
Instance: reg5:reg21	(BUFFER_SIZE = 4)
Instance: reg5:reg20	(BUFFER_SIZE = 4)
Instance: reg5:reg19	(BUFFER_SIZE = 4)
Instance: reg5:reg18	(BUFFER_SIZE = 4)
Instance: reg5:reg17	(BUFFER_SIZE = 4)
Instance: reg5:reg16	(BUFFER_SIZE = 4)
Instance: reg5:reg15	(BUFFER_SIZE = 4)
Instance: reg5:reg14	(BUFFER_SIZE = 4)
Instance: reg5:reg13	(BUFFER_SIZE = 3)
Writing specification cell gtdff#o6, compiler gtdff	
Instance: reg5:reg12	(BUFFER_SIZE = 3)
Instance: reg5:reg11	(BUFFER_SIZE = 3)
Instance: reg5:reg10	(BUFFER_SIZE = 4)
Instance: reg5:reg9	(BUFFER_SIZE = 4)
Instance: reg5:reg8	(BUFFER_SIZE = 4)
Instance: reg5:reg7	(BUFFER_SIZE = 4)
Instance: reg5:reg6	(BUFFER_SIZE = 4)
Instance: reg5:reg5	(BUFFER_SIZE = 5)
Instance: reg5:reg4	(BUFFER_SIZE = 4)
Instance: reg5:reg3	(BUFFER_SIZE = 5)
Instance: reg5:reg2	(BUFFER_SIZE = 5)
Instance: reg5:reg1	(BUFFER_SIZE = 4)
Instance: reg5:reg0	(BUFFER_SIZE = 4)
Instance: reg4:reg42	(BUFFER_SIZE = 2)
Instance: reg4:reg40	(BUFFER_SIZE = 2)
Instance: reg4:reg26	(BUFFER_SIZE = 2)
Instance: reg4:reg9	(BUFFER_SIZE = 2)
Instance: reg4:reg0	(BUFFER_SIZE = 2)
Instance: reg3:reg87	(BUFFER_SIZE = 2)
Instance: reg3:reg81	(BUFFER_SIZE = 2)
Instance: reg3:reg80	(BUFFER_SIZE = 2)
Instance: reg3:reg79	(BUFFER_SIZE = 2)
Instance: reg3:reg77	(BUFFER_SIZE = 2)
Instance: reg3:reg76	(BUFFER_SIZE = 2)
Instance: reg3:reg75	(BUFFER_SIZE = 2)
Instance: reg3:reg74	(BUFFER_SIZE = 2)
Instance: reg3:reg73	(BUFFER_SIZE = 2)
Instance: reg3:reg72	(BUFFER_SIZE = 2)
Instance: reg3:reg70	(BUFFER_SIZE = 2)
Instance: reg3:reg67	(BUFFER_SIZE = 2)
Instance: reg3:reg66	(BUFFER_SIZE = 2)

Instance: reg3:reg65	(BUFFER_SIZE = 2)
Instance: reg3:reg64	(BUFFER_SIZE = 2)
Instance: reg3:reg63	(BUFFER_SIZE = 2)
Instance: reg3:reg62	(BUFFER_SIZE = 2)
Instance: reg3:reg61	(BUFFER_SIZE = 2)
Instance: reg3:reg60	(BUFFER_SIZE = 2)
Instance: reg3:reg59	(BUFFER_SIZE = 2)
Instance: reg3:reg58	(BUFFER_SIZE = 2)
Instance: reg3:reg57	(BUFFER_SIZE = 2)
Instance: reg3:reg56	(BUFFER_SIZE = 2)
Instance: reg3:reg54	(BUFFER_SIZE = 2)
Instance: reg3:reg53	(BUFFER_SIZE = 2)
Instance: reg3:reg51	(BUFFER_SIZE = 2)
Instance: reg3:reg50	(BUFFER_SIZE = 2)
Instance: reg3:reg49	(BUFFER_SIZE = 2)
Instance: reg3:reg48	(BUFFER_SIZE = 2)
Instance: reg3:reg47	(BUFFER_SIZE = 2)
Instance: reg3:reg45	(BUFFER_SIZE = 2)
Instance: reg3:reg42	(BUFFER_SIZE = 2)
Instance: reg3:reg41	(BUFFER_SIZE = 2)
Instance: reg3:reg40	(BUFFER_SIZE = 2)
Instance: reg3:reg39	(BUFFER_SIZE = 2)
Instance: reg3:reg38	(BUFFER_SIZE = 2)
Instance: reg3:reg35	(BUFFER_SIZE = 2)
Instance: reg3:reg34	(BUFFER_SIZE = 2)
Instance: reg3:reg33	(BUFFER_SIZE = 2)
Instance: reg3:reg32	(BUFFER_SIZE = 2)
Instance: reg2i:reg70	(BUFFER_SIZE = 2)
Instance: reg2i:reg69	(BUFFER_SIZE = 2)
Instance: reg2i:reg50	(BUFFER_SIZE = 2)
Instance: reg2i:reg40	(BUFFER_SIZE = 2)
Instance: reg2i:reg39	(BUFFER_SIZE = 2)
Instance: reg2i:reg38	(BUFFER_SIZE = 2)
Instance: reg2i:reg37	(BUFFER_SIZE = 2)
Instance: reg2i:reg36	(BUFFER_SIZE = 2)
Instance: reg2i:reg35	(BUFFER_SIZE = 2)
Instance: reg2i:reg34	(BUFFER_SIZE = 2)
Instance: reg2i:reg33	(BUFFER_SIZE = 2)
Instance: reg2i:reg32	(BUFFER_SIZE = 2)
Instance: reg2i:reg31	(BUFFER_SIZE = 2)
Instance: reg2i:reg30	(BUFFER_SIZE = 2)
Instance: reg2i:reg29	(BUFFER_SIZE = 2)
Instance: reg2i:reg28	(BUFFER_SIZE = 2)
Instance: reg2i:reg27	(BUFFER_SIZE = 2)
Instance: reg2i:reg26	(BUFFER_SIZE = 2)

Instance: reg2i:reg25 (BUFFER_SIZE = 2)
 Instance: reg2i:reg24 (BUFFER_SIZE = 2)
 Instance: reg2i:reg23 (BUFFER_SIZE = 2)
 Instance: reg2i:reg22 (BUFFER_SIZE = 2)
 Instance: reg2i:reg21 (BUFFER_SIZE = 2)
 Instance: reg2i:reg20 (BUFFER_SIZE = 2)
 Instance: reg2i:reg19 (BUFFER_SIZE = 2)
 Instance: reg2i:reg18 (BUFFER_SIZE = 2)
 Instance: reg2i:reg17 (BUFFER_SIZE = 2)
 Instance: reg2i:reg16 (BUFFER_SIZE = 2)
 Instance: reg2i:reg5 (BUFFER_SIZE = 9)
 Writing specification cell gtdff#o7, compiler gtdff
 Instance: reg2i:reg4 (BUFFER_SIZE = 9)
 Instance: reg2i:reg3 (BUFFER_SIZE = 9)
 Instance: reg2i:reg2 (BUFFER_SIZE = 9)
 Instance: reg2i:reg1 (BUFFER_SIZE = 9)
 Instance: reg2i:reg0 (BUFFER_SIZE = 9)
 Instance: reg2:reg64 (BUFFER_SIZE = 2)
 Instance: reg2:reg62 (BUFFER_SIZE = 2)
 Instance: reg2:reg59 (BUFFER_SIZE = 2)
 Instance: reg2:reg57 (BUFFER_SIZE = 2)
 Instance: reg2:reg51 (BUFFER_SIZE = 2)
 Instance: reg2:reg46 (BUFFER_SIZE = 2)
 Instance: reg2:reg15 (BUFFER_SIZE = 2)
 Instance: reg2:reg14 (BUFFER_SIZE = 2)
 Instance: reg2:reg13 (BUFFER_SIZE = 2)
 Instance: reg2:reg12 (BUFFER_SIZE = 2)
 Instance: reg2:reg11 (BUFFER_SIZE = 2)
 Instance: reg2:reg10 (BUFFER_SIZE = 2)
 Instance: reg2:reg9 (BUFFER_SIZE = 2)
 Instance: reg2:reg8 (BUFFER_SIZE = 2)
 Instance: reg2:reg7 (BUFFER_SIZE = 2)
 Instance: reg2:reg6 (BUFFER_SIZE = 2)
 Instance: reg2:reg5 (BUFFER_SIZE = 2)
 Instance: reg2:reg4 (BUFFER_SIZE = 2)
 Instance: reg2:reg3 (BUFFER_SIZE = 2)
 Instance: reg2:reg2 (BUFFER_SIZE = 2)
 Instance: reg2:reg1 (BUFFER_SIZE = 2)
 Instance: reg2:reg0 (BUFFER_SIZE = 2)
 Instance: reg1:reg71 (BUFFER_SIZE = 2)
 Instance: reg1:reg70 (BUFFER_SIZE = 2)
 Instance: reg1:reg59 (BUFFER_SIZE = 2)
 Instance: reg1:reg58 (BUFFER_SIZE = 2)
 Instance: reg1:reg57 (BUFFER_SIZE = 2)
 Instance: reg1:reg56 (BUFFER_SIZE = 2)

Instance: reg1:reg55 (BUFFER_SIZE = 2)
 Instance: reg1:reg50 (BUFFER_SIZE = 2)
 Instance: reg1:reg48 (BUFFER_SIZE = 2)
 Instance: reg1:reg47 (BUFFER_SIZE = 2)
 Instance: reg1:reg42 (BUFFER_SIZE = 2)
 Instance: reg1:reg41 (BUFFER_SIZE = 2)
 Instance: reg1:reg39 (BUFFER_SIZE = 2)
 Instance: reg1:reg38 (BUFFER_SIZE = 2)
 Instance: reg1:reg37 (BUFFER_SIZE = 2)
 Instance: reg1:reg36 (BUFFER_SIZE = 2)
 Instance: reg1:reg35 (BUFFER_SIZE = 2)
 Instance: reg1:reg34 (BUFFER_SIZE = 2)
 Instance: reg1:reg32 (BUFFER_SIZE = 2)
 Instance: reg1:reg31 (BUFFER_SIZE = 2)
 Instance: reg1:reg30 (BUFFER_SIZE = 2)
 Instance: reg1:reg22 (BUFFER_SIZE = 2)
 Instance: reg1:reg20 (BUFFER_SIZE = 2)
 Instance: reg1:reg17 (BUFFER_SIZE = 2)
 Instance: reg1:reg15 (BUFFER_SIZE = 2)
 Instance: reg1:reg14 (BUFFER_SIZE = 2)
 Instance: reg1:reg13 (BUFFER_SIZE = 2)
 Instance: reg1:reg10 (BUFFER_SIZE = 2)
 Instance: reg1:reg6 (BUFFER_SIZE = 2)
 Instance: reg1:reg5 (BUFFER_SIZE = 2)
 Instance: reg1:reg4 (BUFFER_SIZE = 2)
 Instance: reg1:reg2 (BUFFER_SIZE = 2)
 Instance: reg1:reg1 (BUFFER_SIZE = 2)
 Instance: reg1:reg0 (BUFFER_SIZE = 2)
 Instance: or11 (BUFFER_SIZE = 2)
 Writing specification cell stdor2#o2, compiler gtgate
 Instance: overshift:p3_Izin_37__#bar (BUFFER_SIZE = 2)
 Instance: overshift:p3_Izin_27__#bar (BUFFER_SIZE = 2)
 Instance: M_Comparator:invbga (BUFFER_SIZE = 4)
 Writing specification cell gtgate#o8, compiler gtgate
 Instance: Incrementer1:tc_inv (BUFFER_SIZE = 4)
 Instance: Incrementer3:tc_inv (BUFFER_SIZE = 4)
 Instance: Incrementer2:tc_inv (BUFFER_SIZE = 4)
 Instance: and1 (BUFFER_SIZE = 6)
 Instance: man_out_inv:p4_Izsignii52 (BUFFER_SIZE = 3)
 Writing specification cell stdinv#o2, compiler gtgate
 Instance: man_out_inv:p4_Izsignii42 (BUFFER_SIZE = 3)
 Instance: man_out_inv:p4_Izsignii33 (BUFFER_SIZE = 2)
 Instance: Incrementer1:incdec25 (BUFFER_SIZE_COUT = 4)
 Writing specification cell gtadd#o3, compiler gtadd
 Instance: man_invB:p7_Izsignii50 (BUFFER_SIZE = 3)

Instance: man_invB:p7_Izsignii40 (BUFFER_SIZE = 3)
 Instance: man_invB:p7_Izsignii31 (BUFFER_SIZE = 2)
 Instance: man_invA:p7_Izsignii50 (BUFFER_SIZE = 3)
 Instance: man_invA:p7_Izsignii40 (BUFFER_SIZE = 3)
 Instance: man_invA:p7_Izsignii31 (BUFFER_SIZE = 2)
 Instance: man_out_inv:p4_Izout_2__ (BUFFER_SIZE = 2)
 Writing specification cell stdxnor2#o1, compiler gtgate
 Instance: man_out_inv:p4_Izout_1__ (BUFFER_SIZE = 2)
 Instance: man_out_inv:p4_Izout_0__ (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux22 (BUFFER_SIZE = 2)
 Writing specification cell gtmux#o7, compiler gtmux
 Instance: Rounder:MUX:mux5 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux4 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux1 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux0 (BUFFER_SIZE = 2)
 Instance: overshift:p3_Izout_49__ (BUFFER_SIZE = 2)
 Writing specification cell stdnor2#o1, compiler gtgate
 Instance: overshift:p3_Izout_31__ (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux472 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux412 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux352 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux272 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux252 (BUFFER_SIZE = 2)
 Instance: Exp_mux2:mux7 (BUFFER_SIZE = 2)
 Instance: Exp_mux2:mux4 (BUFFER_SIZE = 2)
 Instance: Exp_mux2:mux3 (BUFFER_SIZE = 2)
 Instance: Exp_mux2:mux1 (BUFFER_SIZE = 2)
 Instance: Exp_mux2:mux0 (BUFFER_SIZE = 2)
 Instance: cleanup:p9_Izinf_out (BUFFER_SIZE = 2)
 Writing specification cell stdand4#o1, compiler gtgate
 Instance: or15 (BUFFER_SIZE = 6)
 Writing specification cell stdor3#o2, compiler gtgate
 Instance: or8 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux24 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux22 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux20 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux19 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux17 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux15 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux13 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux12 (BUFFER_SIZE = 2)
 Instance: Man_mux1:mux8 (BUFFER_SIZE = 2)
 Instance: man_out_inv:p4_Izout_25__ (BUFFER_SIZE = 3)
 Writing specification cell stdxnor2#o2, compiler gtgate
 Instance: cleanup:p9_Ic#p9#W#3#bar (BUFFER_SIZE = 2)

Instance: cleanup:p9_Ic#p9#W#2#bar (BUFFER_SIZE = 2)
 Instance: cleanup:p9_Ic#p9#W#1 (BUFFER_SIZE = 2)
 Instance: cleanup:p9_Ic#p9#W#0#bar (BUFFER_SIZE = 2)
 Instance: Value_Test_B:p5_Izstate_0__ (BUFFER_SIZE = 2)
 Instance: Value_Test_B:p6_Ic#p6#W#1 (BUFFER_SIZE = 2)
 Instance: Value_Test_B:p6_Ic#p6#T#32 (BUFFER_SIZE = 2)
 Writing specification cell stdnor4#o1, compiler gtgate
 Instance: Value_Test_A:p5_Izstate_0__ (BUFFER_SIZE = 2)
 Instance: Value_Test_A:p6_Ic#p6#W#1 (BUFFER_SIZE = 2)
 Instance: Value_Test_A:p6_Ic#p6#T#32 (BUFFER_SIZE = 2)
 Instance: Rounder:Incrementer:incdec24 (BUFFER_SIZE_SUM = 5)
 Writing specification cell gtadd#o4, compiler gtadd
 Instance: Rounder:p1_Izaddlsb (BUFFER_SIZE = 2)
 Writing specification cell stdnor3#o1, compiler gtgate
 Instance: M_Comparator:ygate5 (BUFFER_SIZE = 2)
 Writing specification cell gtgate#o9, compiler gtgate
 Instance: M_Comparator:xgate5 (BUFFER_SIZE = 2)
 Instance: Incrementer2:incdec0 (BUFFER_SIZE_COUT = 2)
 Writing specification cell gtadd#o5, compiler gtadd
 Instance: s0_IzExp_mux1_S0 (BUFFER_SIZE = 7)
 Writing specification cell stdinv#o3, compiler gtgate
 Instance: Bitshift1:p8_Izin_25__ii98 (BUFFER_SIZE = 3)
 Instance: Bitshift1:p8_Izin_25__ii89 (BUFFER_SIZE = 2)
 Instance: Bitshift1:p8_Izin_25__ii80 (BUFFER_SIZE = 2)
 Instance: M_Comparator:invagb (BUFFER_SIZE = 2)
 Writing specification cell gtgate#o10, compiler gtgate
 Creating new modules for composite cell: fadder_group
 Instance: Adder1:fadder#7 (BUFFER_SIZE = 4)
 Writing specification cell dpfadder_bit#o26, compiler dpfadder_bit
 Instance: exp_compl:compl7 (BUFFER_SIZE = 4)
 Writing specification cell dpcompl#o2, compiler dpcompl
 Creating new modules for composite cell: fpmplier
 Instance: and2 (BUFFER_SIZE = 2)
 Writing specification cell stdand2#o2, compiler gtgate
 Instance: shiftsel:p11_Izsft_dirii28 (BUFFER_SIZE = 3)
 Instance: shiftsel:p11_Ic#p11#W#7 (BUFFER_SIZE = 2)
 Instance: shiftsel:p11_Ic#p11#W#3 (BUFFER_SIZE = 2)
 Instance: shiftsel:p11_Ic#p11#W#17#bar (BUFFER_SIZE = 2)
 Instance: shiftsel:p11_Ic#p11#W#16#bar (BUFFER_SIZE = 3)
 Instance: shiftsel:p11_Ic#p11#W#0 (BUFFER_SIZE = 3)
 Writing specification cell stdnor3#o2, compiler gtgate
 Instance: shifter:bs_mux151 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux141 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux131 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux121 (BUFFER_SIZE = 2)

Instance: shifter:bs_mux111	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux101	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux91	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux81	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux71	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux61	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux51	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux41	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux31	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux21	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux11	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux360	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux350	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux340	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux330	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux320	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux310	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux300	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux290	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux280	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux270	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux260	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux250	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux240	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux230	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux220	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux210	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux200	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux190	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux180	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux170	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux160	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux150	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux140	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux130	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux120	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux110	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux100	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux90	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux80	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux70	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux60	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux50	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux40	(BUFFER_SIZE = 2)
Instance: shifter:bs_mux30	(BUFFER_SIZE = 2)

Instance: shifter:bs_mux20 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux10 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux00 (BUFFER_SIZE = 2)
 Instance: encoder:p10_Izout_4__ (BUFFER_SIZE = 2)
 Instance: encoder:p10_Izout_3__ (BUFFER_SIZE = 2)
 Instance: encoder:p10_Izout_2__ (BUFFER_SIZE = 2)
 Instance: encoder:p10_Izout_1__ (BUFFER_SIZE = 2)
 Instance: encoder:p10_Izout_0__ (BUFFER_SIZE = 2)
 Writing specification cell stdao211#o1, compiler gtgate
 Instance: reg5:reg36 (BUFFER_SIZE = 2)
 Instance: reg5:reg35 (BUFFER_SIZE = 2)
 Instance: reg5:reg34 (BUFFER_SIZE = 4)
 Instance: reg5:reg33 (BUFFER_SIZE = 2)
 Instance: reg5:reg32 (BUFFER_SIZE = 3)
 Instance: reg5:reg31 (BUFFER_SIZE = 4)
 Instance: reg5:reg30 (BUFFER_SIZE = 4)
 Instance: reg5:reg29 (BUFFER_SIZE = 4)
 Instance: reg5:reg28 (BUFFER_SIZE = 4)
 Instance: reg5:reg27 (BUFFER_SIZE = 4)
 Instance: reg5:reg26 (BUFFER_SIZE = 4)
 Instance: reg5:reg25 (BUFFER_SIZE = 4)
 Instance: reg5:reg24 (BUFFER_SIZE = 4)
 Instance: reg5:reg23 (BUFFER_SIZE = 5)
 Instance: reg5:reg22 (BUFFER_SIZE = 4)
 Instance: reg5:reg21 (BUFFER_SIZE = 4)
 Instance: reg5:reg20 (BUFFER_SIZE = 4)
 Instance: reg5:reg19 (BUFFER_SIZE = 4)
 Instance: reg5:reg18 (BUFFER_SIZE = 3)
 Instance: reg5:reg17 (BUFFER_SIZE = 4)
 Instance: reg5:reg16 (BUFFER_SIZE = 4)
 Instance: reg5:reg15 (BUFFER_SIZE = 4)
 Instance: reg5:reg14 (BUFFER_SIZE = 4)
 Instance: reg5:reg13 (BUFFER_SIZE = 4)
 Instance: reg5:reg12 (BUFFER_SIZE = 4)
 Instance: reg5:reg11 (BUFFER_SIZE = 4)
 Instance: reg5:reg10 (BUFFER_SIZE = 4)
 Instance: reg5:reg9 (BUFFER_SIZE = 4)
 Instance: reg5:reg8 (BUFFER_SIZE = 4)
 Instance: reg5:reg7 (BUFFER_SIZE = 4)
 Instance: reg5:reg6 (BUFFER_SIZE = 4)
 Instance: reg5:reg5 (BUFFER_SIZE = 4)
 Instance: reg5:reg4 (BUFFER_SIZE = 4)
 Instance: reg5:reg3 (BUFFER_SIZE = 4)
 Instance: reg5:reg2 (BUFFER_SIZE = 4)
 Instance: reg5:reg1 (BUFFER_SIZE = 4)

Instance: reg5:reg0	(BUFFER_SIZE = 4)
Instance: reg4i:reg62	(BUFFER_SIZE = 2)
Instance: reg4i:reg60	(BUFFER_SIZE = 2)
Instance: reg4i:reg59	(BUFFER_SIZE = 2)
Instance: reg4i:reg58	(BUFFER_SIZE = 2)
Instance: reg4i:reg57	(BUFFER_SIZE = 2)
Instance: reg4i:reg56	(BUFFER_SIZE = 2)
Instance: reg4i:reg55	(BUFFER_SIZE = 3)
Instance: reg4i:reg54	(BUFFER_SIZE = 2)
Instance: reg4i:reg53	(BUFFER_SIZE = 2)
Instance: reg4i:reg52	(BUFFER_SIZE = 2)
Instance: reg4i:reg51	(BUFFER_SIZE = 2)
Instance: reg4i:reg50	(BUFFER_SIZE = 3)
Instance: reg4i:reg49	(BUFFER_SIZE = 2)
Instance: reg4i:reg48	(BUFFER_SIZE = 2)
Instance: reg4i:reg47	(BUFFER_SIZE = 3)
Instance: reg4i:reg46	(BUFFER_SIZE = 3)
Instance: reg4i:reg45	(BUFFER_SIZE = 3)
Instance: reg4i:reg44	(BUFFER_SIZE = 3)
Instance: reg4i:reg43	(BUFFER_SIZE = 3)
Instance: reg4i:reg42	(BUFFER_SIZE = 2)
Instance: reg4i:reg41	(BUFFER_SIZE = 2)
Instance: reg4i:reg40	(BUFFER_SIZE = 2)
Instance: reg4i:reg39	(BUFFER_SIZE = 2)
Instance: reg4i:reg38	(BUFFER_SIZE = 2)
Instance: reg4i:reg37	(BUFFER_SIZE = 2)
Instance: reg4i:reg36	(BUFFER_SIZE = 3)
Instance: reg4i:reg35	(BUFFER_SIZE = 2)
Instance: reg4i:reg34	(BUFFER_SIZE = 2)
Instance: reg4i:reg33	(BUFFER_SIZE = 2)
Instance: reg4i:reg32	(BUFFER_SIZE = 3)
Instance: reg4i:reg31	(BUFFER_SIZE = 2)
Instance: reg4i:reg30	(BUFFER_SIZE = 2)
Instance: reg4i:reg29	(BUFFER_SIZE = 2)
Instance: reg4i:reg28	(BUFFER_SIZE = 2)
Instance: reg4i:reg27	(BUFFER_SIZE = 2)
Instance: reg4i:reg26	(BUFFER_SIZE = 2)
Instance: reg4i:reg25	(BUFFER_SIZE = 2)
Instance: reg4i:reg24	(BUFFER_SIZE = 2)
Instance: reg4i:reg23	(BUFFER_SIZE = 2)
Instance: reg4i:reg22	(BUFFER_SIZE = 2)
Instance: reg4i:reg21	(BUFFER_SIZE = 2)
Instance: reg4i:reg20	(BUFFER_SIZE = 2)
Instance: reg4i:reg19	(BUFFER_SIZE = 2)
Instance: reg4i:reg18	(BUFFER_SIZE = 2)

Instance: reg4i:reg17	(BUFFER_SIZE = 2)
Instance: reg4i:reg16	(BUFFER_SIZE = 2)
Instance: reg4i:reg15	(BUFFER_SIZE = 2)
Instance: reg4i:reg14	(BUFFER_SIZE = 9)
Instance: reg4i:reg13	(BUFFER_SIZE = 9)
Instance: reg4i:reg12	(BUFFER_SIZE = 9)
Instance: reg4i:reg11	(BUFFER_SIZE = 9)
Instance: reg4i:reg10	(BUFFER_SIZE = 9)
Instance: reg4i:reg8	(BUFFER_SIZE = 2)
Instance: reg4i:reg6	(BUFFER_SIZE = 2)
Instance: reg4i:reg1	(BUFFER_SIZE = 2)
Instance: reg4:reg62	(BUFFER_SIZE = 2)
Instance: reg4:reg59	(BUFFER_SIZE = 2)
Instance: reg4:reg58	(BUFFER_SIZE = 2)
Instance: reg4:reg50	(BUFFER_SIZE = 2)
Instance: reg4:reg49	(BUFFER_SIZE = 2)
Instance: reg4:reg48	(BUFFER_SIZE = 2)
Instance: reg4:reg46	(BUFFER_SIZE = 2)
Instance: reg4:reg45	(BUFFER_SIZE = 2)
Instance: reg4:reg44	(BUFFER_SIZE = 2)
Instance: reg4:reg43	(BUFFER_SIZE = 2)
Instance: reg4:reg41	(BUFFER_SIZE = 2)
Instance: reg4:reg39	(BUFFER_SIZE = 2)
Instance: reg4:reg9	(BUFFER_SIZE = 2)
Instance: reg4:reg3	(BUFFER_SIZE = 2)
Instance: reg4:reg0	(BUFFER_SIZE = 3)
Instance: reg3:reg72	(BUFFER_SIZE = 2)
Instance: reg3:reg71	(BUFFER_SIZE = 2)
Instance: reg3:reg70	(BUFFER_SIZE = 2)
Instance: reg3:reg69	(BUFFER_SIZE = 2)
Instance: reg3:reg68	(BUFFER_SIZE = 2)
Instance: reg3:reg58	(BUFFER_SIZE = 2)
Instance: reg3:reg57	(BUFFER_SIZE = 2)
Instance: reg3:reg49	(BUFFER_SIZE = 2)
Instance: reg3:reg41	(BUFFER_SIZE = 2)
Instance: reg3:reg40	(BUFFER_SIZE = 2)
Instance: reg3:reg38	(BUFFER_SIZE = 2)
Instance: reg3:reg37	(BUFFER_SIZE = 2)
Instance: reg3:reg14	(BUFFER_SIZE = 3)
Instance: reg3:reg13	(BUFFER_SIZE = 3)
Instance: reg3:reg12	(BUFFER_SIZE = 3)
Instance: reg3:reg11	(BUFFER_SIZE = 2)
Instance: reg3:reg10	(BUFFER_SIZE = 2)
Instance: reg3:reg9	(BUFFER_SIZE = 3)
Instance: reg3:reg8	(BUFFER_SIZE = 3)

Instance: reg3:reg7 (BUFFER_SIZE = 2)
 Instance: reg3:reg6 (BUFFER_SIZE = 3)
 Instance: reg3:reg5 (BUFFER_SIZE = 3)
 Instance: reg3:reg4 (BUFFER_SIZE = 3)
 Instance: reg3:reg3 (BUFFER_SIZE = 3)
 Instance: reg3:reg2 (BUFFER_SIZE = 3)
 Instance: reg2:reg8 (BUFFER_SIZE = 2)
 Instance: reg2:reg5 (BUFFER_SIZE = 2)
 Instance: reg2:reg4 (BUFFER_SIZE = 2)
 Instance: reg2:reg3 (BUFFER_SIZE = 2)
 Instance: reg2:reg2 (BUFFER_SIZE = 2)
 Instance: reg1:reg65 (BUFFER_SIZE = 6)
 Writing specification cell gtdff#o8, compiler gtdff
 Instance: reg1:reg64 (BUFFER_SIZE = 5)
 Instance: reg1:reg63 (BUFFER_SIZE = 5)
 Instance: reg1:reg62 (BUFFER_SIZE = 5)
 Instance: reg1:reg61 (BUFFER_SIZE = 5)
 Instance: reg1:reg60 (BUFFER_SIZE = 5)
 Instance: reg1:reg59 (BUFFER_SIZE = 6)
 Instance: reg1:reg58 (BUFFER_SIZE = 5)
 Instance: reg1:reg57 (BUFFER_SIZE = 6)
 Instance: reg1:reg56 (BUFFER_SIZE = 6)
 Instance: reg1:reg55 (BUFFER_SIZE = 5)
 Instance: reg1:reg54 (BUFFER_SIZE = 6)
 Instance: reg1:reg53 (BUFFER_SIZE = 6)
 Instance: reg1:reg52 (BUFFER_SIZE = 6)
 Instance: reg1:reg51 (BUFFER_SIZE = 6)
 Instance: reg1:reg50 (BUFFER_SIZE = 6)
 Instance: reg1:reg49 (BUFFER_SIZE = 6)
 Instance: reg1:reg48 (BUFFER_SIZE = 6)
 Instance: reg1:reg47 (BUFFER_SIZE = 6)
 Instance: reg1:reg46 (BUFFER_SIZE = 6)
 Instance: reg1:reg45 (BUFFER_SIZE = 6)
 Instance: reg1:reg44 (BUFFER_SIZE = 6)
 Instance: reg1:reg43 (BUFFER_SIZE = 6)
 Instance: reg1:reg42 (BUFFER_SIZE = 6)
 Instance: reg1:reg41 (BUFFER_SIZE = 6)
 Instance: reg1:reg40 (BUFFER_SIZE = 2)
 Instance: reg1:reg39 (BUFFER_SIZE = 3)
 Instance: reg1:reg38 (BUFFER_SIZE = 2)
 Instance: reg1:reg37 (BUFFER_SIZE = 2)
 Instance: reg1:reg36 (BUFFER_SIZE = 2)
 Instance: reg1:reg35 (BUFFER_SIZE = 3)
 Instance: reg1:reg34 (BUFFER_SIZE = 2)
 Instance: reg1:reg33 (BUFFER_SIZE = 2)

Instance: reg1:reg32 (BUFFER_SIZE = 2)
 Instance: reg1:reg31 (BUFFER_SIZE = 2)
 Instance: reg1:reg30 (BUFFER_SIZE = 2)
 Instance: reg1:reg29 (BUFFER_SIZE = 2)
 Instance: reg1:reg28 (BUFFER_SIZE = 2)
 Instance: reg1:reg27 (BUFFER_SIZE = 2)
 Instance: reg1:reg26 (BUFFER_SIZE = 2)
 Instance: reg1:reg25 (BUFFER_SIZE = 2)
 Instance: reg1:reg24 (BUFFER_SIZE = 2)
 Instance: reg1:reg23 (BUFFER_SIZE = 2)
 Instance: reg1:reg22 (BUFFER_SIZE = 2)
 Instance: reg1:reg21 (BUFFER_SIZE = 2)
 Instance: reg1:reg20 (BUFFER_SIZE = 2)
 Instance: reg1:reg19 (BUFFER_SIZE = 2)
 Instance: reg1:reg18 (BUFFER_SIZE = 2)
 Instance: reg1:reg17 (BUFFER_SIZE = 2)
 Instance: reg1:reg16 (BUFFER_SIZE = 2)
 Instance: reg1:reg15 (BUFFER_SIZE = 2)
 Instance: reg1:reg14 (BUFFER_SIZE = 2)
 Instance: reg1:reg11 (BUFFER_SIZE = 2)
 Instance: reg1:reg10 (BUFFER_SIZE = 2)
 Instance: reg1:reg9 (BUFFER_SIZE = 2)
 Instance: reg1:reg8 (BUFFER_SIZE = 2)
 Instance: reg1:reg7 (BUFFER_SIZE = 2)
 Instance: reg1:reg6 (BUFFER_SIZE = 2)
 Instance: reg1:reg5 (BUFFER_SIZE = 2)
 Instance: reg1:reg3 (BUFFER_SIZE = 2)
 Instance: reg1:reg2 (BUFFER_SIZE = 2)
 Instance: reg1:reg1 (BUFFER_SIZE = 2)
 Instance: reg1:reg0 (BUFFER_SIZE = 2)
 Instance: or2 (BUFFER_SIZE = 2)
 Instance: Adder4:bit4col0 (BUFFER_SIZE_COUT = 4)
 (BUFFER_SIZE_SUM = 2)
 Writing specification cell gtadd#o6, compiler gtadd
 Instance: Incrementer:tc_inv (BUFFER_SIZE = 4)
 Instance: Incrementer2:tc_inv (BUFFER_SIZE = 4)
 Instance: mansel:p12_Izsft_dirii189#3# (BUFFER_SIZE = 2)
 Instance: mansel:p12_Izsft_dirii189#2# (BUFFER_SIZE = 3)
 Instance: mansel:p12_Izsft_dirii189#1# (BUFFER_SIZE = 3)
 Instance: mansel:p12_Izsft_dirii189#0# (BUFFER_SIZE = 3)
 Instance: mansel:p12_Izsft_dirii189 (BUFFER_SIZE = 3)
 Instance: mansel:p12_Izmanin_9__#bar (BUFFER_SIZE = 2)
 Instance: mansel:p12_Izmanin_32__#bar (BUFFER_SIZE = 2)
 Instance: mansel:p12_Izmanin_28__#bar (BUFFER_SIZE = 2)
 Instance: mansel:p12_Izmanin_25__#bar (BUFFER_SIZE = 2)

Instance: Rounder:MUX:mux22 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux20 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux19 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux18 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux13 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux9 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux5 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux4 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux3 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux2 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux1 (BUFFER_SIZE = 2)
 Instance: Rounder:MUX:mux0 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux312 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux302 (BUFFER_SIZE = 2)
 Instance: shifter:bs_mux292 (BUFFER_SIZE = 2)
 Instance: mansel:p12_Izmanout_31__ (BUFFER_SIZE = 2)
 Writing specification cell stdoi22#o1, compiler gtgate
 Instance: mansel:p12_Izmanout_23__ (BUFFER_SIZE = 2)
 Instance: reg4i:reg68 (BUFFER_SIZE = 9)
 Instance: Comparator:invbga (BUFFER_SIZE = 2)
 Instance: shiftsel:p11_Izexp_fin_5__ (BUFFER_SIZE = 2)
 Writing specification cell stdoi222#o1, compiler gtgate
 Instance: shiftsel:p11_Izexp_fin_3__ (BUFFER_SIZE = 2)
 Instance: shiftsel:p11_Izexp_fin_2__ (BUFFER_SIZE = 2)
 Instance: shiftsel:p11_Izexp_fin_0__ (BUFFER_SIZE = 2)
 Instance: Value_Test_B:p6_Izexp_6__ (BUFFER_SIZE = 2)
 Writing specification cell stdao21#o1, compiler gtgate
 Instance: Value_Test_B:p6_Izexp_4__ (BUFFER_SIZE = 2)
 Instance: Value_Test_B:p6_Izexp_1__ (BUFFER_SIZE = 2)
 Instance: encoder:p10_Ic#p10#W#58#bar (BUFFER_SIZE = 2)
 Instance: encoder:p10_Ic#p10#W#31 (BUFFER_SIZE = 2)
 Writing specification cell stdaoi21#o1, compiler gtgate
 Instance: encoder:p10_Ic#p10#W#19 (BUFFER_SIZE = 2)
 Instance: encoder:p10_Ic#p10#W#0 (BUFFER_SIZE = 2)
 Instance: encoder:p10_Ic#p10#T#83 (BUFFER_SIZE = 2)
 Instance: encoder:p10_Ic#p10#T#66i (BUFFER_SIZE = 2)
 Writing specification cell stdnand2#o1, compiler gtgate
 Instance: encoder:p10_Ic#p10#T#110 (BUFFER_SIZE = 2)
 Instance: cleanup:p9_Ic#p9#W#3#bar (BUFFER_SIZE = 2)
 Instance: cleanup:p9_Ic#p9#W#2#bar (BUFFER_SIZE = 2)
 Instance: cleanup:p9_Ic#p9#W#1 (BUFFER_SIZE = 2)
 Instance: cleanup:p9_Ic#p9#W#0#bar (BUFFER_SIZE = 2)
 Instance: Value_Test_B:p5_Izstate_0__ (BUFFER_SIZE = 3)
 Writing specification cell stdor4#o2, compiler gtgate
 Instance: Value_Test_B:p6_Ic#p6#W#1 (BUFFER_SIZE = 3)

Writing specification cell stdnor2#o2, compiler gtgate
 Instance: Value_Test_A:p5_Izstate_0__ (BUFFER_SIZE = 3)
 Instance: Value_Test_A:p6_Ic#p6#W#1 (BUFFER_SIZE = 3)
 Instance: Rounder:p1_Ic#p1#W#1i (BUFFER_SIZE = 2)
 Writing specification cell stdmux2#o1, compiler gtmux
 Instance: Rounder:Incrementer:incdec24 (BUFFER_SIZE_SUM = 6)
 Writing specification cell gtadd#o7, compiler gtadd
 Instance: Rounder:Incrementer:incdec23 (BUFFER_SIZE_SUM = 2)
 Writing specification cell gtadd#o8, compiler gtadd
 Instance: Adder4:bit3col0 (BUFFER_SIZE_SUM = 2)
 Writing specification cell gtadd#o9, compiler gtadd
 Instance: Adder4:bit2col0 (BUFFER_SIZE_SUM = 2)
 Instance: Adder4:bit1col0 (BUFFER_SIZE_SUM = 2)
 Instance: Adder4:bit0col0 (BUFFER_SIZE_SUM = 2)
 Creating new modules for composite cell: hsmult2pipcd_c_0#o0
 Instance: dpadder#o0_965 (BUFFER_SIZE = 2)
 Writing specification cell dpadder#o1, compiler dpadder
 Instance: dpadder#o0_957 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_955 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_954 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_946 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_945 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_944 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_943 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_935 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_839 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_934 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_933 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_819 (BUFFER_SIZE = 2)
 Writing specification cell dpdff#o1, compiler dpdff
 Instance: dpadder#o0_925 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_811 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_804 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_924 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_797 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_795 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_794 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_793 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_923 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_774 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_773 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_770 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_764 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_756 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_915 (BUFFER_SIZE = 2)

Instance: dpadder#o0_1102	(BUFFER_SIZE = 2)
Instance: dpadder#o0_914	(BUFFER_SIZE = 2)
Instance: dpadder#o0_907	(BUFFER_SIZE = 2)
Instance: dpadder#o0_906	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1108	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1053	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1022	(BUFFER_SIZE = 2)
Instance: dpadder#o0_999	(BUFFER_SIZE = 2)
Instance: dpadder#o0_975	(BUFFER_SIZE = 2)
Instance: dpadder#o0_963	(BUFFER_SIZE = 2)
Instance: dpadder#o0_952	(BUFFER_SIZE = 2)
Instance: dpadder#o0_941	(BUFFER_SIZE = 2)
Instance: dpadder#o0_931	(BUFFER_SIZE = 2)
Instance: dpadder#o0_921	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1113	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1107	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1087	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1079	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1061	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1042	(BUFFER_SIZE = 2)
Instance: dpadder#o0_897	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1021	(BUFFER_SIZE = 2)
Instance: dpadder#o0_962	(BUFFER_SIZE = 2)
Instance: dpadder#o0_912	(BUFFER_SIZE = 2)
Instance: dpadder#o0_903	(BUFFER_SIZE = 2)
Instance: dpadder#o0_895	(BUFFER_SIZE = 2)
Instance: dpadder#o0_887	(BUFFER_SIZE = 2)
Instance: dpadder#o0_837	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1131	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1123	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1118	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1069	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1020	(BUFFER_SIZE = 2)
Instance: dpadder#o0_883	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1009	(BUFFER_SIZE = 2)
Instance: dpadder#o0_985	(BUFFER_SIZE = 2)
Instance: dpadder#o0_973	(BUFFER_SIZE = 2)
Instance: dpadder#o0_961	(BUFFER_SIZE = 2)
Instance: dpadder#o0_882	(BUFFER_SIZE = 2)
Instance: dpadder#o0_833	(BUFFER_SIZE = 2)
Instance: dpadder#o0_877	(BUFFER_SIZE = 2)
Instance: dpadder#o0_879	(BUFFER_SIZE = 2)
Instance: dpadder#o0_872	(BUFFER_SIZE = 2)
Instance: dpadder#o0_867	(BUFFER_SIZE = 2)
Instance: dpadder#o0_861	(BUFFER_SIZE = 2)

Instance: dpadder#o0_856	(BUFFER_SIZE = 2)
Instance: dpadder#o0_851	(BUFFER_SIZE = 2)
Instance: dpadder#o0_834	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1137	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1134	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1130	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1117	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1085	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1077	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1049	(BUFFER_SIZE = 2)
Instance: dpadder#o0_870	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1040	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1028	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1017	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1007	(BUFFER_SIZE = 2)
Instance: dpadder#o0_1006	(BUFFER_SIZE = 2)
Instance: dpadder#o0_869	(BUFFER_SIZE = 2)
Instance: dpadder#o0_995	(BUFFER_SIZE = 2)
Instance: dpadder#o0_994	(BUFFER_SIZE = 2)
Instance: dpadder#o0_982	(BUFFER_SIZE = 2)
Instance: dpadder#o0_981	(BUFFER_SIZE = 2)
Instance: dpadder#o0_971	(BUFFER_SIZE = 2)
Instance: dpadder#o0_958	(BUFFER_SIZE = 2)
Instance: dpadder#o0_948	(BUFFER_SIZE = 2)
Instance: dpadder#o0_928	(BUFFER_SIZE = 2)
Instance: dpadder#o0_918	(BUFFER_SIZE = 2)
Instance: dpadder#o0_863	(BUFFER_SIZE = 2)
Instance: dpadder#o0_917	(BUFFER_SIZE = 2)
Instance: dpadder#o0_909	(BUFFER_SIZE = 2)
Instance: dpadder#o0_901	(BUFFER_SIZE = 2)
Instance: dpadder#o0_900	(BUFFER_SIZE = 2)
Instance: dpadder#o0_893	(BUFFER_SIZE = 2)
Instance: dpadder#o0_885	(BUFFER_SIZE = 2)
Instance: dpadder#o0_884	(BUFFER_SIZE = 2)
Instance: dpadder#o0_878	(BUFFER_SIZE = 2)
Instance: dpadder#o0_859	(BUFFER_SIZE = 2)
Instance: dpadder#o0_871	(BUFFER_SIZE = 2)
Instance: dpadder#o0_866	(BUFFER_SIZE = 2)
Instance: dpadder#o0_860	(BUFFER_SIZE = 2)
Instance: dpadder#o0_855	(BUFFER_SIZE = 2)
Instance: dpadder#o0_850	(BUFFER_SIZE = 2)
Instance: dpadder#o0_847	(BUFFER_SIZE = 2)
Instance: dpadder#o0_843	(BUFFER_SIZE = 2)
Instance: dpadder#o0_831	(BUFFER_SIZE = 2)
Instance: dpadder#o0_858	(BUFFER_SIZE = 2)

Instance: dpadder#o0_1141 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1125 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_829 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1115 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1109 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1103 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1096 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1089 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1083 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1082 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1074 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1073 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1066 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1065 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1064 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1056 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_849 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1055 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1045 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1035 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_846 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1024 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1015 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1014 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1013 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_1004 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_992 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_990 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_989 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_978 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_977 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_968 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_967 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_966 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_842 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_733 (BUFFER_SIZE = 2)
 Instance: dpdff#o0_732 (BUFFER_SIZE = 3)
 Writing specification cell dpdff#o2, compiler dpdff
 Instance: dpdff#o0_731 (BUFFER_SIZE = 3)
 Instance: dpdff#o0_730 (BUFFER_SIZE = 3)
 Instance: dpdff#o0_729 (BUFFER_SIZE = 4)
 Writing specification cell dpdff#o3, compiler dpdff
 Instance: dpdff#o0_728 (BUFFER_SIZE = 3)
 Instance: dpadder#o0_832 (BUFFER_SIZE = 2)
 Instance: dpadder#o0_828 (BUFFER_SIZE = 2)

Instance: dpdff#o0_641	(BUFFER_SIZE = 2)
Instance: dpdff#o0_640	(BUFFER_SIZE = 2)
Instance: dpdff#o0_639	(BUFFER_SIZE = 2)
Instance: dpdff#o0_636	(BUFFER_SIZE = 2)
Instance: dpdff#o0_635	(BUFFER_SIZE = 2)
Instance: dpdff#o0_634	(BUFFER_SIZE = 2)
Instance: dpdff#o0_630	(BUFFER_SIZE = 2)
Instance: dpdff#o0_629	(BUFFER_SIZE = 2)
Instance: dpdff#o0_628	(BUFFER_SIZE = 2)
Instance: dpdff#o0_627	(BUFFER_SIZE = 2)
Instance: dpdff#o0_624	(BUFFER_SIZE = 2)
Instance: dpdff#o0_623	(BUFFER_SIZE = 2)
Instance: dpdff#o0_618	(BUFFER_SIZE = 2)
Instance: dpdff#o0_616	(BUFFER_SIZE = 2)
Instance: dpdff#o0_614	(BUFFER_SIZE = 2)
Instance: dpdff#o0_613	(BUFFER_SIZE = 2)
Instance: dpdff#o0_612	(BUFFER_SIZE = 2)
Instance: dpdff#o0_611	(BUFFER_SIZE = 2)
Instance: dpdff#o0_610	(BUFFER_SIZE = 2)
Instance: dpdff#o0_608	(BUFFER_SIZE = 2)
Instance: dpdff#o0_607	(BUFFER_SIZE = 2)
Instance: dpdff#o0_606	(BUFFER_SIZE = 2)
Instance: dpdff#o0_605	(BUFFER_SIZE = 2)
Instance: dpdff#o0_603	(BUFFER_SIZE = 2)
Instance: dpdff#o0_602	(BUFFER_SIZE = 2)
Instance: dpdff#o0_601	(BUFFER_SIZE = 2)
Instance: dpdff#o0_600	(BUFFER_SIZE = 2)
Instance: dpdff#o0_599	(BUFFER_SIZE = 2)
Instance: dpdff#o0_595	(BUFFER_SIZE = 2)
Instance: dpdff#o0_592	(BUFFER_SIZE = 2)
Instance: dpdff#o0_591	(BUFFER_SIZE = 2)
Instance: dpdff#o0_590	(BUFFER_SIZE = 2)
Instance: dpdff#o0_583	(BUFFER_SIZE = 2)
Instance: dpdff#o0_580	(BUFFER_SIZE = 2)
Instance: dpdff#o0_579	(BUFFER_SIZE = 2)
Instance: dpdff#o0_576	(BUFFER_SIZE = 2)
Instance: dpdff#o0_569	(BUFFER_SIZE = 2)
Instance: dpdff#o0_567	(BUFFER_SIZE = 2)
Instance: dpdff#o0_563	(BUFFER_SIZE = 2)
Instance: dpdff#o0_562	(BUFFER_SIZE = 2)
Instance: dpdff#o0_560	(BUFFER_SIZE = 2)
Instance: dpdff#o0_555	(BUFFER_SIZE = 2)
Instance: dpdff#o0_549	(BUFFER_SIZE = 2)
Instance: dpdff#o0_543	(BUFFER_SIZE = 2)
Instance: dpdff#o0_540	(BUFFER_SIZE = 2)

Instance: dpdff#o0_539	(BUFFER_SIZE = 2)
Instance: dpdff#o0_538	(BUFFER_SIZE = 2)
Instance: dpdff#o0_533	(BUFFER_SIZE = 2)
Instance: dpdff#o0_532	(BUFFER_SIZE = 2)
Instance: dpdff#o0_530	(BUFFER_SIZE = 2)
Instance: dpdff#o0_529	(BUFFER_SIZE = 2)
Instance: dpdff#o0_527	(BUFFER_SIZE = 2)
Instance: dpdff#o0_524	(BUFFER_SIZE = 2)
Instance: dpdff#o0_523	(BUFFER_SIZE = 2)
Instance: dpdff#o0_522	(BUFFER_SIZE = 2)
Instance: dpdff#o0_521	(BUFFER_SIZE = 2)
Instance: dpdff#o0_519	(BUFFER_SIZE = 2)
Instance: dpdff#o0_510	(BUFFER_SIZE = 2)
Instance: dpdff#o0_506	(BUFFER_SIZE = 2)
Instance: dpdff#o0_504	(BUFFER_SIZE = 2)
Instance: dpdff#o0_502	(BUFFER_SIZE = 2)
Instance: dpdff#o0_501	(BUFFER_SIZE = 2)
Instance: dpdff#o0_500	(BUFFER_SIZE = 2)
Instance: dpdff#o0_499	(BUFFER_SIZE = 2)
Instance: dpdff#o0_498	(BUFFER_SIZE = 2)
Instance: dpdff#o0_497	(BUFFER_SIZE = 2)
Instance: dpdff#o0_496	(BUFFER_SIZE = 2)
Instance: dpdff#o0_495	(BUFFER_SIZE = 2)
Instance: dpdff#o0_494	(BUFFER_SIZE = 2)
Instance: dpdff#o0_493	(BUFFER_SIZE = 2)
Instance: dpdff#o0_492	(BUFFER_SIZE = 2)
Instance: dpdff#o0_491	(BUFFER_SIZE = 2)
Instance: dpdff#o0_490	(BUFFER_SIZE = 2)
Instance: dpdff#o0_489	(BUFFER_SIZE = 2)
Instance: dpdff#o0_488	(BUFFER_SIZE = 2)
Instance: dpdff#o0_487	(BUFFER_SIZE = 2)
Instance: dpdff#o0_486	(BUFFER_SIZE = 2)
Instance: dpdff#o0_485	(BUFFER_SIZE = 2)
Instance: dpdff#o0_484	(BUFFER_SIZE = 2)
Instance: dpdff#o0_483	(BUFFER_SIZE = 2)
Instance: dpdff#o0_482	(BUFFER_SIZE = 2)
Instance: dpdff#o0_481	(BUFFER_SIZE = 2)
Instance: dpdff#o0_480	(BUFFER_SIZE = 2)
Instance: dpdff#o0_479	(BUFFER_SIZE = 2)
Instance: dpdff#o0_478	(BUFFER_SIZE = 2)
Instance: dpdff#o0_477	(BUFFER_SIZE = 2)
Instance: dpdff#o0_473	(BUFFER_SIZE = 2)
Instance: dpdff#o0_472	(BUFFER_SIZE = 2)
Instance: dpdff#o0_471	(BUFFER_SIZE = 2)
Instance: dpdff#o0_470	(BUFFER_SIZE = 2)

Instance: dpdff#o0_469	(BUFFER_SIZE = 2)
Instance: dpdff#o0_467	(BUFFER_SIZE = 2)
Instance: dpdff#o0_464	(BUFFER_SIZE = 2)
Instance: dpdff#o0_463	(BUFFER_SIZE = 2)
Instance: dpdff#o0_460	(BUFFER_SIZE = 2)
Instance: dpdff#o0_459	(BUFFER_SIZE = 2)
Instance: dpdff#o0_458	(BUFFER_SIZE = 2)
Instance: dpdff#o0_457	(BUFFER_SIZE = 2)
Instance: dpdff#o0_456	(BUFFER_SIZE = 2)
Instance: dpdff#o0_455	(BUFFER_SIZE = 2)
Instance: dpdff#o0_454	(BUFFER_SIZE = 2)
Instance: dpdff#o0_453	(BUFFER_SIZE = 2)
Instance: dpdff#o0_452	(BUFFER_SIZE = 2)
Instance: dpdff#o0_451	(BUFFER_SIZE = 2)
Instance: dpdff#o0_449	(BUFFER_SIZE = 2)
Instance: dpdff#o0_448	(BUFFER_SIZE = 2)
Instance: dpdff#o0_446	(BUFFER_SIZE = 2)
Instance: dpdff#o0_445	(BUFFER_SIZE = 2)
Instance: dpdff#o0_444	(BUFFER_SIZE = 2)
Instance: dpdff#o0_443	(BUFFER_SIZE = 2)
Instance: dpdff#o0_440	(BUFFER_SIZE = 2)
Instance: dpdff#o0_439	(BUFFER_SIZE = 2)
Instance: dpdff#o0_438	(BUFFER_SIZE = 2)
Instance: dpdff#o0_437	(BUFFER_SIZE = 2)
Instance: dpdff#o0_436	(BUFFER_SIZE = 2)
Instance: dpdff#o0_434	(BUFFER_SIZE = 2)
Instance: dpdff#o0_432	(BUFFER_SIZE = 2)
Instance: dpdff#o0_430	(BUFFER_SIZE = 2)
Instance: dpdff#o0_429	(BUFFER_SIZE = 2)
Instance: dpdff#o0_426	(BUFFER_SIZE = 2)
Instance: dpdff#o0_423	(BUFFER_SIZE = 2)
Instance: dpdff#o0_422	(BUFFER_SIZE = 2)
Instance: dpdff#o0_420	(BUFFER_SIZE = 2)
Instance: dpdff#o0_723	(BUFFER_SIZE = 2)
Instance: dpdff#o0_721	(BUFFER_SIZE = 2)
Instance: dpdff#o0_720	(BUFFER_SIZE = 2)
Instance: dpdff#o0_712	(BUFFER_SIZE = 2)
Instance: dpdff#o0_708	(BUFFER_SIZE = 2)
Instance: dpdff#o0_704	(BUFFER_SIZE = 2)
Instance: dpdff#o0_702	(BUFFER_SIZE = 2)
Instance: dpdff#o0_700	(BUFFER_SIZE = 2)
Instance: dpdff#o0_690	(BUFFER_SIZE = 2)
Instance: dpdff#o0_689	(BUFFER_SIZE = 2)
Instance: dpdff#o0_688	(BUFFER_SIZE = 2)
Instance: dpdff#o0_687	(BUFFER_SIZE = 2)

Instance: dpdff#o0_686	(BUFFER_SIZE = 2)
Instance: dpdff#o0_685	(BUFFER_SIZE = 2)
Instance: dpdff#o0_684	(BUFFER_SIZE = 2)
Instance: dpdff#o0_683	(BUFFER_SIZE = 2)
Instance: dpdff#o0_682	(BUFFER_SIZE = 2)
Instance: dpdff#o0_681	(BUFFER_SIZE = 2)
Instance: dpdff#o0_680	(BUFFER_SIZE = 2)
Instance: dpdff#o0_679	(BUFFER_SIZE = 2)
Instance: dpdff#o0_678	(BUFFER_SIZE = 2)
Instance: dpdff#o0_677	(BUFFER_SIZE = 2)
Instance: dpdff#o0_672	(BUFFER_SIZE = 2)
Instance: dpdff#o0_670	(BUFFER_SIZE = 2)
Instance: dpdff#o0_668	(BUFFER_SIZE = 2)
Instance: dpdff#o0_667	(BUFFER_SIZE = 2)
Instance: dpdff#o0_652	(BUFFER_SIZE = 2)
Instance: dpdff#o0_650	(BUFFER_SIZE = 2)
Instance: dpdff#o0_418	(BUFFER_SIZE = 2)
Instance: dpdff#o0_417	(BUFFER_SIZE = 2)
Instance: dpdff#o0_416	(BUFFER_SIZE = 2)
Instance: dpdff#o0_415	(BUFFER_SIZE = 2)
Instance: dpdff#o0_414	(BUFFER_SIZE = 2)
Instance: dpdff#o0_412	(BUFFER_SIZE = 2)
Instance: dpdff#o0_411	(BUFFER_SIZE = 2)
Instance: dpdff#o0_410	(BUFFER_SIZE = 2)
Instance: dpdff#o0_409	(BUFFER_SIZE = 2)
Instance: dpdff#o0_408	(BUFFER_SIZE = 2)
Instance: dpdff#o0_407	(BUFFER_SIZE = 2)
Instance: dpdff#o0_406	(BUFFER_SIZE = 2)
Instance: dpdff#o0_405	(BUFFER_SIZE = 2)
Instance: dpdff#o0_404	(BUFFER_SIZE = 2)
Instance: dpdff#o0_403	(BUFFER_SIZE = 2)
Instance: dpdff#o0_402	(BUFFER_SIZE = 2)
Instance: dpdff#o0_400	(BUFFER_SIZE = 2)
Instance: dpdff#o0_399	(BUFFER_SIZE = 2)
Instance: dpdff#o0_398	(BUFFER_SIZE = 2)
Instance: dpdff#o0_397	(BUFFER_SIZE = 2)
Instance: dpdff#o0_396	(BUFFER_SIZE = 2)
Instance: dpdff#o0_395	(BUFFER_SIZE = 2)
Instance: dpdff#o0_394	(BUFFER_SIZE = 2)
Instance: dpdff#o0_393	(BUFFER_SIZE = 2)
Instance: dpdff#o0_392	(BUFFER_SIZE = 2)
Instance: dpdff#o0_390	(BUFFER_SIZE = 2)
Instance: dpdff#o0_389	(BUFFER_SIZE = 2)
Instance: dpdff#o0_388	(BUFFER_SIZE = 2)
Instance: dpdff#o0_387	(BUFFER_SIZE = 2)

```

Instance: dpdff#o0_386      (BUFFER_SIZE = 2)
Instance: dpdff#o0_385      (BUFFER_SIZE = 2)
Instance: dpdff#o0_384      (BUFFER_SIZE = 2)
Instance: dpdff#o0_383      (BUFFER_SIZE = 2)
Instance: dpdff#o0_382      (BUFFER_SIZE = 2)
Instance: dpdff#o0_381      (BUFFER_SIZE = 2)
Instance: dpdff#o0_380      (BUFFER_SIZE = 2)
Instance: dpdff#o0_379      (BUFFER_SIZE = 2)
Instance: dpdff#o0_378      (BUFFER_SIZE = 2)
Instance: dpdff#o0_377      (BUFFER_SIZE = 2)
Instance: dpdff#o0_376      (BUFFER_SIZE = 2)
Instance: dpdff#o0_375      (BUFFER_SIZE = 2)
Instance: dpdff#o0_374      (BUFFER_SIZE = 2)
Instance: dpdff#o0_373      (BUFFER_SIZE = 2)
Instance: dpdff#o0_372      (BUFFER_SIZE = 2)
Instance: dpdff#o0_371      (BUFFER_SIZE = 2)
Instance: dpdff#o0_370      (BUFFER_SIZE = 2)
Instance: dpdff#o0_369      (BUFFER_SIZE = 2)
Instance: dpdff#o0_368      (BUFFER_SIZE = 2)
Instance: dpdff#o0_367      (BUFFER_SIZE = 2)
Instance: dpdff#o0_366      (BUFFER_SIZE = 2)
Instance: dpdff#o0_365      (BUFFER_SIZE = 2)
Instance: dpdff#o0_364      (BUFFER_SIZE = 2)
Instance: dpdff#o0_643      (BUFFER_SIZE = 2)
Instance: dpdff#o0_587      (BUFFER_SIZE = 2)
Instance: dpdff#o0_559      (BUFFER_SIZE = 2)
Instance: dpdff#o0_531      (BUFFER_SIZE = 2)
Instance: dpdff#o0_391      (BUFFER_SIZE = 2)
Creating new modules for composite cell: mplier
Instance: Complementor2:compl4      (BUFFER_SIZE = 4)
Instance: Complementor:compl4      (BUFFER_SIZE = 4)
Instance: Adder5:fadder#7      (BUFFER_SIZE = 4)
Instance: Adder2:fadder#7      (BUFFER_SIZE = 4)
Instance: Adder1:fadder#7      (BUFFER_SIZE = 4)
Instance: Adder3:fadder#7      (BUFFER_SIZE = 4)
Creating new modules for composite cell: complex_adder_4_input
Instance: or10      (BUFFER_SIZE = 5)
Writing specification cell stdor3#o3, compiler gtgate
Instance: or9      (BUFFER_SIZE = 5)
Instance: or8      (BUFFER_SIZE = 5)
Instance: or7      (BUFFER_SIZE = 5)
Instance: or6      (BUFFER_SIZE = 5)
/local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW

```

```

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
+trans dpdff#o1 dpdff#o2 dpdff#o3
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell dpdff#o1...
=====> Generating Layout.
Writing cell dpdff#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing dpdff#t1 -- (t74s1)
Writing dpdff#s1 -- (s74s1)
=====> End Simulation Model.
Calling compiler for cell dpdff#o2...
=====> Generating Layout.
Writing cell dpdff#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s13 -- (s107s13)
Writing sprim_inv#t13 -- (t107s13)
Writing dpdff#t2 -- (t74s2)
Writing dpdff#s2 -- (s74s2)
=====> End Simulation Model.
Calling compiler for cell dpdff#o3...
=====> Generating Layout.
Writing cell dpdff#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing dpdff#t3 -- (t74s3)
Writing dpdff#s3 -- (s74s3)
=====> End Simulation Model.
No errors. No warnings.
/local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
+trans dpadder#o1
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell dpadder#o1...
=====> Generating Layout.

```


Writing cell dpadder#g1 to disk
 =====> End Layout.
 =====> Generating Simulation Model.
 Writing dpadder#s1 -- (s119s1)
 =====> End Simulation Model.
 No errors. No warnings.
 /local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
 CDA1u2m1p +msgclear -msgfile +ewpstfEW
 /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
 +trans dpcompl#o2
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 Calling compiler for cell dpcompl#o2...
 =====> Generating Layout.
 Writing cell dpcompl#g2 to disk
 =====> End Layout.
 =====> Generating Simulation Model.
 Writing sprim_inv#s14 -- (s107s14)
 Writing sprim_inv#t14 -- (t107s14)
 Writing dpcompl#s2 -- (s212s2)
 =====> End Simulation Model.
 No errors. No warnings.
 /local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
 CDA1u2m1p +msgclear -msgfile +ewpstfEW
 /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
 +trans dpfadder_bit#o26
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 Calling compiler for cell dpfadder_bit#o26...
 =====> Generating Layout.
 Writing cell d_dpadder_carryend#g2 to disk
 Writing cell d_dpadder_sumtyp#g3 to disk
 Writing cell dpfadder_bit#g26 to disk
 =====> End Layout.
 =====> Generating Simulation Model.
 Writing d_dpadder_carryend#s2 -- (s96s2)
 Writing d_dpadder_sumtyp#s3 -- (s26s3)
 Writing dpfadder_bit#s26 -- (s123s26)
 =====> End Simulation Model.

No errors. No warnings.

```
/local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
+trans gtadd#o2 gtadd#o3 gtadd#o4 gtadd#o5 gtadd#o6 gtadd#o7 gtadd#o8 gtadd#o9
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell gtadd#o2...
=====> Generating Layout.
Writing cell gtadd#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s15 -- (s107s15)
Writing sprim_inv#t15 -- (t107s15)
Writing gtadd#s2 -- (s186s2)
Writing gtadd#t1 -- (t186s1)
=====> End Simulation Model.
Calling compiler for cell gtadd#o3...
=====> Generating Layout.
Writing cell gtadd#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtadd#s3 -- (s186s3)
Writing gtadd#t2 -- (t186s2)
=====> End Simulation Model.
Calling compiler for cell gtadd#o4...
=====> Generating Layout.
Writing cell gtadd#g4 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s16 -- (s107s16)
Writing sprim_inv#t16 -- (t107s16)
Writing gtadd#s4 -- (s186s4)
Writing gtadd#t3 -- (t186s3)
=====> End Simulation Model.
Calling compiler for cell gtadd#o5...
=====> Generating Layout.
Writing cell gtadd#g5 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtadd#s5 -- (s186s5)
Writing gtadd#t4 -- (t186s4)
```

```

=====> End Simulation Model.
Calling compiler for cell gtadd#o6...
=====> Generating Layout.
Writing cell gtadd#g6 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtadd#s6 -- (s186s6)
=====> End Simulation Model.
Calling compiler for cell gtadd#o7...
=====> Generating Layout.
Writing cell gtadd#g7 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtadd#s7 -- (s186s7)
Writing gtadd#t5 -- (t186s5)
=====> End Simulation Model.
Calling compiler for cell gtadd#o8...
=====> Generating Layout.
Writing cell gtadd#g8 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtadd#s8 -- (s186s8)
Writing gtadd#t6 -- (t186s6)
=====> End Simulation Model.
Calling compiler for cell gtadd#o9...
=====> Generating Layout.
Writing cell gtadd#g9 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtadd#s9 -- (s186s9)
=====> End Simulation Model.
No errors. No warnings.
/local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
+trans gtmux#o2 gtmux#o3 gtmux#o4 gtmux#o5 gtmux#o6 gtmux#o7 stdmux2#o1
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell gtmux#o2...
=====> Generating Layout.
Writing cell gtmux#g2 to disk
=====> End Layout.

```

```

=====> Generating Simulation Model.
Writing gtmux#s2 -- (s67s2)
=====> End Simulation Model.
Calling compiler for cell gtmux#o3...
=====> Generating Layout.
Writing cell gtmux#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtmux#s3 -- (s67s3)
=====> End Simulation Model.
Calling compiler for cell gtmux#o4...
=====> Generating Layout.
Writing cell gtmux#g4 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s17 -- (s107s17)
Writing sprim_inv#t17 -- (t107s17)
Writing gtmux#s4 -- (s67s4)
=====> End Simulation Model.
Calling compiler for cell gtmux#o5...
=====> Generating Layout.
Writing cell gtmux#g5 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtmux#s5 -- (s67s5)
=====> End Simulation Model.
Calling compiler for cell gtmux#o6...
=====> Generating Layout.
Writing cell gtmux#g6 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtmux#s6 -- (s67s6)
=====> End Simulation Model.
Calling compiler for cell gtmux#o7...
=====> Generating Layout.
Writing cell gtmux#g7 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtmux#s7 -- (s67s7)
=====> End Simulation Model.
Calling compiler for cell stdmux2#o1...
=====> Generating Layout.
Writing cell stdmux2#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.

```

```

Writing stdmux2#s1 -- (s63s1)
=====> End Simulation Model.
No errors. No warnings.
/local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
+trans gtlatch#o1 gtlatch#o2 gtlatch#o3 gtlatch#o4 gtlatch#o5
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell gtlatch#o1...
=====> Generating Layout.
Writing cell gtlatch#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s18 -- (s107s18)
Writing sprim_inv#t18 -- (t107s18)
Writing d_latch_latch#s1 -- (s172s1)
Writing gtlatch#s1 -- (s194s1)
=====> End Simulation Model.
Calling compiler for cell gtlatch#o2...
=====> Generating Layout.
Writing cell gtlatch#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_latch_latch#s2 -- (s172s2)
Writing gtlatch#s2 -- (s194s2)
=====> End Simulation Model.
Calling compiler for cell gtlatch#o3...
=====> Generating Layout.
Writing cell gtlatch#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_latch_latch#s3 -- (s172s3)
Writing gtlatch#s3 -- (s194s3)
=====> End Simulation Model.
Calling compiler for cell gtlatch#o4...
=====> Generating Layout.
Writing cell gtlatch#g4 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_latch_latch#s4 -- (s172s4)
Writing gtlatch#s4 -- (s194s4)

```

```

=====> End Simulation Model.
Calling compiler for cell gtlatch#o5...
=====> Generating Layout.
Writing cell gtlatch#g5 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_latch_latch#s5 -- (s172s5)
Writing gtlatch#s5 -- (s194s5)
=====> End Simulation Model.
No errors. No warnings.
/local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
+trans stdinv#o1 stdand2#o1 stdor2#o1 stdor3#o1 stdor4#o1 stdand3#o1 stdor2#o2
gtgate#o8 stdinv#o2 stdxnor2#o1
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell stdinv#o1...
=====> Generating Layout.
Writing cell stdinv#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdinv#s1 -- (s187s1)
=====> End Simulation Model.
Calling compiler for cell stdand2#o1...
=====> Generating Layout.
Writing cell stdand2#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdand2#s1 -- (s191s1)
=====> End Simulation Model.
Calling compiler for cell stdor2#o1...
=====> Generating Layout.
Writing cell stdor2#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdor2#s1 -- (s95s1)
=====> End Simulation Model.
Calling compiler for cell stdor3#o1...
=====> Generating Layout.
Writing cell stdor3#g1 to disk
=====> End Layout.

```

```

=====> Generating Simulation Model.
Writing sprim_inv#s19 -- (s107s19)
Writing sprim_inv#t19 -- (t107s19)
Writing stdor3#s1 -- (s99s1)
=====> End Simulation Model.
Calling compiler for cell stdor4#o1...
=====> Generating Layout.
Writing cell stdor4#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdor4#s1 -- (s102s1)
=====> End Simulation Model.
Calling compiler for cell stdand3#o1...
=====> Generating Layout.
Writing cell stdand3#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdand3#s1 -- (s195s1)
=====> End Simulation Model.
Calling compiler for cell stdor2#o2...
=====> Generating Layout.
Writing cell stdor2#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdor2#s2 -- (s95s2)
=====> End Simulation Model.
Calling compiler for cell gtgate#o8...
=====> Generating Layout.
Writing cell gtgate#g8 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtgate#s8 -- (s131s8)
=====> End Simulation Model.
Calling compiler for cell stdinv#o2...
=====> Generating Layout.
Writing cell stdinv#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdinv#s2 -- (s187s2)
=====> End Simulation Model.
Calling compiler for cell stdxnor2#o1...
=====> Generating Layout.
Writing cell stdxnor2#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.

```

Writing d_gate_xnor#s0 -- (s7s0)
 Writing stdxnor2#s1 -- (s106s1)
 =====> End Simulation Model.
 No errors. No warnings.
 /local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
 CDA1u2m1p +msgclear -msgfile +ewpstfEW
 /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
 +trans stdnor2#o1 stdand4#o1 stdor3#o2 stdxnor2#o2 stdnor4#o1 stdnor3#o1 gtgate#o9
 stdinv#o3 gtgate#o10 stdand2#o2 stdnor3#o2 stdao211#o1 stdoi22#o1 stdoi222#o1
 stdao21#o1 stdoi21#o1 stdnand2#o1 stdor4#o2 stdnor2#o2 stdor3#o3
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 Calling compiler for cell stdnor2#o1...
 =====> Generating Layout.
 Writing cell stdnor2#g1 to disk
 =====> End Layout.
 =====> Generating Simulation Model.
 Writing sprim_nor2#s4 -- (s167s4)
 Writing sprim_nor2#t4 -- (t167s4)
 Writing stdnor2#s1 -- (s171s1)
 =====> End Simulation Model.
 Calling compiler for cell stdand4#o1...
 =====> Generating Layout.
 Writing cell stdand4#g1 to disk
 =====> End Layout.
 =====> Generating Simulation Model.
 Writing stdand4#s1 -- (s199s1)
 =====> End Simulation Model.
 Calling compiler for cell stdor3#o2...
 =====> Generating Layout.
 Writing cell stdor3#g2 to disk
 =====> End Layout.
 =====> Generating Simulation Model.
 Writing stdor3#s2 -- (s99s2)
 =====> End Simulation Model.
 Calling compiler for cell stdxnor2#o2...
 =====> Generating Layout.
 Writing cell stdxnor2#g2 to disk
 =====> End Layout.
 =====> Generating Simulation Model.
 Writing d_gate_xnor#s1 -- (s7s1)
 Writing stdxnor2#s2 -- (s106s2)


```

=====> End Simulation Model.
Calling compiler for cell stdnor4#o1...
=====> Generating Layout.
Writing cell stdnor4#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_nor4#s1 -- (s173s1)
Writing sprim_nor4#t1 -- (t173s1)
Writing stdnor4#s1 -- (s178s1)
Writing stdnor4#t0 -- (t178s0)
=====> End Simulation Model.
Calling compiler for cell stdnor3#o1...
=====> Generating Layout.
Writing cell stdnor3#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_nor3#s1 -- (s170s1)
Writing sprim_nor3#t1 -- (t170s1)
Writing stdnor3#s1 -- (s175s1)
Writing stdnor3#t0 -- (t175s0)
=====> End Simulation Model.
Calling compiler for cell gtgate#o9...
=====> Generating Layout.
Writing cell gtgate#g9 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtgate#s9 -- (s131s9)
=====> End Simulation Model.
Calling compiler for cell stdinv#o3...
=====> Generating Layout.
Writing cell stdinv#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s20 -- (s107s20)
Writing sprim_inv#t20 -- (t107s20)
Writing sprim_inv#s21 -- (s107s21)
Writing sprim_inv#t21 -- (t107s21)
Writing sprim_inv#s22 -- (s107s22)
Writing sprim_inv#t22 -- (t107s22)
Writing d_buffer_extra#s0 -- (s109s0)
Writing stdinv#s3 -- (s187s3)
=====> End Simulation Model.
Calling compiler for cell gtgate#o10...
=====> Generating Layout.
Writing cell gtgate#g10 to disk

```

```

=====> End Layout.
=====> Generating Simulation Model.
Writing  gtgate#s10 -- (s131s10)
=====> End Simulation Model.
Calling compiler for cell stdand2#o2...
=====> Generating Layout.
Writing cell stdand2#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing  stdand2#s2 -- (s191s2)
=====> End Simulation Model.
Calling compiler for cell stdnor3#o2...
=====> Generating Layout.
Writing cell d_stdcell_gatebuffer#g1 to disk
Writing cell stdnor3#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing  d_gate_nor3#s0 -- (s224s0)
Writing  stdnor3#s2 -- (s175s2)
=====> End Simulation Model.
Calling compiler for cell stdao211#o1...
=====> Generating Layout.
Writing cell d_stdcell_gatebuffer#g2 to disk
Writing cell stdao211#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing  stdao211#s1 -- (s35s1)
=====> End Simulation Model.
Calling compiler for cell stdoi22#o1...
=====> Generating Layout.
Writing cell d_stdcell_gatebuf#g0 to disk
Writing cell stdoi22#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing  stdoi22#s1 -- (s220s1)
=====> End Simulation Model.
Calling compiler for cell stdoi222#o1...
=====> Generating Layout.
Writing cell stdoi222#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing  stdoi222#s1 -- (s121s1)
=====> End Simulation Model.
Calling compiler for cell stdao21#o1...
=====> Generating Layout.

```

```

Writing cell stdao21#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdao21#s1 -- (s114s1)
=====> End Simulation Model.
Calling compiler for cell stdaoi21#o1...
=====> Generating Layout.
Writing cell stdaoi21#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdaoi21#s1 -- (s154s1)
=====> End Simulation Model.
Calling compiler for cell stdnand2#o1...
=====> Generating Layout.
Writing cell stdnand2#g1 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_nand2#s3 -- (s130s3)
Writing sprim_nand2#t3 -- (t130s3)
Writing stdnand2#s1 -- (s163s1)
=====> End Simulation Model.
Calling compiler for cell stdor4#o2...
=====> Generating Layout.
Writing cell stdor4#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdor4#s2 -- (s102s2)
=====> End Simulation Model.
Calling compiler for cell stdnor2#o2...
=====> Generating Layout.
Writing cell stdnor2#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_nor2#s5 -- (s167s5)
Writing sprim_nor2#t5 -- (t167s5)
Writing stdnor2#s2 -- (s171s2)
Writing stdnor2#t0 -- (t171s0)
=====> End Simulation Model.
Calling compiler for cell stdor3#o3...
=====> Generating Layout.
Writing cell stdor3#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing stdor3#s3 -- (s99s3)
=====> End Simulation Model.

```

No errors. No warnings.

```

/local/epoch/bin/msicells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/msicells.log +geo -sym +sim
+trans gtdff#o2 gtdff#o3 gtdff#o4 gtdff#o5 gtdff#o6 gtdff#o7 gtdff#o8
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell gtdff#o2...
=====> Generating Layout.
Writing cell gtdff#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtdff#t2 -- (t216s2)
Writing gtdff#s2 -- (s216s2)
=====> End Simulation Model.
Calling compiler for cell gtdff#o3...
=====> Generating Layout.
Writing cell gtdff#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtdff#t3 -- (t216s3)
Writing sprim_inv#s23 -- (s107s23)
Writing sprim_inv#t23 -- (t107s23)
Writing gtdff#s3 -- (s216s3)
=====> End Simulation Model.
Calling compiler for cell gtdff#o4...
=====> Generating Layout.
Writing cell gtdff#g4 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtdff#t4 -- (t216s4)
Writing gtdff#s4 -- (s216s4)
=====> End Simulation Model.
Calling compiler for cell gtdff#o5...
=====> Generating Layout.
Writing cell gtdff#g5 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtdff#t5 -- (t216s5)
Writing gtdff#s5 -- (s216s5)
=====> End Simulation Model.
Calling compiler for cell gtdff#o6...

```

```

=====> Generating Layout.
Writing cell gtdff#g6 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing gtdff#t6 -- (t216s6)
Writing gtdff#s6 -- (s216s6)
=====> End Simulation Model.
Calling compiler for cell gtdff#o7...
=====> Generating Layout.
Writing cell gtdff#g7 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s24 -- (s107s24)
Writing sprim_inv#t24 -- (t107s24)
Writing gtdff#t7 -- (t216s7)
Writing gtdff#s7 -- (s216s7)
=====> End Simulation Model.
Calling compiler for cell gtdff#o8...
=====> Generating Layout.
Writing cell gtdff#g8 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing sprim_inv#s25 -- (s107s25)
Writing sprim_inv#t25 -- (t107s25)
Writing gtdff#t8 -- (t216s8)
Writing gtdff#s8 -- (s216s8)
=====> End Simulation Model.
No errors. No warnings.
/local/epoch/bin/padcells -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r
CDA1u2m1p +msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/padcells.log +geo -sym +sim
+trans pad_driver#o2 pad_driver#o3 pad_driver#o4 pad_driver#o5 pad_driver#o6
pad_driver#o7
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Calling compiler for cell pad_driver#o2...
=====> Generating Layout.
Writing cell d_pad_ctrl_buffer#g2 to disk
Writing cell pad_driver#g2 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_pad_ctrl_buffer#s2 -- (s206s2)

```

```

Writing pad_driver#s2 -- (s215s2)
=====> End Simulation Model.
Calling compiler for cell pad_driver#o3...
=====> Generating Layout.
Writing cell d_pad_ctrl_buffer#g3 to disk
Writing cell pad_driver#g3 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_pad_ctrl_buffer#s3 -- (s206s3)
Writing pad_driver#s3 -- (s215s3)
=====> End Simulation Model.
Calling compiler for cell pad_driver#o4...
=====> Generating Layout.
Writing cell d_pad_ctrl_buffer#g4 to disk
Writing cell pad_driver#g4 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_pad_ctrl_buffer#s4 -- (s206s4)
Writing pad_driver#s4 -- (s215s4)
=====> End Simulation Model.
Calling compiler for cell pad_driver#o5...
=====> Generating Layout.
Writing cell d_pad_ctrl_buffer#g5 to disk
Writing cell pad_driver#g5 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_buffer_extra#s1 -- (s109s1)
Writing d_pad_ctrl_buffer#s5 -- (s206s5)
Writing pad_driver#s5 -- (s215s5)
=====> End Simulation Model.
Calling compiler for cell pad_driver#o6...
=====> Generating Layout.
Writing cell d_pad_ctrl_buffer#g6 to disk
Writing cell pad_driver#g6 to disk
=====> End Layout.
=====> Generating Simulation Model.
Writing d_buffer_extra#s2 -- (s109s2)
Writing d_pad_ctrl_buffer#s6 -- (s206s6)
Writing pad_driver#s6 -- (s215s6)
=====> End Simulation Model.
Calling compiler for cell pad_driver#o7...
=====> Generating Layout.
Writing cell d_pad_ctrl_buffer#g7 to disk
Writing cell pad_driver#g7 to disk
=====> End Layout.

```

=====> Generating Simulation Model.

Writing d_buffer_extra#s3 -- (s109s3)

Writing d_pad_ctrl_buffer#s7 -- (s206s7)

Writing pad_driver#s7 -- (s215s7)

=====> End Simulation Model.

No errors. No warnings.

PDABS: Estimated Static Power: 0 microWatts

PDABS: Estimated Total Power: 762555 microWatts

Writing cell fpadder_group to disk

Writing cell fpadder to disk

Writing cell hsmult2piped_c_0#g0 to disk

Writing cell mplier to disk

Writing cell fpmplier to disk

Writing cell complex_mplier to disk

Writing cell complex_adder_4_input to disk

Writing cell fft to disk

Writing cell fftchip_core to disk

Writing cell fftchip to disk

PDABS: No errors. 104 warnings.

PDABS: Done.

/local/epoch/bin/dpath5 -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p

+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/dpath5.log -nplace

fpadder_group

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

DATAPATH: Reading datapath geometry.

DATAPATH: Reading footprints.

DATAPATH: Initial bit cells placement.

DATAPATH: Routing datapath.

DATAPATH 3.0

size: 26 X 3

Horizontal channels

Contour Router V1.1

channel #26

channel #25

channel #24

channel #23
channel #22
channel #21
channel #20
channel #19
channel #18
channel #17
channel #16
channel #15
channel #14
channel #13
channel #12
channel #11
channel #10
channel #9
channel #8
channel #7
channel #6
channel #5
channel #4
channel #3
channel #2
channel #1
channel #0
Vertical channels
channel #3
channel #2
channel #1
channel #0

DATAPATH: Creating geometry.
Writing cell fpadder_group to disk

/local/epoch/bin/glueup -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/glueup.log fpadder
GLUE ROUTER version 6.1.
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Glue: reading in geometry file fpadder
Initialize power connections.
Read in placement expression.
Position export pins.

Route power nets. (auto_size_rails)

-- GND

-- VDD

Read in footprint information.

Over the cell routing gates.

Over the cell routing blocks.

Overcell Router V2.0

Generating global router input.

Mixed-Mode Global Router V1.1

Routing group [0]

Routing group [1]

Routing group [2]

Routing group [3]

Routing group [4]

Assemble Standard cell groups

Assemble Group std_group_0

Contour Router V1.1

Channel no. 0

Channel no. 1

Assemble Group std_group_1

Channel no. 2

Channel no. 3

Channel no. 4

Channel no. 5

Channel no. 6

Channel no. 7

Channel no. 8

Assemble Group rnder

Channel no. 9

Channel no. 10

Channel no. 11

Channel no. 12

Assemble Group std_group_3

Channel no. 13

Channel no. 14

Channel no. 15

Channel no. 16

Channel no. 17

Channel no. 18

Channel no. 19

Channel no. 20

Channel no. 21

Assemble Group std_group_2

Channel no. 22

Channel no. 23

Channel no. 24
 Channel no. 25
 Channel no. 26
 Channel no. 27
 Channel no. 28
 Channel no. 29
 Channel no. 30
 Channel no. 31
 Channel no. 32
 Assemble Blocks
 Channel no. 33
 Channel no. 34
 Channel no. 35
 Channel no. 36
 Channel no. 37
 Route perimeter channels.
 Ripping External Channels
 Channel no. 38
 Channel no. 39
 Channel no. 40
 Channel no. 41
 Generating Geometry.
 Writing cell fpadder to disk
 Glue: fpadder completed, Real time: 00:06:24, CPU time: 291.08 seconds.

```

/local/epoch/bin/glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/glue.log
complex_adder_4_input
GLUE ROUTER version 6.1.
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Glue: reading in geometry file complex_adder_4_input .....
Initialize power connections.
Read in placement expression.
Position export pins.
Route power nets. (auto_size_rails)
-- GND
-- VDD
Read in footprint information.
Over the cell routing gates.
Over the cell routing blocks.
  
```

Overcell Router V2.0
OCR: All standard cell design.
Generating global router input.
Mixed-Mode Global Router V1.1

Routing group [0]
Assemble Standard cell groups
Assemble Group std_group_0
Assemble Blocks
Contour Router V1.1

Channel no. 0
Channel no. 1
Channel no. 2
Channel no. 3
Channel no. 4
Channel no. 5

Route perimeter channels.
Ripping External Channels
Channel no. 6
Channel no. 7
Channel no. 8
Channel no. 9

Generating Geometry.
Writing cell complex_adder_4_input to disk
Glue: complex_adder_4_input completed, Real time: 00:03:40, CPU time: 205.30
seconds.

/local/epoch/bin/dpath5 -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/dpath5.log -nplace
hsmult2pipcd_c_0#o0

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

DATAPATH: Reading datapath geometry.

DATAPATH: Reading footprints.

DATAPATH: Initial bit cells placement.

DATAPATH: Routing datapath.

DATAPATH 3.0

size: 28 X 44

Horizontal channels

Contour Router V1.1

channel #28
channel #27
channel #26
channel #25
channel #24
channel #23
channel #22
channel #21
channel #20
channel #19
channel #18
channel #17
channel #16
channel #15
channel #14
channel #13
channel #12
channel #11
channel #10
channel #9
channel #8
channel #7
channel #6
channel #5
channel #4
channel #3
channel #2
channel #1
channel #0

Vertical channels

channel #44
channel #43
channel #42
channel #41
channel #40
channel #39
channel #38
channel #37
channel #36
channel #35
channel #34
channel #33
channel #32

channel #31
channel #30
channel #29
channel #28
channel #27
channel #26
channel #25
channel #24
channel #23
channel #22
channel #21
channel #20
channel #19
channel #18
channel #17
channel #16
channel #15
channel #14
channel #13
channel #12
channel #11
channel #10
channel #9
channel #8
channel #7
channel #6
channel #5
channel #4
channel #3
channel #2
channel #1
channel #0

DATAPATH: Creating geometry.

Writing cell hsmult2pipd_c_0#g0 to disk

/local/epoch/bin/dpath5 -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/dpath5.log -nplace mplier

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

DATAPATH: Reading datapath geometry.

DATAPATH: Reading footprints.

DATAPATH: Initial bit cells placement.
DATAPATH: Routing datapath.

DATAPATH 3.0

size: 8 X 6

Horizontal channels
Contour Router V1.1

channel #8
channel #7
channel #6
channel #5
channel #4
channel #3
channel #2
channel #1
channel #0

Vertical channels

channel #6
channel #5
channel #4
channel #3
channel #2
channel #1
channel #0

DATAPATH: Creating geometry.
Writing cell mplier to disk

```
/local/epoch/bin/gluepn -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/gluepn.log fpmplier
GLUE ROUTER version 6.1.
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Glue: reading in geometry file fpmplier .....
Initialize power connections.
Read in placement expression.
Position export pins.
Route power nets. (auto_size_rails)
-- GND
```

-- VDD
Read in footprint information.
Over the cell routing gates.
Over the cell routing blocks.
Overcell Router V2.0
Generating global router input.
Mixed-Mode Global Router V1.1
 Routing group [0]
 Routing group [1]
 Routing group [2]
 Routing group [3]
 Routing group [4]
Assemble Standard cell groups
Assemble Group std_group_1
Merging multiple external standard cell connections for net clear
Merging multiple external standard cell connections for net clock
Merging multiple external standard cell connections for net lost_bit
Assemble Group std_group_3
Contour Router V1.1
Channel no. 0
Channel no. 1
Channel no. 2
Channel no. 3
Channel no. 4
Channel no. 5
Channel no. 6
Channel no. 7
Channel no. 8
Channel no. 9
Channel no. 10
Assemble Group rnder
Channel no. 11
Assemble Group std_group_2
Channel no. 12
Channel no. 13
Channel no. 14
Channel no. 15
Channel no. 16
Assemble Group std_group_0
Channel no. 17
Channel no. 18
Channel no. 19
Channel no. 20
Channel no. 21
Channel no. 22

Channel no. 23
 Channel no. 24
 Assemble Blocks
 Channel no. 25
 Channel no. 26
 Channel no. 27
 Channel no. 28
 Channel no. 29
 Channel no. 30
 Route perimeter channels.
 Ripping External Channels
 Channel no. 31
 Channel no. 32
 Channel no. 33
 Channel no. 34
 Generating Geometry.
 Writing cell fpmplier to disk
 Glue: fpmplier completed, Real time: 00:09:06, CPU time: 469.83 seconds.

/local/epoch/bin/glue -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
 +msgclear -msgfile +ewpstfEW
 /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/glue.log complex_mplier
 GLUE ROUTER version 6.1.
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 Glue: reading in geometry file complex_mplier
 Initialize power connections.
 Read in placement expression.
 Position export pins.
 Route power nets. (auto_size_rails)
 -- GND
 -- VDD
 Read in footprint information.
 Over the cell routing gates.
 Over the cell routing blocks.
 Overcell Router V2.0
 OCR: All standard cell design.
 Generating global router input.
 Mixed-Mode Global Router V1.1
 Routing group [0]
 Assemble Standard cell groups
 Assemble Group std_group_0

Assemble Blocks
Contour Router V1.1
Channel no. 0
Channel no. 1
Channel no. 2
Channel no. 3
Channel no. 4
Channel no. 5
Route perimeter channels.
Ripping External Channels
Channel no. 6
Channel no. 7
Channel no. 8
Channel no. 9
Generating Geometry.
Writing cell complex_mplier to disk
Glue: complex_mplier completed, Real time: 00:03:34, CPU time: 192.91 seconds.

/local/epoch/bin/glueup -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
+msgclear -msgfile +ewpstfEW
/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/glueup.log fft
GLUE ROUTER version 6.1.
Copyright 1992, 1993 Cascade Design Automation Corporation.
CASCADE base location: /local/epoch
Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
Ruleset: CDA1u2m1p
Technology: cmos
Glue: reading in geometry file fft
Initialize power connections.
Read in placement expression.
Position export pins.
Route power nets. (auto_size_rails)
-- GND
-- VDD
Read in footprint information.
Over the cell routing gates.
Over the cell routing blocks.
Overcell Router V2.0
OCR: All standard cell design.
Generating global router input.
Mixed-Mode Global Router V1.1
 Routing group [0]
 Routing group [1]
Assemble Standard cell groups
Assemble Group fft_top

Contour Router V1.1

Channel no. 0

Channel no. 1

Channel no. 2

Channel no. 3

Channel no. 4

Channel no. 5

Channel no. 6

Channel no. 7

Channel no. 8

Channel no. 9

Channel no. 10

Channel no. 11

Channel no. 12

Assemble Group std_group_0

Assemble Blocks

Channel no. 13

Channel no. 14

Channel no. 15

Route perimeter channels.

Ripping External Channels

Channel no. 16

Channel no. 17

Channel no. 18

Channel no. 19

Generating Geometry.

Writing cell fft to disk

Glue: fft completed, Real time: 00:13:37, CPU time: 722.35 seconds.

/local/epoch/bin/gluep -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p

+msgclear -msgfile +ewpstfEW

/tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/gluep.log fftchip_core

GLUE ROUTER version 6.1.

Copyright 1992, 1993 Cascade Design Automation Corporation.

CASCADE base location: /local/epoch

Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2

Ruleset: CDA1u2m1p

Technology: cmos

Glue: reading in geometry file fftchip_core

Initialize power connections.

Read in placement expression.

Position export pins.

Route power nets. (auto_size_rails)

-- GND

-- VDD

Read in footprint information.
 Over the cell routing gates.
 Over the cell routing blocks.
 Overcell Router V2.0
 OCR: All standard cell design.
 Generating global router input.
 Mixed-Mode Global Router V1.1
 Routing group [0]
 Assemble Standard cell groups
 Assemble Group std_group_0
 Assemble Blocks
 Contour Router V1.1
 Channel no. 0
 Route perimeter channels.
 Ripping External Channels
 Channel no. 1
 Channel no. 2
 Channel no. 3
 Channel no. 4
 Generating Geometry.
 Writing cell fftchip_core to disk
 Glue: fftchip_core completed, Real time: 00:04:16, CPU time: 230.83 seconds.

/local/epoch/bin/plop5 -p /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2 -r CDA1u2m1p
 +msgclear -msgfile +ewpstfEW
 /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2/chip_logs/plop5.log fftchip
 Copyright 1992, 1993 Cascade Design Automation Corporation.
 CASCADE base location: /local/epoch
 Project: /tmp_mnt/h/kepler_u2/kljackso/projects/thesis2
 Ruleset: CDA1u2m1p
 Technology: cmos
 PACKAGER: Reading top level chip data.
 PACKAGER: No package has been specified, automatically selecting package.
 -----> packcell->name = "cda100p433".
 -----> packcell->name = "cda100p540".
 -----> packcell->name = "cda120p433".
 -----> packcell->name = "cda124p335".
 -----> packcell->name = "cda132p400".
 -----> packcell->name = "cda132p410".
 -----> packcell->name = "cda132p540".
 -----> packcell->name = "cda16p120".
 -----> packcell->name = "cda16p315".
 -----> packcell->name = "cda180p480".
 -----> packcell->name = "cda20p160".
 -----> packcell->name = "cda22p140".

```

-----> packcell->name = "cda24p350".
-----> packcell->name = "cda28p250".
-----> packcell->name = "cda28p310".
-----> packcell->name = "cda40p250".
-----> packcell->name = "cda40p300".
-----> packcell->name = "cda48p250".
-----> packcell->name = "cda48p310".
-----> packcell->name = "cda48p350".
-----> packcell->name = "cda64p400".
-----> packcell->name = "cda68p400".
-----> packcell->name = "cda72p340".
-----> packcell->name = "cda84p400".
-----> packcell->name = "cda84p450".
-----> packcell->name = "cda84p470".

```

----->Can't find any appropriate package, place pads without package.

PACKAGER: Placing pads.

MOAT ROUTER: Routing East channel.

Contour Router V1.1

MOAT ROUTER: Routing West channel.

MOAT ROUTER: Routing North channel.

MOAT ROUTER: Routing South channel.

MOAT ROUTER: Done.

Writing cell fitchip to disk

Die size : 441 x 616 (mils) Scribe spacing : 0 (mils)

Die size : 11190 x 15642 (um) Scribe spacing : 1 (um)

Module size : 391 x 567 (mils) NONE

Module size : 9943 x 14414 (um)

Number of nets : 408

Max. routing space : 15 (mils) Min. routing space : 10 (mils)

Max. routing space : 391 (um) Min. routing space : 248 (um)

Number of pads : 210

Max. pad height : 6 (mils) Min. pad height :6 (mils)

Max. pad height : 139 (um) Min. pad height :139 (um)

APPENDIX E FLOATING POINT ADDER TEST OUTPUT

Host command: verilog

Command arguments:

```
-f adder.vc
-y /tmp_mnt/local/epoch/models/cmos/verilog
+libext+.v+
-s
addertst.v
bitshift.v
overshift.v
inverter.v
inverter2.v
rounder.v
mulclnup.v
vcheck.v
fpadder.v
-s
```

VERILOG-XL 1.8 log file created Sep 18, 1994 01:47:30

VERILOG-XL 1.8 Sep 18, 1994 01:47:30

```
* Copyright Cadence Design Systems, Inc. 1985, 1988.  *
* All Rights Reserved. Licensed Software.  *
* Confidential and proprietary information which is the  *
* property of Cadence Design Systems, Inc.  *

* For technical assistance please contact the Cadence  *
* Response Center at 1-800-CADENC2 or send email to  *
* crc_customers@cadence.com  *
```

```
* For more information on Cadence's Verilog-XL product  *
* line send email to talkverilog@cadence.com  *
```

Compiling source file "addertst.v"

Compiling included source file "header.v"

Continuing compilation of source file "addertst.v"

Compiling source file "bitshift.v"

Compiling source file "overshift.v"

Compiling source file "inverter.v"

Compiling source file "inverter2.v"

Compiling source file "rounder.v"

```
Warning! Text macro (lsb_bit) redefined - replaced with
new definition [Verilog-TMREN]
```

"rounder.v", 27:

Warning! Text macro (guard_bit) redefined - replaced with
new definition [Verilog-TMREN]

"rounder.v", 28:

Warning! Text macro (round_bit) redefined - replaced with
new definition [Verilog-TMREN]

"rounder.v", 29:

Warning! Text macro (sticky_bit) redefined - replaced
with new definition [Verilog-TMREN]

"rounder.v", 30:

Compiling source file "mulclnup.v"

Compiling source file "vcheck.v"

Compiling source file "fpadder.v"

Warning! Text macro (group) redefined - replaced with new
definition [Verilog-TMREN]

"fpadder.v", 11:

Warning! Text macro (imptype) redefined - replaced with
new definition [Verilog-TMREN]

"fpadder.v", 12:

Scanning library directory "/tmp_mnt/local/epoch/models/cmos/verilog"

Highest level modules:

addertst

Type ? for help

C1 > .

***** 2x2=4 *****

0 clock: 0

A sign: x

A mantissa: xxxxxxxxxxxxxxxxxxxxxxxx

A exponent: xxxxxxxx

B sign: x

B mantissa: xxxxxxxxxxxxxxxxxxxxxxxx

B exponent: xxxxxxxx

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: xx Clear: x

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

50 clock: 1

A sign: x

A mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

A exponent: xxxxxxxx

B sign: x

B mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

B exponent: xxxxxxxx

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: xx Clear: x

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

***** Positive X Positive (Normalized) /pos + pos exp *****

100 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

150 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

200 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 11100011

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

250 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 11100011

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

***** Pos x Pos / neg +neg exp *****

300 clock: 0

A sign: 0

A mantissa: 100000000000000000000000

A exponent: 11100111

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

350 clock: 1

A sign: 0

A mantissa: 100000000000000000000000

A exponent: 11100111

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

*****Positive (Norm) X Positive (Unnorm)*****

400 clock: 0

A sign: 0

A mantissa: 0000000000000000000000

A exponent: 00000111

B sign: 0

B mantissa: 0000000000000000000000

B exponent: 00000111

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

450 clock: 1

A sign: 0

A mantissa: 0000000000000000000000

A exponent: 00000111

B sign: 0

B mantissa: 0000000000000000000000

B exponent: 00000111

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

*****Positive X Positive (Unnorm) *****

500 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000001

B sign: 0

B mantissa: 100000000000000000000000

B exponent: 00000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

550 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000001

B sign: 0

B mantissa: 100000000000000000000000

B exponent: 00000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

***** Negative X Positive *****

600 clock: 0

A sign: 0

A mantissa: 1000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 1000000000000000000000

B exponent: 00000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

650 clock: 1

A sign: 0

A mantissa: 1000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 1000000000000000000000

B exponent: 00000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

652 clock: 1

A sign: 0
A mantissa: 1000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 1000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 0000000000000000000000
Res exponent: 10000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Negative X Negative (underflow) *****

700 clock: 0

A sign: 1
A mantissa: 000000100010001000000000
A exponent: 00000011

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000011

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 10000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

750 clock: 1

A sign: 1
A mantissa: 000000100010001000000000
A exponent: 00000011

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000011

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 10000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

752 clock: 1

A sign: 1
A mantissa: 000000100010001000000000
A exponent: 00000011

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000011

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 11100011

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Negative x Positive *****

800 clock: 0

A sign: 1
A mantissa: 000000000000000100010001
A exponent: 00000011

B sign: 1
B mantissa: 000000100010001000000000
B exponent: 00000011

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 11100011

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

850 clock: 1

A sign: 1
A mantissa: 000000000000000100010001
A exponent: 00000011

B sign: 1
B mantissa: 000000100010001000000000
B exponent: 00000011

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 11100011

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

852 clock: 1

A sign: 1
A mantissa: 000000000000000100010001
A exponent: 00000011

B sign: 1
B mantissa: 000000100010001000000000
B exponent: 00000011

Res sign: 0
Res mantissa: 100000000000000000000000
Res exponent: 11100111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** 2x2=4 *****

900 clock: 0

A sign: 0

A mantissa: 000000000010001000000000

A exponent: 10001000

B sign: 1

B mantissa: 000000000000000000000000

B exponent: 00000111

Res sign: 0

Res mantissa: 100000000000000000000000

Res exponent: 11100111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

950 clock: 1

A sign: 0

A mantissa: 000000000010001000000000

A exponent: 10001000

B sign: 1

B mantissa: 000000000000000000000000

B exponent: 00000111

Res sign: 0

Res mantissa: 100000000000000000000000

Res exponent: 11100111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

952 clock: 1

A sign: 0
A mantissa: 000000000010001000000000
A exponent: 10001000

B sign: 1
B mantissa: 000000000000000000000000
B exponent: 00000111

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 00001000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Negative x Negative *****

1000 clock: 0

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 10001000

B sign: 1
B mantissa: 000000000000000000000000
B exponent: 10111100

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 00001000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1050 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 10001000

B sign: 1
B mantissa: 000000000000000000000000
B exponent: 10111100

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 00001000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1052 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 10001000

B sign: 1
B mantissa: 000000000000000000000000
B exponent: 10111100

Res sign: 0
Res mantissa: 100000000000000000000000
Res exponent: 00000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Denor x Denorm *****

1100 clock: 0

A sign: 1
A mantissa: 000000000000000000000000
A exponent: 10111100

B sign: 1
B mantissa: 000000000000000000000000
B exponent: 10111100

Res sign: 0
Res mantissa: 1000000000000000000000
Res exponent: 00000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1150 clock: 1

A sign: 1
A mantissa: 000000000000000000000000
A exponent: 10111100

B sign: 1
B mantissa: 000000000000000000000000
B exponent: 10111100

Res sign: 0
Res mantissa: 100000000000000000000000
Res exponent: 00000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1152 clock: 1

A sign: 1
A mantissa: 000000000000000000000000
A exponent: 10111100

B sign: 1
B mantissa: 000000000000000000000000
B exponent: 10111100

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 00000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Inf x pos *****

1200 clock: 0

A sign: 0

A mantissa: 0010000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 0010000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 0000000000000000000000

Res exponent: 00000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1250 clock: 1

A sign: 0

A mantissa: 0010000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 0010000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 0000000000000000000000

Res exponent: 00000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1252 clock: 1

A sign: 0
A mantissa: 0010000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 0010000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 00000010001000100000000
Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1300 clock: 0

A sign: 0
A mantissa: 0000000000000000000010000
A exponent: 11111111

B sign: 0
B mantissa: 0000000000000000000010000
B exponent: 10000000

Res sign: 1
Res mantissa: 00000010001000100000000
Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1350 clock: 1

A sign: 0
A mantissa: 0000000000000000000010000
A exponent: 11111111

B sign: 0
B mantissa: 0000000000000000000010000

B exponent: 10000000

Res sign: 1

Res mantissa: 000000100010001000000000

Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1352 clock: 1

A sign: 0

A mantissa: 0000000000000000000010000

A exponent: 11111111

B sign: 0

B mantissa: 0000000000000000000010000

B exponent: 10000000

Res sign: 1

Res mantissa: 00000001000100100001000

Res exponent: 00000100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1400 clock: 0

A sign: 0

A mantissa: 10000010001000100010001

A exponent: 11111111

B sign: 0

B mantissa: 00000010001000100010001

B exponent: 10000000

Res sign: 1

Res mantissa: 00000001000100100001000

Res exponent: 00000100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1450 clock: 1

A sign: 0

A mantissa: 10000010001000100010001

A exponent: 11111111

B sign: 0

B mantissa: 00000010001000100010001

B exponent: 10000000

Res sign: 1

Res mantissa: 00000001000100100001000

Res exponent: 00000100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1452 clock: 1

A sign: 0

A mantissa: 10000010001000100010001

A exponent: 11111111

B sign: 0

B mantissa: 00000010001000100010001

B exponent: 10000000

Res sign: 0

Res mantissa: 0000000000010001000000000

Res exponent: 10001000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1500 clock: 0

A sign: 0

A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 000000000010001000000000
Res exponent: 10001000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1550 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 000000000010001000000000
Res exponent: 10001000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1552 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 000000000000000000000001
Res exponent: 10111100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1600 clock: 0

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 000000000000000000000001
Res exponent: 10111100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1650 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 000000000000000000000001
Res exponent: 10111100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1652 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 1

Res mantissa: 000000000000000000000000

Res exponent: 10111101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1700 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 1

Res mantissa: 000000000000000000000000

Res exponent: 10111101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1750 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 000000000000000000000000
Res exponent: 10111101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1752 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 010000000000000000000000
Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1800 clock: 0

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 010000000000000000000000

Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1850 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 010000000000000000000000

Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1852 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 0000000000000000000010000

Res exponent: 11111111

Rounding Mode: 00 Clear: 1

NAN: 1 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1900 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 000000000000000000010000

Res exponent: 11111111

Rounding Mode: 00 Clear: 1

NAN: 1 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1950 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 000000000000000000010000

Res exponent: 11111111

Rounding Mode: 00 Clear: 1

NAN: 1 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1952 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 10000010001000100010001
Res exponent: 11111111

Rounding Mode: 00 Clear: 1

NAN: 1 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

L79 "addertst.v": \$finish at simulation time 2000

6 warnings

13294 simulation events + 2478 accelerated events + 4918 timing check events

CPU time: 0.6 secs to compile + 1.1 secs to link + 1.5 secs in simulation

End of VERILOG-XL 1.8 Sep 18, 1994 01:47:37

APPENDIX F FLOATING POINT MULTIPLIER TEST OUTPUT

Host command: verilog

Command arguments:

```
-f mplier.vc
-y /tmp_mnt/local/epoch/models/cmos/verilog
+libext+.v+
-s
mpltst.v
shiftsel.v
encoder.v
rounder.v
vcheck.v
mansel.v
mulcnp.v
fpmplier.v
-s
```

VERILOG-XL 1.8 log file created Sep 18, 1994 01:43:49

VERILOG-XL 1.8 Sep 18, 1994 01:43:49

```
* Copyright Cadence Design Systems, Inc. 1985, 1988.  *
* All Rights Reserved. Licensed Software.  *
* Confidential and proprietary information which is the *
* property of Cadence Design Systems, Inc.  *
```

```
* For technical assistance please contact the Cadence  *
* Response Center at 1-800-CADENC2 or send email to  *
* crc_customers@cadence.com  *
```

```
* For more information on Cadence's Verilog-XL product  *
* line send email to talkverilog@cadence.com  *
```

Compiling source file "mpltst.v"

Compiling source file "shiftsel.v"

Compiling source file "encoder.v"

Compiling source file "rounder.v"

Compiling source file "vcheck.v"

Compiling source file "mansel.v"

Compiling source file "mulcnp.v"

Compiling source file "fpmplier.v"

Scanning library directory "/tmp_mnt/local/epoch/models/cmos/verilog"

Highest level modules:

mpltst

Type ? for help

C1 > .

***** 2x2=4 *****

0 clock: 0

A sign: x

A mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

A exponent: xxxxxxxx

B sign: x

B mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

B exponent: xxxxxxxx

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: xx Clear: x

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

50 clock: 1

A sign: x

A mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

A exponent: xxxxxxxx

B sign: x

B mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

B exponent: xxxxxxxx

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: xx Clear: x

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

***** Positive X Positive (Normalized) /pos + pos exp *****

100 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

150 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

200 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 11100011

B sign: 0

B mantissa: 000000000000000000000000
B exponent: 10000000

Res sign: x
Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx
Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

250 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 1110011

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 10000000

Res sign: x
Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx
Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

***** Pos x Pos / neg +neg exp *****

300 clock: 0

A sign: 0
A mantissa: 100000000000000000000000
A exponent: 1110011

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx
Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

350 clock: 1

A sign: 0
A mantissa: 1000000000000000000000
A exponent: 11100111

B sign: 0
B mantissa: 0000000000000000000000
B exponent: 10000000

Res sign: x
Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx
Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

*****Positive (Norm) X Positive (Unnorm)*****

400 clock: 0

A sign: 0
A mantissa: 0000000000000000000000
A exponent: 00000111

B sign: 0
B mantissa: 0000000000000000000000
B exponent: 00000111

Res sign: x
Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxx
Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

450 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000111

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000111

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

*****Positive X Positive (Unnorm)*****

500 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10000001

B sign: 0

B mantissa: 100000000000000000000000

B exponent: 00000010

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

550 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10000001

B sign: 0

B mantissa: 100000000000000000000000

B exponent: 00000010

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

***** Negative X Positive *****

600 clock: 0

A sign: 0

A mantissa: 100000000000000000000000

A exponent: 00000010

B sign: 0

B mantissa: 100000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

650 clock: 1

A sign: 0

A mantissa: 100000000000000000000000

A exponent: 00000010

B sign: 0

B mantissa: 1000000000000000000000

B exponent: 10000000

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

***** Negative X Negative (underflow) *****

700 clock: 0

A sign: 1

A mantissa: 00000010001000100000000

A exponent: 00000011

B sign: 0

B mantissa: 00000000000000000000000

B exponent: 10000001

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

750 clock: 1

A sign: 1

A mantissa: 00000010001000100000000

A exponent: 00000011

B sign: 0

B mantissa: 00000000000000000000000

B exponent: 10000001

Res sign: x

Res mantissa: xxxxxxxxxxxxxxxxxxxxxxxxx

Res exponent: xxxxxxxx

Rounding Mode: 00 Clear: 1

NAN: x INF: x OVFL: x UNFL: x INEXACT: x

752 clock: 1

A sign: 1

A mantissa: 00000010001000100000000

A exponent: 00000011

B sign: 0

B mantissa: 00000000000000000000000

B exponent: 10000001

Res sign: 0

Res mantissa: 00000000000000000000000

Res exponent: 10000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Negative x Positive *****

800 clock: 0

A sign: 1

A mantissa: 00000000000000100010001

A exponent: 01000011

B sign: 1

B mantissa: 00000010001000100000000

B exponent: 01000011

Res sign: 0

Res mantissa: 00000000000000000000000

Res exponent: 10000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

850 clock: 1

A sign: 1

A mantissa: 000000000000000100010001

A exponent: 01000011

B sign: 1

B mantissa: 000000100010001000000000

B exponent: 01000011

Res sign: 0

Res mantissa: 000000000000000000000000

Res exponent: 10000001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

852 clock: 1

A sign: 1

A mantissa: 000000000000000100010001

A exponent: 01000011

B sign: 1

B mantissa: 000000100010001000000000

B exponent: 01000011

Res sign: 0

Res mantissa: 000000000000000000000000

Res exponent: 11100100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** 2x2=4 *****

900 clock: 0

A sign: 0

A mantissa: 00000000001000100000000

A exponent: 10001000

B sign: 1

B mantissa: 00000000000000000000000

B exponent: 00000111

Res sign: 0

Res mantissa: 00000000000000000000000

Res exponent: 11100100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

950 clock: 1

A sign: 0

A mantissa: 00000000001000100000000

A exponent: 10001000

B sign: 1

B mantissa: 00000000000000000000000

B exponent: 00000111

Res sign: 0

Res mantissa: 00000000000000000000000

Res exponent: 11100100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

952 clock: 1

A sign: 0

A mantissa: 00000000001000100000000

A exponent: 10001000

B sign: 1

B mantissa: 000000000000000000000000

B exponent: 00000111

Res sign: 0

Res mantissa: 100000000000000000000000

Res exponent: 11101000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Negative x Negative *****

1000 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10001000

B sign: 1

B mantissa: 000000000000000000000000

B exponent: 10111100

Res sign: 0

Res mantissa: 100000000000000000000000

Res exponent: 11101000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1050 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 10001000

B sign: 1

B mantissa: 000000000000000000000000

B exponent: 10111100

Res sign: 0

Res mantissa: 1000000000000000000000

Res exponent: 11101000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1052 clock: 1

A sign: 0

A mantissa: 0000000000000000000000

A exponent: 10001000

B sign: 1

B mantissa: 0000000000000000000000

B exponent: 10111100

Res sign: 0

Res mantissa: 0000000000000000000000

Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 1 INEXACT: 0

***** Denor x Denorm *****

1100 clock: 0

A sign: 1

A mantissa: 0000000000000000000000

A exponent: 10111100

B sign: 1

B mantissa: 0000000000000000000000

B exponent: 10111100

Res sign: 0

Res mantissa: 0000000000000000000000

Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 1 INEXACT: 0

1150 clock: 1

A sign: 1

A mantissa: 000000000000000000000000

A exponent: 10111100

B sign: 1

B mantissa: 000000000000000000000000

B exponent: 10111100

Res sign: 0

Res mantissa: 000000000000000000000000

Res exponent: 00000000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 1 INEXACT: 0

1152 clock: 1

A sign: 1

A mantissa: 000000000000000000000000

A exponent: 10111100

B sign: 1

B mantissa: 000000000000000000000000

B exponent: 10111100

Res sign: 0

Res mantissa: 100000000000000000000000

Res exponent: 00000100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

***** Inf x pos *****

1200 clock: 0

A sign: 0

A mantissa: 0010000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 0000000000000000000000

B exponent: 10001000

Res sign: 0

Res mantissa: 1000000000000000000000

Res exponent: 00000100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1250 clock: 1

A sign: 0

A mantissa: 0010000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 0000000000000000000000

B exponent: 10001000

Res sign: 0

Res mantissa: 1000000000000000000000

Res exponent: 00000100

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1252 clock: 1

A sign: 0

A mantissa: 0010000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 10001000

Res sign: 0

Res mantissa: 001000000000000000000000

Res exponent: 00000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 1 OVFL: 0 UNFL: 0 INEXACT: 1

1300 clock: 0

A sign: 0

A mantissa: 0000000000000000000010000

A exponent: 11111111

B sign: 0

B mantissa: 0000000000000000000010000

B exponent: 10000000

Res sign: 0

Res mantissa: 001000000000000000000000

Res exponent: 00000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 1 OVFL: 0 UNFL: 0 INEXACT: 1

1350 clock: 1

A sign: 0

A mantissa: 0000000000000000000010000

A exponent: 11111111

B sign: 0

B mantissa: 0000000000000000000010000

B exponent: 10000000

Res sign: 0

Res mantissa: 001000000000000000000000

Res exponent: 00000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 1 OVFL: 0 UNFL: 0 INEXACT: 1

1352 clock: 1

A sign: 0

A mantissa: 0000000000000000000010000

A exponent: 11111111

B sign: 0

B mantissa: 0000000000000000000010000

B exponent: 10000000

Res sign: 1

Res mantissa: 000000100010001000000000

Res exponent: 00000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1400 clock: 0

A sign: 0

A mantissa: 10000010001000100010001

A exponent: 11111101

B sign: 0

B mantissa: 00000010001000100010001

B exponent: 10000000

Res sign: 1

Res mantissa: 000000100010001000000000

Res exponent: 00000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1450 clock: 1

A sign: 0

A mantissa: 10000010001000100010001

A exponent: 11111101

B sign: 0

B mantissa: 00000010001000100010001

B exponent: 10000000

Res sign: 1

Res mantissa: 00000010001000100000000

Res exponent: 00000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1452 clock: 1

A sign: 0

A mantissa: 10000010001000100010001

A exponent: 11111101

B sign: 0

B mantissa: 00000010001000100010001

B exponent: 10000000

Res sign: 0

Res mantissa: 00000010001001000010011

Res exponent: 00000111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1500 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 00000010001001000010011
Res exponent: 00000111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1550 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 00000010001001000010011
Res exponent: 00000111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 1

1552 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 000000000010001000000000
Res exponent: 00010000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1600 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 1

Res mantissa: 000000000010001000000000

Res exponent: 00010000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1650 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 1

Res mantissa: 000000000010001000000000

Res exponent: 00010000

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1652 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 000000000000000000000000
Res exponent: 11000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1700 clock: 0

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 1
Res mantissa: 000000000000000000000000
Res exponent: 11000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1750 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 1

Res mantissa: 000000000000000000000000

Res exponent: 11000101

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1752 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 000000000000000000000000

Res exponent: 11111001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1800 clock: 0

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 000000000000000000000000

Res exponent: 11111001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1850 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 000000000000000000000000

Res exponent: 11111001

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1852 clock: 1

A sign: 0

A mantissa: 000000000000000000000000

A exponent: 00000000

B sign: 0

B mantissa: 000000000000000000000000

B exponent: 00000000

Res sign: 0

Res mantissa: 000000000000000000000000

Res exponent: 00000111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1900 clock: 0

A sign: 0

A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 00000111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1950 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 00000111

Rounding Mode: 00 Clear: 1

NAN: 0 INF: 0 OVFL: 0 UNFL: 0 INEXACT: 0

1952 clock: 1

A sign: 0
A mantissa: 000000000000000000000000
A exponent: 00000000

B sign: 0
B mantissa: 000000000000000000000000
B exponent: 00000000

Res sign: 0
Res mantissa: 000000000000000000000000
Res exponent: 11111111

Rounding Mode: 00 Clear: 1

NAN: 1 INF: 0 OVFL: 1 UNFL: 0 INEXACT: 1

L114 "mplertst.v": \$finish at simulation time 2000
13027 simulation events + 2387 accelerated events + 4822 timing check events
CPU time: 0.8 secs to compile + 1.8 secs to link + 1.2 secs in simulation
End of VERILOG-XL 1.8 Sep 18, 1994 01:43:57

APPENDIX G LAYOUT PLATES

Project Manager View Select Placement Package Route Miscellaneous

အစီရင်ခံာ် အကျဉ်းချုပ်

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

fftchip_core (fftchip_core)

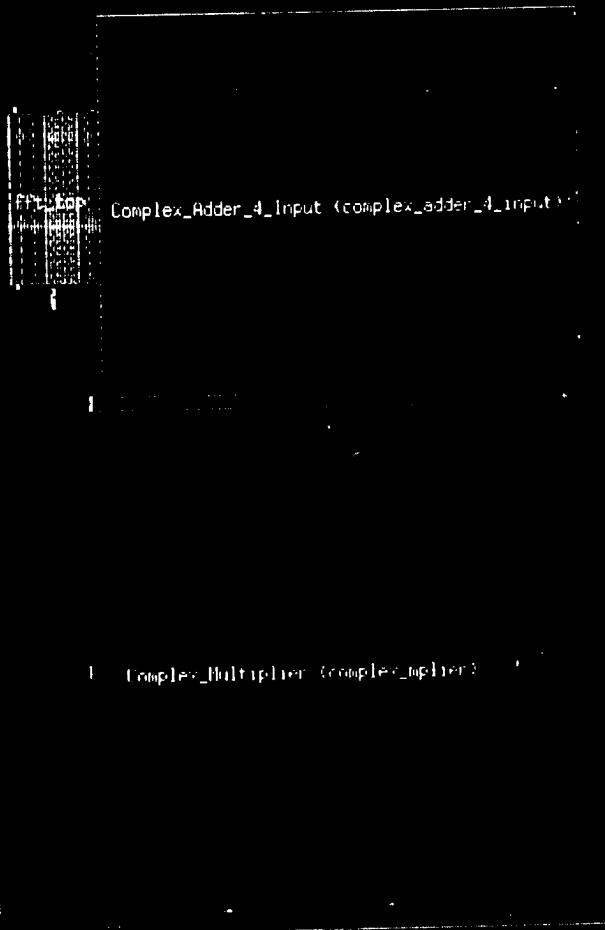
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

[illegible]

Project: /tnp_mnt/h/kepler_u2/kljacksn/projects/thesis2 (CDR1u2n1p) Cell: fftchip (440.553 x 615.835) = 2.71e+05 sq nils

Cascade Design Automation

Project Manager View Select Channel Placemnt Route Miscellaneous



Project: /tmp/mnt/h/kepler_u2/kljackson/projects/thesis2 (CDR1u2n1p) Cell: fft (372.61 x 548.665) = 2.04e+05 sq mils

edit-cell

Cascade Design Automation

Project Manager View Select Channel Placenet Route Miscellaneous

FP_Adder1 (fpadder)

FP_Adder5 (fpadder)

FP_Adder3 (fpadder)

FP_Adder6 (fpadder)

FP_Adder4 (fpadder)

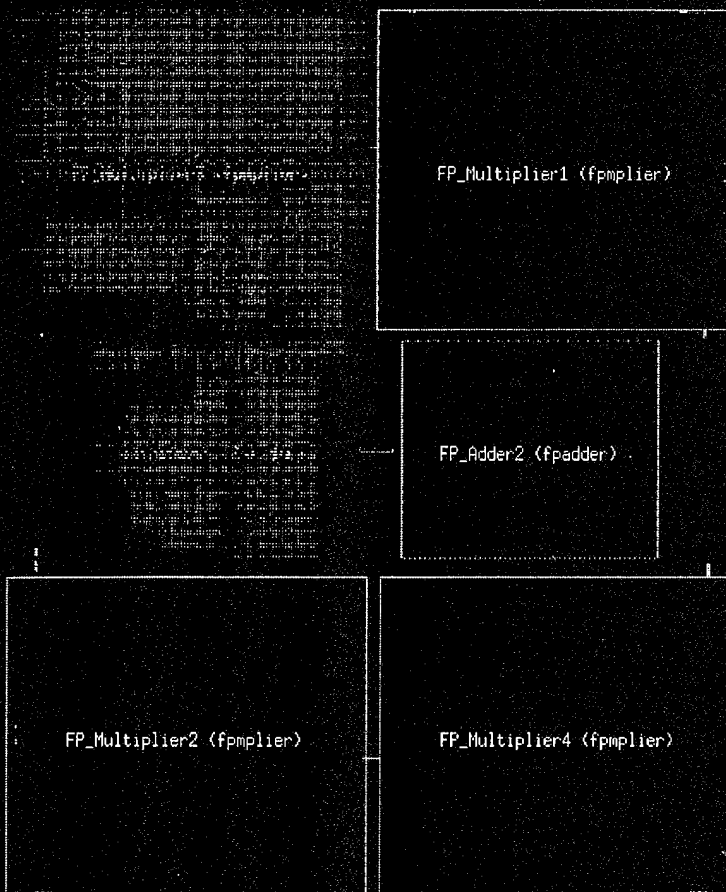
FP_Adder2 (fpadder)

Project: /cnp_rml/h/kepler_u2/kl.jacksn/projects/thesis2 (COR1u2nlp) Cell: complex_adder_4 input (287,486 x 283,744) = 6.65e+04 sq nls

window

Cascade Design Automation

Project Manager View Select Channel Placement Route Miscellaneous



Project: /tmp_wnt/h/kepler_u2/k1.jackso/projects/thesis2 (CDR1u2n1p) Cell: complex_nplier (311,862 x 372,61) = 1.16e+05 sq mils

n1 05.150260 105.290019

Help Project Setup View Select View Network Edit

Mode clock
Mode clock on set with Buffer none of possible none type design 0.00% of Total Delay
Project: /tmp/.../k1jackson/projects/thesis Ruleset: CDRin2xip Design: Fowler

x1 -0.15000 0.00000
view set
window
view all
select clear

TRACTIC > Copyright 1992, 1993 Cascade Design Automation Corporation.

APPENDIX H PART SPECIFICATIONS (SELECTED)

DATA TABLE for part mux4

BUFFER_SIZE = 2

bits = 64

dpath_flag = 0

Ruleset: CDA1u2m1p

Ruleset Version: 3.00

Process Version: 3.00

Process Variation: nominal

Derating Factor: 1.000

Voltage: 5.0 V

Temp: 25 C

Switching Factor: 50%

Power consumption: 984.400uW at 10 MHz: Duty cycle 100%

Height: 655.500u Width: 502.800u Area: 329585.400u²

Delay parameters calculated for loads from 0.216 to 0.541 pF

Load range is determined by module buffer-size of 2

Delay Equation: Output delay = intrinsic + ((ns/pF) * output load)

Inputs

Pin Name	Input Capacitance (fF)
IN0	52
IN1	55
IN2	56
IN3	57
S0	8176
S1	7553

Outputs

Pin Name	Output Capacitance (fF)	Zpu (ohms)	Zpd (ohms)
Y	44	1398	1539

Propagation Delays

	Output Rising		Output Falling		Output delay for 0.000 pf load	
	intrinsic delay (ns)	load dependent delay (ns/pF)	intrinsic delay (ns)	load dependent delay (ns/pF)	Rising (ns)	Falling (ns)
IN0-Y	1.444	1.508	1.174	1.668	1.444	1.174
IN1-Y	1.424	1.508	1.186	1.668	1.424	1.186
IN2-Y	1.418	1.508	1.191	1.668	1.418	1.191
IN3-Y	1.410	1.508	1.192	1.668	1.410	1.192
S0-Y	1.852	1.508	1.559	1.668	1.852	1.559
S1-Y	1.933	1.508	1.273	1.668	1.933	1.273

Attributes

Name	Type	Value	Description

Verilog information

mux4

```

#(N, DPFLAG, GROUP)
(IN0, IN1, IN2, IN3, S0, S1, Y);
N : integer
DPFLAG : integer
GROUP : string
wire IN0[N-1 : 0];
wire IN1[N-1 : 0];
wire IN2[N-1 : 0];
wire IN3[N-1 : 0];
wire S0;
wire S1;
wire Y[N-1 : 0];

```

VHDL information

mux4

```

GENERIC MAP (N, DPFLAG, GROUP)
PORT MAP (IN0, IN1, IN2, IN3, S0, S1, Y);

```

(
N : integer
DPFLAG : integer
GROUP : string
IN0: std_logic_vector(N-1 downto 0);
IN1: std_logic_vector(N-1 downto 0);
IN2: std_logic_vector(N-1 downto 0);
IN3: std_logic_vector(N-1 downto 0);
S0: std_logic;
S1: std_logic;
Y: std_logic_vector(N-1 downto 0);

DATA TABLE for part inc

BUFFER_SIZE = 2

bits = 28

dpath_flag = 0

Ruleset: CDA1u2m1p

Ruleset Version: 3.00

Process Version: 3.00

Process Variation: nominal

Derating Factor: 1.000

Voltage: 5.0 V

Temp: 25 C

Switching Factor: 50%

Power consumption: 2220.175uW at 10 MHz: Duty cycle 100%

Height: 758.200u Width: 768.600u Area: 582752.520u²

Delay parameters calculated for loads from 0.216 to 0.541 pF

Load range is determined by module buffer-size of 2

Delay Equation: Output delay = intrinsic + ((ns/pF) * output load)

Inputs

Pin Name	Input Capacitance (fF)
IN0	208

Outputs

Pin Name	Output Capacitance (fF)	Zpu (ohms)	Zpd (ohms)
Y	50	698	770

Propagation Delays

	Output Rising		Output Falling		Output delay for 0.000 pf load	
	intrinsic delay (ns)	load dependent delay (ns/pF)	intrinsic delay (ns)	load dependent delay (ns/pF)	Rising (ns)	Falling (ns)
IN0-Y	5.773	0.985	5.204	1.055	5.773	5.204

Attributes

Name	Type	Value	Description

Verilog information

inc

```
#(N, DPFLAG, GROUP)
(EN, IN0, TC, TCBAR, Y);
N : integer
DPFLAG : integer
GROUP : string
wire EN;
wire IN0[N-1 : 0];
wire TC;
wire TCBAR;
wire Y[N-1 : 0];
```

VHDL information

inc

```
GENERIC MAP (N, DPFLAG, GROUP)
PORT MAP (EN, IN0, TC, TCBAR, Y);
N : integer
DPFLAG : integer
GROUP : string
EN: std_logic;
IN0: std_logic_vector(N-1 downto 0);
TC: std_logic;
TCBAR: std_logic;
Y: std_logic_vector(N-1 downto 0);
```

DATA TABLE for part barreleft

BUFFER_SIZE = 4

bits = 48

dpath_flag = 0

sels = 6

Ruleset: CDA1u2m1p

Ruleset Version: 3.00

Process Version: 3.00

Process Variation: nominal

Derating Factor: 1.000

Voltage: 5.0 V

Temp: 25 C

Switching Factor: 50%

Power consumption: 2220.300uW at 10 MHz: Duty cycle 100%

Height: 760.300u Width: 764.600u Area: 581325.380u²

Delay parameters calculated for loads from 0.811 to 1.081 pF

Load range is determined by module buffer-size of 4

Delay Equation: Output delay = intrinsic + ((ns/pF) * output load)

Inputs

Pin Name	Input Capacitance (fF)
IN0	206
S	5531

Outputs

Pin Name	Output Capacitance (fF)	Zpu (ohms)	Zpd (ohms)
Y	31	698	770

Propagation Delays

	Output Rising		Output Falling		Output delay for 0.000 pf load	
	intrinsic delay (ns)	load dependent delay (ns/pF)	intrinsic delay (ns)	load dependent delay (ns/pF)	Rising (ns)	Falling (ns)
IN0-Y	5.831	0.781	5.368	0.848	5.831	5.368
S-Y	6.384	0.781	5.474	0.848	6.384	5.474

Attributes

Name	Type	Value	Description

Verilog information

barrelleft

```
 #(N, M, DPFLAG, GROUP)
 (IN0, S, Y);
 N : integer
 M : integer
 DPFLAG : integer
 GROUP : string
 wire IN0[N-1 : 0];
 wire S[M-1 : 0];
 wire Y[N-1 : 0];
```

VHDL information

barrelleft

```
 GENERIC MAP (N, M, DPFLAG, GROUP)
 PORT MAP (IN0, S, Y);
 N : integer
 M : integer
 DPFLAG : integer
 GROUP : string
 IN0: std_logic_vector(N-1 downto 0);
 S: std_logic_vector(M-1 downto 0);
 Y: std_logic_vector(N-1 downto 0);
```

DATA TABLE for part addhs

BUFFER_SIZE = 2

bits = 8

dpath_flag = 1

Ruleset: CDA1u2m1p

Ruleset Version: 3.00

Process Version: 3.00

Process Variation: nominal

Derating Factor: 1.000

Voltage: 5.0 V

Temp: 25 C

Switching Factor: 50%

Power consumption: 816.931uW at 10 MHz: Duty cycle 100%

Height: 324.200u Width: 191.650u Area: 62132.930u²

Delay parameters calculated for loads from 0.216 to 0.541 pF

Load range is determined by module buffer-size of 2

Delay Equation: Output delay = intrinsic + ((ns/pF) * output load)

Inputs

Pin Name	Input Capacitance (fF)
A	112
B	128
CIN	36

Outputs

Pin Name	Output Capacitance (fF)	Zpu (ohms)	Zpd (ohms)
COUT	40	1398	1539
SUM	34	1398	1539

Propagation Delays

	Output Rising		Output Falling		Output delay for 0.000 pf load	
	intrinsic delay (ns)	load dependent delay (ns/pF)	intrinsic delay (ns)	load dependent delay (ns/pF)	Rising (ns)	Falling (ns)
A-COUT	2.328	1.305	2.961	1.514	2.328	2.961
B-COUT	2.372	1.305	3.005	1.514	2.372	3.005
CIN-COUT	2.474	1.305	3.107	1.514	2.474	3.107
A-SUM	3.329	1.311	3.332	1.517	3.329	3.332
B-SUM	3.373	1.311	3.376	1.517	3.373	3.376
CIN-SUM	3.475	1.311	3.478	1.517	3.475	3.478

Attributes

Name	Type	Value	Description

Verilog information

addhs

```
#(N, DPFLAG, GROUP)
(A, B, CIN, COUT, SUM);
N : integer
DPFLAG : integer
GROUP : string
wire A[N-1 : 0];
wire B[N-1 : 0];
wire CIN;
wire COUT;
wire SUM[N-1 : 0];
```

VHDL information

addhs

```
GENERIC MAP (N, DPFLAG, GROUP)
PORT MAP (A, B, CIN, COUT, SUM);
N : integer
DPFLAG : integer
```



```
GROUP : string  
A: std_logic_vector(N-1 downto 0);  
B: std_logic_vector(N-1 downto 0);  
CIN: std_logic;  
COUT: std_logic;  
SUM: std_logic_vector(N-1 downto 0);
```

DATA TABLE for part inc

BUFFER_SIZE = 4

bits = 8

dpath_flag = 0

Ruleset: CDA1u2m1p

Ruleset Version: 3.00

Process Version: 3.00

Process Variation: nominal

Derating Factor: 1.000

Voltage: 5.0 V

Temp: 25 C

Switching Factor: 50%

Power consumption: 207.062uW at 10 MHz: Duty cycle 100%

Height: 149.350u Width: 172.800u Area: 25807.680u²

Delay parameters calculated for loads from 0.811 to 1.081 pF

Load range is determined by module buffer-size of 4

Delay Equation: Output delay = intrinsic + ((ns/pF) * output load)

Inputs

Pin Name	Input Capacitance (fF)
EN	56
IN0	49

Outputs

Pin Name	Output Capacitance (fF)	Zpu (ohms)	Zpd (ohms)
TC	59	698	770
TCBAR	18	2795	3077
Y	33	698	770

Propagation Delays

	Output Rising		Output Falling		Output delay for 0.000 pf load	
	intrinsic delay (ns)	load dependent delay (ns/pF)	intrinsic delay (ns)	load dependent delay (ns/pF)	Rising (ns)	Falling (ns)
EN-TC	5.382	0.652	5.212	0.752	5.382	5.212
IN0-TC	5.367	0.652	5.241	0.752	5.367	5.241
IN0-TCBAR	6.142	2.670	6.187	3.044	6.142	6.187
EN-TCBAR	6.113	2.670	6.202	3.044	6.113	6.202
EN-Y	5.660	0.656	5.818	0.752	5.660	5.818
IN0-Y	5.689	0.656	5.803	0.752	5.689	5.803

Attributes

Name	Type	Value	Description

Verilog information

```
inc
  #(N, DPFLAG, GROUP)
  (EN, IN0, TC, TCBAR, Y);
  N : integer
  DPFLAG : integer
  GROUP : string
  wire EN;
  wire IN0[N-1 : 0];
  wire TC;
  wire TCBAR;
  wire Y[N-1 : 0];
```

VHDL information

```
inc
  GENERIC MAP (N, DPFLAG, GROUP)
  PORT MAP (EN, IN0, TC, TCBAR, Y);
  N : integer
  DPFLAG : integer
```

GROUP : string
EN: std_logic;
IN0: std_logic_vector(N-1 downto 0);
TC: std_logic;
TCBAR: std_logic;
Y: std_logic_vector(N-1 downto 0);

DATA TABLE for part hsmult2piped_c

A_operand_width = 26
BUFFER_SIZE = 6
B_operand_width = 26
C_operand_width = 0
PROD_operand_width = 52

Ruleset: CDA1u2m1p
Ruleset Version: 3.00
Process Version: 3.00
Process Variation: nominal
Derating Factor: 1.000
Voltage: 5.0 V
Temp: 25 C
Switching Factor: 50%
Power consumption: 38323.449uW at 10 MHz: Duty cycle 100%
Height: 1618.700u Width: 3631.350u Area: 5878066.245u²

Delay parameters calculated for loads from 1.351 to 1.621 pF
Load range is determined by module buffer-size of 6
Delay Equation: Output delay = intrinsic + ((ns/pF) * output load)

Inputs

Pin Name	Input Capacitance (fF)
A	1152
B	68
CLK	19996
CLR	27866

Outputs

Pin Name	Output Capacitance (fF)	Zpu (ohms)	Zpd (ohms)
PROD	376	559	616

Propagation Delays

	Output Rising		Output Falling		Output delay for 0.000 pf load	
	intrinsic delay (ns)	load dependent delay (ns/pF)	intrinsic delay (ns)	load dependent delay (ns/pF)	Rising (ns)	Falling (ns)
CLK-PROD	10.893	0.522	10.900	0.604	10.893	10.900

Attributes

Name	Type	Value	Description
hold	int	0.108 ns	data hold time for CLK-D
mpwh	int	0.676 ns	minimum pulse width high for CLK-D
mpwl	int	0.676 ns	minimum pulse width low for CLR-VDD
setup	int	0.676 ns	data setup time for CLK-D

Verilog information

```
hsmult2pipcd_c
#(AA, BB, P)
(A, B, CLK, CLR, PROD);
AA : integer
BB : integer
P : integer
wire A[AA-1 : 0];
wire B[BB-1 : 0];
wire CLK;
wire CLR;
wire PROD[P-1 : 0];
```

VHDL information

```
hsmult2pipcd_c
GENERIC MAP (AA, BB, P)
PORT MAP (A, B, CLK, CLR, PROD);
AA : integer
BB : integer
P : integer
```

```
A: std_logic_vector(AA-1 downto 0);  
B: std_logic_vector(BB-1 downto 0);  
CLK: std_logic;  
CLR: std_logic;  
PROD: std_logic_vector(P-1 downto 0);
```

DATA TABLE for part barrelright

BUFFER_SIZE = 4

bits = 50

dpath_flag = 0

sels = 6

Ruleset: CDA1u2m1p

Ruleset Version: 3.00

Process Version: 3.00

Process Variation: nominal

Derating Factor: 1.000

Voltage: 5.0 V

Temp: 25 C

Switching Factor: 50%

Power consumption: 2312.812uW at 10 MHz: Duty cycle 100%

Height: 816.950u Width: 748.600u Area: 611568.770u²

Delay parameters calculated for loads from 0.811 to 1.081 pF

Load range is determined by module buffer-size of 4

Delay Equation: Output delay = intrinsic + ((ns/pF) * output load)

Inputs

Pin Name	Input Capacitance (fF)
IN0	58
S	5841

Outputs

Pin Name	Output Capacitance (fF)	Zpu (ohms)	Zpd (ohms)
Y	74	698	770

Propagation Delays

	Output Rising		Output Falling		Output delay for 0.000 pf load	
	intrinsic delay (ns)	load dependent delay (ns/pF)	intrinsic delay (ns)	load dependent delay (ns/pF)	Rising (ns)	Falling (ns)
IN0-Y	6.066	0.789	5.602	0.856	6.066	5.602
S-Y	6.583	0.789	5.686	0.856	6.583	5.686

Attributes

Name	Type	Value	Description

Verilog information

barrelright

```
#(N, M, DPFLAG, GROUP)
(IN0, S, Y);
N : integer
M : integer
DPFLAG : integer
GROUP : string
wire IN0[N-1 : 0];
wire S[M-1 : 0];
wire Y[N-1 : 0];
```

VHDL information

barrelright

```
GENERIC MAP (N, M, DPFLAG, GROUP)
PORT MAP (IN0, S, Y);
N : integer
M : integer
DPFLAG : integer
GROUP : string
IN0: std_logic_vector(N-1 downto 0);
S: std_logic_vector(M-1 downto 0);
Y: std_logic_vector(N-1 downto 0);
```

LIST OF REFERENCES

- [1] Michael Lee Zimmer, "A Design of a Radix-4 Floating Point FFT Butterfly," Master's Dissertation. Naval Postgraduate School 1991.
- [2] William A. Brown. and Herschel H. Loomis, Jr., "Digital Implementation of Spectral Correlation Analyzers," *IEEE Transactions on Signal Processing.*, Vol 41, No. 2 February, 1993, p.703 .
- [3] W. A. Gardner, *Statistical Spectral Analysis: A Nonprobabilistic Theory* , Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [4] Randy S. Roberts, William A. Brown, and Herschel H. Loomis, Jr., " Computationally Efficient Algorithms for Cyclic Spectral Analysis," *IEEE SP Magazine*, April 1991.
- [5] Raymond F. Bernstein, Jr. and Herschel H. Loomis, Jr., "Cyclostationary Processing Using Butterfly Machines," 28th Conference on Information Sciences and Systems, Princeton University, March 18, 1994.
- [6] John A. Eldon, " Fourier on a Chip," *VLSI Signal Processing, III*, Robert W. Brodersen and Howard S. Moscovitz, editors. IEEE Press: New York, NY 1988, p. 15.
- [7] *Verilog-XL Reference Manual*, Volume 1, Version 1.6, Cadence Design Systems Inc., March 1991.
- [8] *Epoch User's Manual*, Cascade Design Automation, Bellevue, WA, January 10, 1994.
- [9] *MC68881/MC68882 User's Manual*, Motorola, Inc, Prentis-Hall:Englewood Cliffs, NJ.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CS 93943-5121	1
4.	Professor H. H. Loomis Jr., Code EC/Lm Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	2
5.	Professor Ray Bernstein, Code EC/Be Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CS 93943-5121	1
6.	Professor Douglas J. Fouts, Code EC/Fs Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CS 93943-5121	1
7.	Curricular Office, Code 3A Naval Postgraduate School Monterey, CA 93943-5000	1
8.	Capt. R. Fisher Naval Research Laboratory Code 9110 4555 Overlook Ave. SW Washington, DC 20375	1
9.	Director, Navy Space Systems Division Space and Electronic Warfare Directorate ATTN: Code N63C Chief of Naval Operations (N63) 2000 Navy Pentagon Room 4C671 Washington, DC 20350-2000	1

10.	Hiram Cooke Computer Center, Code 051 Naval Postgraduate School Monterey, CA 93943-2000	1
11.	Kevin L. Jackson 9838 Arrowood Dr. Manassas, VA 22111	2