Rapid Prototyping
using
Field Programmable Gate Array (FPGA)
and
Field Programmable Interconnect Devices (FPID)

by 53

Hung Nguyen

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

ISBN 0-612-13403-2

Name ___

Dissertation Abstracts International and *Masters Abstracts International* are arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation or thesis. Enter the corresponding four-digit code in the spaces provided.

ELECTRONIC AND ELECTRICAL
SUBJECT TERM

0524
SUBJECT CODE

UMI

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

RAPID PROTOTYPING USING FIELD PROGRAMMABLE GATE ARRAY (FPGA) AND FIELD PROGRAMMABLE INTERCONNECT DEVICES (FPID)

BY

HUNG NGUYEN

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

I here by declare that I am the sole author of this thesis.

I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Manitoba to reproduce this thesis by photocopying or other means, in whole or in part, at the request of other institutions or individuals for the purpose of scholarly research.

ABSTRACT

Field Programmable Interconnect Devices (FPIDs) focus on design verification solutions to enable user programmable circuit boards for system and ASIC prototyping. Applications for the FPIDs address a broad spectrum of interconnect needs; including factory reconfigurable PCBs for flexible manufacturing, in-system reconfigurable PCBs for multi-function designs and fault tolerant systems, application configurable processors, alterable test interfaces, as well as programmable connector and switching matrix applications. FPIDs are also effective for ASIC prototyping when used with Field Programmable Gate Arrays (FPGA) to minimized the time , risk and cost associated with such designs. The designer enters the ASIC design, partitions and maps the logic to multiple FPGAs via external software. The FPGA I/Os are then programmably interconnected by one or more multiple FPIDs. Since FPIDs provide all pin to pin interconnects, the I/Os of the FPGAs can be left unassigned allowing for optimized routing performance. In addition to the logic in FPGAs, ASIC prototypes should also accommodate standard components like memories, processors and peripherals. ASIC or system designs using FPIDs and FPGAs provide designers with many advantages such as; instant prototypes from schematic, quick changes for experimentation, full signals observability and automated debug, early system and software integration, and faster design verification.

This thesis describes the development of an ASIC prototyping environment exploiting the attributes of reprogrammable devices such as FPGAs and FPIDs. In addition, the prototyping environment accomodates a wide variety of standard components through the programmable interconnect devices. The system is capable of supporting ASIC prototyping of up to 50,000 gates as well as embeded system prototyping.

ACKNOWLEDGEMENTS

I would like to express my thanks to my advisor Professor Bob McLeod, whose continual enthusiasm, wealth of inspirations, and direction were invaluable in the completion of this thesis. I would also like to thank my friends and colleagues who have been a valuable source of information and assistance throughout my graduate program. Most notably, Doug Martens, Derek Ross, Alex McIlraith, Dave Fletcher, and Mr. Brad Brown. I would like to thank IDers Incorporate for allowing me to develope this thesis into a commercial product.

TABLE OF CONTENTS

CHAPTER 6

CHAPTER 7

CHAPTER 8

CHAPTER 9

REFERENCES

GLOSSARY

APPENDIX A : Interconnection Cases.

APPENDIX B : The Prototyping System Schematic Capture.

APPENDIX C : Hardcopy of Performance Test Result and Schematic Capture of Walking Ones and Zeros with the Output Files of Routing Software.

LIST OF FIGURES

CHAPTER 1

1.0 Introduction

Today, design verification is the largest bottleneck in the system design process. Until all components of the system including hardware, boards and or ASIC chips have been physically integrated together, it cannot be said that the system has been truly verified. Reaching the integration point earlier in the design cycle not only finds any major problems while there is time to fix them, but also accelerates software development. In general, the software development could proceed faster, if a hardware or ASIC prototype was available earlier in the development cycle. In most cases, the availability of working hardware prototypes (either board level or ASIC), is the limiting factor.

The system proposed and developed here makes use of FPID and FPGA devices to implement a rapid prototyping environment. This system represents an excellent opportunity to assess the role of FPGA and FPID technologies in the rapid prototyping stage of system development.

1.1 Prototyping Systems Background

FPGA-based logic emulation systems have been developed for design ranging in complexity from several thousand to several million gates. Quickturn System Inc. has developed emulation systems which interconnect FPGAs in a two-dimensional mesh and, more recently, in a partial crossbar topology. The RPM Emulation System from Quickturn uses FPGAs as the prototype technology. The FPGAs are hard-wired together on large printed circuit boards called Emulation Modules with a printed-wiring scheme that connects each FPGA to all of its neighbour FPGAs. When additional connections are required to implement the ASIC netlist, programmed interconnect within FPGAs are used to supplement the hard-wired connections [1]. A basic version sells for $125,000, with a capacity of 25,000 gates [2].

The AXB-GP2 system is a General Purpose Field Programmable Circuit Board (FPCB) designed by Aptix Corporation [3] using two Field Programmable Interconnect Components (FPIC) to provide component interconnect and diagnostic observability for FPCB. The GP2 provides the component interconnect area with over 1700 through-holes for component attachment.

The interconnect architecture of the GP2 is divided into two regions, each with centralized 32x32 pad array for mounting an FPIC device. All component pins within a given region are pre-routed to individual pads within the array. Once mounted, the FPIC device can be programmed to provide pin-to-pin interconnect of any components within its region. Each region of the GP2 has approximately 785 I/O pins available for component interconnect, a total of 1570 I/Os for the two regions on the board. The through hole pattern on the GP2 is arranged with repeated rows spaced at 100 mils, 300 mils, 100 mils, 300 mils, and so on. This pattern supports direct insertion of standard 300 mils DIP components. Packages such as PGAs, QFPs, PLCCs and others can be mounted on the GP2 through the use of component adapters available from various manufacturers. This system provides a reasonable interconnect for 300 mils DIP components. If other component packages are used the component adapter is required. The component adapter is expensive and soldering the component to the component adapter is time consuming.

The Institut National Polytechnique de Grenoble in France [4] also developed a rapid prototyping system using the Xilinx 4010 FPGAs for logic designs and Actel 1280 FPGAs for interconnect the 4010 FPGAs. The problem with this system is that the Actel FPGAs are one time programmable device, if there is an error in the interconnection netlist a new set of Actel FPGAs need to be programmed. This system could be expensive if the designer needs to do more than one iteration.

1.2 Initial Design of The Proposed Prototyping System

In January 1994, the idea of a more flexible rapid prototyping system was introduced when I-Cube Inc. came out with the Field Programmable Interconnect Device (FPID) IQ160. IQ160 device provides 160 I/Os for interconnect applications. At the time the IQ160 has the highest I/O count FPID available. The idea of the initial system was to utilized the FPID and FPGA technology to provide a rapid prototyping system that was flexible and relatively inexpensive.

The initial prototyping system was designed using a wire-wrap board. The schematic capture was done in Orcad, while the board was manually hand wire-wrapped. The initial prototyping system was done in a wire-wrap version is to provide a fast turn around for system trouble

shooting and software integration. The purpose of the initial prototyping system was to learn how the FPID devices functioned.

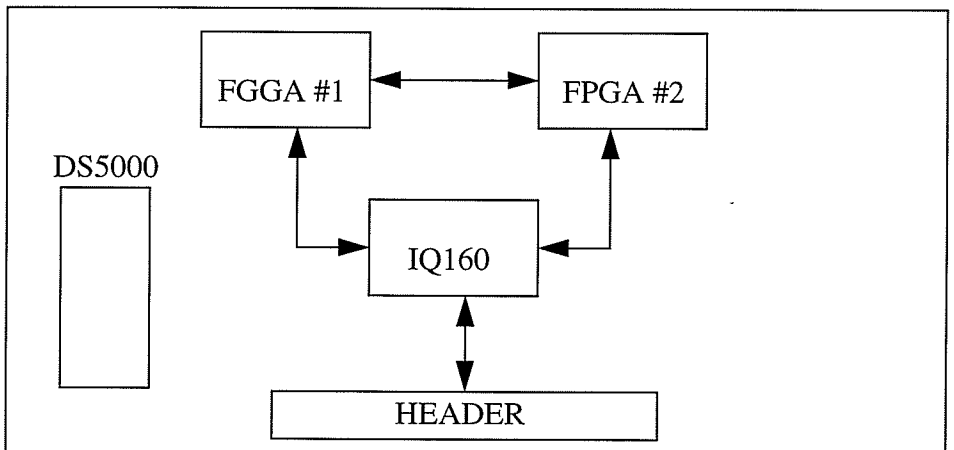


Figure 1: The Architecture of The Initial Prototyping System

The architecture of the initial prototyping system used XC3090 FPGA and IQ160 FPID devices as the main components for the prototyping system and the DS5000 micro-controller used to download configuration bit files to both the FPGAs and FPIDs. Figure 1 shows the architecture of the initial prototyping system. The initial system provided an adequate routing path between the FPGAs and the Target System which was interfaced to the header through the FPID device. Problems arose when the design used more than 40% of the logic capacity of the FPGAs. Test designs with specific pin assignments to interconnect the two FPGAs running the Xilinx place and route software could not complete . While the same test design running the Xilinx place and route software without specific pin assignments completed successfully. With this architecture the performance density of the FPGAs cannot be maximized and the designer has to manually edit the pin assignment to interface between the FPGA#1 and FPGA#2. This process is very time consuming and error prone. The initial architecture of the prototyping system proved to be not suitable for a rapid prototyping system.

A second version of the prototyping system came out of this process. Figure 2 shows the architecture of the second version of the prototyping system. The second version used XC4003-84 FPGA and IQ160 FPID devices. This system was also done through a wire-wrap board. It

connects all of the I/Os of both FPGAs to the FPID allowing the FPID to interconnect all of the pins of the two FPGAs. The second version system provided a better result than the previous version. The utilization of the FPGAs went up to 90% when running Xilinx place and route software when there was no specific pin assignment. Interconnect software was written to translate both the FPGAs and the Target System to the I-Cube netlist.

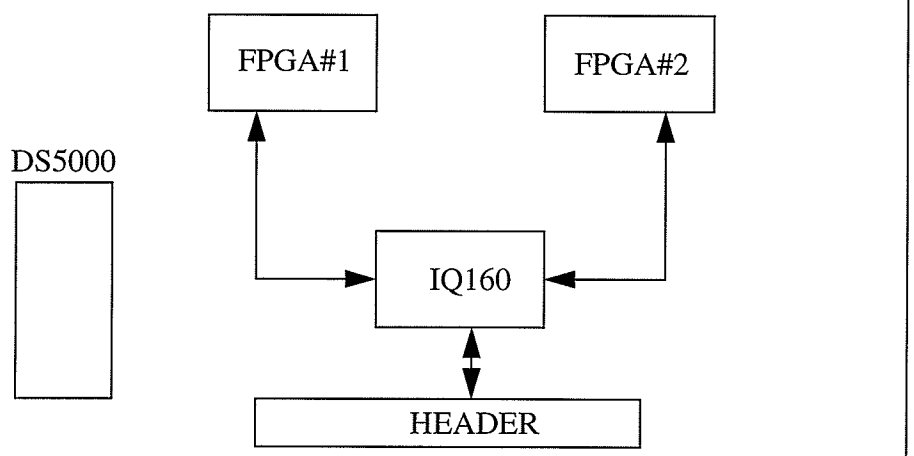


Figure 2: The 2nd Version of The Prototyping System

The second version of the prototyping system can be consider as a rapid prototyping environment. It provided the designer with an automated process from the FPGA design netlist to the interconnection of the FPGAs to the target system. The advantage of this system is that it provides the designer a smooth design flow and a fairly easy routing software. The problem with this system is that the gate density and the I/Os interfaced to the target system are very limited. In order to emulate a fairly complex design (ASIC) we need more gates and I/Os interfaced to the target system.

The third and the final version of the prototyping system has a different hardware architecture to provide the designer with more gates and I/Os interfaced to the target system. The hardware and software architecture will be explained in more detail in the following sections.

CHAPTER 2

2.0 I-Cube FPID (IQ320) Description

The IQ320s are designed for use in programmable switching, interfacing, and wiring applications [5]. In switching applications, these devices are used to dynamically switch or multiplex a large number of signals. When used in interfacing applications, FPID devices allow a common board or system to satisfy different interfacing requirements such as different bus types, communication protocols or multiple standards. In wiring applications, these devices emulate a trace on a PCB and can be used to change point-to-point connections between components on a board or signals on the backplane.

At the heart of these devices is a non-blocking , globally-connected crossbar switch, allowing total flexibility in routing signals. Every signal in the crossbar switch can be connected to one or more other signals. Each I/O port is identical and can be programmed as an input, output, or bidirectional port. The FPIDs support either flow-through or clock signal flow. The delays through the devices are identical and predictable, thereby simplifying hardware design.

The crossbar array connections are programmed and the I/O port attributes are configured by storing data in the internal SRAM cells and registers. These devices permit in-system configuration, thereby making them suitable for applications that require static or dynamic (on the fly) reconfiguration. Also these devices offer two different configuration modes; the JTAG-based serial mode for static connections, and a Rapid Connect mode for fast configuration changes. The crossbar connections can be programmed incrementally in either mode.

2.1 I-Cube FPID Architecture

The I-Cube FPID devices are configured by storing appropriate data in internal static RAM and various registers. As shown in Figure 3, the main functional blocks of the device are the crossbar switch array, I/O ports, and configuration controller.

The external signals pass through the I/O ports. The crossbar switch array is used inter-

nally to connect these I/O ports to one another, thereby establishing connections between the external signals.

The JTAG-based configuration controller decodes the incoming configuration bitstream and stores the data in the internal SRAM and registers to establish the desired configuration. Additionally, by switching the device to Rapid Connect mode, one can directly access the internal crossbar SRAM to change (make or break) crossbar connections incrementally.

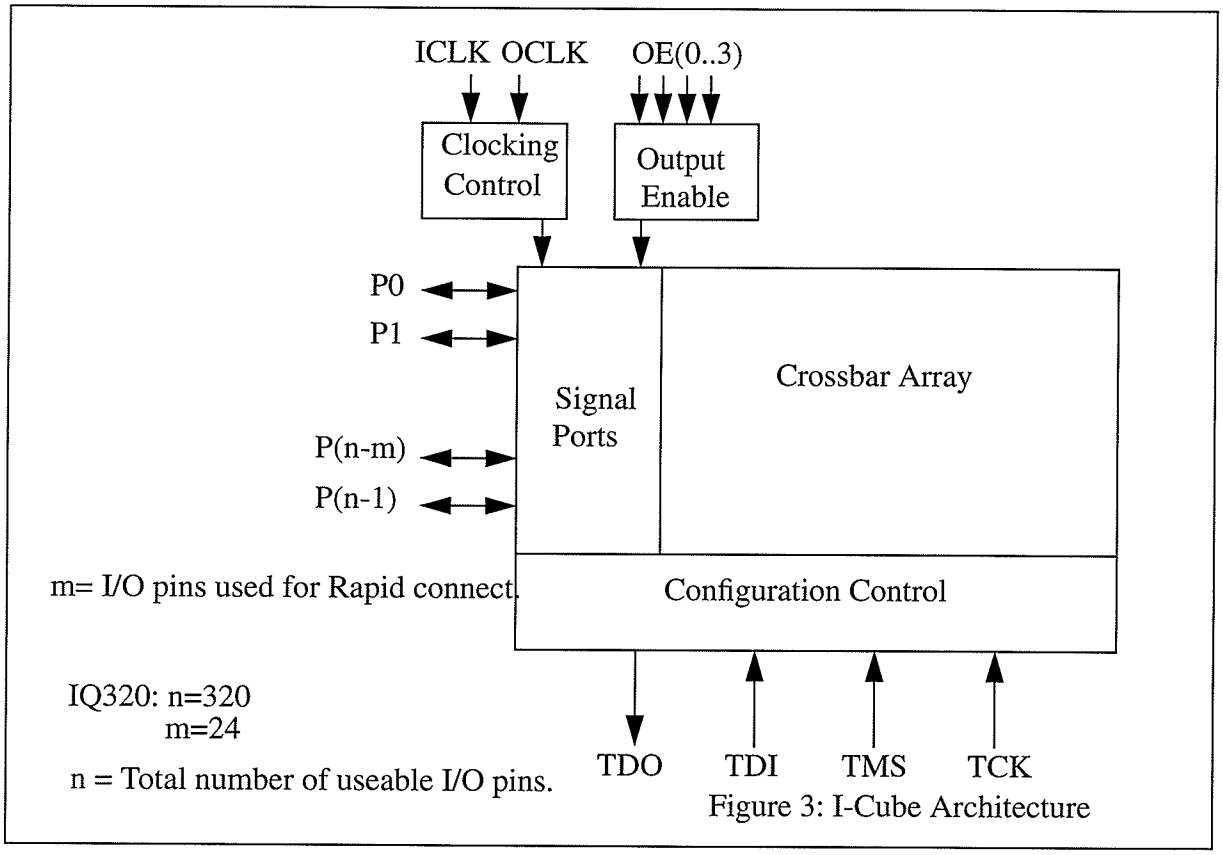


Figure 3: I-Cube Architecture

2.2 Crossbar Array

The crossbar array is an array of pass transistor switches, each programmable with an SRAM cell. Each switch, when programmed to be in the ON state, connects a unique pair of signal lines in the array. The external I/O signals are connected to the lines in the crossbar array.

A connection between two I/O ports is made by closing the transistor switch at the intersection of the crossbar signal lines. The array is globally connected, and therefore a connection

can always be made between any two I/O ports. More over only one transistor switch needs to be closed in order to make a connection between any two I/O ports. This FPID provides a 10 ns Port-to-Port delay in a flow through mode.

2.3 I/O Ports

The attributes of each I/O port is individually programmable. The attributes include its I/O function, output voltage level and pull-up current. Each I/O port is buffered to provide low capacitive loading (in input mode) and low impedance and high current drive (in output mode). The I/O buffer is tristateable using a predefined Output Enable control signal. There are 4 tristate control signals, each controlling an equal number of I/O ports; 80 I/O ports each in the case of the IQ320.

There are 9 different types of I/O attributes.

- **Input (IN):** In this mode, the external signal at I/O port pin is connected to the corresponding crossbar line through a buffer.
- **Register Input (RI):** In this mode, the external signal at the I/O port pin is connected to the corresponding crossbar line through a register. The register is controlled by an external clock signal, ICLK.
- **Output (OP):** In this mode, the corresponding crossbar line is connected to the I/O port pin through a buffer.
- **Register Output (RO):** In this mode, the corresponding crossbar line is connected to the I/O port pin through a register. The register is controlled by an external clock signal, OCLK.
- **Output Force 0 (F0):** In this mode, the I/O port pin is forced low (logic 0), regardless of the state of the signal on the corresponding crossbar line.
- **Output Force 1 (F1):** In this mode, the I/O port pin is forced high (logic 1), regardless of the state of the signal on the corresponding crossbar line.
- **Bus Repeater (BR):** This mode is used for connecting two external signals that are bidirectional but the direction control signal for them is not available. When in this mode, the device automatically detects the driving ports and passes the signal to the receiving ports.

•**Non-Buffer (NB):** In this mode, the I/O buffer is bypassed and the I/O port pin is directly connected to the corresponding crossbar line. This mode can be used to pass an analog signal if certain conditions are met.

•**No Connect (NC):** In this mode, the I/O port pin is isolated from the crossbar array.

When the I/O port is programmed to be in one of the output modes, the output high voltage level can be programmed as TTL or CMOS. For those I/Os programmed for CMOS output levels, an additional pull-up current of 2mA or 13.5mA can be programmed. The normal pull-up current is 8mA.

2.4 Configuration Control

The FPID devices are programmed using the JTAG (IEEE 1149.1) serial bus. The JTAG serial bus uses four pins : Test Data In (TDI), Test Data Out (TDO), Test Clock (TCK), and Test Mode Select (TMS). TCK is used to clock data in and out of TDI and TDO. TMS, in conjunction with TDI implement the state machine that controls the various operations of the JTAG protocol. In addition the reset signal TRST is used to reset the FPID devices. Figure 4 illustrates the JTAG timing diagram, the data is latched on the rising edge of the TCK.

For the FPID device, the I/O attributes and the crossbar connections can be programmed using the JTAG serial bus. Additionally, a special mode call Rapid Connect can be enabled or disabled using the JTAG serial bus.

The Rapid Connect mode is used for real time switching applications where crossbar connections need to be altered dynamically within the user's system. In this mode, a designated number of I/O ports are used to directly address the internal crossbar SRAM for writing, allowing the contents of the SRAM to be altered very quickly, resulting in fast connections changes.

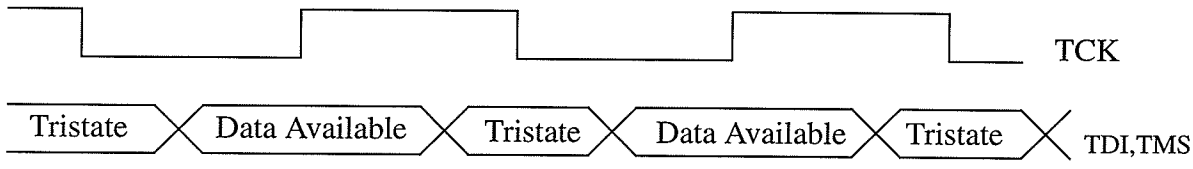


Figure 4: JTAG Timing

CHAPTER 3

3.0 Xilinx FPGA Architecture

The Xilinx XC4013 FPGA is a high density user programmable gate array comprised of three major configurable elements [6]; Configurable Logic Blocks (CLB), Input/Ouput Blocks (IOB), and Interconnect. The CLBs provide the functional elements for constructing the user logic. The IOBs provide the interface between the package pins and the internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs into appropriate networks. Customized configuration is established by programming internal static memory cells that determine the logic functions and interconnections implemented in the Logic Cell Array (LCA) .

3.1 FPGA Configurable Logic Blocks (CLB)

The principal CLB elements are shown in Figure 5. Each CLB contains a pair of flip flops and two independent 4-input function generators. Thirteen CLB inputs and four CLB outputs provide access to the function generators and flip flops. These inputs and outputs are connected to the programmable interconnect resources outside the block. Four independent inputs are provided to each of the two function generators (F1-F4 and G1-G4). These function generators, whose outputs are labeled F' and G', are each capable of implementing any arbitrarily defined Boolean function of their four inputs. A third function generator, labelled H', can implement any Boolean function of its three inputs: F', G' and a third input from outside the block H1. Signals from the function generators can exit the CLB on two outputs; F' or H' can be connected to the X output, and G' or H' can be connected to the Y output. Thus, a CLB can be used to implement any two independent functions of up-to-four variables, or any single function of five variables, or any function of four variables together with some functions of five variables, or it can implement some functions of up to nine variables. Implementing wide functions in a single block reduces both the number of blocks required and the delay in the signal path.

The two storage elements in the CLB are edge-triggered D-type flip flops with common clock(K) and clock enable(EC) inputs. A third common input(S/R) can be programmed as either

an asynchronous set or reset signal independently for each of the two registers

Each flip flop can be triggered on either the rising or falling clock edge. The source of a flip flop data input is programmable: it is driven either by the functions F', G', and H' or the Direct In (DIN) block input. The flip flops drive the XQ and YQ CLB outputs.

Multiplexers in the CLB map the four control inputs, labelled C1-C4 into four internal control signals (H1, DIN, S/R, and EC) in any arbitrary manner.

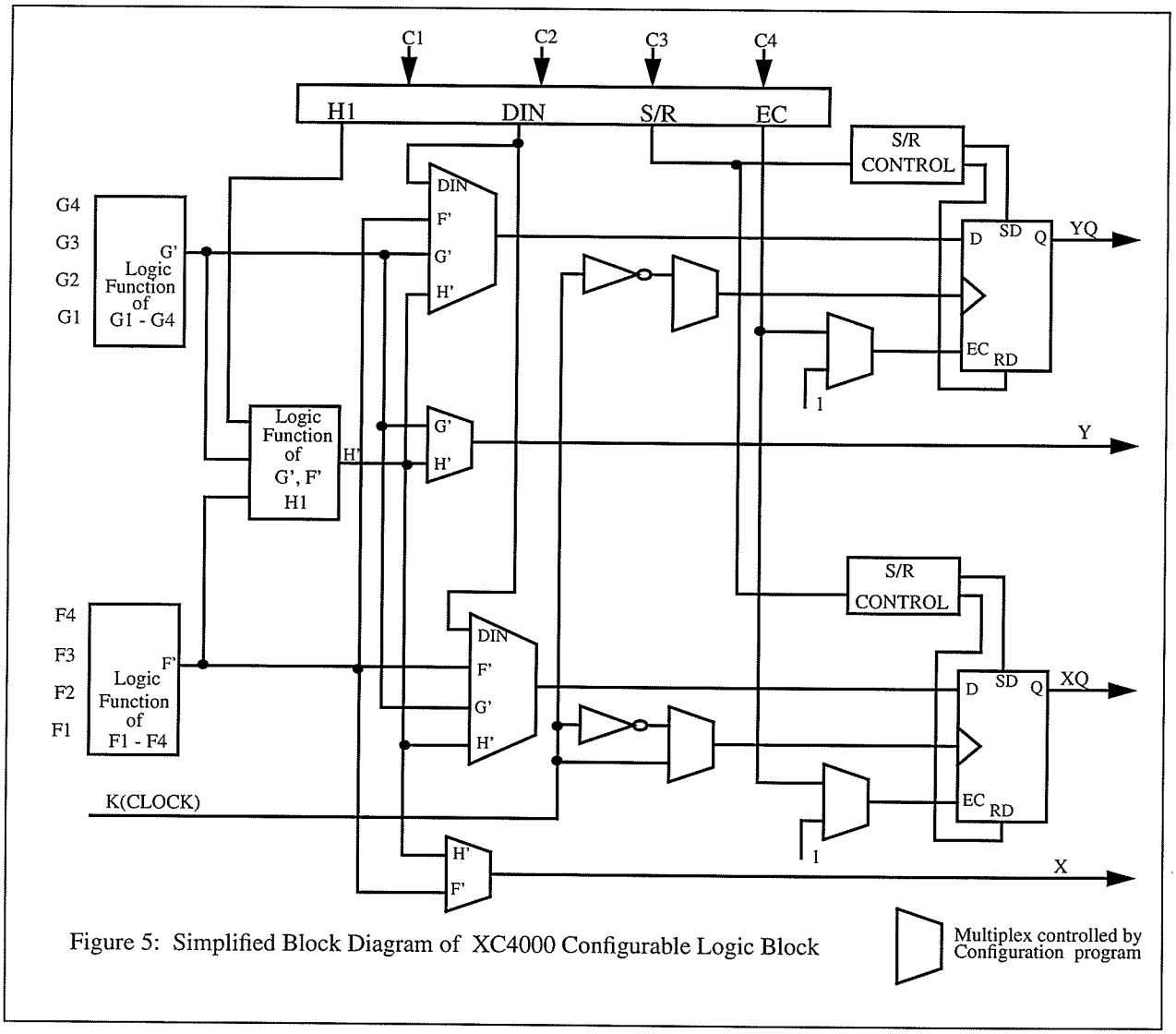


Figure 5: Simplified Block Diagram of XC4000 Configurable Logic Block

3.2 Input/Output Blocks (IOB)

User-configurable IOBs provide the interface between external package pins and the internal logic as shown in Figure 6. Each IOB controls one package pin and can be defined for input, output, or bidirectional signals.

Two paths, labelled I1 and I2, bring input signals into the array. Inputs are routed to an input register that can be programmed as either an edged-triggered flip flop or level sensitive transparent latch.

Outputs signals can be inverted or not inverted, and can be pass directly to the pad or be stored in an edge-triggered flip flop. Optionally an output enable signal can be used to place the output buffer in a high-impedance state, implementing 3-state outputs or bidirectional I/O. There are a number of other programmable options in the IOB. Programmable pull-up and pull-down resistors are useful for tying unused pins to power or ground to minimize power consumption.

3.3 Programmable Interconnect

There are three main types of interconnect, distinguished by the relative length of their segments: single-length lines, double-length lines, and long-length lines. The single-length lines are a grid of horizontal and vertical lines that intersect at a Switching Matrix between each block. The single-length interconnect lines surround each CLB in the array. Each Switching Matrix consists of programmable n-channel pass transistors used to establish connections between the single length lines. Single-length lines are normally used to conduct signals within a localized area and to provide the branching for nets with fanout greater than one.

Double-length lines consists of a grid of metal segments twice as long as the single-length lines. A double-length line runs past two CLBs before entering a Switching Matrix. Double-length lines are grouped in pairs through a Switch Matrix at every other CLB location in that row or column. Double-length line provides the most efficient implementation of intermediate length, point-to-point interconnection.

Long-length lines form a grid of metal interconnect segments that run the entire length or width of the array. Additional vertical longlines can be driven by special global buffers, designed

to distribute clocks and other high fanout control signals throughout the array with minimal skew. Longlines are intended for high fan-out, time-critical signal nets.

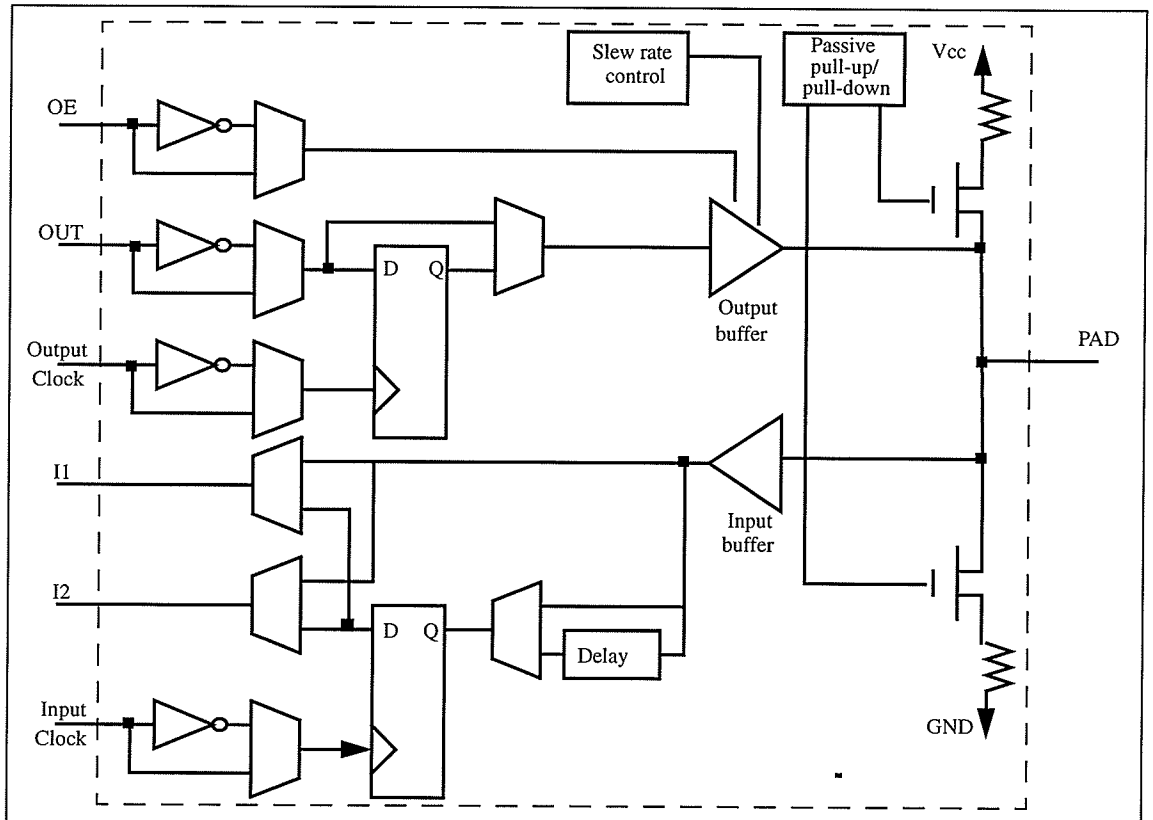


Figure 6: Input/Output Blocks

CHAPTER 4

4.0 Features of the Rapid Prototyping System

The Rapid Prototyping System was designed to facilitate ASIC emulation . Every ASIC design methodology includes a system validation phase. This is a phase of the design cycle where all of the hardware and software are brought together for the first time. Since it is unlikely that the entire system will function as expected the first time it is critical that the system designers have the flexibility to modify the ASIC designs during this phase without incurring high costs in time or tooling expenses. If the ASIC design has already been committed to mask-based silicon at this point, design changes will lead to an extended development schedule and associated cost [10].

The only other real alternative to silicon prototypes for system validation is to build ASIC breadboards prototypes in the traditional way. However breadboards for today's major design projects are no longer economical nor practical although they represent a well established design validation technique.

ASIC in-circuit emulation is a new prototyping alternative for ASIC system validation. It is made possible through the use of FPGA and FPID technologies. This approach gives the designer the capability of automatically creating a functional image of the chip in reprogrammable hardware. The Rapid Prototyping System developed here allows a designer to create a prototype rapidly and automatically. This hardware prototype can then be utilized to fully validate the functional design of the user's logic.

The Rapid Prototyping System runs under a PC environment. It combines both FPGA and FPID development tools ; Xilinx Development System for FPGA design and I-Cube Development System for FPID routing . This system provides a designer the following features as an emulating tool:

- Emulation of ASIC designs realized utilizing FPGA and FPID technologies.
- FPIDs provide all pin-to-pin interconnects leaving FPGA I/O unassigned for optimized performance.

•Interconnects between the target system and the prototype design (ASIC) are fully automated.

•Provides a designer with up to 256 I/Os channels for interfacing to the target system, or to a logic analyzer for monitoring waveforms.

•Full control of system clock during emulation.

•System clock stepping can be varied from 1 to 65,000 cycles, or used as a continuous clock.

•Modification of interconnects can be done on the fly.

•Gate emulation ranges from 26,000 to 50,000 gates.

•Provides a designer an XCHECKER capability, for use with the Xilinx Development System (optional).

4.1 Rapid Prototyping System Interface

The prototyping system interface consists of two main elements; the PC environment and Rapid Prototyping Board (RPB). Figure 7 shows the interface between the RPB, PC environment, and the Target System.

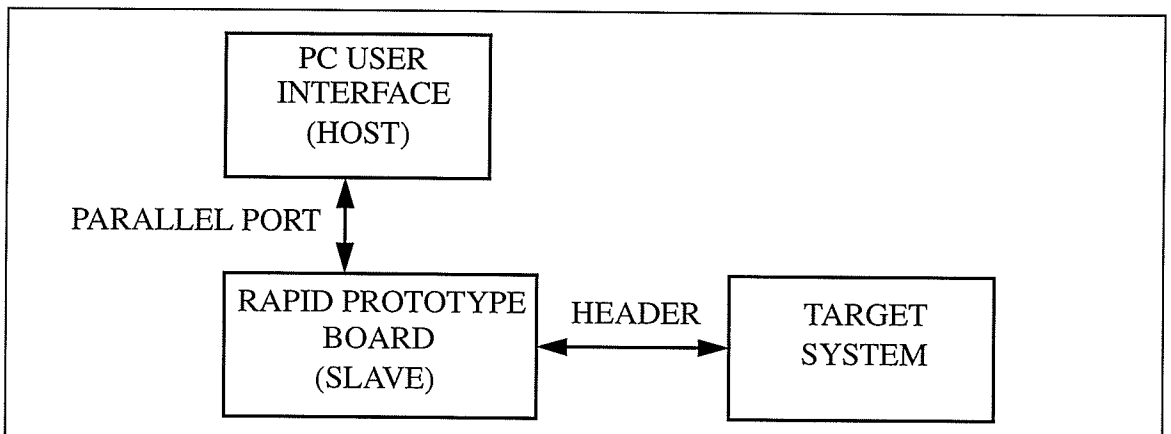


Figure 7: Rapid Prototype System Interface

In this system the PC serves as a user interface and a host to the RPB which acts as a slave. The RPB only executes upon a request from the host. All of the communication from the

PC to the RPB is done through the parallel port. The commands are sent from the host to the slave. Once a slave has received a command it will execute the requested command and send the result back to the host.

The role of the host is to analyze and display the results obtained from the slave. The PC environment is where a designer creates ASIC designs using a schematic capture tool, VHDL design tools, or other netlist creation tools. Another function of the host is to compile the netlists created by ASIC design tools and the Target System to determine the interconnection from the ASIC design to the Target System. Once the host has completed determining the interconnects between the ASIC design and the Target System, it then generates the configuration bit file for downloading to the slave. When the system is in operation, the host can be used to modify the connections between an ASIC design and the Target System without having to undergo a full system reset.

The RPB on the other hand acts as a slave to the host. This is where ASIC design emulation takes place. The ASIC design is implemented in the FPGAs while all of the physical connections are done through the FPID devices. The headers provide 256 general purpose I/Os for interfacing to the Target System. The functionality of the RPB will be discussed in more detail in the next section.

The Target System can be any generic system. Its interfacing to the Rapid Prototype Board is through the I/Os from the headers. Usually the Target System is the system that interfaces to an ASIC to produce a final product. For example a designer is designing an VME bus controller. The controller functionality is implemented in the RPB with all the necessary I/Os routed out to the I/Os in the header. The Target System may contain microprocessors, memories, and other necessary peripheral components with I/Os interfaced to the ASIC design routed to designated I/Os on the header. Once this process is completed, a compiler will determine the interconnections and generate an appropriate configuration bit file for down loading to the FPID devices. The compiler will be discussed in more detail in section 6.

Figure 8 shows the FPD component side and Figure 9 shows the micro-controller side of the prototyping board. The schematics for the prototyping board are shown in Appendix B.

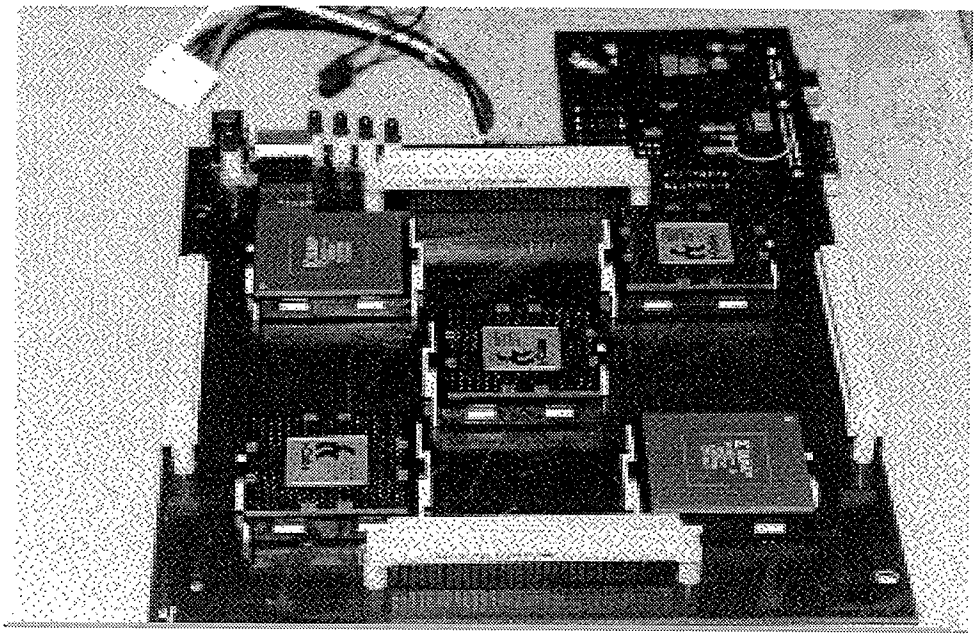


Figure 8: FPD Component Side of The Prototyping Board

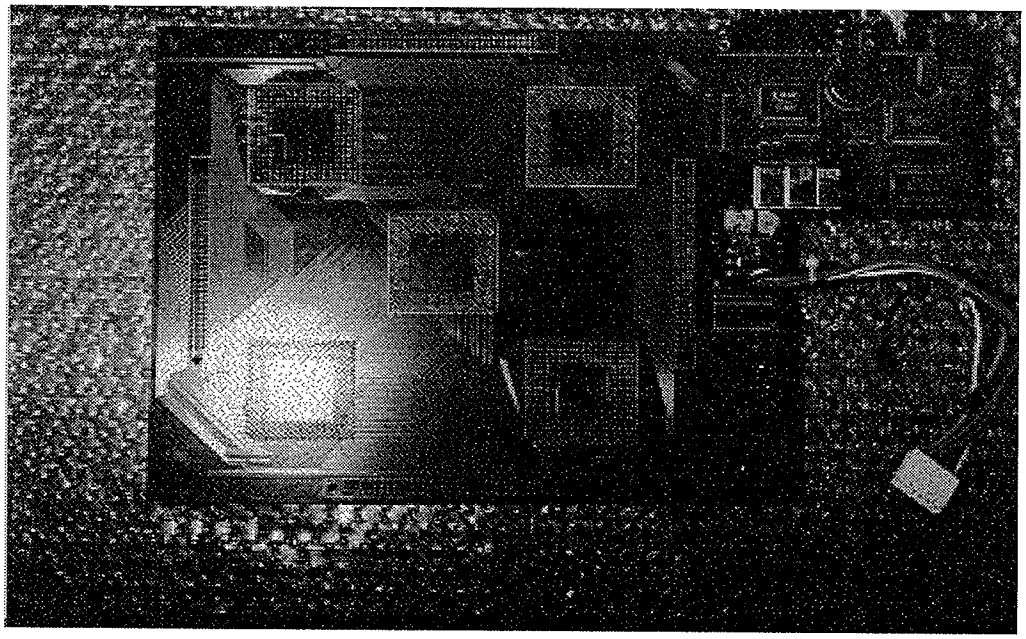


Figure 9: Micro-controller Component Side of The Prototyping Board

CHAPTER 5

5.0 Rapid Prototype Board Architecture

When logic is partitioned into multiple SRAM based FPGAs for prototyping purposes, the FPGA ultilization becomes a major design issue. The current FPGA automatic placement and routing technology is such that if the I/O pin locations are constrained, the ability of the placement and route tools to sucessfully route the chip is greatly reduced. This problem gets worse as the higher density FPGAs are used for prototyping larger designs. The manner in which the FPGAs are connected on the board has a significant impact on the overall gate ultilization of FPGAs and the maximum performance (density) a designer can achieve [11].

The Rapid Prototyping System shown in Figure 10 greatly reduces the problems mentioned above. The separation of interconnect functions from the logic functions; each to be served by a different type of device allieviates the place and route problems. The interconnect function can be provided either by FPGAs or by dedicated FPIDs . In the architecture proposed here, all interconnections go through one crossbar array and therefore the delays are more uniform. The symmetry of the architecture also simplifies partitioning and routing. In this approach, the interconnection routing is a simple, repetitive table-driven-task. The FPGAs have full routing performance without having I/O pin assignment constraint. All interconnections will be done through the FPID devices. In this system the FPID acts as a programmable PCB for the FPGA devices interfaced to the Target System.

In this system there are four main functional blocks. They are the FPGA, FPID, Headers, Interfacing FPGA, and the DS5001 micro-controller. Each one of these functional block plays a unique role.

5.1 FPGA Functional Block and Design Methodology

The FPGA devices used in this Rapid Prototyping System are the Xilinx component XC4013 devices each having approximately 13,000 gates. These are arranged in a 24x24 Configure Logic Blocks (CLB) matrix which provides 576 CLBs, and consists of 192 general purpose

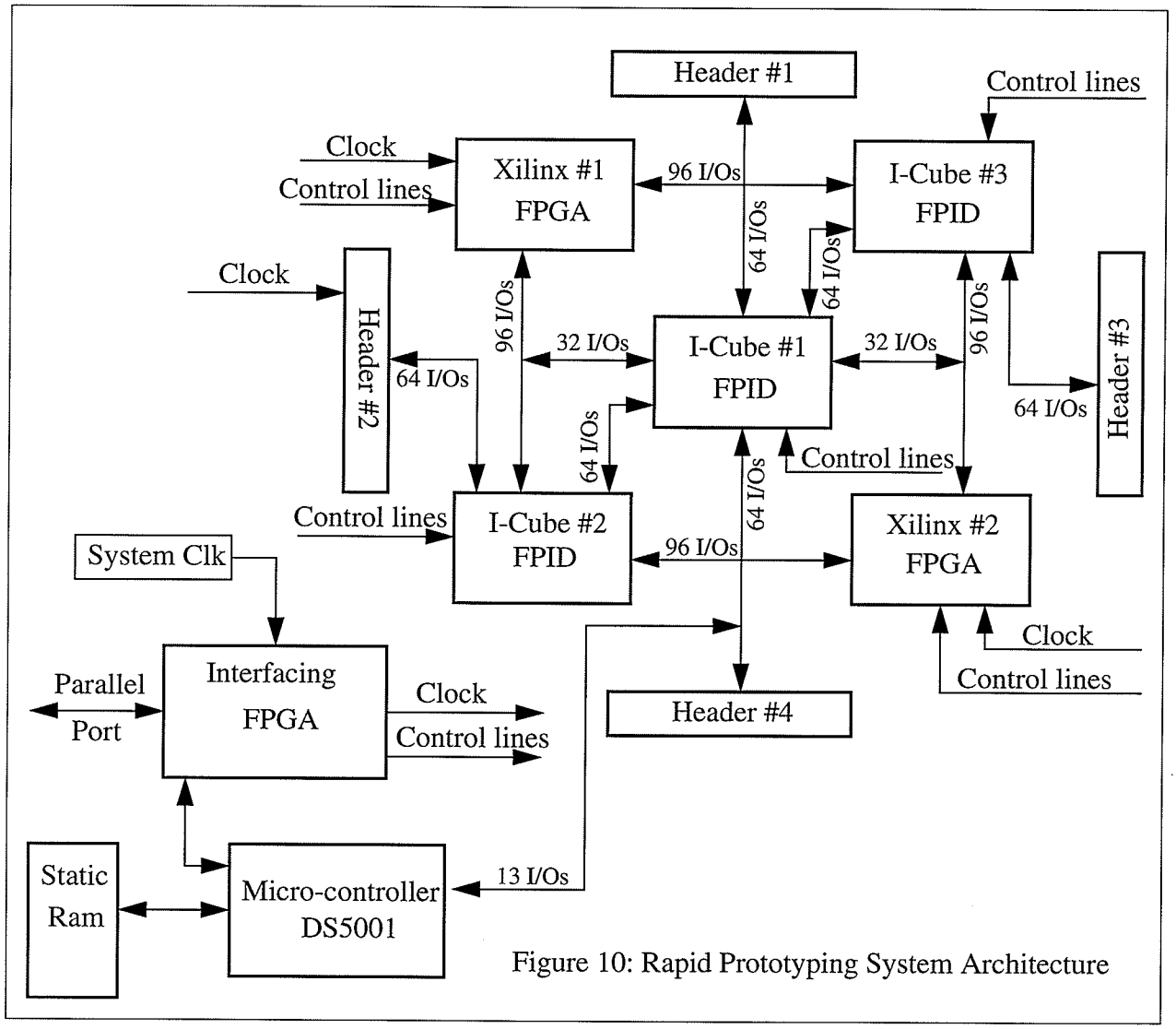


Figure 10: Rapid Prototyping System Architecture

I/Os. The XC4013 package is a 223 pin grid array [1]. The reason this type of package was chosen because it is pin to pin compatible with the XC4025 which is almost double the density of the XC4013. This allows a designer to have added flexibility by doubling the FPGA density by simply inserting the XC4025 parts. This system provides a designer approximately 26,000 gates with the XC4013 and up to 50,000 gates with the XC4025.

The methodology for designing into the XC4013 consists of two interrelated processes: design entry and design implementation. Figure 11 illustrates Xilinx Development System Design Entry, and Design Implementation Flowchart. The design entry could be any third party software such as OrCad, Viewlogic, Mentor Graphics, and or a number of HDL/VHDL tools. OrCad was chosen for design entry in this prototype system but any one of above approaches

could be used for design entry. Once the design has been completed and compiled to created its own schematic netlist, it is ready for design implementation. In the design implementation the schematic netlist is converted into Xilinx Netlist Format (XNF). This is followed by logic reduction, automatic place and route, LCA netlisting, and creation of a configuration bit file.

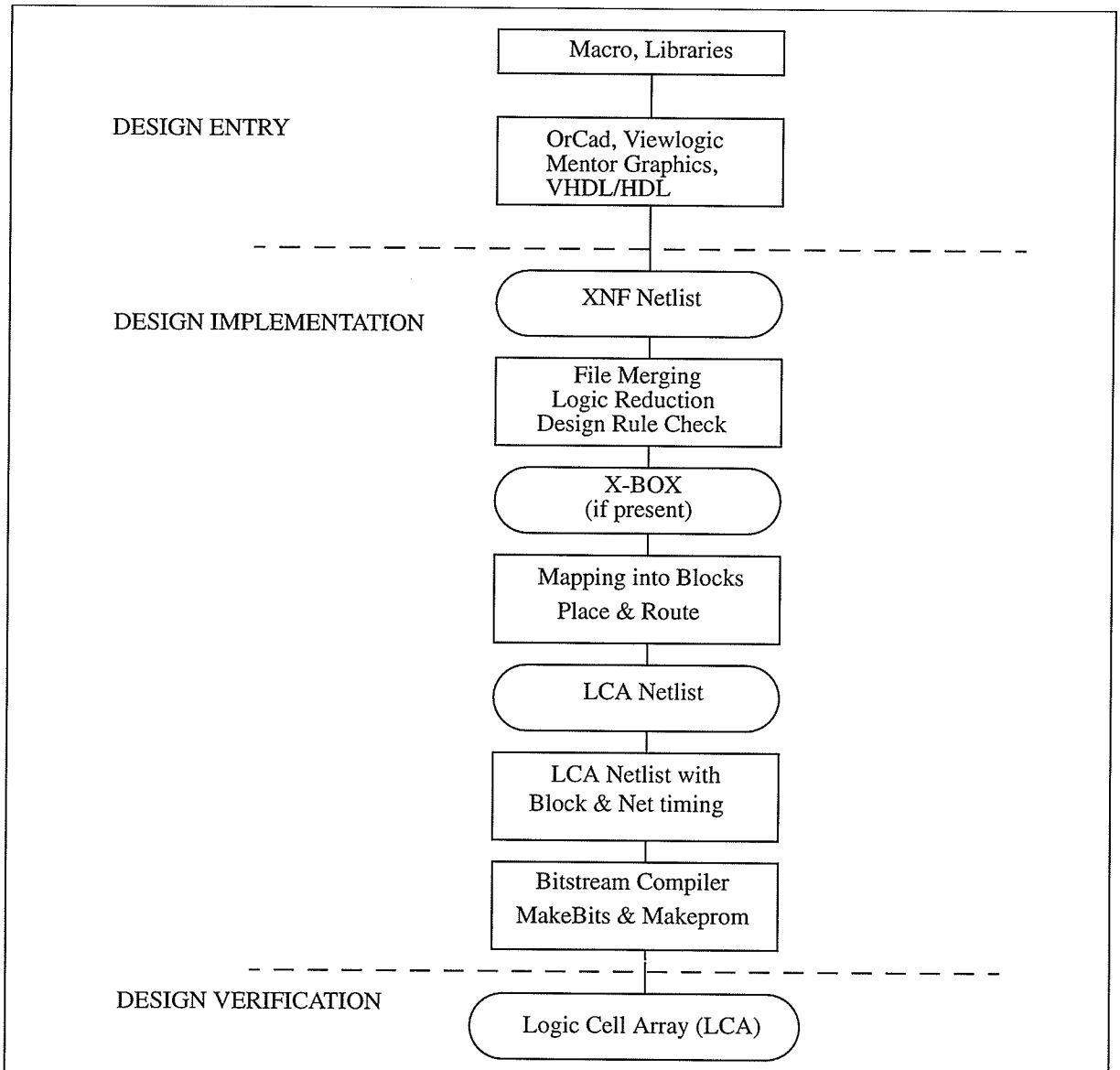


Figure 11: FPGA Design Flow

5.2 FPID Functional Block

The FPID Functional Block in this system is used to optimize ASIC prototyping and emla-

tion applications. It acts as a Field Programmable Circuit Board primarily for emulating and debugging the ASIC design in "realtime". Simply put, it provides a physical realization for the ASIC and system design. The fully programmable and observable features of the Rapid Prototyping System is made possible by the I-Cube IQ320 FPIDs which can link two Xilinx XC4013 FPGAs to emulate an ASIC design. The three IQ320s provide the necessary connections to the two XC4013s and provide 256 general purpose I/Os for interfacing the ASIC design to the target system. The IQ320s have 320 reprogrammable I/Os provide routing from one FPGA to another FPGA, or from one FPID to another FPID, or from an FPGA to the target system.

FPGAs are reprogrammable high-density devices that contain the emulated ASIC design. The combination of FPGA and FPID devices allows the designer to place and route virtually any logic design on the Rapid Prototyping System. In this scenario the designer enters the ASIC design, and partitions and maps the logic to multiple FPGAs via external CAE software. The FPGA I/Os are then programmably interconnected by one or multiple FPIDs . The FPIDs in the prototyping board automatically routes for the fastest possible performance between the FPGAs . Since the FPID devices provide all the pin to pin interconnects, the I/Os of the FPGAs can be left unassigned during place and route resulting in an optimized design.

5.3 DS5001FP Micro-controller and SRAM Functional Block

The DS5001FP is an 8051 compatible micro-controller based on non-volatile RAM technology [7]. It is designed for systems that need large quantities of non-volatile memory. It provides full compatibility with the 8051 instruction set, timers, serial port, and parallel I/O ports. By using NVRAM instead of ROM, the user can program, and subsequently reprogram the microcontroller while in-system. The application software can even change its own operation. This allows for frequent software updates, adaptive programs, customized systems, etc.

The DS5001FP provides the benefits of NVRAM without using its I/O resources. It uses a non-multiplexed byte-wide address and data bus for memory access. This bus can perform all memory access and provides decoded chip enables for the SRAM. This leaves the 32 I/O port pins free for other application use. The DS5001FP uses ordinary SRAM and battery backs the memory contents.

A user loads programs into the DS5001FP via its on chip serial bootstrap loader. This function supervises the loading of software into the SRAM, validates it, then becomes transparent to the user. The software can be stored in multiples of 32K or one block of 128K bytes. Using its internal partitioning, the DS5001FP can divide a common RAM into user selectable program and data segments. This partition can be selected at program load time, and can be modified anytime later. The micro-controller will decode memory access to the SRAM, access the memory via its Byte-wide bus and write-protect the memory portion designated as ROM.

On this prototype board the DS5001FP is configured as 128K bytes for program and data segments. The microcontroller is used to download the configuration bit file to the FPGA and the FPID devices. It also executes the commands received from the host. The SRAM on the other hand is used to store program and configuration bit files for FPGA and FPID devices. One advantage of having a battery backed SRAM is that once the ASIC has been fully debugged and verified, the configuration bit files are stored in memory, the hardware designer can hand-off this system to the software engineer for software verification.

5.4 Interface FPGA and I/O Header Functional Blocks

The interface FPGA provides the handshaking interface between the PC and the microcontroller. This is where the handshaking protocol is implemented. The interface FPGA also provides other functions; such as controlling the system clock for debugging, downloading configuration bit file to FPGA or FPID devices, and enabling the XChecker capability.

The I/O headers provide the interface between the ASIC and the target system, they also provide a logic analyzer port for debugging and testing purposes for the ASIC design.

5.5 Interconnections between the FPGA and FPID Devices

Figure 12 illustrates the interconnects between FPGA and FPID devices and the Target System through the I/O headers.

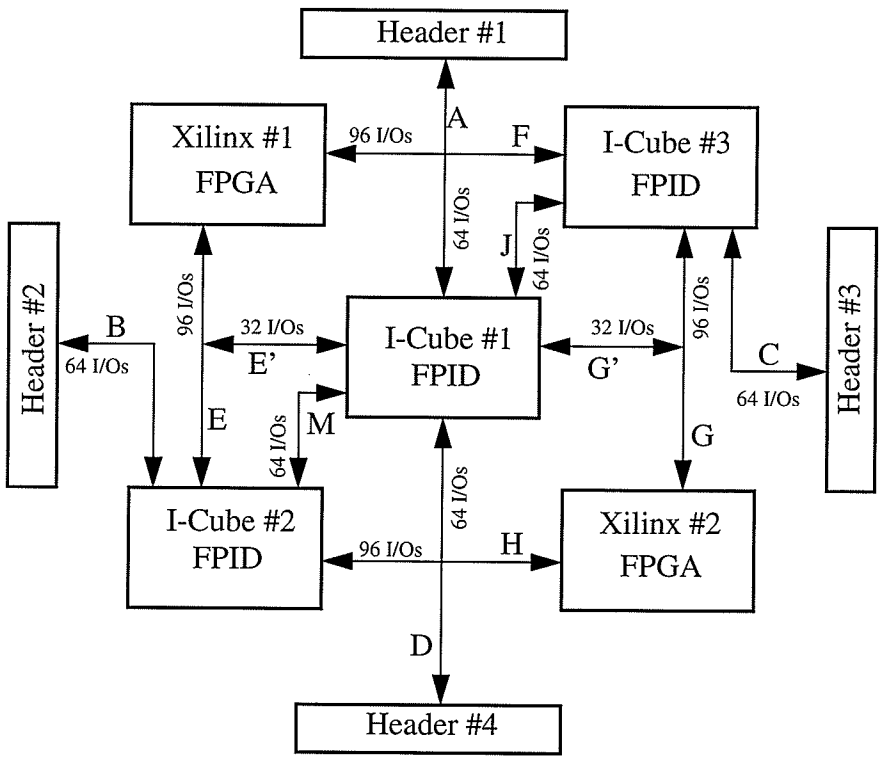


Figure 12 : Interconnects between FPGA and FPID Devices and the Target System
Note: The letters designate bus name.

The prototyping system uses the FPID devices to handle all the interconnect between the FPGAs and the I/O headers. It uses three FPID devices to do the routing. In the next section the routing software will be explained in greater detail. In this section we concentrate more on how these components are interconnected together. The FPID and FPGA devices have 320 and 192 I/Os respectively. The system provides 256 I/Os that can be interfaced to the Target System through the headers. The unique feature about this type of set up is that any two or more signals can be connected together with a 10ns, 20 or 30ns pin-to-pin delay.

When a designer has decided on the application of the ASIC and the design requires more gates than can be implemented in one FPGA; there are some rules that a designer must consider.

1. A designer must partition the ASIC design, a certain portion of the logic will be implemented in one FPGA and the remainder implemented in the other.

2. The signal name that interfaces the two FPGAs must be identical.

3. The designer does not need to assign a pin number to each signal.

4. Any signal name between the two FPGAs that are not the same will be assumed to be not connected.

5. Any signal name that corresponds to the Target System net name will be routed to the appropriate I/O header. For example, the ASIC design has an address bus A(0..31) that needs to be interfaced to the Target System; the Target System net name for the same address bus must also be assigned as A(0..31) and connected to specific I/O headers.

The FPIDs will connect all signals of the same name. As illustrated in Figure 12, half of the I/Os in FPGA#1 are connected to FPID#3 and the other half are connected to the FPID#1 with the bus name 'F' and 'E' respectively. The FPGA#2 on the other hand has half of its I/Os connected to FPID#3 and the other half connected to FPID#1 with the bus name 'G' and 'H' respectively. FPID#3 and #1 each provide 64 I/Os to Header#2 and #4 with the bus name 'B' and 'D' respectively. FPID#2 provides Header#1 and #3 each with 64 I/Os with the bus name 'A' and 'C' respectively. FPID#2 also provides FPID#3 and #1 each with 64 I/Os with the bus name 'J' and 'I' respectively.

This architecture provides predictable delays. The following describes the path connection and approximate delays for bus 'F' to other bus signals.

- F <-> G via FPID#3 with 10ns pin-to-pin delay.
- F <-> C via FPID#3 with 10ns pin-to-pin delay.
- F <-> D via FPID#3 to FPID#1 with 20ns pin-to-pin delay.
- F <-> B via FPID#3 to FPID#1 to FPID#2 with 30ns pin-to-pin delay.
- F <-> H via FPID#3 to FPID#1 to FPID#2 with 30ns pin-to-pin delay.
- F <-> A via FPID#3 to FPID#1 with 20 ns pin-to-pin delay.

This examples illustrate how the signals are interconnected together. Analogous routing paths can be applied for other bus names.

CHAPTER 6

6.0 Software Overview

The software for this prototyping system consists of three separate software components. They are routing , host interface , and DS5001 micro-controller codes. The routing software determines how all the signals are interconnected together and generates an I-Cube format netlist. The host software determines the handshaking between the host and the DS5001 micro-controller. It also generates a set of commands to be send to the micro-controller. The DS5001 microcontroller software determines the handshaking between the host and the micro-controller. It is responsible for downloading configuration bit files to both the FPGA and FPID devices. It also receives and executes the commands which are send by the host.

6.1 Routing Software

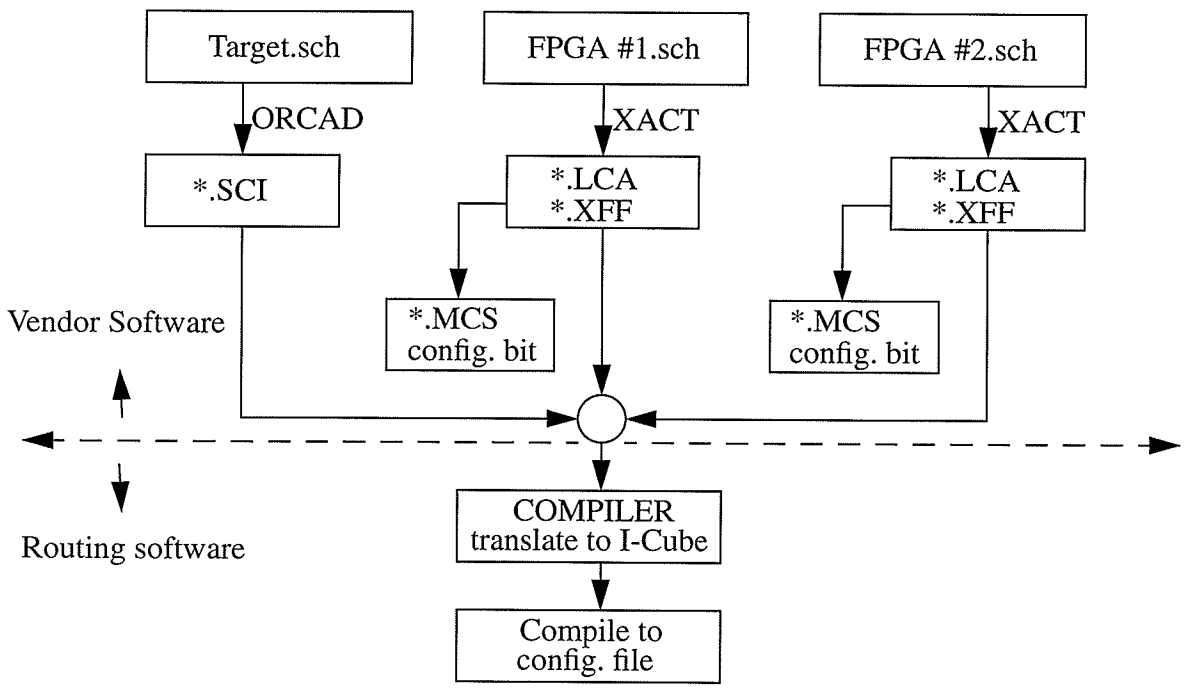


Figure 13. Translation from Schematic to System Prototyping Level

Figure 13 shows the design flow from schematic capture to the I-Cube routing configuration format. For this thesis, Orcad was chosen to be a front end capture tool to generate the

design. The XACT development system [8] was chosen as the FPGA design place and route software tool. For FPGA design capture other tools can be used in place of Orcad such as VHDL or Verilog HDL compilers. Similarly the schematic capture for target system can be created in Mentor Graphics, Cadence, or other schematic capture tools.

The design flow for the prototyping system is as follows:

- The design is captured using Orcad with the use of the Xilinx 4000 family library.
- Orcad compiles the schematic file into an Xilinx Netlist Format (XNF file).
- The XACT development system reads the XNF and begins to execute place and route. During this process XACT generates two files; they are the LCA and XFF files which will later be used by the routing program. At this time the configuration bit file is generated (MCS file) which will be downloaded to the FPGA later on.
- Orcad compiles the target system schematic capture into Scisscard format (SCI file) which will also be used by routing program.
- The routing program reads the LCA, XFF and SCI netlist files to generate an I-Cube interconnect format.
- The I-Cube development software reads this file to generate a configuration bit file for FPID devices.

The routing software requires three different netlist files which are created by the XACT and Orcad netlist translators. By reading these three files, the routing software can determine how the system is going to be interconnected together. Each of these three netlist files provide unique information to the routing program. The XFF and LCA both are created by XACT. The XFF file provides the signal names, and the signal attributes such as input, output, or bidirectional. The LCA file provides signal name , and the pin location of that signal. These two files provide the routing software with three important pieces of information to determine the interconnection of the FPGAs . They are the signal name, pin location of the signal , and the signal attributes. The SCI file on the other hand provides the signal names, pin locations on the header of those signals, which headers the signals are located, and the signal attributes. With this information the routing software can determine how to connect the signals between the FPGAs and the Target System.

The above list gives an overview of the design flow for the routing software. The routing software consists of several stages of development. Each stage of the routing software generates specific information which will be used later by the next stage. The reason for breaking up the routing software in different stages is that this allows for easy software debugging. There are seven different stages of the routing software. They are Netlist Translation, Netlist Error Checking, Interconnect Translation, Net Routing, Portmap Configuration Attribute (PCA) Translation, Translation from PCA to I-Cube netlist format, and FPID bitstream compiler.

6.2 Netlist Translation

The netlist translation software reads in the LCA, XFF, and the SCI files to generate a connection format file which is the ITC file. Note: when LCA, XFF, SCI, or ITC is mentioned it means the extension of the filename. Figure 14 shows the design flow of this netlist translation process.

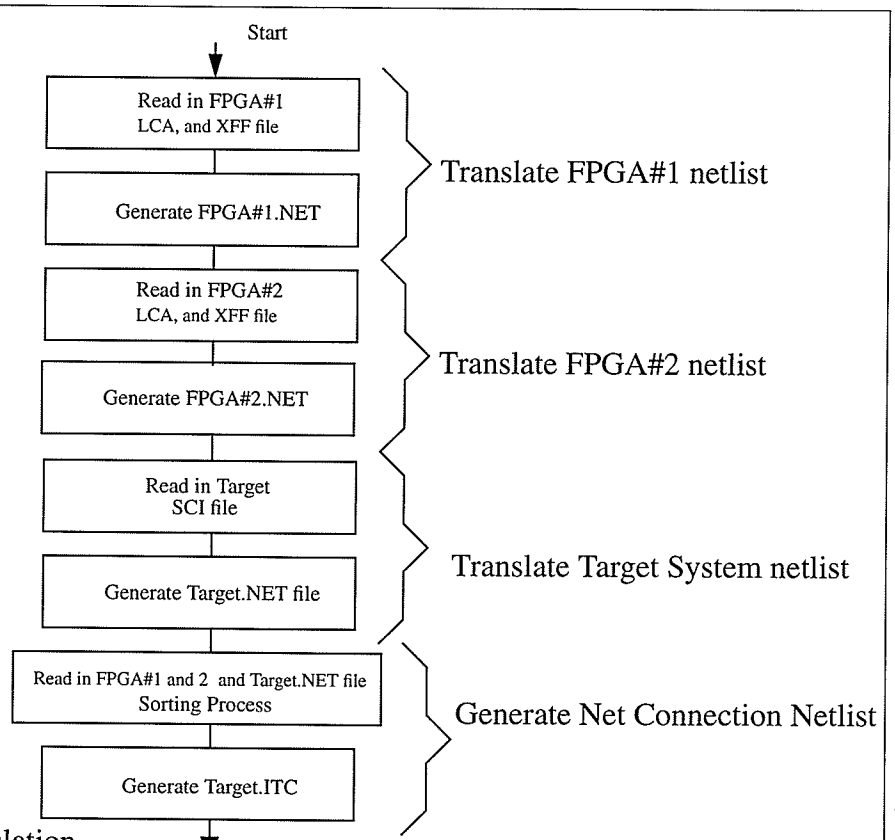


Figure 14: Netlist Translation Flowchart

As shown in Figure 14 the netlist translation software consists of four different processes. They are translating FPGA#1 netlist, FPGA#2 netlist, Target System netlist, and generation of a Net Connection Netlist.

6.2.1 FPGA Netlist Translation

The FPGA netlist translation for both FPGAs are similar. The only difference is that one indicates FPGA#1 and the other indicates FPGA#2. As mentioned earlier the LCA file provides the signal names, and the pin locations of particular signals. The XFF file on the other hand provides the signal names, and the attributes of those signals.

When the netlist translation reads in the LCA file, the only information that needs to be scanned is a line which contains "Nameblk". This line contains the signal name and the pin location. The software will read in the LCA file and scan each line of this file until it reaches a line which contains "Nameblk". Once this event has been detected it saves the signal name and the pin location information. The next step is to read in the XFF file and scan each line of this file until it reaches a line which contains "EXT" which provides signal attribute information. Once this event has been detected, it compares the signal name in the LCA file to the signal name detected in the XFF file. If the signal name is identical between the two files, it will save the signal name , the signal pin location, and the signal attribute information in a FPGA#1.NET or FPGA#2.NET file. If the signal detected in the XFF file is not the same as in the LCA file, it will continue to scan until this condition is detected. The software will scan the LCA file for the next line that contains "Nameblk" and the whole process described above will continue until the software reading the LCA file detects an end of file. The order in which the information is stored in the FPGA.NET file is as follows:

"aa pin_location signal_name signal_attribute:" Where aa indicates the FPGA#1 or 2

For example: "x 200 wr_enable I:", represents FPGA#1 with a wr_enable signal located at pin location 200 and its attribute direction is input (X maps to FPGA#1).

6.2.2 Target System Netlist Translation

The Target System netlist provides the information on how the Target System is intercon-

nected. The only information that concerns us are the signals which are interfaced to the FPGA (ASIC) signals. The software will read in the SCI file and scan each line of this file until it reaches a line which contains "Net". Once this event has been detected, it will scan that line and determine if that particular signal is interfaced to the FPGA or not. This is accomplished by determining if the signal is connected to the header I/O. It is either Header#1, Header#2, Header#3, or Header#4. When the software has detected this condition, it saves the signal name, the header I/O number along with the pin location in that particular header, and the signal attribute. The order in which the information is stored in the Target.Net file is as follows:

"signal_name aa bb signal_attribute:" Where aa defines header no. and bb defines pin location in that particular header.

For example: "asic_wr J 23 O:" , represents header J pin 23 has asic_wr signal with attribute direction output. Note: J, K, L, M represent Header #1, 2, 3 and 4 respectively.

6.2.3 Generating the Interconnet Netlist

This part of the netlist translation software generates the information which defines the interconnect between FPGA#1 to FPGA#2 to the Target System, FPGA#1 to FPGA#2, FPGA#1 to the Target System, or FPGA#2 to the Target System.

The software reads in three NET files which were generated as mentioned above or two NET files if only one FPGA is used. Figure 15 shows the flowchart of this particular subroutine. The results of this netlist translation software are the signal names, pin locations on each device with a corresponding attribute. For example:

```
1    Net D0
2    X 100 O
3    Y 25  I
4    J 23  I
5    End
```

The above example represents; Signal D0 is an output signal at FPGA#1 pin 100 connected to an input at FPGA#2 pin 25 and connected to an input at Header#1 pin 23. The netlist translation software generates an ITC file. For each of the nets, the 'source' is always at line 2 which simplifies the algorithm for the next software section.

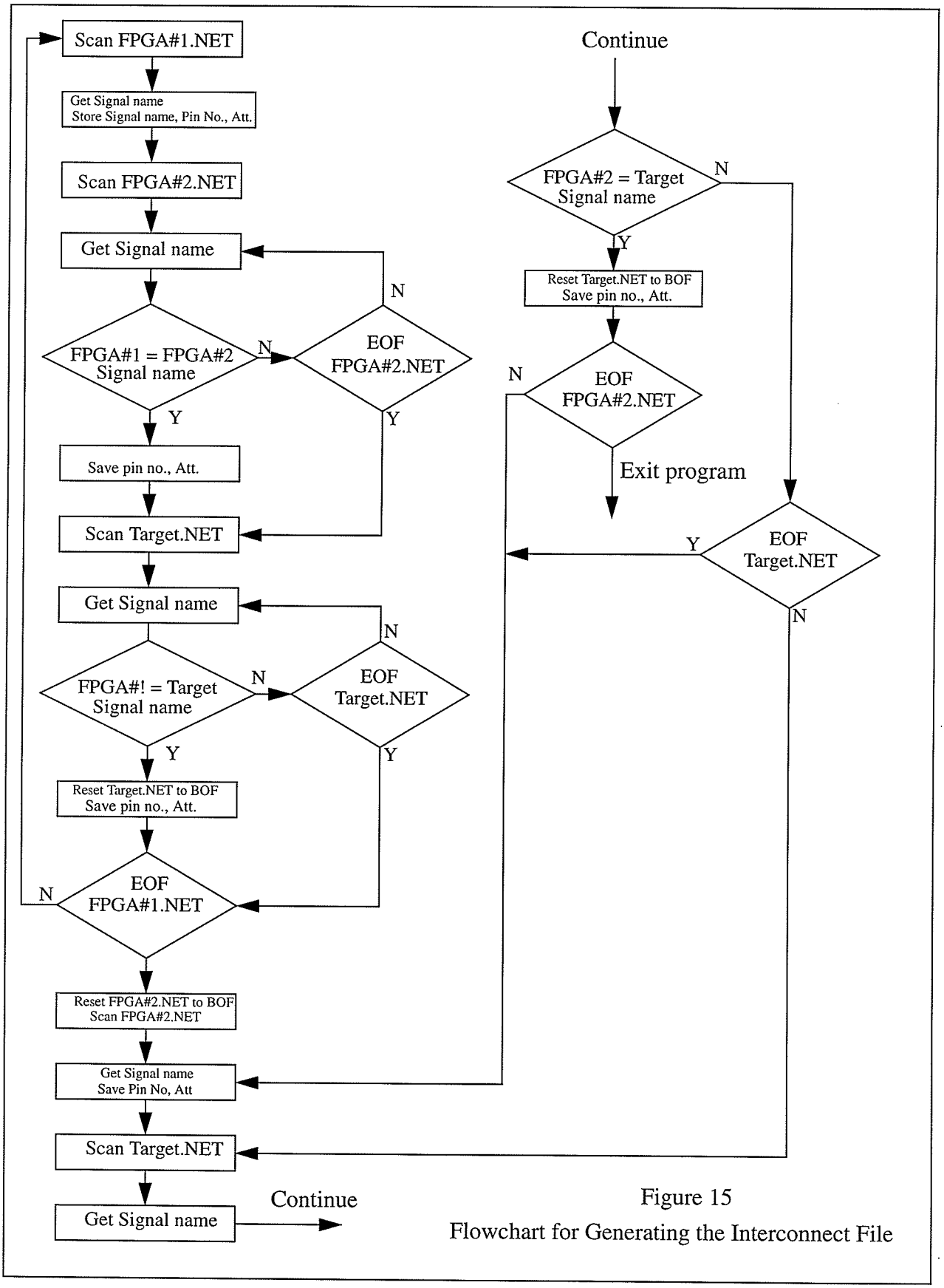


Figure 15
Flowchart for Generating the Interconnect File

6.3 Netlist Error Checking

The error checking routine reads in the ITC file and processes it for errors; if errors are detected the result is saved in the LOG file which can be viewed later. It also checks for warning conditions. The following errors and warnings are checked for.

Errors:

- A net cannot have more than one output (drive source). This produces a contention message indicating multiple sources.
- A net cannot have only an input signal. This produces a no source message.
- A net cannot have only an output signal . This produces a no destination message.

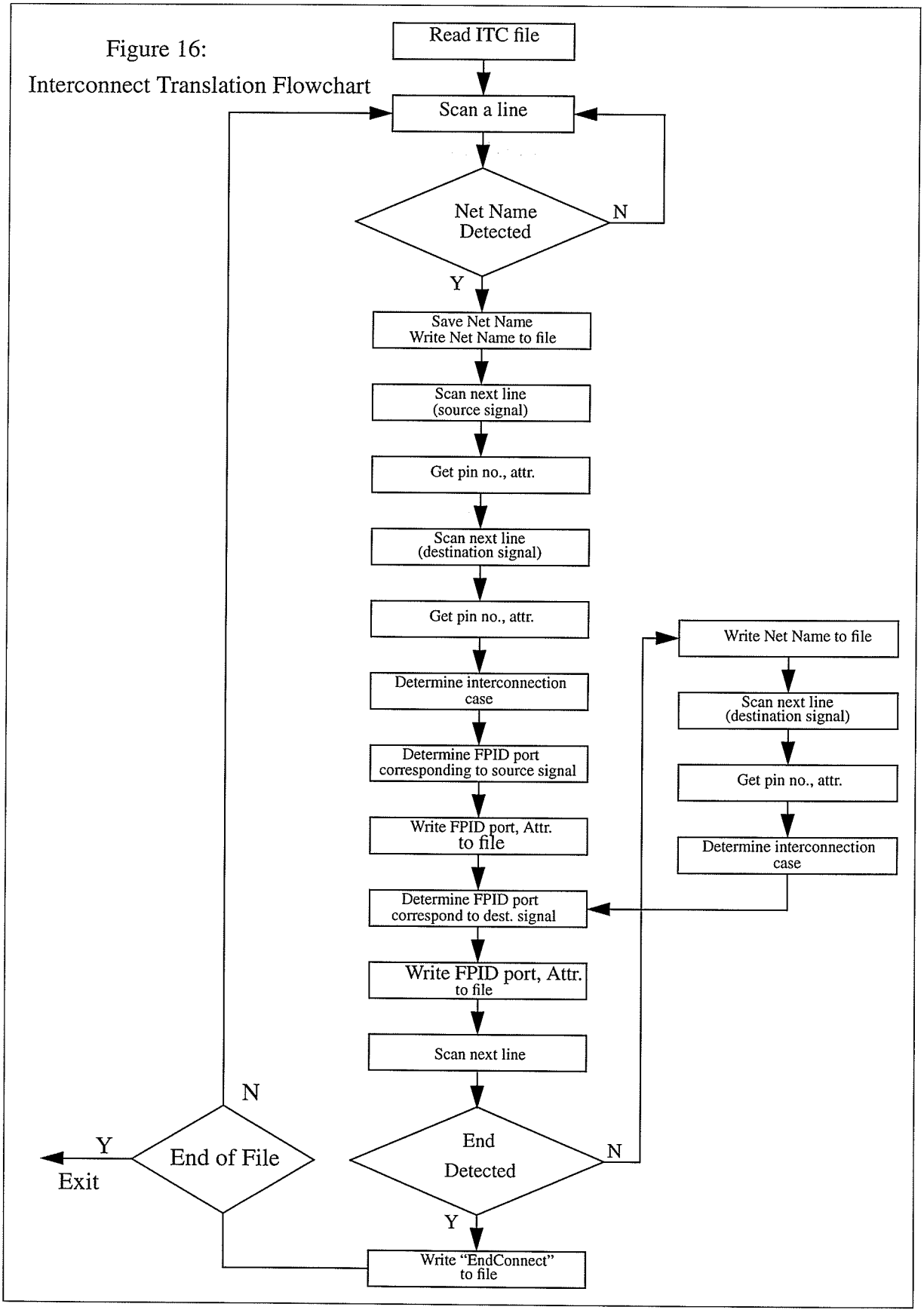
Warnings:

- A net that has both output and bidirectional attributes. This produces a contention may occur message.

6.4 Interconnect Translation

The interconnection translation software reads in the ITC file and determines how to interconnect each signal. Refering back to Figure 12, we can see that each of the I/O signals on both FPGA devices and the I/O headers are routed to specific ports of the FPID devices. This information and the information which was generated by the netlist translation software allow us to determine the interconnections for the FPIDs. Figures 12 shows that depending on a net origin and destination, a net can be interconnected through one , two, or three FPID devices. The way the prototyping system is set up , it allows us to interconnect a net in a specific manner. Note: for example G -> F represent a signal on bus 'G' which is a source to a signal on bus 'F' which is an input as shown in Figure 12. Appendix A lists different cases of interconnect the prototyping system supports. Figure 16 shows the flow chart for this interconnect translation program.

Figure 16:
Interconnect Translation Flowchart



For example a net wr_enable which was generated from an ITC file is shown below:

```
Net  wr_enable
X  125  O          X = FPGA#1
J   20  I          J = Header#1
K   10  I          K = Header#2
M   2   I          M = Header#4
End
```

The interconnect translation software reads in a file with an "ITC" extension. It then scans a line until a "Net" is detected. Once this condition is true, it writes the " Net wr_enable" statement to a file to indicate the beginning of interconnect for net wr_enable. It then scans in the next line which is always a source or the origin of a net and saves the pin number, the attribute, and the component identification. The component identification defines on which device the signal is located. The processing of the next line which is a destination of a net is done the same way. Next, the program retrieves the component identification and the pin number for the source and determines on which bus and FPID port the source is located. (Note: refer to Figure 12 for bus name). This is accomplished by comparing the component identification and the pin number to a look-up table. Once this is completed, the information consisting of bus name, on which FPID is the signal is located, and the FPID port is saved away. The same process is done on the destination signal. Next, the program retreives both the bus name for the source and the destination and determines the interconnect cases. (Refer to Appendix A for the listing of the cases.) Once this is done, the program will write the connection's case, bus name, FPID devices, and FPID ports for both the source and the destination, followed by a "EndConnect" statement. It will then scan the next line; if an "End" statement is detected, the program will terminate the process of determining the interconnect cases for net wr_enable. If the result is false, scanning for the next destination within this net will be processed. The result for the above example is shown below.

```
1   Net wr_enable
2   Connect Case 37
3   F  A 23   O
4   A  C 45   I
5   EndConnect
6   Connect  Case  39
7   F  A  23  O
8   B  B  10  I
9   EndConnect
10  Connect Case 43
11  F  A   23  O
12  D  C   100 I
13  EndConnect
14  EndNet
```

Result are stored in a CON file

The net wr_enable consists of a single source and three destinations. The following will explain each of the line numbers shown above.

- Line 1: Net name definition.
- Line 2: Determine the connection case for that particular node. Refer to Appendix A for the connecting cases listing.
- Line 3: The source is located on bus 'F' at port 23 of FPID#1 with an output attribute.
- Line 4: The destination is located on bus 'A' at port 45 of FPID#3 with an input attribute.
- Line 5: Indicate end of connection for this particular node within net wr_enable.
- Line 6 - 13 are similar to line 2 - 5.
- Line 14: Indicate the end of connection for this particular net 'wr_enable'.

This "CON" file provides us with the information in terms of how a net is interconnected among the FPID devices. It also provides us a connection case for each node in a particular net. As shown above, the example net wr_enable has 3 nodes. By definition the number of nodes equals number of destinations.

6.5 Net Routing

This software determines the routing path for a particular signal. The signal could be routed through one, two, or even three FPID devices depending on the source and the destination.

The worst case for routing any signal is through three FPID devices. Figure 17 shows the flow chart for this routing software.

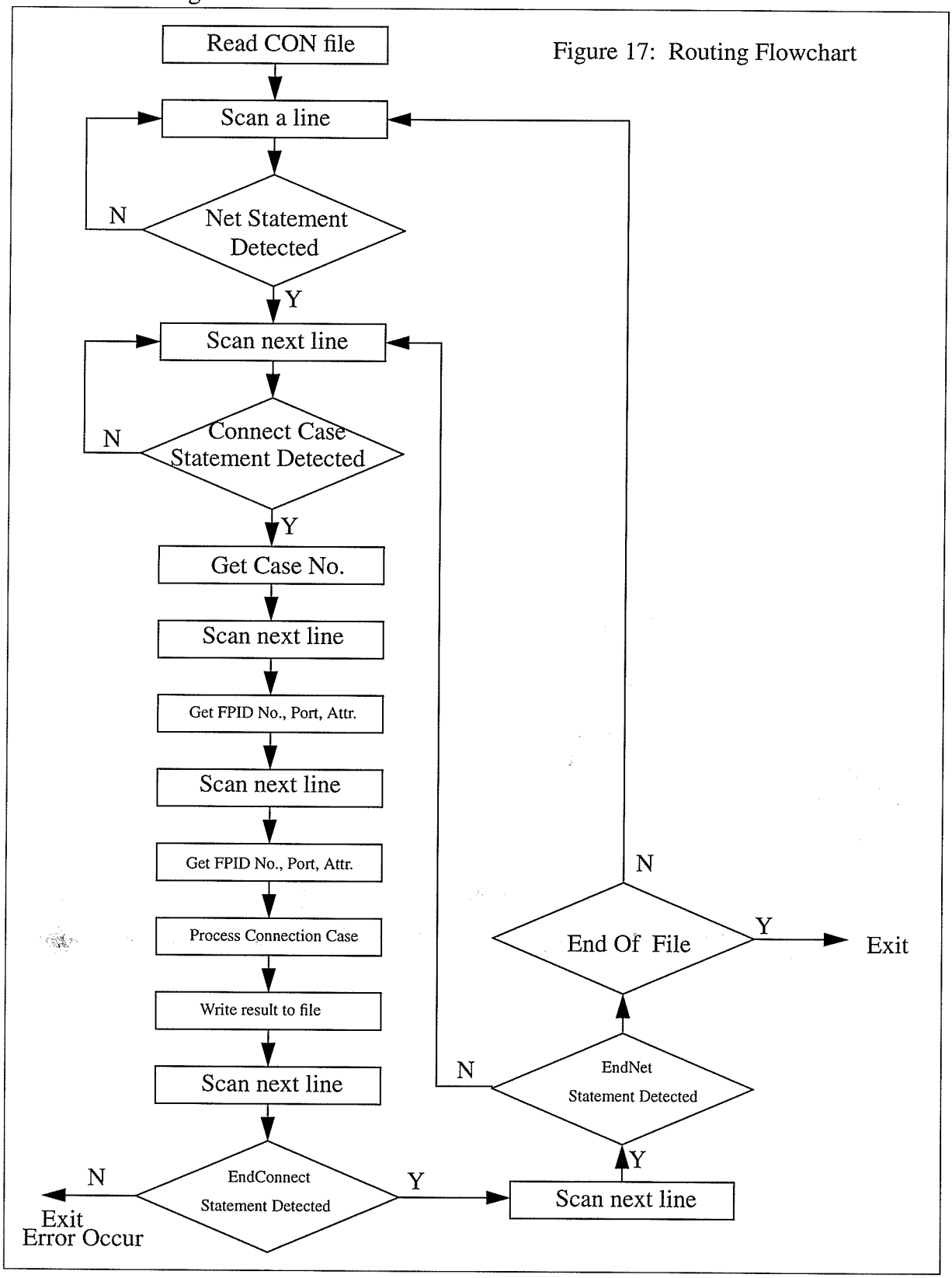


Figure 17: Routing Flowchart

The routing software reads in the CON file which was generated by the interconnect translation software. It then scans a line until detects a Net statement within a line. Once this condition is true, it scans the next line until it detects a Connect Case statement within a line. When this condition is detected the Connection Case number is saved, the next two lines are scanned. Each time the software scans a line, it saves the FPID number, port, and the attribute information. Once this is done the process for determining the connection is begun.

Refer to the result above for Net wr_enable. The process for determining the connection case begins by retrieving the Connection Case NUMBER, for our example the first connection case is 37. This indicates that the source originates at FPGA#1 locate on bus 'F' connected to FPID#1 through bus 'J' to FPID#3 and ends up on bus 'A' connected to Header#1. This is illustrated in Figure 10. Once the software detects the specific case number for interconnecting a particular node; it knows the connection for that node is either through one, two or three FPIDs. The system is set up in such a way that each connection case represents a specific routing path. This keeps the routing software simple. Once the software has determined the routing path for each node; it retrieves the FPID number, port, and attribute for both the source and the destination and writes this information to the ROU file. During this process the attributes will be the opposite from when the attribute was first read in. For example if the original source was an output; the ROU source attribute will be input. The only attribute that will not change is bidirection.

Next the software determines if the end of a net connection is detected or not. This is done by scanning the next line and checking if EndConnect is detected. If this condition is true, it means that this is the end of node connections and the next line is scanned in. It then checks for the Endnet statement within this line. If this true; it means that we have reached the end of the net connection for routing purposes. If the result is not true the whole process starts over again. The result for Net wr_enable is as shown below. Note the result of this software is saved in an ROU file. Figure 18 shows the routing path for Net wr_enable with three different nodes.

```
Net wr_enable
Connect
A Portmap 23 [FSEL = IN]
A Portmap 80 [FSEL = OP]
C Portmap 30 [FSEL = IN]
C Portmap 45 [FSEL = OP]
EndConnect
Connect
A Portmap 23 [FSEL = IN]
A Portmap 80 [FSEL = OP]
C Portmap 30 [FSEL = IN]
C Portmap 35 [FSEL = OP]
B Portmap 112 [FSEL = IN]
B Portmap 10 [FSEL = OP]
EndConnect
Connect
A Portmap 23 [FSEL = IN]
A Portmap 80 [FSEL = OP]
C Portmap 30 [FSEL = IN]
C Portmap 100 [FSEL = OP]
EndConnect
EndNet
```

Node 1

Node 2

Node 3

This is a result from the routing software and the result is saved in ROU file.

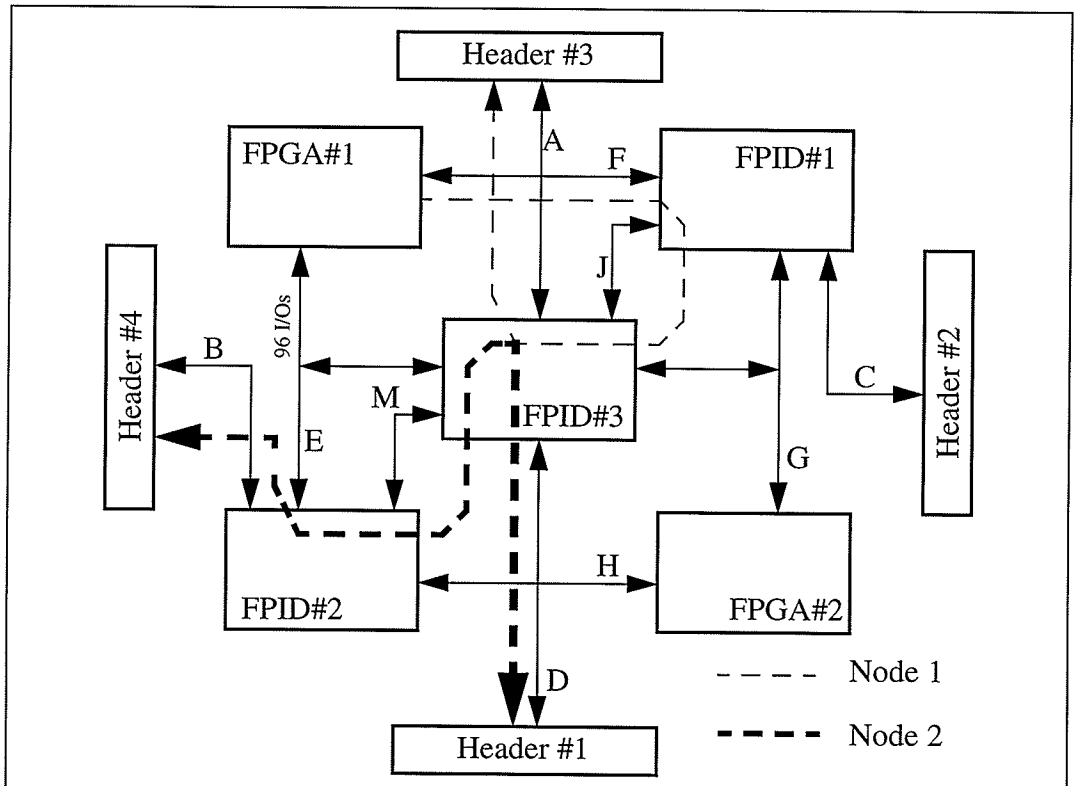


Figure 18: The Routing Path for Net wr_enable

6.6 Portmap Configuration Attribute Translation

This software translates the information stored in the ROU file into an I-Cube Portmap Configuration Attribute format (PCA). Figure 19 shows the software flowchart.

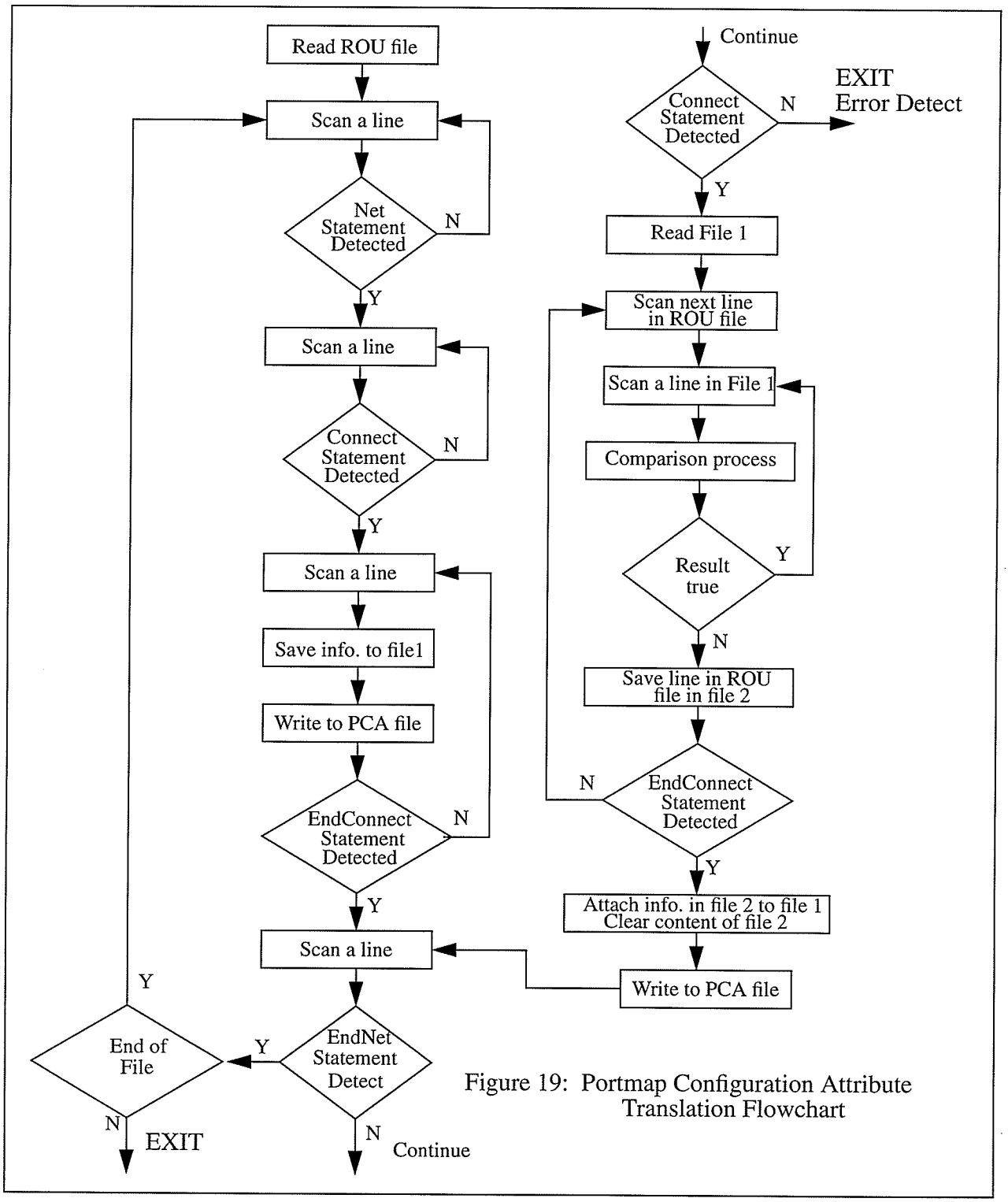


Figure 19: Portmap Configuration Attribute Translation Flowchart

The main purpose of the net routing software is to read in the ROU file and extract out all of the duplicate interconnect information for a particular net. The result of this is saved in a PCA file. Refer to the above example for Net wr_enable; the result for this net is as follow:

```
Net wr_enable
A Portmap 23 [FSEL = IN]
A Portmap 80 [FSEL = OP]
C Portmap 30 [FSEL = IN]
C Portmap 45 [FSEL = OP]
C Portmap 35 [FSEL = IN]
B Portmap 112 [FSEL = OP]
B Portmap 10 [FSEL = OP]
C Portmap 100 [FSEL = OP]
EndNet
```

This PCA file shows how a net is interconnected among the three FPID devices. For example the line "A Portmap 23 [FSEL = IN]" represents FPID#1 at port 23 is to be configured as an input.

6.7 Translation of PCA to I-Cube Netlist Format

This software reads in the information stored in the PCA file and translates this information into an I-Cube netlist format which will be compiled by FPIDComp into configuration bit file. Figure 20 shows the flowchart of this software. The result for the above is illustrated as follows:

```
Device IQ320 U1 U2 U3
JTagChain JC1 U1 U2 U3
DevInst IQ320 U1;
Portmap 23 [FSEL = IN];
Portmap 80 [FSEL = OP];
Connect 23 80;

EndDevinst;
DevInst IQ320 U2;
Portmap 112 [FSEL = IN];
Portmap 10 [FSEL = OP];
Connect 112 10;

DevInst IQ320 U3;
Portmap 30 [FSEL = IN];
Portmap 45 [FSEL = OP];
Portmap 35 {FSEL = IN];
Portmap 100 [FSEL = OP];
EndDevinst;
```

The interpretation of the above netlist format will be explained in detail in the next section. This information is stored in an NLT file.

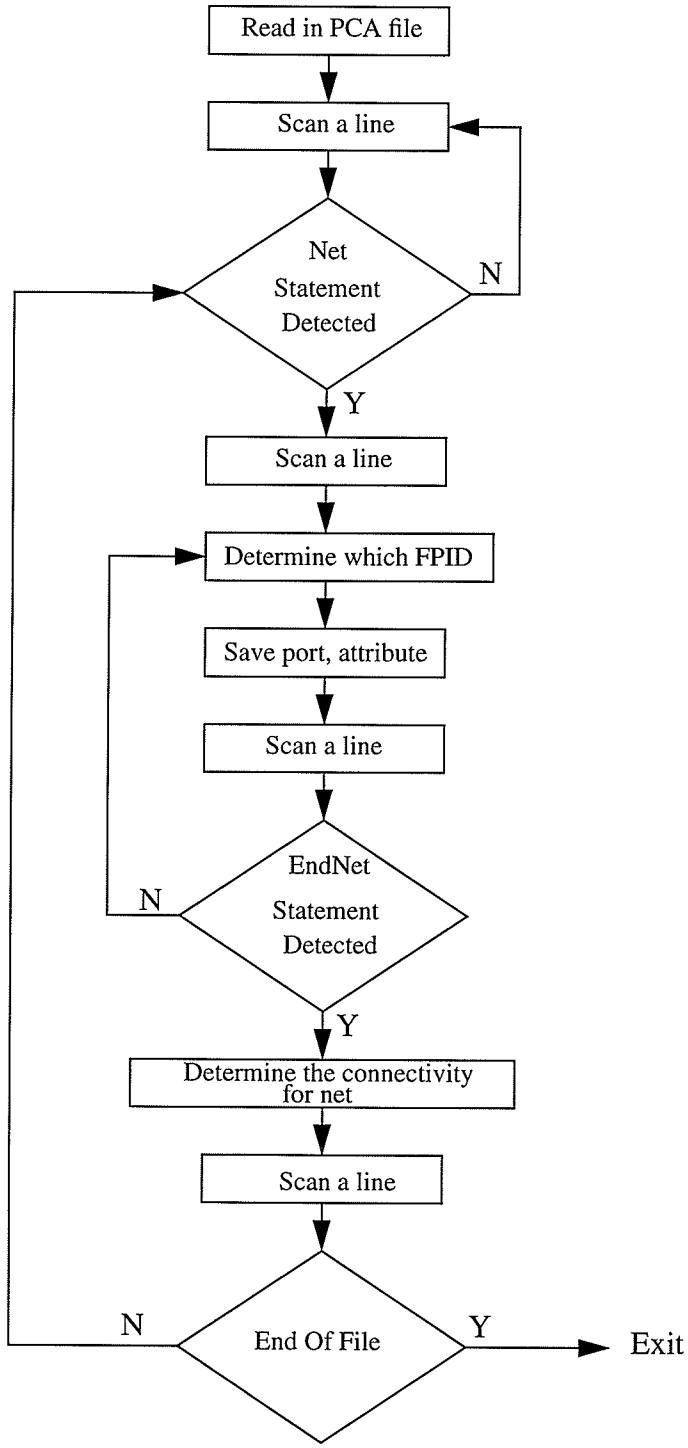


Figure 20 : Flowchart for Translating from PCA to I-Cube Netlists

6.8 FPID Bitstream Compiler

FPIDComp is an FPID "compiler" that accepts a textual netlist file describing the desired connectivity and functionality for one or more FPID devices and compiles the netlist into a bitstream file[9]. When downloaded into the devices, the bitstream will configure the FPIDs to exhibit the desired connectivity and functionality. The input and output files of FPIDComp are shown in Figure 21.

Figure 21: FPIDComp Input and Output Files

The program allows the user to describe the desired connectivity and I/O port functionality in a simple textual netlist file. From the netlist description the system automatically generates the necessary JTAG bitstream to configure the FPID devices. This greatly alleviates the user from having to deal directly with the JTAG mechanism used to configure the devices and the detailed internal operating of the FPID device.

The program can be used to generate bitstreams for a full or incremental configuration. In full configuration the bitstream starts by first initializing all the FPID devices and then fully configuring them. In the incremental mode, the devices are in some operational, non-reset condition. The bitstream will only incrementally reconfigure certain portions of the devices.

For the designs containing more than one FPID, the devices are assumed to be daisy-chained to form a "JTAG chain" as shown in Figure 22.

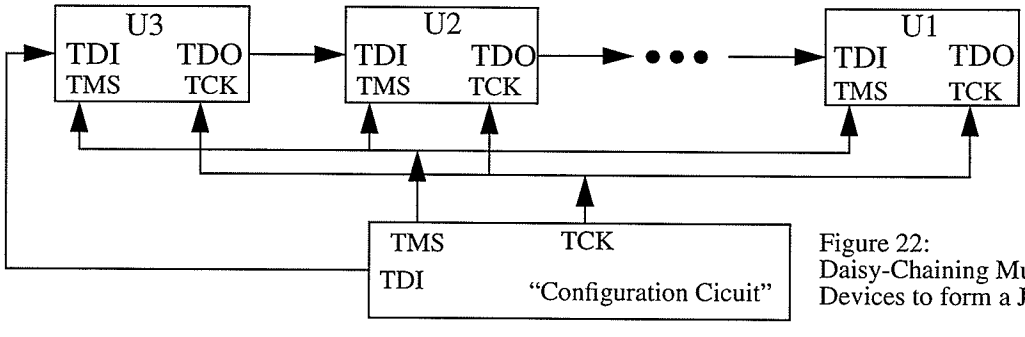


Figure 22:
Daisy-Chaining Multiple FPID Devices to form a JTAG Chain

A design netlist specification file (NLT file) consists of two parts; the configuration specification and the connectivity & attribute specification. The configuration specification defines the number and kind of devices comprising a JTAG chain and the order of chaining. The connectivity and attribute specification defines, for each FPID device, the schematic signals connected to the I/O ports, the attribute(direction and other functions) of each I/O port, and the manner in which the I/O ports are to be connected using the crossbar switch array. The following shows the outline of a design netlist specification:

```
// Configuration specification.
Device  device_ID    inst_name_1    inst_name_2    ......;
Device  device_ID    inst_name_3    inst_name_4    ......;
.....
JTAG    Chain_name
        inst_name_j  inst_name_i   inst_name_k    ......;
// Connectivity & Attribute specification.
DevInst  device_ID    inst_name_a;
   Portmap        ........;
   .......
   .......
   Connect        .........;
EndDevinst;

DevInst   device_ID   inst_name_b;
.....
......
EndDevInst;
....
.....
```

A netlist file consists of a sequence of statements. Each statement starts with a reserved keyword and ends with a semicolon(';'). A statement may span more than one line. A double-slash ('//') is used to indicate a comment. All text following the double-slash, up to the end of the line, is treated as commentary and ignored. All keywords and names used in the netlist file are case-sentive. Statements in the design netlist specification are described as follows:

The "**Device**" statement (" Device device_ID inst_name_1 inst_name_2 ...;"): The Device statement lists, for each device type used in the design, the instances that are of the particular type. Each device instance is identified by its reference designator. There can be as many **Device** statements as there are device types. The following is the list of device types (device_ID) currently supported by I-Cube corporation. They are IQ320, IQ240B, IQ160, IQ128, and IQ96B.

The "**JTAGChain**" statement ("JTAGChain chain_name inst_name_i inst_name_j .."): The JTAGChain statement lists all the devices in the JTAGChain according to their order of cascading. The first device is the one that is connected to the external TDO and the last device is the one that is connected to the external TDI. As an example, the devices in Figure 22 would be specify as follow: " JTAGChain JC1 U1 .... U2 U3 ";

The "**Device & EndDevInst**" statement: The detailed specification of each device instance is contained in a list of statements bracketed by two statements DevInst and EndDevInst. The opening bracket repeats the device type ID and the instance reference designator. There may also be some instance-specific attributes that are given in the attribute specification part. The list of statements ends with an EndDevInst statement.

Within the statement brackets, four statement types are used to specify the way an FPID device is connected to the other parts on board, the desired characteristics of the device's I/O ports, and the way the I/O ports are to be connected using the crossbar switch. These are describe below.

The "**Portmap**" statement ("Portmap Port Net [Attribute}; "): Defines the external connections to FPID I/O ports. It maps a signal net in a schematic drawing to an IO port of the FPID part, or a group of nets to a range of I/O ports. The statement also defines the desired attribute for the I/O ports. The Net part may left empty. In that case, the user is just defining the I/O port attributes. For example " Portmap 0 N1 [FSEL = OP]; " indicates that the signal net N1 in the schematic drawing is connected to port 0 of the FPID and the port is an output.

The "**Connect**" statement (" Connect port port ..... ; "): Specifies how the I/O ports are to be connected through the crossbar array. The specification may be based on the "external" signal nets. For example "Connect 10 20 30 ;", here the desired connectivity is specified directly using the I/O port numbers. Thus three I/O port (10, 20, and 30) are to be tied together using the crossbar array.

The bitstream file format (JTAG bitstream file) generated by FPIDComp is a fully packed binary file. The file starts with a header block containing some identification information. A

multi-byte number in the header block is stored in such a way that the least significant bits occupy the lower addressed byte position. For example, for a 32 bit integer, if the least significant 8 bits are kept in byte "n" , the next 8 bits will occupy byte "n+1",....., etc. The following shows the specification of the JTAG bitstream format.

Byte Number	Contents
0 - 79	Text string identification file type (Printable ASCII string for DOS Type commands).
80 -83	Length (number of bytes) of this header block (32 bit integer).
84 - 87	File format version/revision number (32 bit integer).
88 - 91	Reserved.
92 - 95	Number of clock cycles of JTAG bitstream (32 bit integer).
96 - 127	Reserved.
128	JTAG data containing TMS/TDI data alternating bytes, 8 cycles to one byte. That is, byte 128 contains the first 8 TMS bits, byte 129 contains the first 8 TDI bits, etc. Within each byte the bits are stored from the LSB to MSB, ie., bit 0 of byte 128 is the cycle 0 of TMS value, bit 0 of byte 129 is the cycle 0 of TDI value, etc. There will always be an even number of data bytes. Partial bytes at end are filled with all 0's.

CHAPTER 7

7.0 Host / Micro-controller Software

The host software is responsible for sending commands to the prototyping board. The communication link between the host and a prototyping board is through the parallel port. The commands that the host can performs are as followed:

- Get Status.
- Write Configuration .
- Read Configuration .
- Load Interface FPGA configuration bit file.
- Load FPGA #1 configuration bit file.
- Load FPGA #2 configuration bit file.
- Load FPID configuration bit file.
- Program the Interface FPGA.
- Program FPGA #1.
- Program FPGA #2.
- Program both FPGAs .
- Program FPID .
- Program all FPD devices.
- Xchecker for FPGA #1.
- Xchecker for FPGA #2.
- Step Clock .

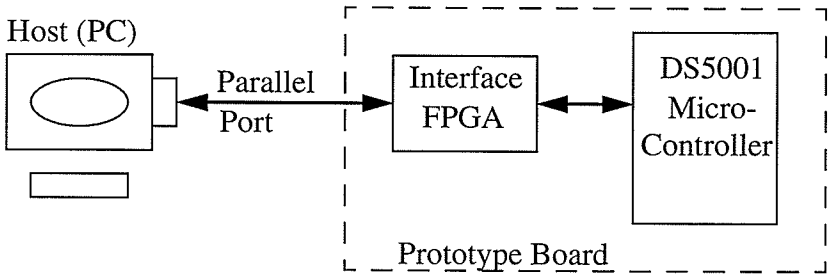


Figure 23: Communication Interface between Host and Prototyping Board

7.1 Communication Handshaking Between the Host and the Prototyping Board

The communication interface between the host and a prototyping board is shown in Figure 23. The standard PC parallel port is a unidirectional port. The parallel port has three independent buses. They are data out bus "D[0...7]" configured as output, status bus "Status(0...3)" configure as input, and control bus "Control(0...3)" configure as output. D[0...7], Status(0...3), and Control(0...4) are located at 0x378, 0x379, and 0x37A respectively at the I/O memory address of the PC. D[0...7] is used to write out a byte to the parallel port, Status(0...3) is used to read back a nibble, therefore in a read cycle there will be two consecutive reads, one for the upper nibble and one for the lower nibble. Control(0...4) is used as a controlling signals interfaced to the prototyping board. The controlling signals are write strobe, read strobe, and address decoding located at bit postionS 0, 1, and 2-4 respectively. The following shows the signal description for Control(0...4).

Bit	Description
0	Write strobe: Active low.
1	Read strobe: Active low.
2 ... 4	000 => Select Control Register.
	001 => Select Write Buffer Register.
	010 => Select Read Buffer Register (lower nibble).
	011 => Select Read Buffer Register (higher nibble).
	100 => Select XCchecker for FPGA #1.
	101 => Select XCchecker for FPGA #2.
	110 => Select Clock Step function.

Write strobe: Whenever a write to a parallel port address occurs this signal should pull low. Its default state is logic high.

Read strobe: Same as write strobe.

Control Register: This register stores the status of the handshaking communication. It is an eight bit register. This is a read only register by either the host or the micro-controller DS5001. The following is the bit definition of this register:

- Bit 0: 'Host Transmit Register Empty'. The Host Transmit Register is empty whenever this bit is set. The host can only write to the parallel port when this bit is set. When a host writes to the parallel port this bit is cleared and it can only be set when the micro-controller does a read.
- Bit 1: 'Host Receive Register Full (lower nibble)'. The Host Receive Register is full when ever this bit is set. The host can only read from the parallel port when this bit is set. When the micro-controller writes to the parallel port, this bit is set and it can only be cleared when the host does a read.
- Bit 2: 'Host Receive Register Full (upper nibble). Same as the above
- Bit 3: 'Slave Transmit Register Empty'. The Slave Transmit Register is empty whenever this bit is set. The micro-controller can only write to the parallel port when this bit is set. This bit can only be cleared when the host does two consecutive reads from the Host Receive Register for the lower and upper nibble.
- Bit 4: 'Slave Receive Register Full'. The Slave Receive Register is full whenever this bit is set. The micro-controller can only read the Slave Receive Register when this bit is set. When a host writes to the parallel port this bit is set and it is clear when the micro-controller reads the Slave Receive Register.
- Bit 5: 'Interface FPGA DONE pin'. This bit indicates the status of the configuration of the Interface FPGA. Logic 1 indicates a proper configuration, logic 0 indicates an error in configuration.
- Bit 6: 'FPGA #1 DONE pin'. This bit indicates the status of the configuration of the FPGA #1. Logic 1 indicates a proper configuration, logic 0 indicates an error in configuration.
- Bit 7: 'FPGA #2 DONE pin'. This bit indicates the status of the configuration of the FPGA #2. Logic 1 indicates a proper configuration, logic 0 indicates an error in configuration.

The handshaking state machines and the registers are designed in the Interface FPGA which is a XC3042 part. The Interface FPGA controls the set and reset of all the bits in the registers and generates interrupts to the micro-controller when ever the host does a write.

Host Write Cycle: The host begins the write cycle by polling bit 0 of the control register until this bit is set. When this is true; the data is latched to D(0...7), the write strobe is pulled low and Write Buffer Register address is selected. This generates an interrupt to micro-controller.

Host Read Cycle: The host begins the write cycle by polling bit 1 of the control register until this bit is set. When this is true; the read strobe is pulled low, the Read Buffer Register (lower nibble) address is selected and the data is read through Status(0...3). The same process is done when the host reads the Read Buffer Register (upper nibble).

7.2 Get Status Command

The Get Status Command allows the user to retrieve information from each of the FPDs on the prototyping board. The information that the user can get back are the Revision number, the Date , the Time , and the Filename of a particular file. This could be any of the configuration bit files of the Interface FPGA, FPGA#1, FPGA#2 or FPID devices.

7.3 Write/Read Configuration Command

The Write Configuration Command allows the user to set the configuration of the programming mode for the FPDs. The configuration can be set so that the configuration bit file for interface FPGA can be downloaded after reset or after power up. Another configuration is to download all of the configuration bit files to all of the FPDs or selectively download any of the configuration bit files. The default state is the configuration to download to all of the FPDs. This configuration is stored in the memory. When the micro-controller comes out of reset the first thing that it looks for is the configuration command information.

The Read Configuration Command allows the user to retrieve the Configuration Command information on how the prototyping board is configured.

7.4 Load Interface FPGA Configuration Bit File

This command allows the user to download the interface FPGA configuration bit file to

the prototyping board. When this command is executed , it reads the MCS file which is the configuration bit file in Intel Hex format and writes a byte at a time to the prototyping board for downloading. This commands does not program the interface FPGA. The configuration is downloaded and stored in the memory of the prototyping board. This allows the user the flexibility to redesign the interface FPGA to suit his application requirement. This is one of the advantages using FPGAs in a prototyping environment. Besides the configuration bit file being downloaded, the header of this file contains the Revision Number, Date, Time , Filename and the number of bytes in the configuration bit file. This information is useful when the user wants to read back the status of the interface FPGA.

7.5 Load FPGA #1 / FPGA #2 Configuration Bit File Command

This command is similar to the above command. The advantage of this command is that it allows the user to modify the FPGA and download the new configuration bit while the system is still in operation. One exception that the user must follow is that the I/O names must be assigned to the pin number as in the previous design.

7.6 Load FPID Configuration File Command

The process for this command is similar to the previous one. It allows the user to download the new interconnect netlist to the prototyping board. For example, some signals that needed to be monitored using a logic analyzer or scope can be made by modifying the NLT file and use FPIDComp to compile a new configuration bit file . This modified bit file is downloaded to the prototyping board within a matter of minutes.

7.7 Program Interface FPGA Command

This command lets the user program the interface FPGA. The programming for the interface FPGA can be made by executing this command. Another way is setting the configuration of the prototyping board. When the micro-controller receivedsthis command; the programming of the interface FPGA is executed until all of the bits are fully downloaded to the FPGA and the interface FPGA DONE pin bit is set in the control register . This is accomplished by monitoring the DONE pin on the FPGA. If this pin transitions to a logic high after all the bits have been

downloaded, then FPGA device is properly configured. During this process the micro-controller is generating the configuration bit DIN and the CCLK signals.

7.8 Program FPGA #1 / FPGA #2 Command

This command is similar to the above command. The only difference is that instead of bit bashing out the configuration bit file , it latches out a byte at a time and a state machine in the Interface FPGA will generate a CCLK and a DIN signals to download the configuration bit file. It takes eight clock cycles to download a byte to the FPGA#1 or FPGA#2 with a 10MHz clock rate. This clock rate is a lot faster than the execution rate of the micro-controller; so we don't need to generate interrupt to the micro-controller every time a byte is shifted out. The micro-controller only needs to get a byte of data and latch that data out until the end of the buffer.

7.9 Program Both FPGAs Command

This command is executed when the user wants to program both FPGAs at the same time. The process of this command is similar to the above. FPGA#1 is programmed first then follow by FPGA#2.

7.10 Program FPID Devices Command

This command let the user download the configuration bit file to the prototyping board. The process for this command is similar to the above. The only difference is that the state machine in the Interface FPGA needs to receive two bytes of data the TMS and TDI byte before it can download the bit file to the FPID devices. The interface FPGA is responsible for generating TMS, TDI and TCK signals.

7.11 Xchecker for FPGA #1 / FPGA #2 Command

This command allows the user to execute the Xchecker functionality provided by Xilinx software. The circuit in the interface FPGA will route the appropriate Xchecker signals to either one of the FPGAs depending on the Command it receives. Figure 24 shows the Xchecker interface between the interface FPGA and both of the FPGAs. The XChecker provides the following capability:

- Allows the user to download a design to the Logic Cell Array (LCA) on the Target System.
- After configuration of an FPGA, the Xchecker can verify its configuration by comparing it to the original design.
- The user can probe an LCA's internal logic with XChecker to debug the design. Probing is the execution of a readback of all the configuration data and extracting the internal logic states of desired signals from it.

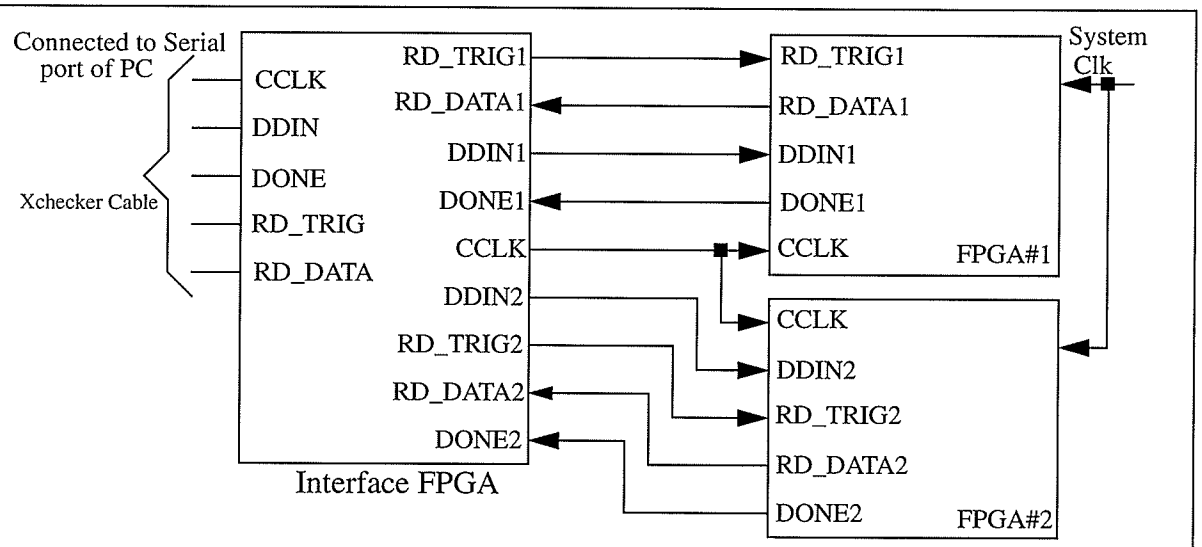


Figure 24: Xchecker Interface between the Interface FPGA and FPGA#1 and FPGA#2

For downloading configuration data, the XChecker can be used with a single FPGA, or several connected in a daisy chain. For the prototyping system developed here there is only one FPGA that can be configured at a time. This is compatible with the XChecker when it is used to read back data or as a logic probe, the XChecker can only be used with one device at a time. The functionality for each signal of the XChecker is described below:

- CCLK: Configuration Clock provides configuration clock to the target system during the configuration and readback.
- DDIN: Data In provides configuration data to the Target System during configuration and is 3-state at all other times.

- DONE: DONE program provides an end of configuration.
- RD_TRIG: Read Trigger initiates a read back by causing a Low-to-High transition at the target LCA RTRIG pin.
- RD_DATA: Reads back data from the target LCA is read at this pin RDATA.

All of the commands for executing the XChecker can be obtained from the XACT Hardware and Peripheral Guide in Chapter 5.

7.12 Clock Stepping Command

This command allows the user to enter the number of clock cycles to be executed. The clock cycle ranges from 1 to 65,000 cycles or runs in a continuous mode. Figure 25 shows the interface of the system clock to both of the FPGAs.

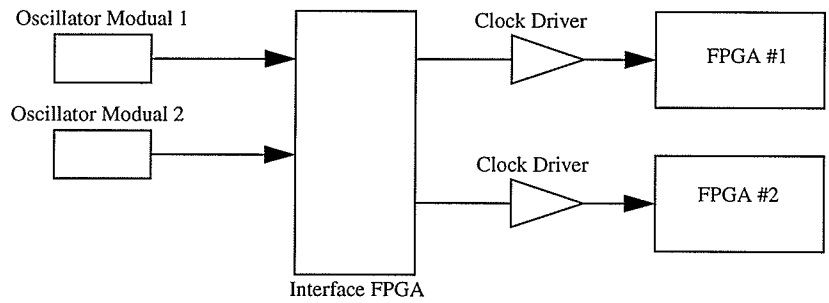


Figure 25: System Clock interface to FPGA

Another option is that the user can choose to run both FPGAs with a single clock source or run each FPGA with an individual clock. The Interface FPGA circuit will determine how many clock cycles will be executed and which oscillator will be routed to the FPGA. This is dependent on the command given by the user. On the prototyping board the oscillator module is not soldered down therefore the user can change the speed of the system with ease.

CHAPTER 8

8.0 The Prototyping System Performance and Testing

The performance of the prototyping system was tested using a divided by two counter with the output of the flip-flop routed through the I-Cube#2 to I-Cube#1 to FPGA#1 and through I-Cube#3 to the input of the flip-flop as shown in Figure 26.

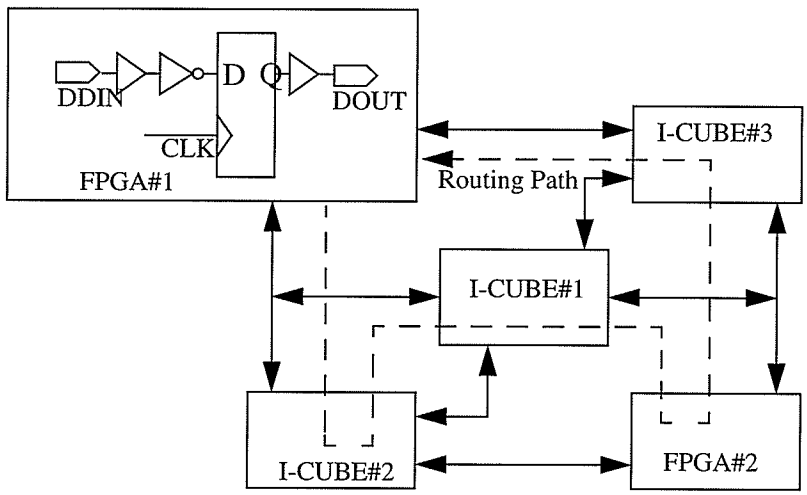


Figure 26: Prototyping System Performance Test Circuit

This test circuit provides the maximum possible delay through the three FPID devices. The divide by two counter is resideNT in FPGA#1 while FPGA#2 only provideS a buffer through input-output The input clock to the flip-flop was generated by the function generator to create various clock frequencies. The DIN, DOUT, and CLOCK signals were captured using a Technotrix TDS510 model capable of storing waveform information. The waveform information will be downloaded to a wave plotter to create a hardcopy. The tested clock frequencies were 5 , 10 , 12 , 13, and 14 MHz. The result of difference frequency test results are shown in Appendix C. From the test result the performance of the prototyping system starts to fail when the clock frequency is at 14 MHz, this is caused by the fact that the input data transition to the input of the flip-flop did not meet the setup time specify by Xilinx. Figure 27 shows the timing diagram of the test circuit.

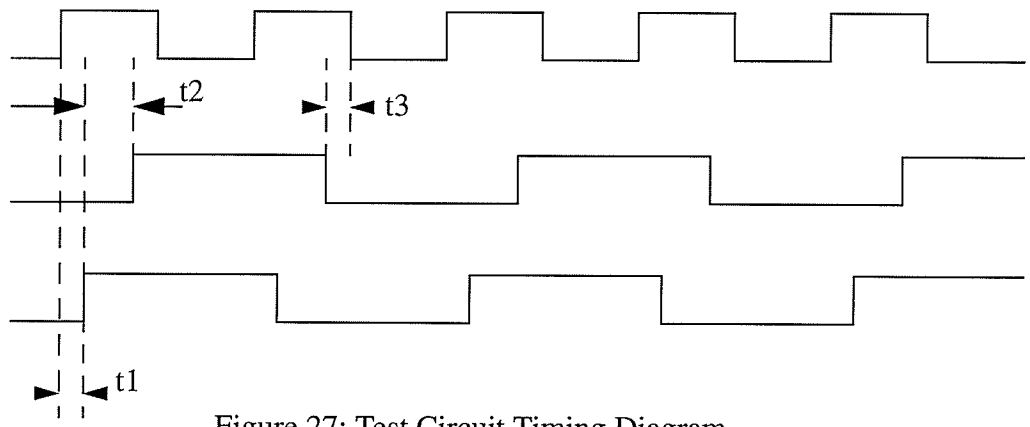


Figure 27: Test Circuit Timing Diagram

t1 = Time delay from an input pad to an output pad of the XC4013-6.
t2 = Time delay from DOUT through three FPID devices to DIN.
t3 = Setup from DIN to positive edge of CLOCK.

Table 1:

Frequency	t1	t2	t3
5 MHz	30 ns	40 ns	30 ns
10 MHz	30 ns	40 ns	30 ns
12 MHz	30 ns	40 ns	13 ns
13 MHz	30 ns	40 ns	7 ns
14 MHz	30 ns	40 ns	2 ns

Table 1 shows the result created by the test circuit. The test circuit begins to fail at 14 MHz clock because of the time violation at t3; for XC4013-6 the minimum for t3 is 6 ns [8]. When there is a time violation at t3 the output of the flip-flop becomes tristate as shown in the plot in the Appendix C. Even though the test circuit passed at 13 MHz, for a more complex design the recommended frequency should somewhere between 8 - 10 MHz allowing more setup time.

8.1 Testing the Prototyping System

The prototyping system was tested for manufacturing by designing a state machine to generate a 96 bit output of walking ones and zeros within an FPGA. The design was running at a 2 Hz internal clock of the FPGA. This is used for the visualization of the LEDs mounted on the header. The test design generate a walking ones pattern for 96 clock cycles then a walking zeros for 96 clock cycles and keeps repeating. The outputs of the FPGA#1 are connected to the FPGA#2 which acts as a buffer to drive the LEDs mounted of the Header#1 and Header#2. Figure 28 shows the routing path of the walking ones and zeros test circuit.

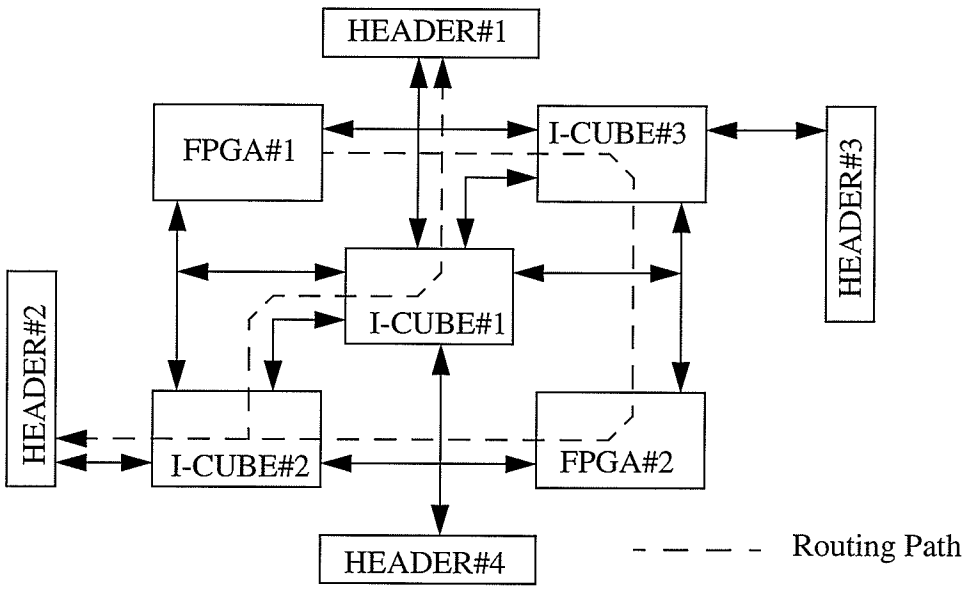


Figure 28: Walking Ones and Zeros Routing Path

The purpose of this testing is to test out the routing software and secondly test out the operation of the hardware. This test provides a testing checklist for the commands provides by the prototyping system. The following commands were sucessfully passed when running the walking ones and zeros test are; get status, write configuration, read configuration, load interface FPGA configuration bit file, load FPGA#1 and #2, program interface FPGA, program FPGA#1, program FPGA#2 , and program all FPD devices. Appendix D shows the schematic capture for both FPGAs and the target system. It also shows the files generated by the routing software. The files are originally in text then reformat to smaller font for easy display.

CHAPTER 9

9.0 Conclusion and Recommendations

Increasing ASIC complexity and relentless time-to-market pressures complicate design prototyping and verification. Fortunately, new solutions are arising that allow designers to test, debug and verify their ASIC designs rapidly, in an environment that closely resembles the final product.

The time tested approaches of hardware breadboarding and simulation are still important methods, but they have some shortcomings. With today's designs, the traditional approach to building a breadboard prototype can be expensive and error-prone. And if bugs or design errors are found, making changes can be difficult and time consuming.

Simulation is also a powerful approach to verify a design, since it provides the flexibility that breadboarding lacks. But the effectiveness of simulation depends heavily on models, and models for a newly introduced device may not be immediately available. Simulation also take time, and there are many real-world applications that can not be effectively simulated.

The prototyping system developed here provides the designer with new prototyping solutions through the use of FPGA and FPID devices. This prototyping solution can be considered as an in-circuit emulation technology. The reprogrammable hardware is automatically configured to emulate the functionality represented by an ASIC netlist. The capability resembles that provided by microprocessor in-circuit emulation except that the functionality of the system can be reprogrammed quickly to emulate various ASIC designs from their netlist.

The prototyping system developed is a low cost ASIC in-circuit emulator. It provides the designer a flexible and a generic testing environment due to the fact that the components are all reprogrammable.

Figure 29 shows the front view and Figure 30 shows the rear view of the prototyping system as a complete product.

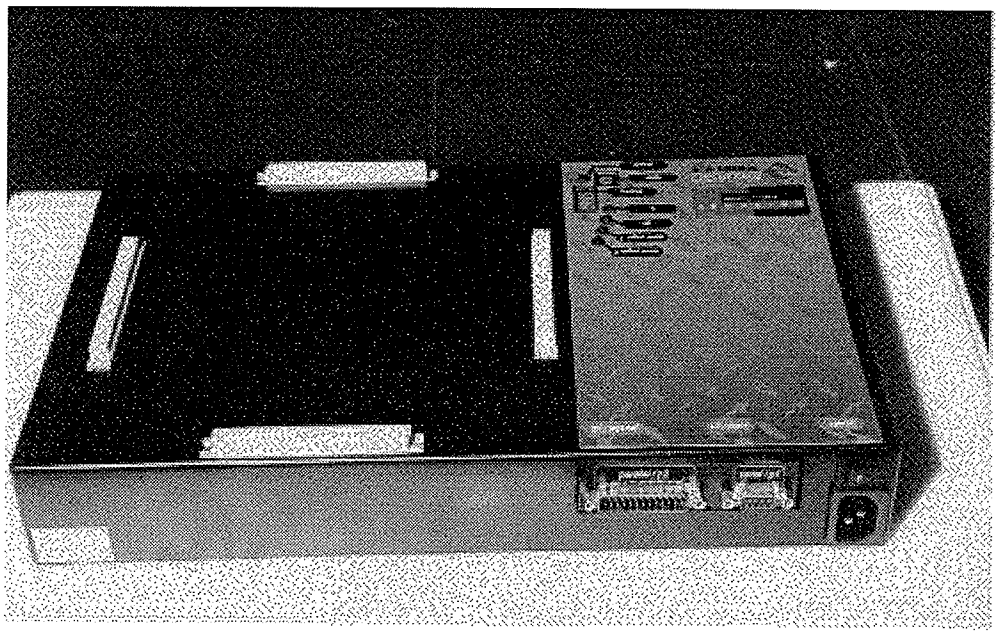


Figure 29: Front View of the Prototyping System

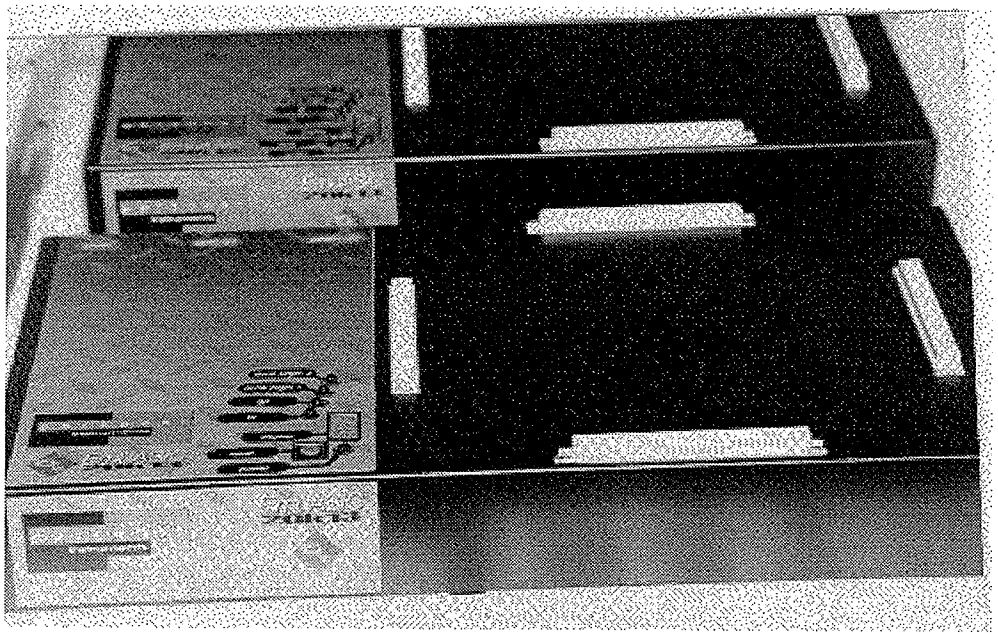


Figure 25: Rear View of thePrototyping System

Recommendations

After a period of writing software and debugging this prototyping system I've found that there are some areas where this system could be modified. The areas that I think need improvement are:

- Using an enhanced parallel port : This is a bi-directional parallel port where the data bus is bidirection. This will speed up downloading the configuration bit file considerably. Using the enhanced parallel port would reduce the gate count on the handshaking circuitry in the Interface FPGA.
- Reconnect the way the FPGAs and the FPIDs are interconnected together. The current system presents some routing problems. Refer to Figure 10, for example after the place and route on both of the FPGAs, the result of the netlist tells us that all the signals on bus 'F' must be connected to all of the signals on bus 'H'. This will provide a fatal routing path. We can see that in order to route from bus 'F' to 'H', the signal path must travel from bus 'F' through FPID #1 to bus 'J' through FPID #3 to bus 'M' through FPID #2 then terminate on bus 'H'. The problem arose when the signals travel through bus "J' and "M' which only provides 64 I/Os for interconnect while the bus 'F' can commodate upto 96 I/Os. Since we may need to connect 96 I/Os on bus 'F' to 96 I/Os on bus 'H' we are 32 I/Os short. This is the bottle neck of the routing resources on the prototyping board. I recommend that we reconnect 32 I/Os on bus G' to bus 'J' increasing this bus to 96 I/Os and the same solution on bus 'E' and 'M'. When we make this modification the I/Os on both Header #2 and #4 will reduceD to 32 I/Os each and providing 192 I/Os to connect to the target system.
- UsE a bigger part for Interface FPGA device. We could increase this part to a XC3064 or even some XC4000 series device. At this time the Interface FPGA is about 85% utilized.
- Connect to the Interface FPGA to at least 16 I/Os from either one of the Headers. This will give us the option of triggering on certain events. With this option the prototyping system is functioning as a logic analyzer. There will be a state machine implementing this function in the Interface FPGA. This function can be used in conjunction with the

XChecker software. For example when the system detects the triggering event, it stops the system clock and using the XChecker reads back the internal states of the design in the FPGA.

- A software routine to translate a state logic of a signal or a bus into waveform format for the ease of visualize.
- Routing software to translate the schematic netlist from Mentor Graphic , Cadence or View Logic to I-Cube configuration bit format.

REFERENCES

[1] Walters, S., "Reprogrammable Hardware Emulation Automates System Level ASIC Validation", Electronics, May, 1990, pp120.

[2] Wolf, H., "How Quickturn is Filling the Gap", Electronics, April, 1990, pp 70.

[3] Aptix Corporation, Aptix Programmable Interconnect Product System Data Book, 1993.

[4] Slimane-Kadi, M., Brasery, D., and Saucier, G., "A Fast-FPGA Prototyping System That Uses Inexpensive High-Performance FPIC", Presented at The 2nd International ACM/ SIGDA Workshop on FPGAs, Feb. 13-15, 1994.

[5] I-Cube Inc., The FPID Family Data Sheet, 1994.

[6] Xilinx Inc., The Xilinx Data Book, 1994.

[7] Dalas Semiconductor Inc., The DS5001 Micro-controller Handbook, 1994

[8] Xilinx Inc., XACT Reference Guide, Volume 1-3, 1994.

[9] I-Cube Inc., IDS100 User's Guide, 1994.

[10] Stephen S., Michael P., D'Amour R., , and Payne T., " Apparatus for Emulation of Electronic Hardware System," Quickturn Design System, Inc.,

[11] I-Cube Inc., "Using FPID Devices in FPGA base Prototyping", Application Notes, Feb., 1994.

[12] Butts, M., Jon Batcheller, and Joseph Varghese, "An Efficient Logic Emulation System," 1992 IEEE International Conference. on Computer Design, pp. 138 ~ 141.

[13] Butts, M., and J. Batcheller, "Method of Using Electronically Reconfigurable Logic Circuits." Mentor Graphic Corporation, July 30, 1991.

[14] I-Cube Inc., "mFPID Design Tools Specification".

[15] Butts, M., "Applications of SRAM-Based FPGAs: Use FPGAs for ASIC Implementation" ICCAD '93 Tutorial.

[16] Pasternak, D. and Hike, T. "In-Circuit Emulation in ASIC Architectural Core Designs. " presented at the Second Annual IEEE Seminar and Exhibit, Rochester, New York, Setember 25-28, 1989.

[17] Small, Charles, "User-Programmable Gate Arrays, " EDN, April 27, 1989, pp 146.

[18] K. Shirai and T. Takezawa, and S. Ueno, "Knowledge Based Techniques on a Development System of ASIC," Proc. of IEEE/SICE Int. Workshop on Artificial Intelligence for

Industrial Applications, May 1988.

[19] J. Babb, R. Tessier, and A. Agarwal, "Virtual Wires: Overcoming Pin Limitations in FPGA-based Logic Emulators", In Proceedings, IEEE Workshop on FPGA-based Custom Computing Machines, pp 142 ~ 151, Napa, CA, April 1993. IEEE.

[13] J. Babb, R. Tessier, and A. Agarwal, "The Virtual Wires Emulation System: A Gate Efficient ASIC Prototyping Environment", 1994 ACM International Workshop on Field Programmable Gate Arrays, Feb. 1994..

GLOSSARY

FPID	Field Programmable Interconnect Device.
FPGA	Field Programmable Gate Array.
FPD	Field Programmable Device.
ASIC	Application Specific Integrated Circuit.
PCB	Printed Circuit Board.
I/O	Input / Output.
SRAM	Static Random Access Memory.
TDI	Test Data In.
TMS	Test Mode Select.
TCK	Test Clock.
TRST	Test Reset.
CLB	Configurable Logic Block
IOB	Input Output Block.
LCA	Logic Cell Array.
RPB	Rapid Prototyping Board.
PC	Personal Computer.
VHDL	(VHSIC) Hardware Description Language.
NVRAM	Non-volatile Random Access Memory.
ROM	Read Only Memory.

APPENDIX A

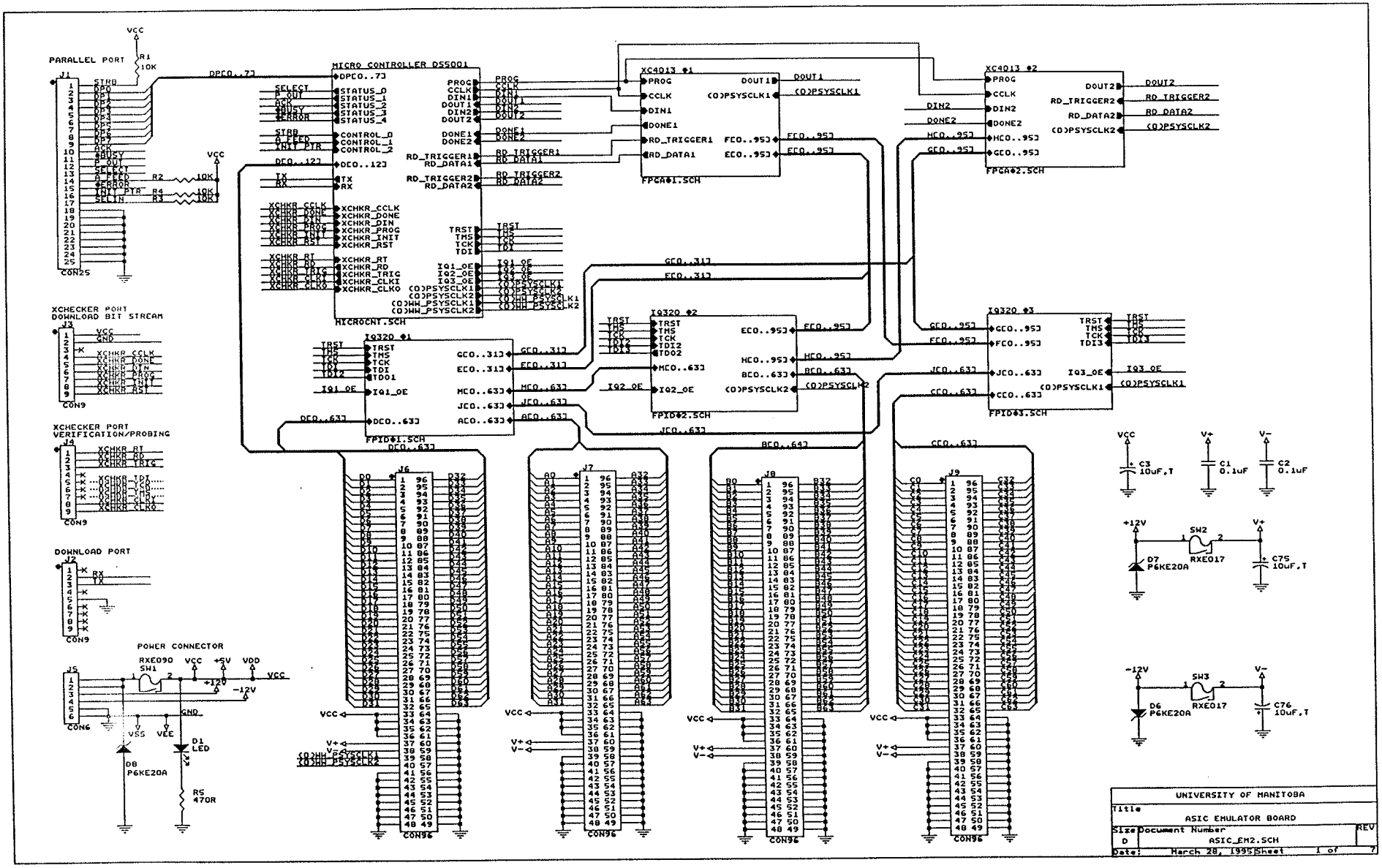
Interconnection Cases

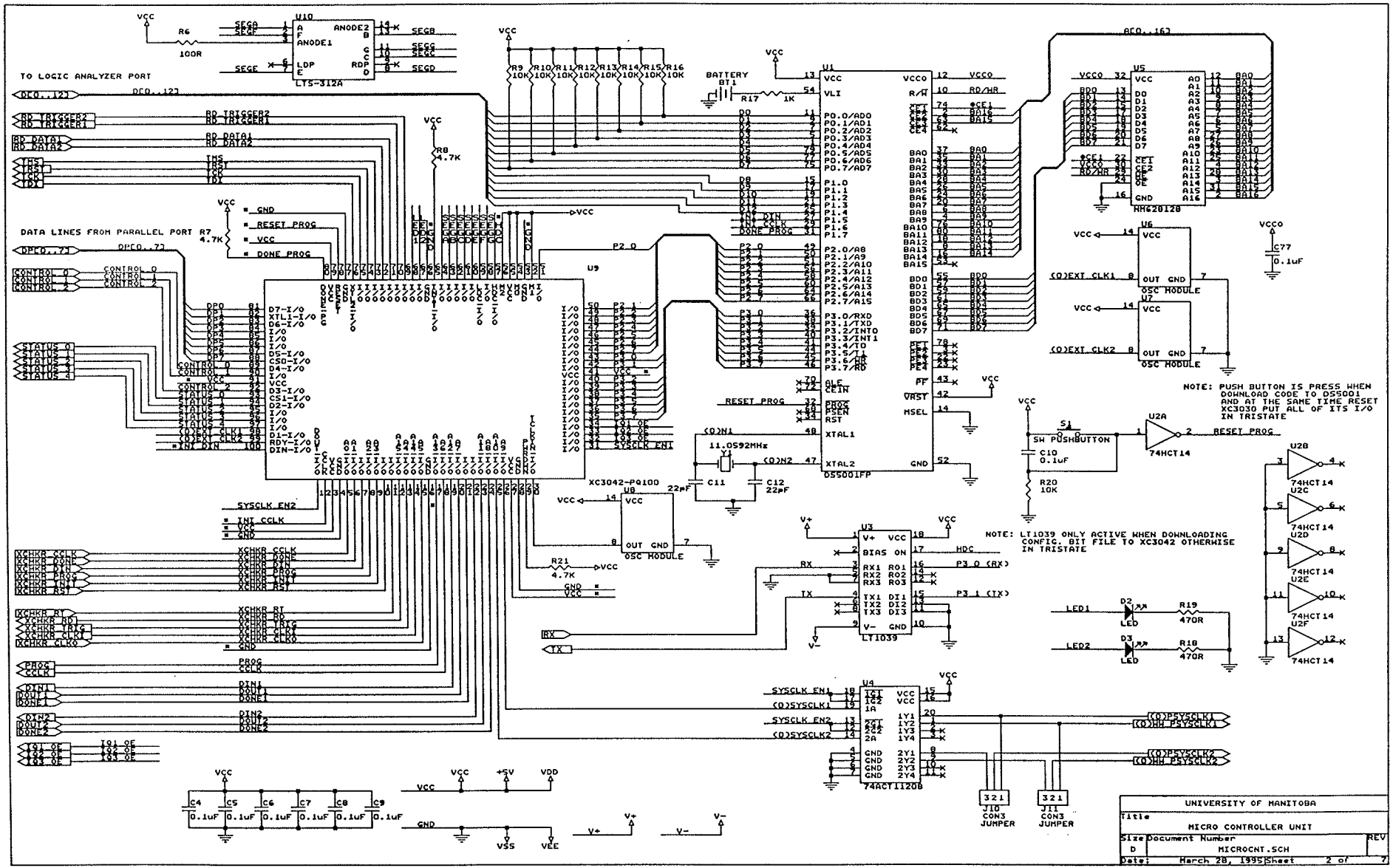
The followings lists all of the possible connection case in the prototyping board.

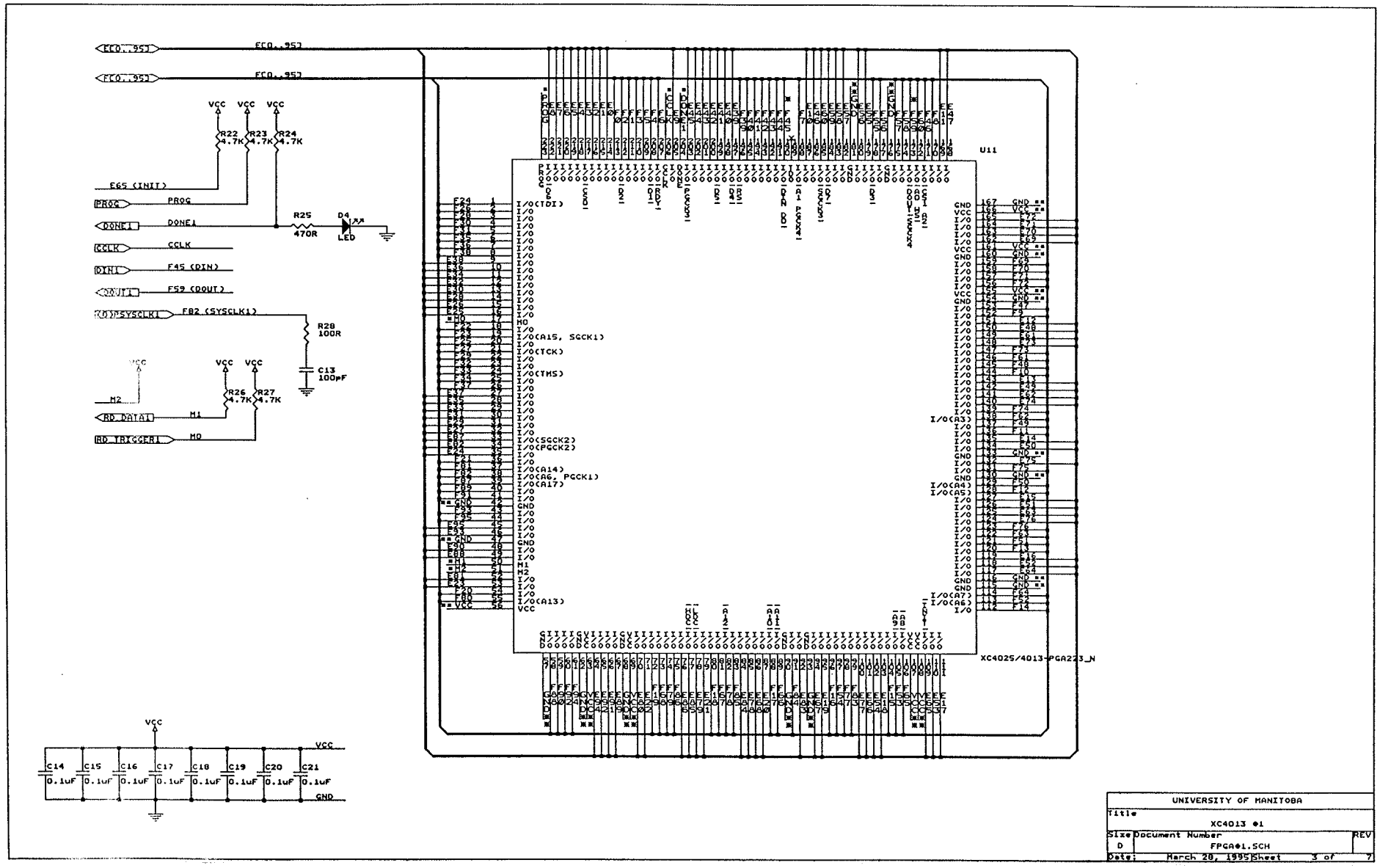
Case 1	G -> F	Case 2	F - > G	Case 3	G - > E
Case 4	E -> G	Case 5	G' - > E	Case 6	E - > G'
Case 7	G -> E'	Case 8	E' - > G	Case 9	G' - > E'
Case 10	E' -> G'	Case 11	H - > F	Case 12	F - > H
Case 13	H -> E	Case 14	E - > H	Case 15	G -> A
Case 16	A -> G	Case 17	G' - > A	Case 18	A - > G'
Case 19	G -> B	Case 20	B - > G	Case 21	G' - > B
Case 22	B -> G'	Case 23	G - > C	Case 24	C - > G
Case 25	G -> D	Case 26	D - > G	Case 27	G' - > D
Case 28	D -> G'	Case 29	H - > A	Case 30	A - > H
Case 31	H -> B	Case 32	B - > H	Case 33	H - > C
Case 34	C -> H	Case 35	H - > D	Case 36	D - > H
Case 37	F -> A	Case 38	A - > F	Case 39	F - > B
Case 40	B -> F	Case 41	F - > C	Case 42	C - > F
Case 43	F -> D	Case 44	D - > F	Case 45	E - > A
Case 46	A -> E	Case 47	E' - > A	Case 48	A - > E'
Case 49	E -> B	Case 50	B - > E	Case 51	E - > C
Case 52	C -> E	Case 53	E' - > C	Case 54	C - > E'
Case 55	E -> D	Case 56	D - > E	Case 57	E' - > D
Case 58	D -> E'				

APPENDIX B

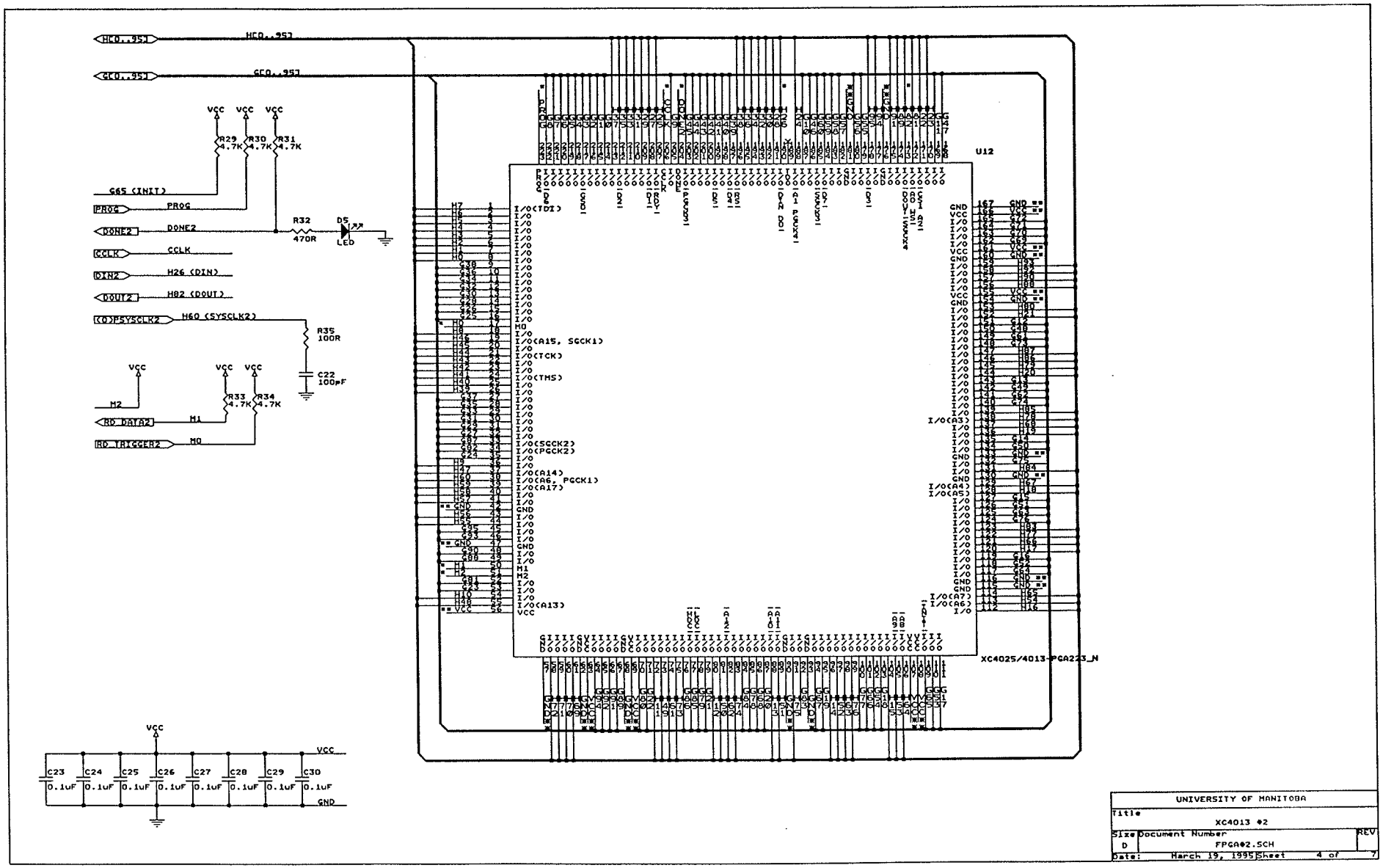
The Prototyping System Schematic

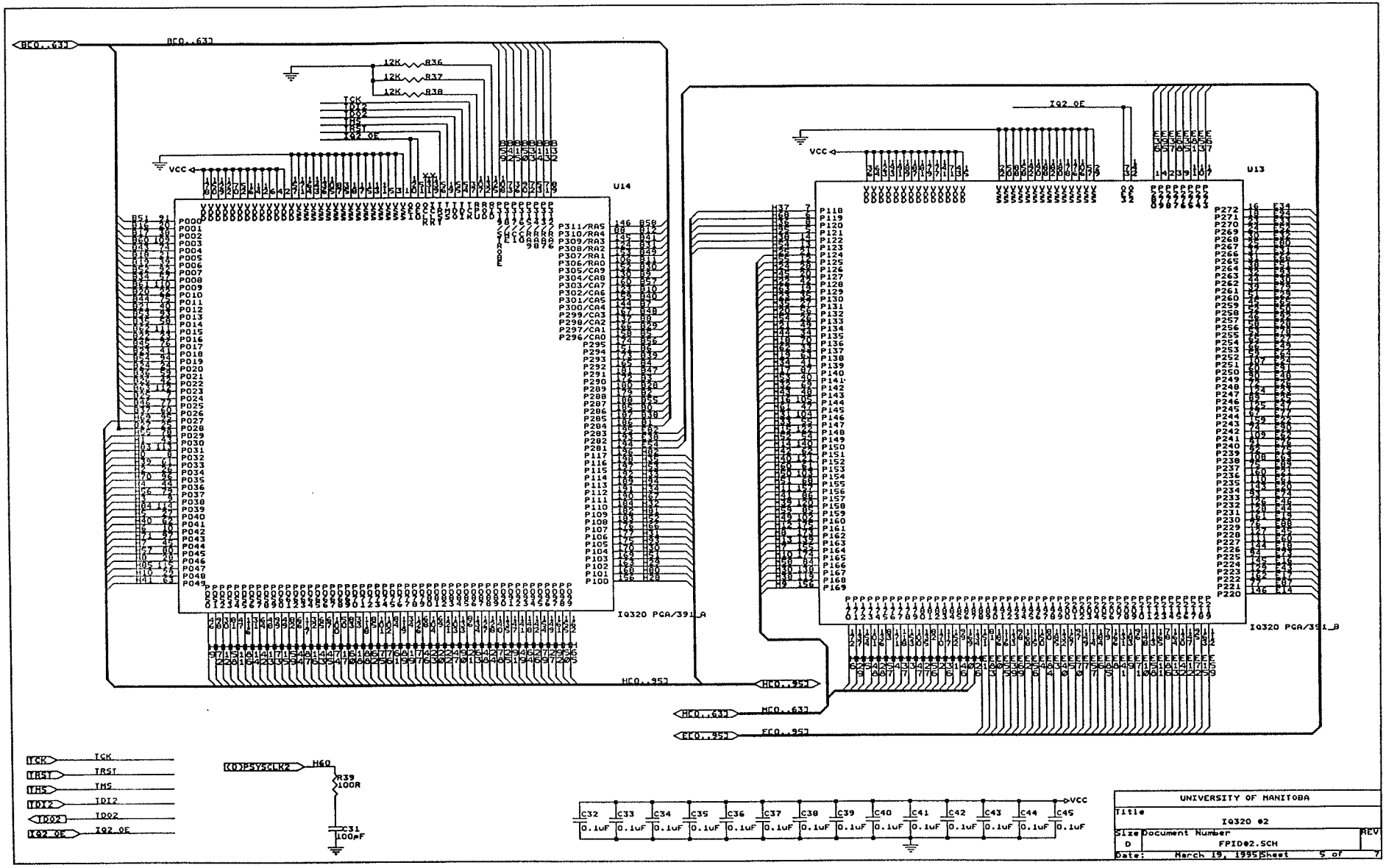

PARALLEL PORT
J1
CON25
MICRO CONTROLLER DS5001
MICROCNT.SCH
XC4013 #1
FPGA#1.SCH
XC4013 #2
FPGA#2.SCH
IQ320 #1
FPID#1.SCH
IQ320 #2
FPID#2.SCH
IQ320 #3
FPID#3.SCH
XCHECKER PORT
DOWNLOAD BIT STREAM
J3
CON9
XCHECKER PORT
VERIFICATION/PROBING
J4
CON9
DOWNLOAD PORT
J2
CON9
POWER CONNECTOR
J5
CON6
J6
J7
J8
J9
CON96
C3
10uF,T
C1
0.1uF
C2
0.1uF
SW2
RXE017
D7
P6KE20A
C75
10uF,T
SW3
RXE017
D6
P6KE20A
C76
10uF,T
D8
P6KE20A
D1
LED
R5
470R
UNIVERSITY OF MANITOBA
Title
ASIC EMULATOR BOARD
Size
D
Document Number
ASIC_EM2.SCH
REV
Date: March 28, 1995
Sheet 1 of 7


TO LOGIC ANALYZER PORT
DATA LINES FROM PARALLEL PORT
U10
LTS-312A
ANODE1
ANODE2
LDP
RDP
R6
100R
R7
4.7K
R8
4.7K
R9 R10 R11 R12 R13 R14 R15 R16
10K
BATTERY
BT1
R17
1K
U1
DS5001FP
U5
HM628128
U6
U7
OSC MODULE
U9
XC3042-PQ100
U8
OSC MODULE
11.0592MHz
Y1
C11
C12
22pF
R21
4.7K
S1
SW PUSHBUTTON
C10
0.1uF
R20
10K
U2A
74HCT14
U2B
U2C
U2D
U2E
U2F
NOTE: PUSH BUTTON IS PRESS WHEN DOWNLOAD CODE TO DS5001 AND AT THE SAME TIME RESET XC3030 PUT ALL OF ITS I/O IN TRISTATE
NOTE: LT1039 ONLY ACTIVE WHEN DOWNLOADING CONFIG. BIT FILE TO XC3042 OTHERWISE IN TRISTATE
U3
LT1039
LED1
LED2
D2
D3
R19
470R
R18
470R
U4
74ACT11208
J10
CON3
JUMPER
J11
CON3
JUMPER
C77
0.1uF
C4 C5 C6 C7 C8 C9
0.1uF
VCC
+5V
VDD
GND
VSS
VEE
UNIVERSITY OF MANITOBA
Title
MICRO CONTROLLER UNIT
Size D
Document Number
MICROCNT.SCH
REV
Date: March 28, 1995
Sheet 2 of 7

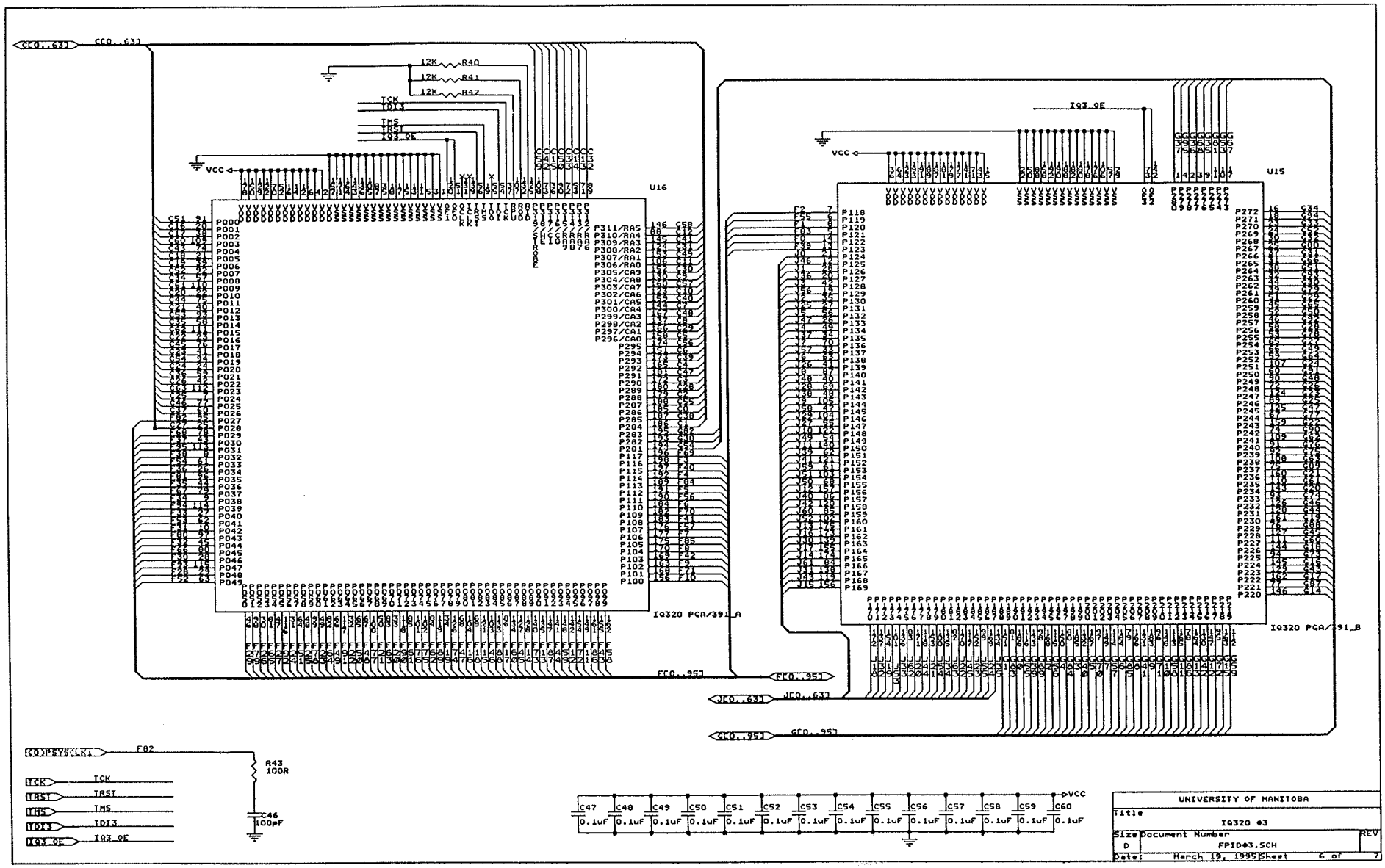


UNIVERSITY OF MANITOBA	
Title	XC4013 #1
Size D	Document Number FPGA#1.SCH REV
Date: March 28, 1995	Sheet 3 of 7


HC0..95
GC0..95
VCC
R29 4.7K
R30 4.7K
R31 4.7K
G65 (INIT)
PROG
DONE2
R32 470R
D5 LED
CCLK
H26 (DIN)
H02 (DOUT)
(CO)SYSCLK2
H60 (SYSCLK2)
R35 100R
C22 100pF
M2
R33 4.7K
R34 4.7K
RD_DATA2
M1
RD_TRIGGER2
M0
U12
XC4025/4013-PGA223_N
C23 0.1uF
C24 0.1uF
C25 0.1uF
C26 0.1uF
C27 0.1uF
C28 0.1uF
C29 0.1uF
C30 0.1uF
GND
UNIVERSITY OF MANITOBA
Title
XC4013 #2
Size D
Document Number
FPGA#2.SCH
REV
Date: March 19, 1995
Sheet 4 of 7



UNIVERSITY OF MANITOBA	
Title	IQ320 #2
Size D	Document Number FPID#2.SCH
REV	
Date: March 19, 1995	Sheet 5 of 7

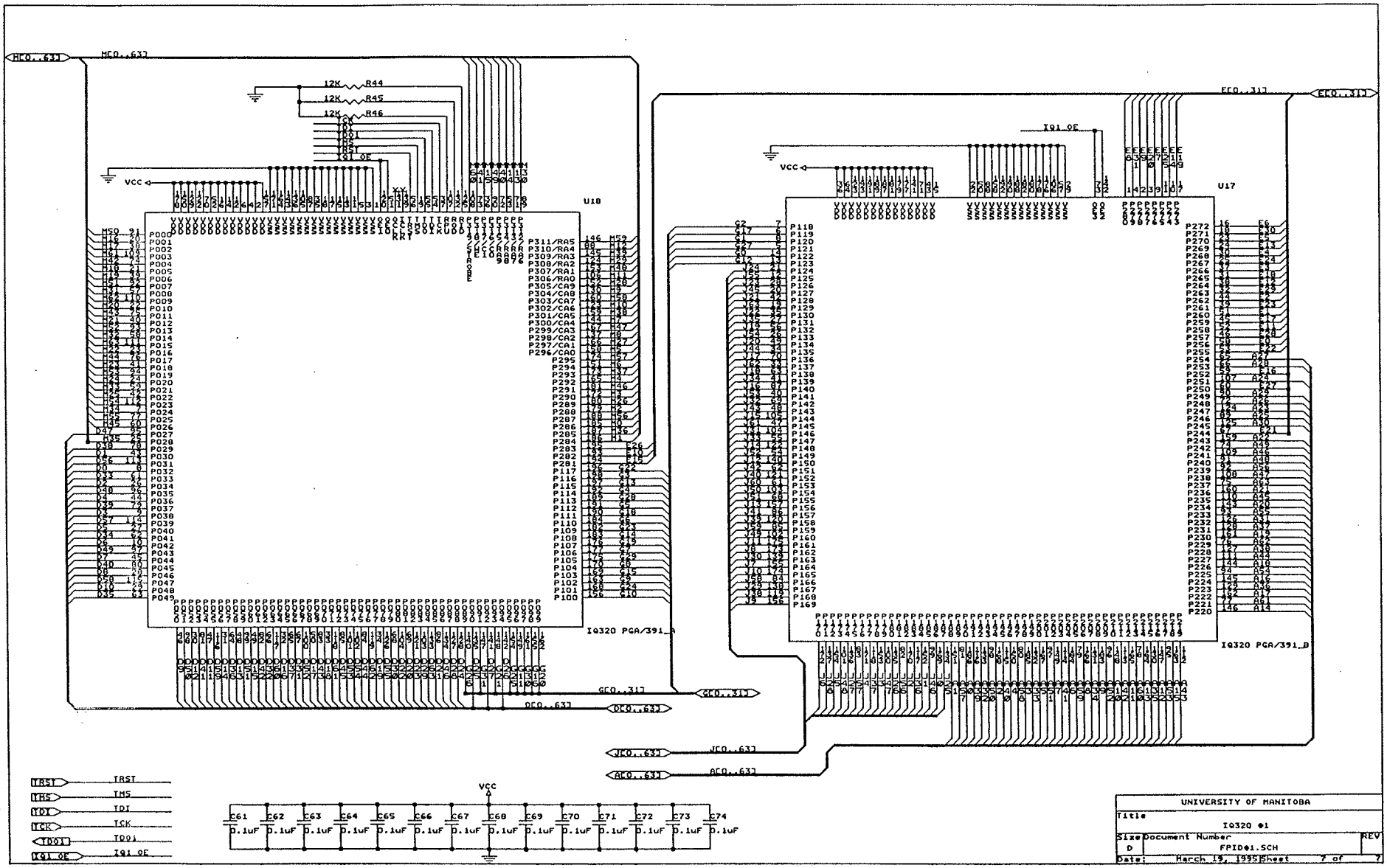


UNIVERSITY OF MANITOBA

Title: IQ320 #3

Size: D | Document Number: FPID#3.SCH | REV

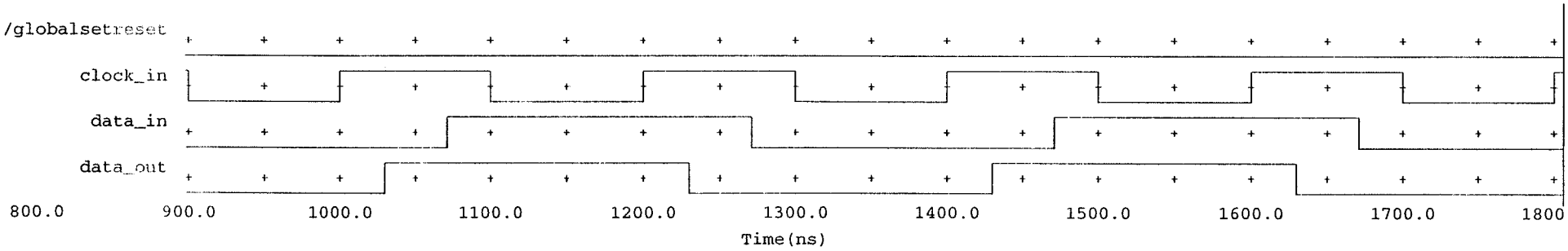
Date: March 19, 1995 | Sheet 6 of 7

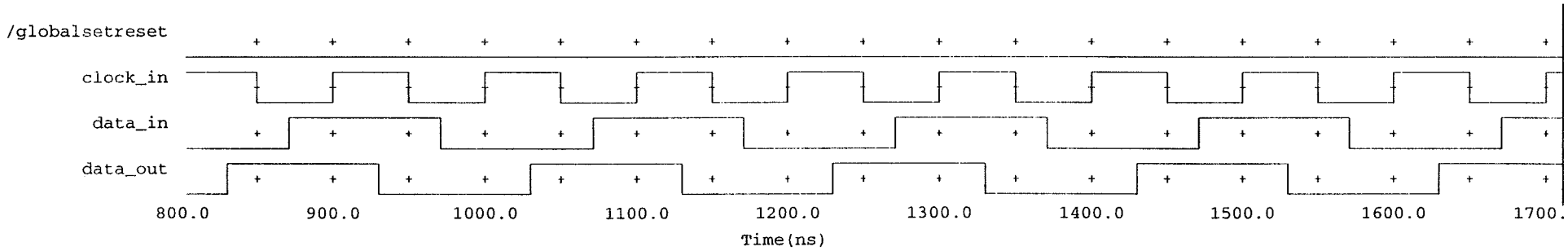

MC0..63
12K R44
12K R45
12K R46
VCC
U18
IQ320 PGA/391_A
FC0..31
U17
IQ320 PGA/391_B
GC0..31
DC0..63
JC0..63
AC0..63
TRST
TMS
TDI
TCK
TDO1
IQ1 OE
C61 C62 C63 C64 C65 C66 C67 C68 C69 C70 C71 C72 C73 C74
0.1uF
UNIVERSITY OF MANITOBA
Title
IQ320 #1
Size D
Document Number
FPID#1.SCH
REV
Date: March 19, 1995
Sheet 7 of 7

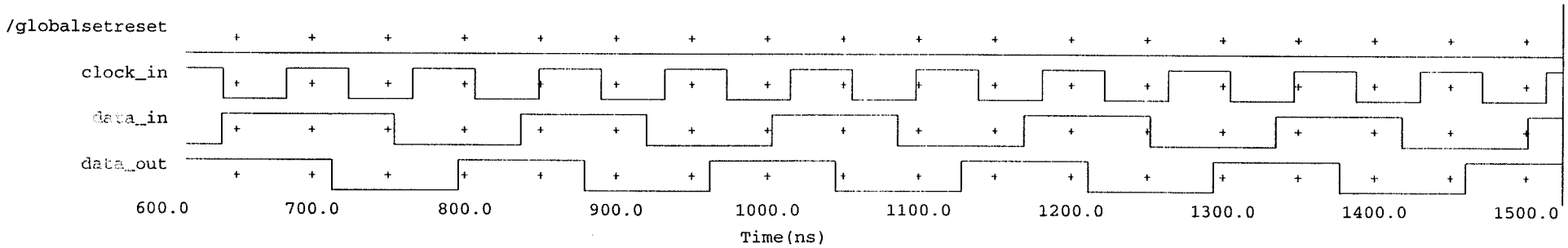
APPENDIX C

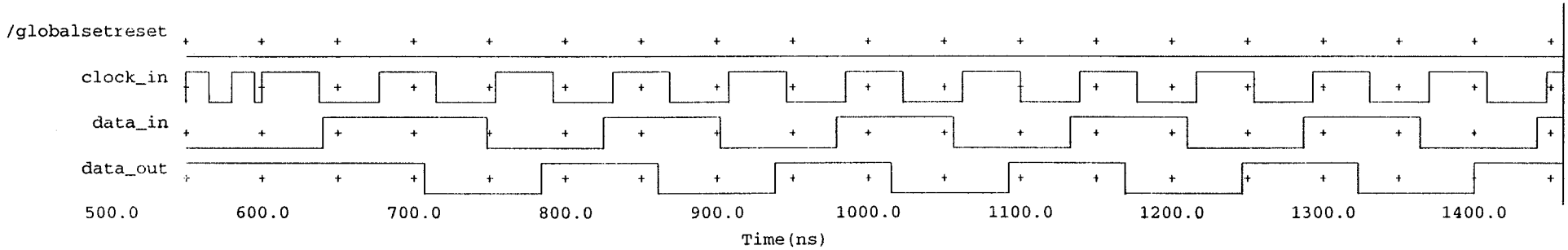
Hardcopy of Performance Test Result
and
Schematic Capture of Walking Ones and Zeros with Output Files of Routing Software

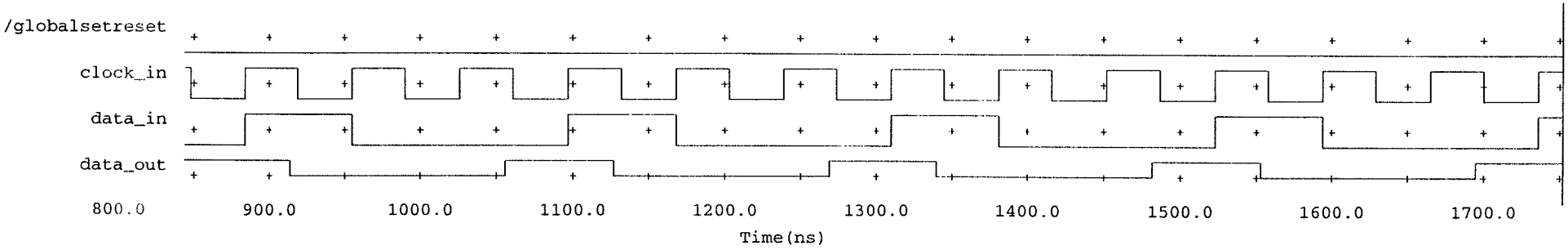
The waveform plots shows the performance test circuit for 5, 10, 12, 13 and 14 MHz clock frequency

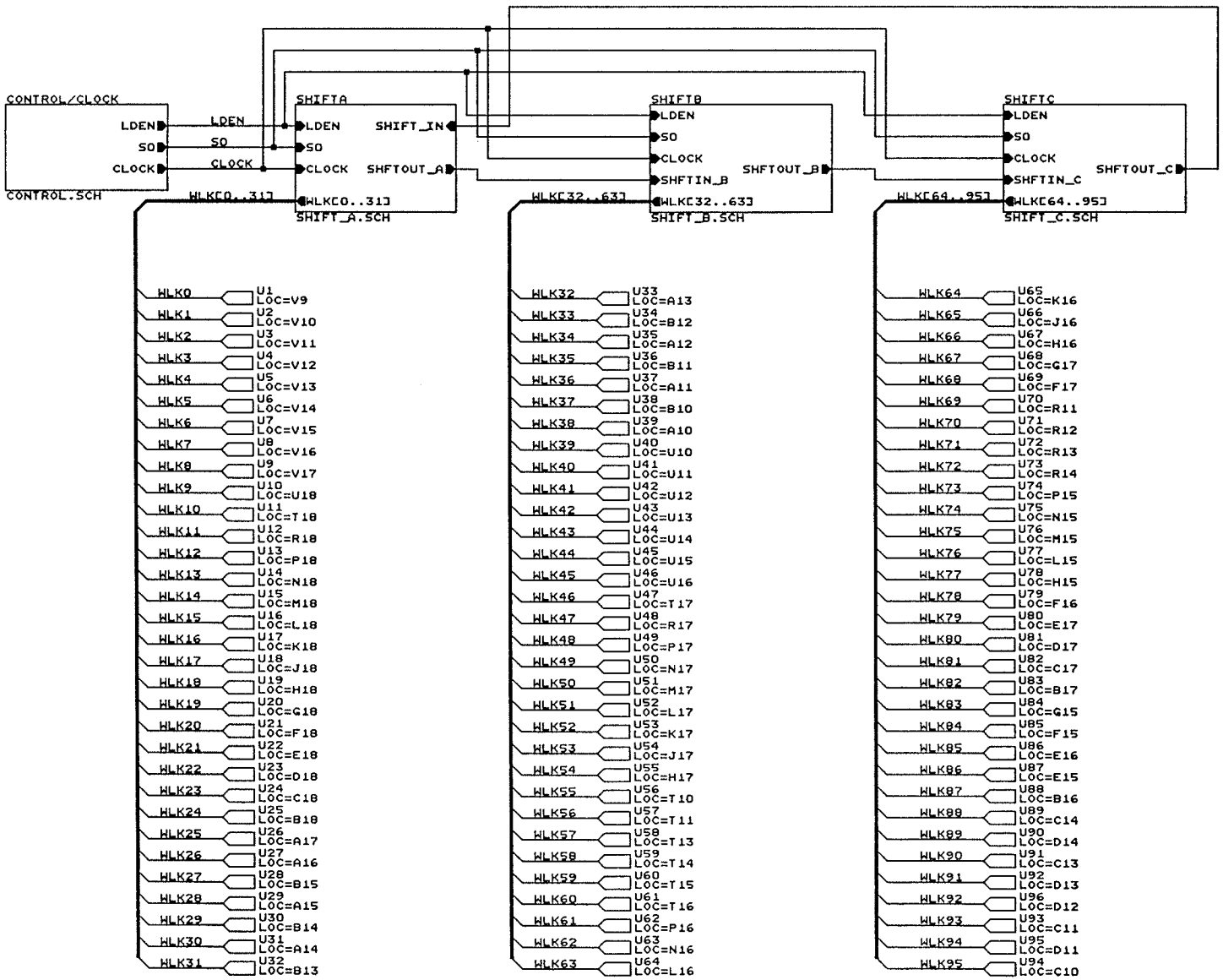

/globalsetreset
clock_in
data_in
data_out
800.0
900.0
1000.0
1100.0
1200.0
1300.0
1400.0
1500.0
1600.0
1700.0
1800
Time(ns)


/globalsetreset
clock_in
data_in
data_out
800.0
900.0
1000.0
1100.0
1200.0
1300.0
1400.0
1500.0
1600.0
1700.
Time(ns)

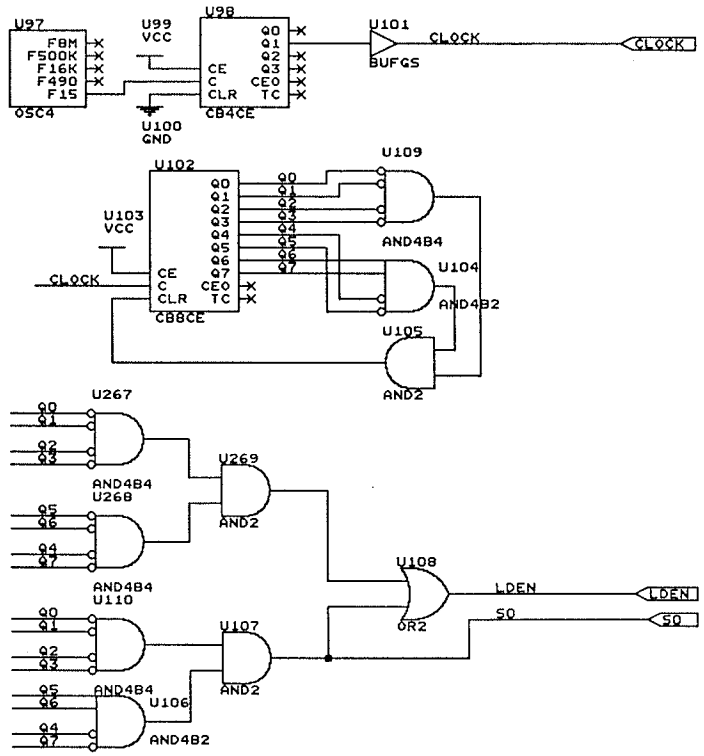

/globalsetreset
clock_in
data_in
data_out
600.0
700.0
800.0
900.0
1000.0
1100.0
1200.0
1300.0
1400.0
1500.0
Time(ns)


/globalsetreset
clock_in
data_in
data_out
500.0
600.0
700.0
800.0
900.0
1000.0
1100.0
1200.0
1300.0
1400.0
Time(ns)

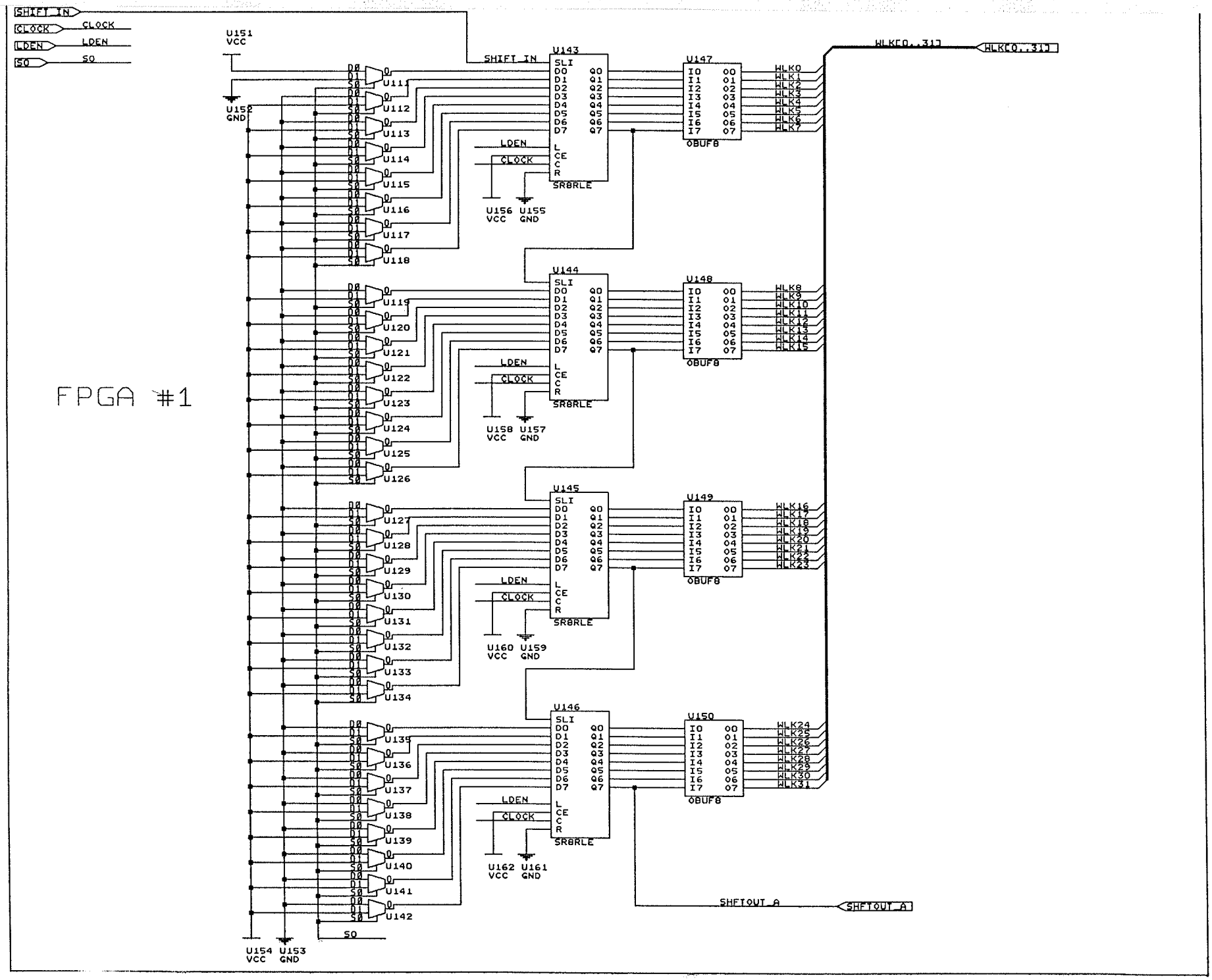

/globalsetreset
clock_in
data_in
data_out
800.0
900.0
1000.0
1100.0
1200.0
1300.0
1400.0
1500.0
1600.0
1700.0
Time(ns)

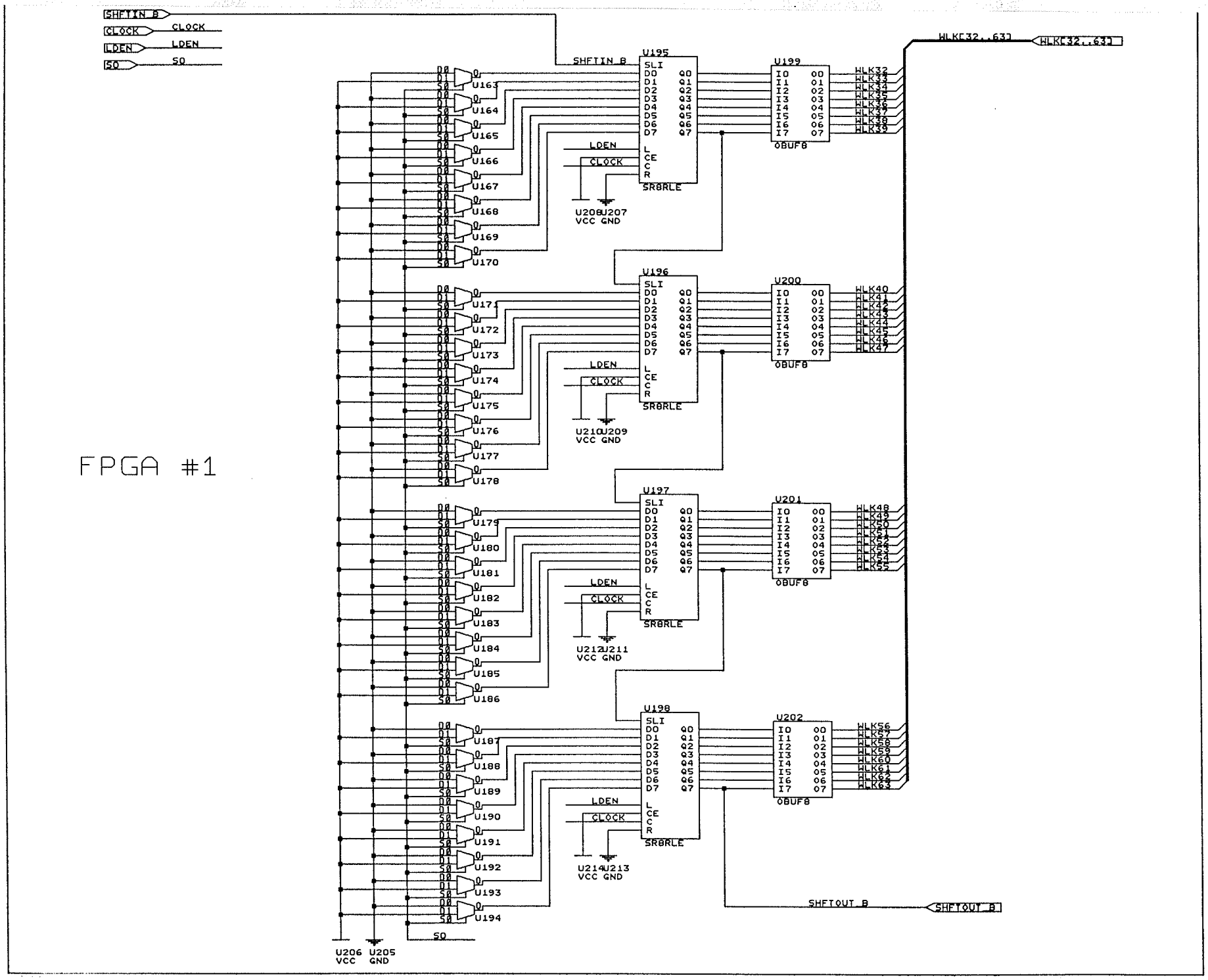

CONTROL/CLOCK
LDEN
SO
CLOCK
CONTROL.SCH
LDEN
SO
CLOCK
WLKC0..31]
SHIFTA
LDEN
SO
CLOCK
WLKC0..31]
SHIFT_IN
SHFTOUT_A
SHIFT_A.SCH
SHIFTB
LDEN
SO
CLOCK
SHFTIN_B
WLKC32..63]
SHFTOUT_B
WLKC32..63]
SHIFT_B.SCH
SHIFTC
LDEN
SO
CLOCK
SHFTIN_C
WLKC64..95]
SHFTOUT_C
WLKC64..95]
SHIFT_C.SCH
WLK0 U1 LOC=V9
WLK1 U2 LOC=V10
WLK2 U3 LOC=V11
WLK3 U4 LOC=V12
WLK4 U5 LOC=V13
WLK5 U6 LOC=V14
WLK6 U7 LOC=V15
WLK7 U8 LOC=V16
WLK8 U9 LOC=V17
WLK9 U10 LOC=U18
WLK10 U11 LOC=T18
WLK11 U12 LOC=R18
WLK12 U13 LOC=P18
WLK13 U14 LOC=N18
WLK14 U15 LOC=M18
WLK15 U16 LOC=L18
WLK16 U17 LOC=K18
WLK17 U18 LOC=J18
WLK18 U19 LOC=H18
WLK19 U20 LOC=G18
WLK20 U21 LOC=F18
WLK21 U22 LOC=E18
WLK22 U23 LOC=D18
WLK23 U24 LOC=C18
WLK24 U25 LOC=B18
WLK25 U26 LOC=A17
WLK26 U27 LOC=A16
WLK27 U28 LOC=B15
WLK28 U29 LOC=A15
WLK29 U30 LOC=B14
WLK30 U31 LOC=A14
WLK31 U32 LOC=B13
WLK32 U33 LOC=A13
WLK33 U34 LOC=B12
WLK34 U35 LOC=A12
WLK35 U36 LOC=B11
WLK36 U37 LOC=A11
WLK37 U38 LOC=B10
WLK38 U39 LOC=A10
WLK39 U40 LOC=U10
WLK40 U41 LOC=U11
WLK41 U42 LOC=U12
WLK42 U43 LOC=U13
WLK43 U44 LOC=U14
WLK44 U45 LOC=U15
WLK45 U46 LOC=U16
WLK46 U47 LOC=T17
WLK47 U48 LOC=R17
WLK48 U49 LOC=P17
WLK49 U50 LOC=N17
WLK50 U51 LOC=M17
WLK51 U52 LOC=L17
WLK52 U53 LOC=K17
WLK53 U54 LOC=J17
WLK54 U55 LOC=H17
WLK55 U56 LOC=T10
WLK56 U57 LOC=T11
WLK57 U58 LOC=T13
WLK58 U59 LOC=T14
WLK59 U60 LOC=T15
WLK60 U61 LOC=T16
WLK61 U62 LOC=P16
WLK62 U63 LOC=N16
WLK63 U64 LOC=L16
WLK64 U65 LOC=K16
WLK65 U66 LOC=J16
WLK66 U67 LOC=H16
WLK67 U68 LOC=G17
WLK68 U69 LOC=F17
WLK69 U70 LOC=R11
WLK70 U71 LOC=R12
WLK71 U72 LOC=R13
WLK72 U73 LOC=R14
WLK73 U74 LOC=P15
WLK74 U75 LOC=N15
WLK75 U76 LOC=M15
WLK76 U77 LOC=L15
WLK77 U78 LOC=H15
WLK78 U79 LOC=F16
WLK79 U80 LOC=E17
WLK80 U81 LOC=D17
WLK81 U82 LOC=C17
WLK82 U83 LOC=B17
WLK83 U84 LOC=G15
WLK84 U85 LOC=F15
WLK85 U86 LOC=E16
WLK86 U87 LOC=E15
WLK87 U88 LOC=B16
WLK88 U89 LOC=C14
WLK89 U90 LOC=D14
WLK90 U91 LOC=C13
WLK91 U92 LOC=D13
WLK92 U96 LOC=D12
WLK93 U93 LOC=C11
WLK94 U95 LOC=D11
WLK95 U94 LOC=C10

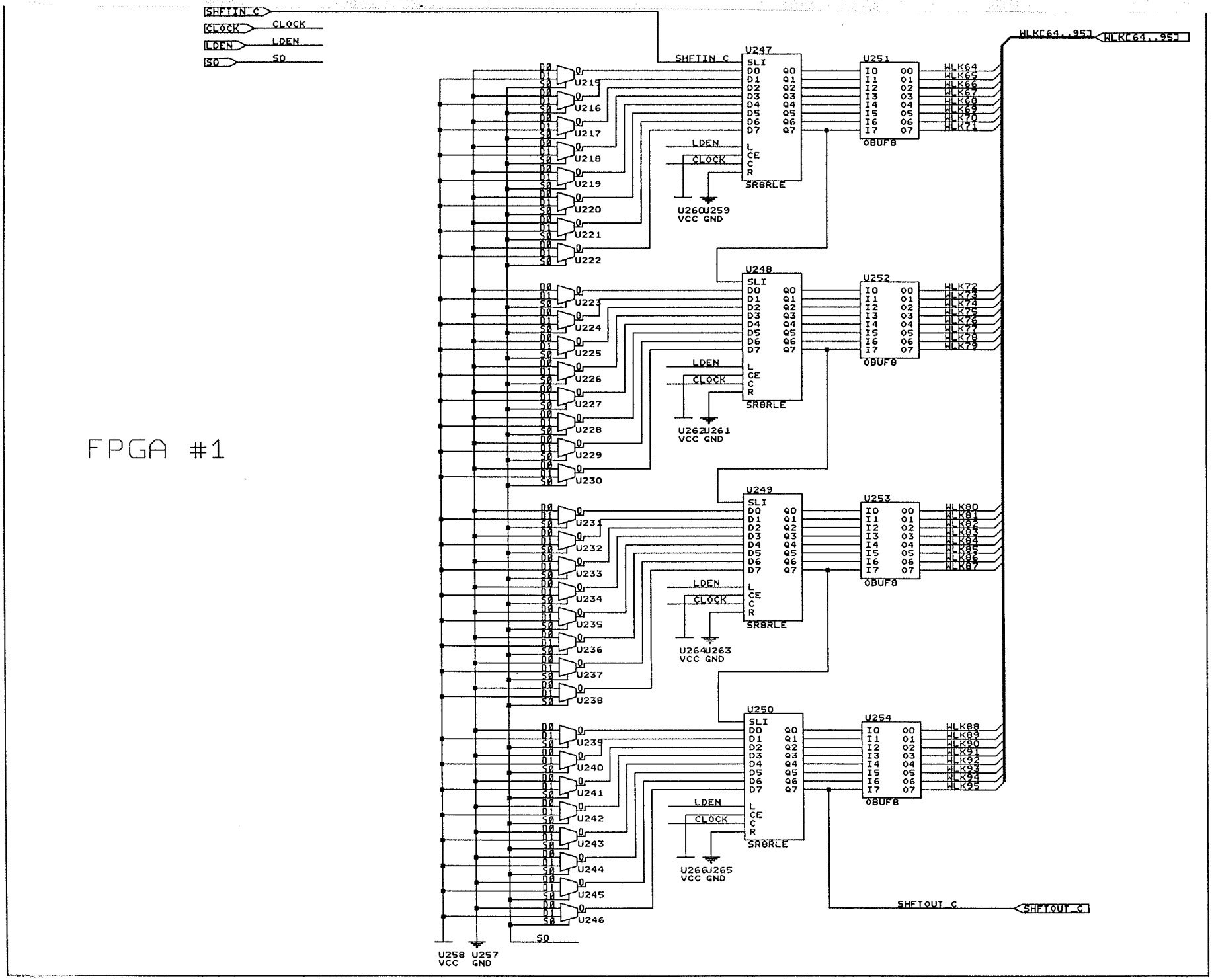
FPGA #1

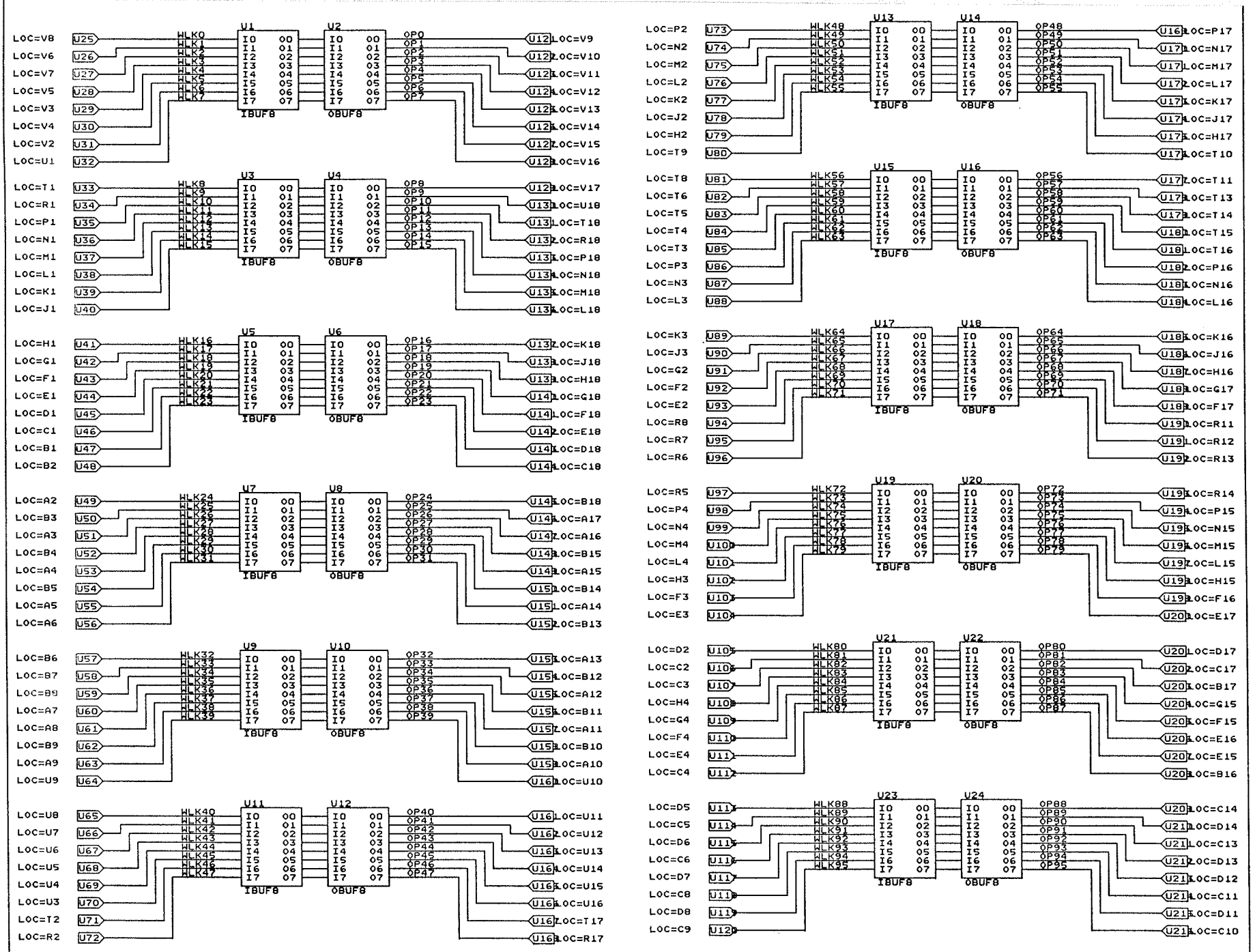

U97
F8M
F500K
F16K
F490
F15
OSC4
U99
VCC
U100
GND
U98
CE
C
CLR
Q0
Q1
Q2
Q3
CEO
TC
CB4CE
U101
BUFGS
CLOCK
CLOCK
U102
U103
VCC
CLOCK
CB8CE
U109
AND4B4
U104
AND4B2
U105
AND2
U267
AND4B4
U268
U269
AND2
U108
OR2
LDEN
LDEN
SO
SO
AND4B4
U110
U107
AND2
AND4B4
U106
AND4B2

FPGA #1

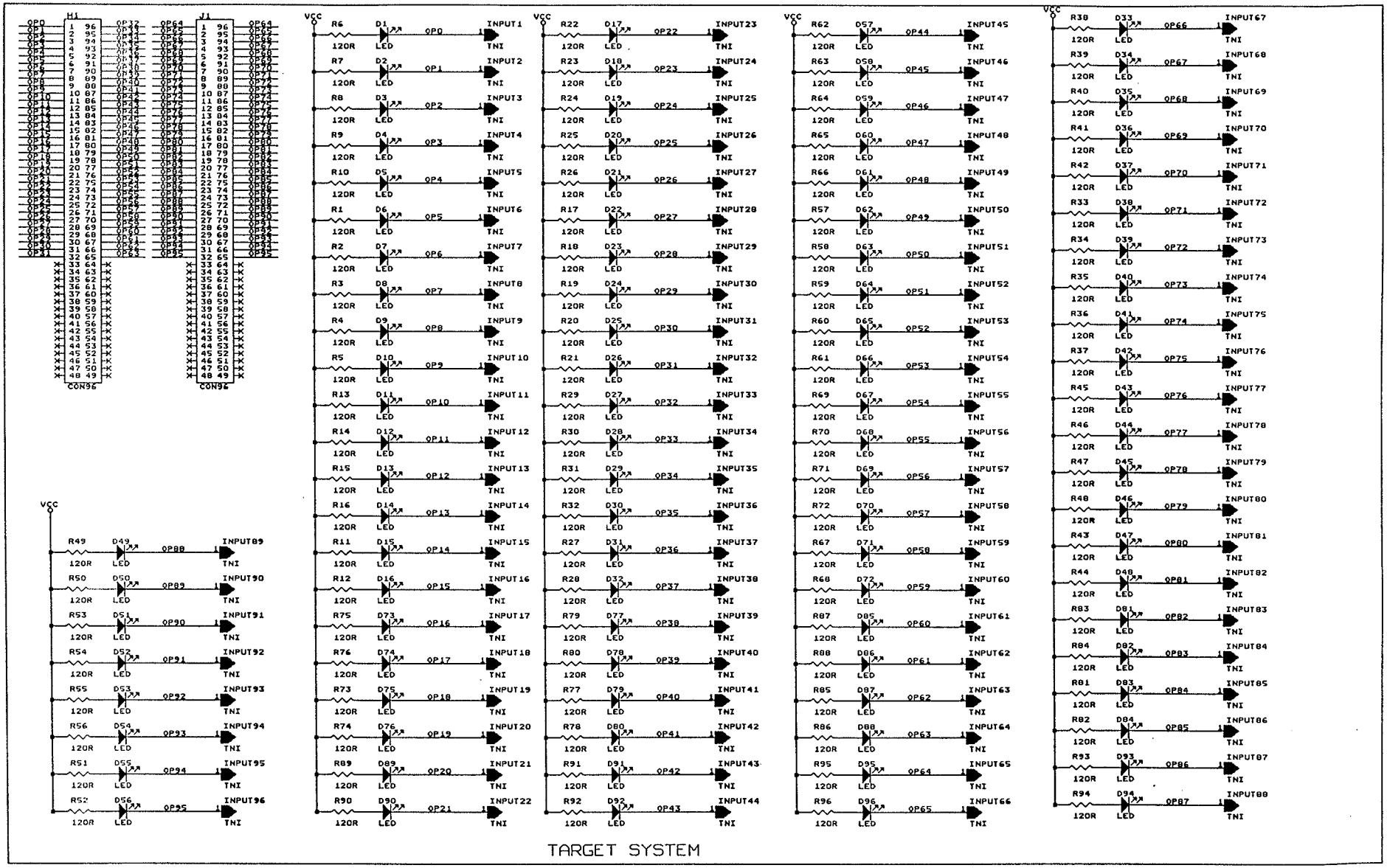

FPGA #1
SHIFT_IN
CLOCK
LDEN
SO
U151 VCC
U152 GND
U154 VCC
U153 GND
U143
U144
U145
U146
SLI
D0 D1 D2 D3 D4 D5 D6 D7
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
L CE C R
SR8RLE
U156 VCC
U155 GND
U158 VCC
U157 GND
U160 VCC
U159 GND
U162 VCC
U161 GND
U147
U148
U149
U150
I0 I1 I2 I3 I4 I5 I6 I7
O0 O1 O2 O3 O4 O5 O6 O7
OBUF8
WLK[0..31]
SHFTOUT_A


FPGA #1
SHFTIN_B
CLOCK
LDEN
SO
WLK[32..63]
SHFTOUT_B
U195
U196
U197
U198
SR8RLE
U199
U200
U201
U202
OBUF8
U163
U164
U165
U166
U167
U168
U169
U170
U171
U172
U173
U174
U175
U176
U177
U178
U179
U180
U181
U182
U183
U184
U185
U186
U187
U188
U189
U190
U191
U192
U193
U194
U206 U205
VCC GND
U208U207
U210U209
U212U211
U214U213


FPGA #1
SHFTIN_C
CLOCK
LDEN
S0
WLK[64..95]
SHFTOUT_C
U247
U248
U249
U250
U251
U252
U253
U254
SR8RLE
OBUF8
U258 VCC
U257 GND


U1 IBUF8 U2 OBUF8
LOC=V8 U25 WLK0 OP0 U12 LOC=V9
LOC=V6 U26 WLK1 OP1 U12 LOC=V10
LOC=V7 U27 WLK2 OP2 U12 LOC=V11
LOC=V5 U28 WLK3 OP3 U12 LOC=V12
LOC=V3 U29 WLK4 OP4 U12 LOC=V13
LOC=V4 U30 WLK5 OP5 U12 LOC=V14
LOC=V2 U31 WLK6 OP6 U12 LOC=V15
LOC=U1 U32 WLK7 OP7 U12 LOC=V16
U3 IBUF8 U4 OBUF8
LOC=T1 U33 WLK8 OP8 U12 LOC=V17
LOC=R1 U34 WLK9 OP9 U13 LOC=U18
LOC=P1 U35 WLK10 OP10 U13 LOC=T18
LOC=N1 U36 WLK11 OP11 U13 LOC=R18
LOC=M1 U37 WLK12 OP12 U13 LOC=P18
LOC=L1 U38 WLK13 OP13 U13 LOC=N18
LOC=K1 U39 WLK14 OP14 U13 LOC=M18
LOC=J1 U40 WLK15 OP15 U13 LOC=L18
U5 IBUF8 U6 OBUF8
LOC=H1 U41 WLK16 OP16 U13 LOC=K18
LOC=G1 U42 WLK17 OP17 U13 LOC=J18
LOC=F1 U43 WLK18 OP18 U13 LOC=H18
LOC=E1 U44 WLK19 OP19 U14 LOC=G18
LOC=D1 U45 WLK20 OP20 U14 LOC=F18
LOC=C1 U46 WLK21 OP21 U14 LOC=E18
LOC=B1 U47 WLK22 OP22 U14 LOC=D18
LOC=B2 U48 WLK23 OP23 U14 LOC=C18
U7 IBUF8 U8 OBUF8
LOC=A2 U49 WLK24 OP24 U14 LOC=B18
LOC=B3 U50 WLK25 OP25 U14 LOC=A17
LOC=A3 U51 WLK26 OP26 U14 LOC=A16
LOC=B4 U52 WLK27 OP27 U14 LOC=B15
LOC=A4 U53 WLK28 OP28 U14 LOC=A15
LOC=B5 U54 WLK29 OP29 U15 LOC=B14
LOC=A5 U55 WLK30 OP30 U15 LOC=A14
LOC=A6 U56 WLK31 OP31 U15 LOC=B13
U9 IBUF8 U10 OBUF8
LOC=B6 U57 WLK32 OP32 U15 LOC=A13
LOC=B7 U58 WLK33 OP33 U15 LOC=B12
LOC=B9 U59 WLK34 OP34 U15 LOC=A12
LOC=A7 U60 WLK35 OP35 U15 LOC=B11
LOC=A8 U61 WLK36 OP36 U15 LOC=A11
LOC=B9 U62 WLK37 OP37 U15 LOC=B10
LOC=A9 U63 WLK38 OP38 U15 LOC=A10
LOC=U9 U64 WLK39 OP39 U16 LOC=U10
U11 IBUF8 U12 OBUF8
LOC=U8 U65 WLK40 OP40 U16 LOC=U11
LOC=U7 U66 WLK41 OP41 U16 LOC=U12
LOC=U6 U67 WLK42 OP42 U16 LOC=U13
LOC=U5 U68 WLK43 OP43 U16 LOC=U14
LOC=U4 U69 WLK44 OP44 U16 LOC=U15
LOC=U3 U70 WLK45 OP45 U16 LOC=U16
LOC=T2 U71 WLK46 OP46 U16 LOC=T17
LOC=R2 U72 WLK47 OP47 U16 LOC=R17
U13 IBUF8 U14 OBUF8
LOC=P2 U73 WLK48 OP48 U16 LOC=P17
LOC=N2 U74 WLK49 OP49 U17 LOC=N17
LOC=M2 U75 WLK50 OP50 U17 LOC=M17
LOC=L2 U76 WLK51 OP51 U17 LOC=L17
LOC=K2 U77 WLK52 OP52 U17 LOC=K17
LOC=J2 U78 WLK53 OP53 U17 LOC=J17
LOC=H2 U79 WLK54 OP54 U17 LOC=H17
LOC=T9 U80 WLK55 OP55 U17 LOC=T10
U15 IBUF8 U16 OBUF8
LOC=T8 U81 WLK56 OP56 U17 LOC=T11
LOC=T6 U82 WLK57 OP57 U17 LOC=T13
LOC=T5 U83 WLK58 OP58 U17 LOC=T14
LOC=T4 U84 WLK59 OP59 U18 LOC=T15
LOC=T3 U85 WLK60 OP60 U18 LOC=T16
LOC=P3 U86 WLK61 OP61 U18 LOC=P16
LOC=N3 U87 WLK62 OP62 U18 LOC=N16
LOC=L3 U88 WLK63 OP63 U18 LOC=L16
U17 IBUF8 U18 OBUF8
LOC=K3 U89 WLK64 OP64 U18 LOC=K16
LOC=J3 U90 WLK65 OP65 U18 LOC=J16
LOC=G2 U91 WLK66 OP66 U18 LOC=H16
LOC=F2 U92 WLK67 OP67 U18 LOC=G17
LOC=E2 U93 WLK68 OP68 U18 LOC=F17
LOC=R8 U94 WLK69 OP69 U19 LOC=R11
LOC=R7 U95 WLK70 OP70 U19 LOC=R12
LOC=R6 U96 WLK71 OP71 U19 LOC=R13
U19 IBUF8 U20 OBUF8
LOC=R5 U97 WLK72 OP72 U19 LOC=R14
LOC=P4 U98 WLK73 OP73 U19 LOC=P15
LOC=N4 U99 WLK74 OP74 U19 LOC=N15
LOC=M4 U10 WLK75 OP75 U19 LOC=M15
LOC=L4 U10 WLK76 OP76 U19 LOC=L15
LOC=H3 U10 WLK77 OP77 U19 LOC=H15
LOC=F3 U10 WLK78 OP78 U19 LOC=F16
LOC=E3 U10 WLK79 OP79 U20 LOC=E17
U21 IBUF8 U22 OBUF8
LOC=D2 U10 WLK80 OP80 U20 LOC=D17
LOC=C2 U10 WLK81 OP81 U20 LOC=C17
LOC=C3 U10 WLK82 OP82 U20 LOC=B17
LOC=H4 U10 WLK83 OP83 U20 LOC=G15
LOC=G4 U10 WLK84 OP84 U20 LOC=F15
LOC=F4 U11 WLK85 OP85 U20 LOC=E16
LOC=E4 U11 WLK86 OP86 U20 LOC=E15
LOC=C4 U11 WLK87 OP87 U20 LOC=B16
U23 IBUF8 U24 OBUF8
LOC=D5 U11 WLK88 OP88 U20 LOC=C14
LOC=C5 U11 WLK89 OP89 U21 LOC=D14
LOC=D6 U11 WLK90 OP90 U21 LOC=C13
LOC=C6 U11 WLK91 OP91 U21 LOC=D13
LOC=D7 U11 WLK92 OP92 U21 LOC=D12
LOC=C8 U11 WLK93 OP93 U21 LOC=C11
LOC=D8 U11 WLK94 OP94 U21 LOC=D11
LOC=C9 U12 WLK95 OP95 U21 LOC=C10

FPGA #2



TARGET SYSTEM

WLKTST1.DOC 1684 December 23, 1995 10:52:36 AM
FPGA#1 NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST1.LCA, WLKTST1.XFF OUTPUT FILES: WLKTST1.NET

```
X34  WLK82  O
X77  WLK85  O
X52  WLK81  O
X70  WLK80  O
X35  WLK24  O
X78  WLK79  O
X85  WLK78  O
X53  WLK23  O
X71  WLK22  O
X86  WLK68  O
X76  WLK86  O
X84  WLK84  O
X79  WLK21  O
X87  WLK20  O
X94  WLK67  O
X95  WLK19  O
X101  WLK66  O
X102  WLK54  O
X92  WLK83  O
X100  WLK77  O
X103  WLK18  O
X111  WLK17  O
X110  WLK53  O
X109  WLK65  O
X117  WLK64  O
X118  WLK52  O
X119  WLK16  O
X127  WLK15  O
X126  WLK51  O
X125  WLK63  O
X124  WLK76  O
X132  WLK75  O
X135  WLK14  O
X134  WLK50  O
X143  WLK13  O
X151  WLK12  O
X140  WLK74  O
X148  WLK73  O
X142  WLK49  O
X169  WLK11  O
X187  WLK10  O
X150  WLK48  O
X141  WLK62  O
X186  WLK46  O
X168  WLK47  O
X149  WLK61  O
X205  WLK9  O
X185  WLK60  O
X33  WLK87  O
X16  WLK25  O
X184  WLK59  O
X203  WLK45  O
X49  WLK88  O
X32  WLK27  O
X183  WLK58  O
X202  WLK44  O
X15  WLK26  O
X31  WLK29  O
X165  WLK72  O
X164  WLK71  O
X48  WLK90  O
X14  WLK28  O
X222  WLK8  O
X221  WLK7  O
X13  WLK30  O
X30  WLK31  O
X182  WLK57  O
X201  WLK43  O
X67  WLK89  O
X66  WLK91  O
X220  WLK6  O
X219  WLK5  O
X12  WLK32  O
X29  WLK33  O
X163  WLK70  O
X162  WLK69  O
X11  WLK34  O
X28  WLK35  O
X200  WLK42  O
X218  WLK4  O
X65  WLK92  O
X64  WLK94  O
X199  WLK41  O
X217  WLK3  O
X46  WLK93  O
X10  WLK36  O
X180  WLK56  O
X198  WLK40  O
X9  WLK38  O
X216  WLK2  O
X215  WLK1  O
X27  WLK37  O
X45  WLK95  O
X197  WLK39  O
X179  WLK55  O
X214  WLK0  O
eof
```

WLKTST1B.DOC 3252 December 23, 1995 10:53:28 AM
FPGA#2 NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST1B.LCA, WLKTST1B.XFF OUTPUT FILES: WLKTST1B.NET

```
Y34  OP82  O
Y77  OP85  O
Y52  OP81  O
Y70  OP80  O
Y35  OP24  O
Y78  OP79  O
Y85  OP78  O
Y53  OP23  O
Y71  OP22  O
Y86  OP68  O
Y76  OP86  O
Y84  OP84  O
Y79  OP21  O
Y87  OP20  O
Y94  OP67  O
Y95  OP19  O
Y101  OP66  O
Y102  OP54  O
Y92  OP83  O
Y100  OP77  O
Y103  OP18  O
Y111  OP17  O
Y110  OP53  O
Y109  OP65  O
Y117  OP64  O
Y118  OP52  O
Y119  OP16  O
Y127  OP15  O
Y126  OP51  O
Y125  OP63  O
Y124  OP76  O
Y132  OP75  O
Y135  OP14  O
Y134  OP50  O
Y143  OP13  O
Y151  OP12  O
Y140  OP74  O
Y148  OP73  O
Y142  OP49  O
Y169  OP11  O
Y187  OP10  O
Y150  OP48  O
Y141  OP62  O
Y186  OP46  O
Y168  OP47  O
Y149  OP61  O
Y205  OP9  O
Y185  OP60  O
Y33  OP87  O
Y16  OP25  O
Y184  OP59  O
Y203  OP45  O
Y49  OP88  O
Y32  OP27  O
Y183  OP58  O
Y202  OP44  O
Y15  OP26  O
Y31  OP29  O
Y165  OP72  O
Y164  OP71  O
Y48  OP90  O
Y14  OP28  O
Y222  OP8  O
Y221  OP7  O
Y13  OP30  O
Y30  OP31  O
Y182  OP57  O
Y201  OP43  O
Y67  OP89  O
Y66  OP91  O
Y220  OP6  O
Y219  OP5  O
Y12  OP32  O
Y29  OP33  O
Y163  OP70  O
Y162  OP69  O
Y11  OP34  O
Y28  OP35  O
Y200  OP42  O
Y218  OP4  O
Y65  OP92  O
Y64  OP94  O
Y199  OP41  O
Y217  OP3  O
Y46  OP93  O
Y10  OP36  O
Y180  OP56  O
Y198  OP40  O
Y9  OP38  O
Y8  WLK38  I
Y216  OP2  O
Y215  OP1  O
Y27  OP37  O
Y45  OP95  O
Y197  OP39  O
Y179  OP55  O
Y44  WLK95  I
Y26  WLK37  I
Y178  WLK55  I
Y196  WLK39  I
Y7  WLK36  I
Y25  WLK34  I
Y214  OP0  O
Y213  WLK0  I
Y6  WLK35  I
Y43  WLK93  I
Y195  WLK40  I
Y177  WLK56  I
Y61  WLK94  I
Y60  WLK92  I
Y212  WLK2  I
Y194  WLK41  I
Y5  WLK31  I
Y24  WLK33  I
Y211  WLK1  I
Y193  WLK42  I
Y4  WLK30  I
Y3  WLK28  I
Y159  WLK69  I
Y158  WLK70  I
Y59  WLK90  I
Y58  WLK88  I
Y157  WLK71  I
Y156  WLK72  I
```

WLKTST1B.DOC 3252 December 23, 1995 10:53:28 AM
FPGA#2 NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST1B.LCA, WLKTST1B.XFF OUTPUT FILES: WLKTST1B.NET

```
Y23  WLK32  I
Y22  WLK29  I
Y210  WLK3  I
Y209  WLK5  I
Y2  WLK26  I
Y41  WLK91  I
Y192  WLK43  I
Y175  WLK57  I
Y21  WLK27  I
Y1  WLK24  I
Y208  WLK4  I
Y207  WLK6  I
Y40  WLK89  I
Y20  WLK25  I
Y191  WLK44  I
Y174  WLK58  I
Y39  WLK87  I
Y38  WLK82  I
Y190  WLK45  I
Y173  WLK59  I
Y19  WLK23  I
Y37  WLK81  I
Y74  WLK79  I
Y18  WLK22  I
Y75  WLK86  I
Y83  WLK85  I
Y55  WLK80  I
Y82  WLK78  I
Y73  WLK68  I
Y36  WLK21  I
Y54  WLK20  I
Y81  WLK67  I
Y72  WLK19  I
Y80  WLK18  I
Y91  WLK84  I
Y99  WLK83  I
Y89  WLK66  I
Y88  WLK17  I
Y98  WLK77  I
Y97  WLK54  I
Y96  WLK16  I
Y104  WLK15  I
Y105  WLK53  I
Y106  WLK65  I
Y114  WLK64  I
Y113  WLK52  I
Y112  WLK14  I
Y120  WLK13  I
Y121  WLK51  I
Y122  WLK63  I
Y128  WLK12  I
Y129  WLK50  I
Y123  WLK76  I
Y131  WLK75  I
Y136  WLK11  I
Y144  WLK10  I
Y137  WLK49  I
Y152  WLK9  I
Y170  WLK8  I
Y145  WLK48  I
Y139  WLK74  I
Y147  WLK73  I
Y138  WLK62  I
Y171  WLK46  I
Y153  WLK47  I
Y146  WLK61  I
Y188  WLK7  I
Y172  WLK60  I
eof
```

WALK_NET.DOC 2027 December 23, 1995 10:51:38 AM
TARGET SYSTEM NETLIST CREATED BY LST2NET.EXE
INPUT FILE: WALK.SCI OUTPUT FILE: WALK.NET

```
OP0  I1  I
OP1  I2  I
OP2  I3  I
OP3  I4  I
OP4  I5  I
OP5  I6  I
OP6  I7  I
OP7  I8  I
OP8  I9  I
OP9  I10  I
OP10  I11  I
OP11  I12  I
OP12  I13  I
OP13  I14  I
OP14  I15  I
OP15  I16  I
OP16  I17  I
OP17  I18  I
OP18  I19  I
OP19  I20  I
OP20  I21  I
OP21  I22  I
OP22  I23  I
OP23  I24  I
OP24  I25  I
OP25  I26  I
OP26  I27  I
OP27  I28  I
OP28  I29  I
OP29  I30  I
OP30  I31  I
OP31  I32  I
OP32  I65  I
OP33  I66  I
OP34  I67  I
OP35  I68  I
OP36  I69  I
OP37  I70  I
OP38  I71  I
OP39  I72  I
OP40  I73  I
OP41  I74  I
OP42  I75  I
OP43  I76  I
OP44  I77  I
OP45  I78  I
OP46  I79  I
OP47  I80  I
OP48  I81  I
OP49  I82  I
OP50  I83  I
OP51  I84  I
OP52  I85  I
OP53  I86  I
OP54  I87  I
OP55  I88  I
OP56  I89  I
OP57  I90  I
OP58  I91  I
OP59  I92  I
OP60  I93  I
OP61  I94  I
OP62  I95  I
OP63  I96  I
OP64  J1  I
OP65  J2  I
OP66  J3  I
OP67  J4  I
OP68  J5  I
OP69  J6  I
OP70  J7  I
OP71  J8  I
OP72  J9  I
OP73  J10  I
OP74  J11  I
OP75  J12  I
OP76  J13  I
OP77  J14  I
OP78  J15  I
OP79  J16  I
OP80  J17  I
OP81  J18  I
OP82  J19  I
OP83  J20  I
OP84  J21  I
OP85  J22  I
OP86  J23  I
OP87  J24  I
OP88  J25  I
OP89  J26  I
OP90  J27  I
OP91  J28  I
OP92  J29  I
OP93  J30  I
OP94  J31  I
OP95  J32  I
OP64  J65  I
OP65  J66  I
OP66  J67  I
OP67  J68  I
OP68  J69  I
OP69  J70  I
OP70  J71  I
OP71  J72  I
OP72  J73  I
OP73  J74  I
OP74  J75  I
OP75  J76  I
OP76  J77  I
OP77  J78  I
OP78  J79  I
OP79  J80  I
OP80  J81  I
OP81  J82  I
OP82  J83  I
OP83  J84  I
OP84  J85  I
OP85  J86  I
OP86  J87  I
OP87  J88  I
OP88  J89  I
OP89  J90  I
OP90  J91  I
OP91  J92  I
```

WALK_NET.DOC 2027 December 23, 1995 10:51:38 AM
TARGET SYSTEM NETLIST CREATED BY LST2NET.EXE
INPUT FILE: WALK.SCI OUTPUT FILE: WALK.NET

```
OP92  J93  I
OP93  J94  I
OP94  J95  I
OP95  J96  I
eof
```

WALK_ITC.DOC 8473 December 23, 1995 10:50:47 AM
INTERCONNECT NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST.NET, WLKTST1B.NET, WALK.NET OUTPUT FILES: WALK.ITC

```
Net  WLK82
X34  O
Y38  I
End
Net  WLK85
X77  O
Y83  I
End
Net  WLK81
X52  O
Y37  I
End
Net  WLK80
X70  O
Y55  I
End
Net  WLK24
X35  O
Y1  I
End
Net  WLK79
X78  O
Y74  I
End
Net  WLK78
X85  O
Y82  I
End
Net  WLK23
X53  O
Y19  I
End
Net  WLK22
X71  O
Y18  I
End
Net  WLK68
X86  O
Y73  I
End
Net  WLK86
X76  O
Y75  I
End
Net  WLK84
X84  O
Y91  I
End
Net  WLK21
X79  O
Y36  I
End
Net  WLK20
X87  O
Y54  I
End
Net  WLK67
X94  O
Y81  I
End
Net  WLK19
X95  O
Y72  I
End
Net  WLK66
X101  O
Y89  I
End
Net  WLK54
X102  O
Y97  I
End
Net  WLK83
X92  O
Y99  I
End
Net  WLK77
X100  O
Y98  I
End
Net  WLK18
X103  O
Y80  I
End
Net  WLK17
X111  O
Y88  I
End
Net  WLK53
X110  O
Y105  I
End
Net  WLK65
X109  O
Y106  I
End
Net  WLK64
X117  O
Y114  I
End
Net  WLK52
X118  O
Y113  I
End
Net  WLK16
X119  O
Y96  I
End
Net  WLK15
X127  O
Y104  I
End
Net  WLK51
X126  O
Y121  I
End
Net  WLK63
X125  O
Y122  I
End
Net  WLK76
X124  O
Y123  I
End
```

INTERCONNECT NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST.NET, WLKTST1B.NET, WALK.NET OUTPUT FILES: WALK.ITC

```
Net  WLK75
X132  O
Y131  I
End
Net  WLK14
X135  O
Y112  I
End
Net  WLK50
X134  O
Y129  I
End
Net  WLK13
X143  O
Y120  I
End
Net  WLK12
X151  O
Y128  I
End
Net  WLK74
X140  O
Y139  I
End
Net  WLK73
X148  O
Y147  I
End
Net  WLK49
X142  O
Y137  I
End
Net  WLK11
X169  O
Y136  I
End
Net  WLK10
X187  O
Y144  I
End
Net  WLK48
X150  O
Y145  I
End
Net  WLK62
X141  O
Y138  I
End
Net  WLK46
X186  O
Y171  I
End
Net  WLK47
X168  O
Y153  I
End
Net  WLK61
X149  O
Y146  I
End
Net  WLK9
X205  O
Y152  I
End
Net  WLK60
X185  O
Y172  I
End
Net  WLK87
X33  O
Y39  I
End
Net  WLK25
X16  O
Y20  I
End
Net  WLK59
X184  O
Y173  I
End
Net  WLK45
X203  O
Y190  I
End
Net  WLK88
X49  O
Y58  I
End
Net  WLK27
X32  O
Y21  I
End
Net  WLK58
X183  O
Y174  I
End
Net  WLK44
X202  O
Y191  I
End
Net  WLK26
X15  O
Y2  I
End
Net  WLK29
X31  O
Y22  I
End
Net  WLK72
X165  O
Y156  I
End
Net  WLK71
X164  O
Y157  I
End
Net  WLK90
X48  O
Y59  I
End
Net  WLK28
X14  O
Y3  I
End
```

WALK_ITC.DOC 8473 December 23, 1995 10:50:47 AM
INTERCONNECT NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST.NET, WLKTST1B.NET, WALK.NET OUTPUT FILES: WALK.ITC

```
Net  WLK8
X222  O
Y170  I
End
Net  WLK7
X221  O
Y188  I
End
Net  WLK30
X13  O
Y4  I
End
Net  WLK31
X30  O
Y5  I
End
Net  WLK57
X182  O
Y175  I
End
Net  WLK43
X201  O
Y192  I
End
Net  WLK89
X67  O
Y40  I
End
Net  WLK91
X66  O
Y41  I
End
Net  WLK6
X220  O
Y207  I
End
Net  WLK5
X219  O
Y209  I
End
Net  WLK32
X12  O
Y23  I
End
Net  WLK33
X29  O
Y24  I
End
Net  WLK70
X163  O
Y158  I
End
Net  WLK69
X162  O
Y159  I
End
Net  WLK34
X11  O
Y25  I
End
Net  WLK35
X28  O
Y6  I
End
Net  WLK42
X200  O
Y193  I
End
Net  WLK4
X218  O
Y208  I
End
Net  WLK92
X65  O
Y60  I
End
Net  WLK94
X64  O
Y61  I
End
Net  WLK41
X199  O
Y194  I
End
Net  WLK3
X217  O
Y210  I
End
Net  WLK93
X46  O
Y43  I
End
Net  WLK36
X10  O
Y7  I
End
Net  WLK56
X180  O
Y177  I
End
Net  WLK40
X198  O
Y195  I
End
Net  WLK38
X9  O
Y8  I
End
Net  WLK2
X216  O
Y212  I
End
Net  WLK1
X215  O
Y211  I
End
Net  WLK37
X27  O
Y26  I
End
Net  WLK95
X45  O
Y44  I
End
```

WALK_ITC.DOC 8473 December 23, 1995 10:50:47 AM
INTERCONNECT NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST.NET, WLKTST1B.NET, WALK.NET OUTPUT FILES: WALK.ITC

```
Net  WLK39
X197  O
Y196  I
End
Net  WLK55
X179  O
Y178  I
End
Net  WLK0
X214  O
Y213  I
End
Net  OP82
Y34  O
J19  I
J83  I
End
Net  OP85
Y77  O
J22  I
J86  I
End
Net  OP81
Y52  O
J18  I
J82  I
End
Net  OP80
Y70  O
J17  I
J81  I
End
Net  OP24
Y35  O
I25  I
End
Net  OP79
Y78  O
J16  I
J80  I
End
Net  OP78
Y85  O
J15  I
J79  I
End
Net  OP23
Y53  O
I24  I
End
Net  OP22
Y71  O
I23  I
End
Net  OP68
Y86  O
J5  I
J69  I
End
Net  OP86
Y76  O
J23  I
J87  I
End
Net  OP84
Y84  O
J21  I
J85  I
End
Net  OP21
Y79  O
I22  I
End
Net  OP20
Y87  O
I21  I
End
Net  OP67
Y94  O
J4  I
J68  I
End
Net  OP19
Y95  O
I20  I
End
Net  OP66
Y101  O
J3  I
J67  I
End
Net  OP54
Y102  O
I87  I
End
Net  OP83
Y92  O
J20  I
J84  I
End
Net  OP77
Y100  O
J14  I
J78  I
End
Net  OP18
Y103  O
I19  I
End
Net  OP17
Y111  O
I18  I
End
Net  OP53
Y110  O
I86  I
End
Net  OP65
Y109  O
J2  I
J66  I
End
Net  OP64
Y117  O
```

INTERCONNECT NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST.NET, WLKTST1B.NET, WALK.NET OUTPUT FILES: WALK.ITC

```
J1  I
J65  I
End
Net  OP52
Y118  O
I85  I
End
Net  OP16
Y119  O
I17  I
End
Net  OP15
Y127  O
I16  I
End
Net  OP51
Y126  O
I84  I
End
Net  OP63
Y125  O
I96  I
End
Net  OP76
Y124  O
J13  I
J77  I
End
Net  OP75
Y132  O
J12  I
J76  I
End
Net  OP14
Y135  O
I15  I
End
Net  OP50
Y134  O
I83  I
End
Net  OP13
Y143  O
I14  I
End
Net  OP12
Y151  O
I13  I
End
Net  OP74
Y140  O
J11  I
J75  I
End
Net  OP73
Y148  O
J10  I
J74  I
End
Net  OP49
Y142  O
I82  I
End
Net  OP11
Y169  O
I12  I
End
Net  OP10
Y187  O
I11  I
End
Net  OP48
Y150  O
I81  I
End
Net  OP62
Y141  O
I95  I
End
Net  OP46
Y186  O
I79  I
End
Net  OP47
Y168  O
I80  I
End
Net  OP61
Y149  O
I94  I
End
Net  OP9
Y205  O
I10  I
End
Net  OP60
Y185  O
I93  I
End
Net  OP87
Y33  O
J24  I
J88  I
End
Net  OP25
Y16  O
I26  I
End
Net  OP59
Y184  O
I92  I
End
Net  OP45
Y203  O
I78  I
End
Net  OP88
Y49  O
J25  I
J89  I
End
Net  OP27
Y32  O
I28  I
```

```
End
Net  OP58
Y183  O
I91  I
End
Net  OP44
Y202  O
I77  I
End
Net  OP26
Y15  O

I27  I
End
Net  OP29
Y31  O
I30  I
End
Net  OP72
Y165  O
J9  I
J73  I
End
Net  OP71
Y164  O
J8  I
J72  I
End
Net  OP90
Y48  O
J27  I
J91  I
End
Net  OP28
Y14  O
I29  I
End
Net  OP8
Y222  O
I9  I
End
Net  OP7
Y221  O
I8  I
End
Net  OP30
Y13  O
I31  I
End
Net  OP31
Y30  O
I32  I
End
Net  OP57
Y182  O
I90  I
End
Net  OP43
Y201  O
I76  I
End
Net  OP89
Y67  O
J26  I
J90  I
End
Net  OP91
Y66  O
J28  I
J92  I
End
Net  OP6
Y220  O
I7  I
End

Net  OP5
Y219  O
I6  I
End
Net  OP32
Y12  O
I65  I
End
Net  OP33
Y29  O
I66  I
End
Net  OP70
Y163  O
J7  I
J71  I
End
Net  OP69
Y162  O
J6  I
J70  I
End
Net  OP34
Y11  O
I67  I
End
Net  OP35
Y28  O
I68  I
End
Net  OP42
Y200  O
I75  I
End
Net  OP4
Y218  O
I5  I
End
Net  OP92
Y65  O
J29  I
J93  I
End
Net  OP94
Y64  O
J31  I
J95  I
End
```

WALK_ITC.DOC 8473 December 23, 1995 10:50:47 AM
INTERCONNECT NETLIST CREATED BY LST2NET.EXE
INPUT FILES: WLKTST.NET, WLKTST1B.NET, WALK.NET OUTPUT FILES: WALK.ITC

```
Net  OP41
Y199  O
I74  I
End
Net  OP3
Y217  O
I4  I
End
Net  OP93
Y46  O
J30  I
J94  I
End
Net  OP36

Y10  O
I69  I
End
Net  OP56
Y180  O
I89  I
End
Net  OP40
Y198  O
I73  I
End
Net  OP38
Y9  O
I71  I
End
Net  OP2
Y216  O
I3  I
End
Net  OP1
Y215  O
I2  I
End
Net  OP37
Y27  O
I70  I
End
Net  OP95
Y45  O
J32  I
J96  I
End
Net  OP39
Y197  O
I72  I
End
Net  OP55
Y179  O
I88  I
End
Net  OP0
Y214  O
I1  I
End

eof
```

WALK_CON.DOC 20765 December 23, 1995 10:48:54 AM
NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
Net WLK82
Connect   Case 1
G  A  283      O
F  A  27      I
EndConnect
EndNet
Net WLK85
Connect   Case 1
G  A  205      O
F  A  105      I
EndConnect
EndNet
Net WLK81
Connect   Case 1
G  A  275      O
F  A  35      I
EndConnect
EndNet
Net WLK80
Connect   Case 1
G  A  267      O
F  A  43      I
EndConnect
EndNet
Net WLK24
Connect   Case 1
G'  A  251  C  101  O
F  A  56      I
EndConnect
EndNet
Net WLK79
Connect   Case 1
G  A  261      O
F  A  51      I
EndConnect
EndNet
Net WLK78
Connect   Case 1
G  A  255      O
F  A  59      I
EndConnect
EndNet
Net WLK23
Connect   Case 1
G'  A  247  C  109  O
F  A  60      I
EndConnect
EndNet
Net WLK22
Connect   Case 1
G'  A  243  C  117  O
F  A  64      I
EndConnect
EndNet
Net WLK68
Connect   Case 1
G  A  277      O
F  A  29      I
EndConnect
EndNet
Net WLK86
Connect   Case 1
G  A  213      O
F  A  97      I
EndConnect
EndNet
Net WLK84
Connect   Case 1
G  A  197      O
F  A  113      I
EndConnect
EndNet
Net WLK21
Connect   Case 1
G'  A  236  C  93  O
F  A  68      I
EndConnect
EndNet
Net WLK20
Connect   Case 1
G'  A  234  C  99  O
F  A  70      I
EndConnect
EndNet
Net WLK67
Connect   Case 1
G  A  273      O
F  A  37      I
EndConnect
EndNet
Net WLK19
Connect   Case 1
G'  A  230  C  107  O
F  A  77      I
EndConnect
EndNet
Net WLK66
Connect   Case 1
G  A  265      O
F  A  45      I
EndConnect
EndNet
Net WLK54
Connect   Case 1
G  A  281      O
F  A  33      I
EndConnect
EndNet
Net WLK83
Connect   Case 1
G  A  189      O
F  A  121      I
EndConnect
EndNet
Net WLK77
Connect   Case 1
G  A  244      O
F  A  67      I
EndConnect
EndNet
Net WLK18
Connect   Case 1
G'  A  226  C  111  O
F  A  81      I
```

```
EndConnect
EndNet
Net WLK17
Connect   Case 1
G'  A  222  C  119  O
F  A  85      I
EndConnect
EndNet
Net WLK53
Connect   Case 1
G  A  274      O
F  A  41      I
EndConnect
EndNet
Net WLK65
Connect   Case 1
G  A  259      O
F  A  53      I
EndConnect
EndNet
Net WLK64
Connect   Case 1
G  A  252      O
F  A  61      I
EndConnect
EndNet
Net WLK52
Connect   Case 1
G  A  269      O
F  A  49      I
EndConnect
EndNet
Net WLK16
Connect   Case 1
G'  A  224  C  98  O
F  A  80      I
EndConnect
EndNet
Net WLK15
Connect   Case 1
G'  A  218  C  103  O
F  A  82      I
EndConnect
EndNet
Net WLK51
Connect   Case 1
G  A  264      O
F  A  57      I
EndConnect
EndNet
Net WLK63
Connect   Case 1
G  A  238      O
F  A  69      I
EndConnect
EndNet
Net WLK76
Connect   Case 1
G  A  240      O
F  A  73      I
EndConnect
EndNet
Net WLK75
Connect   Case 1
G  A  239      O
F  A  74      I
EndConnect
EndNet
Net WLK14
Connect   Case 1
G'  A  220  C  108  O
F  A  88      I
EndConnect
EndNet
Net WLK50
Connect   Case 1
G  A  258      O
F  A  65      I
EndConnect
EndNet
Net WLK13
Connect   Case 1
G'  A  214  C  115  O
F  A  90      I
EndConnect
EndNet
Net WLK12
Connect   Case 1
G'  A  216  C  123  O
F  A  94      I
EndConnect
EndNet
Net WLK74
Connect   Case 1
G  A  233      O
F  A  78      I
EndConnect
EndNet
Net WLK73
Connect   Case 1
G  A  225      O
F  A  89      I
EndConnect
EndNet
Net WLK49
Connect   Case 1
G  A  253      O
F  A  62      I
EndConnect
EndNet
Net WLK11
Connect   Case 1
G'  A  212  C  96  O
F  A  96      I
EndConnect
EndNet
Net WLK10
Connect   Case 1
G'  A  210  C  100  O
F  A  100      I
EndConnect
EndNet
```

NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
Net WLK48
Connect   Case 1
G  A  249      O
F  A  66      I
EndConnect
EndNet
Net WLK62
Connect   Case 1
G  A  241      O
F  A  75      I
EndConnect
EndNet
Net WLK46
Connect   Case 1
G  A  232      O
F  A  83      I
EndConnect
EndNet
Net WLK47
Connect   Case 1
G  A  245      O
F  A  79      I
EndConnect
EndNet
Net WLK61
Connect   Case 1
G  A  235      O
F  A  72      I
EndConnect
EndNet
Net WLK9
Connect   Case 1
G'  A  208  C  102  O
F  A  102      I
EndConnect
EndNet
Net WLK60
Connect   Case 1
G  A  227      O
F  A  86      I
EndConnect
EndNet
Net WLK87
Connect   Case 1
G  A  221      O
F  A  91      I
EndConnect
EndNet
Net WLK25
Connect   Case 1
G'  A  246  C  95  O
F  A  58      I
EndConnect
EndNet
Net WLK59
Connect   Case 1
G  A  219      O
F  A  93      I
EndConnect
EndNet
```

```
Net WLK45
Connect   Case 1
G  A  228      O
F  A  87      I
EndConnect
EndNet
Net WLK88
Connect   Case 1
G  A  229      O
F  A  84      I
EndConnect
EndNet
Net WLK27
Connect   Case 1
G'  A  254  C  121  O
F  A  54      I
EndConnect
EndNet
Net WLK58
Connect   Case 1
G  A  211      O
F  A  99      I
EndConnect
EndNet
Net WLK44
Connect   Case 1
G  A  231      O
F  A  92      I
EndConnect
EndNet
Net WLK26
Connect   Case 1
G'  A  248  C  89  O
F  A  52      I
EndConnect
EndNet
Net WLK29
Connect   Case 1
G'  A  260  C  105  O
F  A  50      I
EndConnect
EndNet
Net WLK72
Connect   Case 1
G  A  217      O
F  A  95      I
EndConnect
EndNet
Net WLK71
Connect   Case 1
G  A  209      O
F  A  101      I
EndConnect
EndNet
Net WLK90
Connect   Case 1
G  A  242      O
F  A  71      I
EndConnect
EndNet
```

WALK_CON.DOC 20765 December 23, 1995 10:48:54 AM
NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
Net WLK28
Connect   Case 1
G'  A  256  C  113  O
F  A  48      I
EndConnect
EndNet
Net WLK8
Connect   Case 1
G'  A  206  C  104  O
F  A  104      I
EndConnect
EndNet
Net WLK7
Connect   Case 1
G'  A  202  C  106  O
F  A  106      I
EndConnect
EndNet
Net WLK30
Connect   Case 1
G'  A  262  C  97  O
F  A  46      I
EndConnect
EndNet
Net WLK31
Connect   Case 1
G'  A  266  C  91  O
F  A  42      I
EndConnect
EndNet
Net WLK57
Connect   Case 1
G  A  203      O
F  A  107      I
EndConnect
EndNet
Net WLK43
Connect   Case 1
G  A  223      O
F  A  98      I
EndConnect
EndNet
Net WLK89
Connect   Case 1
G  A  237      O
F  A  76      I
EndConnect
EndNet
Net WLK91
Connect   Case 1
G  A  250      O
F  A  63      I
EndConnect
EndNet
Net WLK6
Connect   Case 1
G'  A  204  C  110  O
F  A  110      I
EndConnect
EndNet
Net WLK5
Connect   Case 1
G'  A  200  C  112  O
F  A  112      I
EndConnect
EndNet
Net WLK32
Connect   Case 1
G  A  268      O
F  A  44      I
EndConnect
EndNet
Net WLK33
Connect   Case 1
G  A  270      O
F  A  40      I
EndConnect
EndNet
Net WLK70
Connect   Case 1
G  A  201      O
F  A  109      I
EndConnect
EndNet
Net WLK69
Connect   Case 1
G  A  193      O
F  A  117      I
EndConnect
EndNet
Net WLK34
Connect   Case 1
G  A  272      O
F  A  38      I
EndConnect
EndNet
Net WLK35
Connect   Case 1
G  A  276      O
F  A  36      I
EndConnect
EndNet
Net WLK42
Connect   Case 1
G  A  215      O
F  A  103      I
EndConnect
EndNet
Net WLK4
Connect   Case 1
G'  A  196  C  114  O
F  A  114      I
EndConnect
EndNet
Net WLK92
Connect   Case 1
G  A  257      O
F  A  55      I
EndConnect
EndNet
```

WALK_CON.DOC 20765 December 23, 1995 10:48:54 AM
NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
Net WLK94
Connect   Case 1
G  A  271      O
F  A  39      I
EndConnect
EndNet
Net WLK41
Connect   Case 1
G  A  207      O
F  A  108      I
EndConnect
EndNet
Net WLK3
Connect   Case 1
G'  A  198  C  116  O
F  A  116      I
EndConnect
EndNet
Net WLK93
Connect   Case 1
G  A  263      O
F  A  47      I
EndConnect
EndNet
Net WLK36
Connect   Case 1
G  A  278      O
F  A  34      I
EndConnect
EndNet
Net WLK56
Connect   Case 1
G  A  195      O
F  A  111      I
EndConnect
EndNet
Net WLK40
Connect   Case 1
G  A  199      O
F  A  115      I
EndConnect
EndNet
Net WLK38
Connect   Case 1
G  A  282      O
F  A  32      I
EndConnect
EndNet
Net WLK2
Connect   Case 1
G'  A  194  C  118  O
F  A  118      I
EndConnect
EndNet
Net WLK1
Connect   Case 1
G'  A  188  C  120  O
F  A  120      I
EndConnect
EndNet
Net WLK37
Connect   Case 1
G  A  280      O
F  A  30      I
EndConnect
EndNet
Net WLK95
Connect   Case 1
G  A  279      O
F  A  31      I
EndConnect
EndNet
Net WLK39
Connect   Case 1
G  A  192      O
F  A  123      I
EndConnect
EndNet
Net WLK55
Connect   Case 1
G  A  191      O
F  A  119      I
EndConnect
EndNet
Net WLK0
Connect   Case 1
G'  A  190  C  122  O
F  A  122      I
EndConnect
EndNet
Net OP82
Connect   Case 49
E  B  283      O
B  B  5      I
EndConnect
Connect   Case 49
E  B  283      O
B  B  17      I
EndConnect
EndNet
Net OP85
Connect   Case 49
E  B  205      O
B  B  12      I
EndConnect
Connect   Case 49
E  B  205      O
B  B  318      I
EndConnect
EndNet
Net OP81
Connect   Case 49
E  B  275      O
B  B  2      I
EndConnect
Connect   Case 49
E  B  275      O
B  B  25      I
EndConnect
EndNet
Net OP80
Connect   Case 49
E  B  267      O
```

NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
B  B  1      I
EndConnect
Connect   Case 49
E  B  267      O
B  B  291      I
EndConnect
EndNet
Net OP24
Connect   Case 47
E'  B  251  C  267  O
A  C  251      I
EndConnect
EndNet
Net OP79
Connect   Case 49
E  B  261      O
B  B  317      I
EndConnect
Connect   Case 49
E  B  261      O
B  B  299      I
EndConnect
EndNet
Net OP78
Connect   Case 49
E  B  255      O
B  B  314      I
EndConnect
Connect   Case 49
E  B  255      O
B  B  307      I
EndConnect
EndNet
Net OP23
Connect   Case 47
E'  B  247  C  261  O
A  C  247      I
EndConnect
EndNet
Net OP22
Connect   Case 47
E'  B  243  C  255  O
A  C  243      I
EndConnect
EndNet
Net OP68
Connect   Case 49
E  B  277      O
B  B  292      I
EndConnect
Connect   Case 49
E  B  277      O
B  B  319      I
EndConnect
EndNet
Net OP86
Connect   Case 49
E  B  213      O
B  B  16      I
EndConnect
Connect   Case 49
E  B  213      O
B  B  309      I
EndConnect
EndNet
Net OP84
Connect   Case 49
E  B  197      O
B  B  10      I
EndConnect
Connect   Case 49
E  B  197      O
B  B  4      I
EndConnect
EndNet
Net OP21
Connect   Case 47
E'  B  236  C  244  O
A  C  236      I
EndConnect
EndNet
Net OP20
Connect   Case 47
E'  B  234  C  277  O
A  C  234      I
EndConnect
EndNet
Net OP67
Connect   Case 49
E  B  273      O
B  B  290      I
EndConnect
Connect   Case 49
E  B  273      O
B  B  3      I
EndConnect
EndNet
Net OP19
Connect   Case 47
E'  B  230  C  273  O
A  C  230      I
EndConnect
EndNet
Net OP66
Connect   Case 49
E  B  265      O
B  B  288      I
EndConnect
Connect   Case 49
E  B  265      O
B  B  9      I
EndConnect
EndNet
Net OP54
Connect   Case 45
E  B  281      O
A  C  203      I
EndConnect
EndNet
Net OP83
Connect   Case 49
E  B  189      O
B  B  6      I
EndConnect
```

NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
Connect   Case 49
E  B  189      O
B  B  11      I
EndConnect
EndNet
Net OP77
Connect   Case 49
E  B  244      O
B  B  313      I
EndConnect
Connect   Case 49
E  B  244      O
B  B  316      I
EndConnect
EndNet
Net OP18
Connect   Case 47
E'  B  226  C  265  O
A  C  226      I
EndConnect
EndNet
Net OP17
Connect   Case 47
E'  B  222  C  259  O
A  C  222      I
EndConnect
EndNet
Net OP53
Connect   Case 45
E  B  274      O
A  C  211      I
EndConnect
EndNet
Net OP65
Connect   Case 49
E  B  259      O
B  B  284      I
EndConnect
Connect   Case 49
E  B  259      O
B  B  15      I
EndConnect
EndNet
Net OP64
Connect   Case 49
E  B  252      O
B  B  286      I
EndConnect
Connect   Case 49
E  B  252      O
B  B  23      I
EndConnect
EndNet
Net OP52
Connect   Case 45
E  B  269      O
A  C  219      I
EndConnect
EndNet
Net OP16
Connect   Case 47
E'  B  224  C  252  O
A  C  224      I
EndConnect
EndNet
Net OP15
Connect   Case 47
E'  B  218  C  281  O
A  C  218      I
EndConnect
EndNet
Net OP51
Connect   Case 45
E  B  264      O
A  C  227      I
EndConnect
EndNet
Net OP63
Connect   Case 45
E  B  238      O
A  C  192      I
EndConnect
EndNet
Net OP76
Connect   Case 49
E  B  240      O
B  B  310      I
EndConnect
Connect   Case 49
E  B  240      O
B  B  0      I
EndConnect
EndNet
Net OP75
Connect   Case 49
E  B  239      O
B  B  306      I
EndConnect
Connect   Case 49
E  B  239      O
B  B  7      I
EndConnect
EndNet
Net OP14
Connect   Case 47
E'  B  220  C  274  O
A  C  220      I
EndConnect
EndNet
Net OP50
Connect   Case 45
E  B  258      O
A  C  235      I
EndConnect
EndNet
Net OP13
Connect   Case 47
E'  B  214  C  269  O
A  C  214      I
EndConnect
EndNet
Net OP12
Connect   Case 47
```

NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
E'  B  216  C  264  O
A  C  216      I
EndConnect
EndNet
Net OP74
Connect   Case 49
E  B  233      O
B  B  302      I
EndConnect
Connect   Case 49
E  B  233      O
B  B  13      I
EndConnect
EndNet
Net OP73
Connect   Case 49
E  B  225      O
B  B  304      I
EndConnect
Connect   Case 49
E  B  225      O
B  B  19      I
EndConnect
EndNet
Net OP49
Connect   Case 45
E  B  253      O
A  C  241      I
EndConnect
EndNet
Net OP11
Connect   Case 47
E'  B  212  C  258  O
A  C  212      I
EndConnect
EndNet
Net OP10
Connect   Case 47
E'  B  210  C  282  O
A  C  210      I
EndConnect
EndNet
Net OP48
Connect   Case 45
E  B  249      O
A  C  238      I
EndConnect
EndNet
Net OP62
Connect   Case 45
E  B  241      O
A  C  199      I
EndConnect
EndNet
Net OP46
Connect   Case 45
E  B  232      O
A  C  242      I
EndConnect
EndNet
Net OP47
Connect   Case 45
E  B  245      O
A  C  240      I
EndConnect
EndNet
Net OP61
Connect   Case 45
E  B  235      O
A  C  207      I
EndConnect
EndNet
Net OP9
Connect   Case 47
E'  B  208  C  278  O
A  C  208      I
EndConnect
EndNet
Net OP60
Connect   Case 45
E  B  227      O
A  C  215      I
EndConnect
EndNet
Net OP87
Connect   Case 49
E  B  221      O
B  B  18      I
EndConnect
Connect   Case 49
E  B  221      O
B  B  301      I
EndConnect
EndNet
Net OP25
Connect   Case 47
E'  B  246  C  275  O
A  C  246      I
EndConnect
EndNet
Net OP59
Connect   Case 45
E  B  219      O
A  C  223      I
EndConnect
EndNet
Net OP45
Connect   Case 45
E  B  228      O
A  C  193      I
EndConnect
EndNet
Net OP88
Connect   Case 49
E  B  229      O
B  B  20      I
EndConnect
Connect   Case 49
E  B  229      O
B  B  293      I
EndConnect
EndNet
Net OP27
Connect   Case 47
```

NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
E'  B  254  C  250  O
A  C  254      I
EndConnect
EndNet
Net OP58
Connect   Case 45
E  B  211      O
A  C  231      I
EndConnect
EndNet
Net OP44
Connect   Case 45
E  B  231      O
A  C  201      I
EndConnect
EndNet
Net OP26
Connect   Case 47
E'  B  248  C  283  O
A  C  248      I
EndConnect
EndNet
Net OP29
Connect   Case 47
E'  B  260  C  263  O
A  C  249      I
EndConnect
EndNet
Net OP72
Connect   Case 49
E  B  217      O
B  B  298      I
EndConnect
Connect   Case 49
E  B  217      O
B  B  287      I
EndConnect
EndNet
Net OP71
Connect   Case 49
E  B  209      O
B  B  300      I
EndConnect
Connect   Case 49
E  B  209      O
B  B  295      I
EndConnect
EndNet
Net OP90
Connect   Case 49
E  B  242      O
B  B  22      I
EndConnect
Connect   Case 49
E  B  242      O
B  B  26      I
EndConnect
EndNet
Net OP28
Connect   Case 47
E'  B  256  C  257  O
A  C  253      I
EndConnect
EndNet
Net OP8
Connect   Case 47
E'  B  206  C  280  O
A  C  206      I
EndConnect
EndNet
Net OP7
Connect   Case 47
E'  B  202  C  276  O
A  C  202      I
EndConnect
EndNet
Net OP30
Connect   Case 47
E'  B  262  C  271  O
A  C  245      I
EndConnect
EndNet
Net OP31
Connect   Case 47
E'  B  266  C  279  O
A  C  232      I
EndConnect
EndNet
Net OP57
Connect   Case 45
E  B  203      O
A  C  228      I
EndConnect
EndNet
Net OP43
Connect   Case 45
E  B  223      O
A  C  209      I
EndConnect
EndNet
Net OP89
Connect   Case 49
E  B  237      O
B  B  24      I
EndConnect
Connect   Case 49
E  B  237      O
B  B  285      I
EndConnect
EndNet
Net OP91
Connect   Case 49
E  B  250      O
B  B  28      I
EndConnect
Connect   Case 49
E  B  250      O
B  B  21      I
EndConnect
EndNet
Net OP6
Connect   Case 47
E'  B  204  C  272  O
A  C  204      I
```

```
EndConnect
EndNet
Net OP5
Connect   Case 47
E'  B  200  C  270  O
A  C  200      I
EndConnect
EndNet
Net OP32
Connect   Case 45
E  B  268      O
A  C  237      I
EndConnect
EndNet
Net OP33
Connect   Case 45
E  B  270      O
A  C  229      I
EndConnect
EndNet
Net OP70
Connect   Case 49
E  B  201      O
B  B  294      I
EndConnect
Connect   Case 49
E  B  201      O
B  B  303      I
EndConnect
EndNet
Net OP69
Connect   Case 49
E  B  193      O
B  B  296      I
EndConnect
Connect   Case 49
E  B  193      O
B  B  311      I
EndConnect
EndNet
Net OP34
Connect   Case 45
E  B  272      O
A  C  221      I
EndConnect
EndNet
Net OP35
Connect   Case 45
E  B  276      O
A  C  213      I
EndConnect
EndNet
Net OP42
Connect   Case 45
E  B  215      O
A  C  217      I
EndConnect
EndNet
Net OP4
Connect   Case 47
E'  B  196  C  268  O
A  C  196      I
EndConnect
EndNet
Net OP92
Connect   Case 49
E  B  257      O
B  B  289      I
EndConnect
Connect   Case 49
E  B  257      O
B  B  14      I
EndConnect
EndNet
Net OP94
Connect   Case 49
E  B  271      O
B  B  305      I
EndConnect
Connect   Case 49
E  B  271      O
B  B  315      I
EndConnect
EndNet
Net OP41
Connect   Case 45
E  B  207      O
A  C  225      I
EndConnect
EndNet
Net OP3
Connect   Case 47
E'  B  198  C  266  O
A  C  198      I
EndConnect
EndNet
Net OP93
Connect   Case 49
E  B  263      O
B  B  297      I
EndConnect
Connect   Case 49
E  B  263      O
B  B  8      I
EndConnect
EndNet
Net OP36
Connect   Case 45
E  B  280      O
A  C  205      I
EndConnect
EndNet
Net OP56
Connect   Case 45
E  B  195      O
A  C  191      I
EndConnect
EndNet
Net OP40
Connect   Case 45
E  B  199      O
A  C  233      I
EndConnect
EndNet
```

NET CONNECTION NETLIST CREATED BY NET2CON.EXE
INPUT FILES: WALK.ITC OUTPUT FILES: WALK.CON

```
Net OP38
Connect   Case 45
E  B  282      O
A  C  189      I
EndConnect
EndNet
Net OP2
Connect   Case 47
E'  B  194  C  262  O
A  C  194      I
EndConnect
EndNet
Net OP1
Connect   Case 47
E'  B  188  C  260  O
A  C  188      I
EndConnect
EndNet
Net OP37
Connect   Case 45
E  B  278      O
A  C  197      I
EndConnect
EndNet
Net OP95
Connect   Case 49
E  B  279      O
B  B  308      I
EndConnect
Connect   Case 49
E  B  279      O
B  B  312      I
EndConnect
EndNet
Net OP39
Connect   Case 45
E  B  192      O
A  C  239      I
EndConnect
EndNet
Net OP55
Connect   Case 45
E  B  191      O
A  C  195      I
EndConnect
EndNet
Net OP0
Connect   Case 47
E'  B  190  C  256  O
A  C  190      I
EndConnect
EndNet
eof
```

WALK_ROU.DOC 30742 December 23, 1995 10:56:51 AM
NET ROUTING NETLIST CREATED BY CON2ROU.EXE
INPUT FILES: WALK.CON OUTPUT FILES: WALK.ROU

```
Net  WLK82
Connect
A Portmap  283  WLK82_G0  [FSEL= IN]
A Portmap  27  WLK82_F0  [FSEL= OP]
EndConnect
EndNet
Net  WLK85
Connect
A Portmap  205  WLK85_G1  [FSEL= IN]
A Portmap  105  WLK85_F1  [FSEL= OP]
EndConnect
EndNet
Net  WLK81
Connect
A Portmap  275  WLK81_G2  [FSEL= IN]
A Portmap  35  WLK81_F2  [FSEL= OP]
EndConnect
EndNet
Net  WLK80
Connect
A Portmap  267  WLK80_G3  [FSEL= IN]
A Portmap  43  WLK80_F3  [FSEL= OP]
EndConnect
EndNet
Net  WLK24
Connect
A Portmap  251  WLK24_G4  [FSEL= IN]
A Portmap  56  WLK24_F4  [FSEL= OP]
EndConnect
EndNet
Net  WLK79
Connect
A Portmap  261  WLK79_G5  [FSEL= IN]
A Portmap  51  WLK79_F5  [FSEL= OP]
EndConnect
EndNet
Net  WLK78
Connect
A Portmap  255  WLK78_G6  [FSEL= IN]
A Portmap  59  WLK78_F6  [FSEL= OP]
EndConnect
EndNet
Net  WLK23
Connect
A Portmap  247  WLK23_G7  [FSEL= IN]
A Portmap  60  WLK23_F7  [FSEL= OP]
EndConnect
EndNet
Net  WLK22
Connect
A Portmap  243  WLK22_G8  [FSEL= IN]
A Portmap  64  WLK22_F8  [FSEL= OP]
EndConnect
EndNet
Net  WLK68
Connect
A Portmap  277  WLK68_G9  [FSEL= IN]
A Portmap  29  WLK68_F9  [FSEL= OP]
EndConnect
EndNet
Net  WLK86
Connect
A Portmap  213  WLK86_G10  [FSEL= IN]
A Portmap  97  WLK86_F10  [FSEL= OP]
EndConnect
EndNet
Net  WLK84
Connect
A Portmap  197  WLK84_G11  [FSEL= IN]
A Portmap  113  WLK84_F11  [FSEL= OP]
EndConnect
EndNet
Net  WLK21
Connect
A Portmap  236  WLK21_G12  [FSEL= IN]
A Portmap  68  WLK21_F12  [FSEL= OP]
EndConnect
EndNet
Net  WLK20
Connect
A Portmap  234  WLK20_G13  [FSEL= IN]
A Portmap  70  WLK20_F13  [FSEL= OP]
EndConnect
EndNet
Net  WLK67
Connect
A Portmap  273  WLK67_G14  [FSEL= IN]
A Portmap  37  WLK67_F14  [FSEL= OP]
EndConnect
EndNet
Net  WLK19
Connect
A Portmap  230  WLK19_G15  [FSEL= IN]
A Portmap  77  WLK19_F15  [FSEL= OP]
EndConnect
EndNet
Net  WLK66
Connect
A Portmap  265  WLK66_G16  [FSEL= IN]
A Portmap  45  WLK66_F16  [FSEL= OP]
EndConnect
EndNet
Net  WLK54
Connect
A Portmap  281  WLK54_G17  [FSEL= IN]
A Portmap  33  WLK54_F17  [FSEL= OP]
EndConnect
EndNet
Net  WLK83
Connect
A Portmap  189  WLK83_G18  [FSEL= IN]
A Portmap  121  WLK83_F18  [FSEL= OP]
EndConnect
EndNet
Net  WLK77
Connect
A Portmap  244  WLK77_G19  [FSEL= IN]
A Portmap  67  WLK77_F19  [FSEL= OP]
EndConnect
EndNet
Net  WLK18
onnect
A Portmap  226  WLK18_G20  [FSEL= IN]
A Portmap  81  WLK18_F20  [FSEL= OP]
```

```
EndConnect
EndNet
Net WLK17
Connect
A Portmap 222 WLK17_G21 [FSEL= IN]
A Portmap 85 WLK17_F21 [FSEL= OP]
EndConnect
EndNet
Net WLK53
Connect
A Portmap 274 WLK53_G22 [FSEL= IN]
A Portmap 41 WLK53_F22 [FSEL= OP]
EndConnect
EndNet
Net WLK65
Connect
A Portmap 259 WLK65_G23 [FSEL= IN]
A Portmap 53 WLK65_F23 [FSEL= OP]
EndConnect
EndNet
Net WLK64
Connect
A Portmap 252 WLK64_G24 [FSEL= IN]
A Portmap 61 WLK64_F24 [FSEL= OP]
EndConnect
EndNet
Net WLK52
Connect
A Portmap 269 WLK52_G25 [FSEL= IN]
A Portmap 49 WLK52_F25 [FSEL= OP]
EndConnect
EndNet
Net WLK16
Connect
A Portmap 224 WLK16_G26 [FSEL= IN]
A Portmap 80 WLK16_F26 [FSEL= OP]
EndConnect
EndNet
Net WLK15
Connect
A Portmap 218 WLK15_G27 [FSEL= IN]
A Portmap 82 WLK15_F27 [FSEL= OP]
EndConnect
EndNet
Net WLK51
Connect
A Portmap 264 WLK51_G28 [FSEL= IN]
A Portmap 57 WLK51_F28 [FSEL= OP]
EndConnect
EndNet
Net WLK63
Connect
A Portmap 238 WLK63_G29 [FSEL= IN]
A Portmap 69 WLK63_F29 [FSEL= OP]
EndConnect
EndNet
Net WLK76
Connect
A Portmap 240 WLK76_G30 [FSEL= IN]
A Portmap 73 WLK76_F30 [FSEL= OP]
EndConnect
EndNet
Net WLK75
Connect
A Portmap 239 WLK75_G31 [FSEL= IN]
A Portmap 74 WLK75_F31 [FSEL= OP]
EndConnect
EndNet
Net WLK14
Connect
A Portmap 220 WLK14_G32 [FSEL= IN]
A Portmap 88 WLK14_F32 [FSEL= OP]
EndConnect
EndNet
Net WLK50
Connect
A Portmap 258 WLK50_G33 [FSEL= IN]
A Portmap 65 WLK50_F33 [FSEL= OP]
EndConnect
EndNet
Net WLK13
Connect
A Portmap 214 WLK13_G34 [FSEL= IN]
A Portmap 90 WLK13_F34 [FSEL= OP]
EndConnect
EndNet
Net WLK12
Connect
A Portmap 216 WLK12_G35 [FSEL= IN]
A Portmap 94 WLK12_F35 [FSEL= OP]
EndConnect
EndNet
Net WLK74
Connect
A Portmap 233 WLK74_G36 [FSEL= IN]
A Portmap 78 WLK74_F36 [FSEL= OP]
EndConnect
EndNet
Net WLK73
Connect
A Portmap 225 WLK73_G37 [FSEL= IN]
A Portmap 89 WLK73_F37 [FSEL= OP]
EndConnect
EndNet
Net WLK49
Connect
A Portmap 253 WLK49_G38 [FSEL= IN]
A Portmap 62 WLK49_F38 [FSEL= OP]
EndConnect
EndNet
Net WLK11
Connect
A Portmap 212 WLK11_G39 [FSEL= IN]
A Portmap 96 WLK11_F39 [FSEL= OP]
EndConnect
EndNet
Net WLK10
Connect
A Portmap 210 WLK10_G40 [FSEL= IN]
A Portmap 100 WLK10_F40 [FSEL= OP]
EndConnect
EndNet
Net WLK48
Connect
```

WALK_ROU.DOC 30742 December 23, 1995 10:56:51 AM
NET ROUTING NETLIST CREATED BY CON2ROU.EXE
INPUT FILES: WALK.CON OUTPUT FILES: WALK.ROU

```
A Portmap  249  WLK48_G41  [FSEL= IN]
A Portmap  66  WLK48_F41  [FSEL= OP]
EndConnect
EndNet
Net  WLK62
Connect
A Portmap  241  WLK62_G42  [FSEL= IN]
A Portmap  75  WLK62_F42  [FSEL= OP]
EndConnect
EndNet
Net  WLK46
Connect
A Portmap  232  WLK46_G43  [FSEL= IN]
A Portmap  83  WLK46_F43  [FSEL= OP]
EndConnect
EndNet
Net  WLK47
Connect
A Portmap  245  WLK47_G44  [FSEL= IN]
A Portmap  79  WLK47_F44  [FSEL= OP]
EndConnect
EndNet
Net  WLK61
Connect
A Portmap  235  WLK61_G45  [FSEL= IN]
A Portmap  72  WLK61_F45  [FSEL= OP]
EndConnect
EndNet
Net  WLK9
Connect
A Portmap  208  WLK9_G46  [FSEL= IN]
A Portmap  102  WLK9_F46  [FSEL= OP]
EndConnect
EndNet
Net  WLK60
Connect
A Portmap  227  WLK60_G47  [FSEL= IN]
A Portmap  86  WLK60_F47  [FSEL= OP]
EndConnect
EndNet
Net  WLK87
Connect
A Portmap  221  WLK87_G48  [FSEL= IN]
A Portmap  91  WLK87_F48  [FSEL= OP]
EndConnect
EndNet
Net  WLK25
Connect
A Portmap  246  WLK25_G49  [FSEL= IN]
A Portmap  58  WLK25_F49  [FSEL= OP]
EndConnect
EndNet
Net  WLK59
Connect
A Portmap  219  WLK59_G50  [FSEL= IN]
A Portmap  93  WLK59_F50  [FSEL= OP]
EndConnect
EndNet
Net  WLK45
Connect
A Portmap  228  WLK45_G51  [FSEL= IN]
A Portmap  87  WLK45_F51  [FSEL= OP]
EndConnect
EndNet
Net  WLK88
Connect
A Portmap  229  WLK88_G52  [FSEL= IN]
A Portmap  84  WLK88_F52  [FSEL= OP]
EndConnect
EndNet
Net  WLK27
Connect
A Portmap  254  WLK27_G53  [FSEL= IN]
A Portmap  54  WLK27_F53  [FSEL= OP]
EndConnect
EndNet
Net  WLK58
Connect
A Portmap  211  WLK58_G54  [FSEL= IN]
A Portmap  99  WLK58_F54  [FSEL= OP]
EndConnect
EndNet
Net  WLK44
Connect
A Portmap  231  WLK44_G55  [FSEL= IN]
A Portmap  92  WLK44_F55  [FSEL= OP]
EndConnect
EndNet
Net  WLK26
Connect
A Portmap  248  WLK26_G56  [FSEL= IN]
A Portmap  52  WLK26_F56  [FSEL= OP]
EndConnect
EndNet
Net  WLK29
Connect
A Portmap  260  WLK29_G57  [FSEL= IN]
A Portmap  50  WLK29_F57  [FSEL= OP]
EndConnect
EndNet
Net  WLK72
Connect
A Portmap  217  WLK72_G58  [FSEL= IN]
A Portmap  95  WLK72_F58  [FSEL= OP]
EndConnect
EndNet
Net  WLK71
Connect
A Portmap  209  WLK71_G59  [FSEL= IN]
A Portmap  101  WLK71_F59  [FSEL= OP]
EndConnect
EndNet
Net  WLK90
Connect
A Portmap  242  WLK90_G60  [FSEL= IN]
A Portmap  71  WLK90_F60  [FSEL= OP]
EndConnect
EndNet
Net  WLK28
Connect
A Portmap  256  WLK28_G61  [FSEL= IN]
A Portmap  48  WLK28_F61  [FSEL= OP]
EndConnect
EndNet
```

WALK_ROU.DOC 30742 December 23, 1995 10:56:51 AM
NET ROUTING NETLIST CREATED BY CON2ROU.EXE
INPUT FILES: WALK.CON OUTPUT FILES: WALK.ROU

```
Net  WLK8
Connect
A Portmap  206  WLK8_G62  [FSEL= IN]
A Portmap  104  WLK8_F62  [FSEL= OP]
EndConnect
EndNet
Net  WLK7
Connect
A Portmap  202  WLK7_G63  [FSEL= IN]
A Portmap  106  WLK7_F63  [FSEL= OP]
EndConnect
EndNet
Net  WLK30
Connect
A Portmap  262  WLK30_G64  [FSEL= IN]
A Portmap  46  WLK30_F64  [FSEL= OP]
EndConnect
EndNet
Net  WLK31
Connect
A Portmap  266  WLK31_G65  [FSEL= IN]
A Portmap  42  WLK31_F65  [FSEL= OP]
EndConnect
EndNet
Net  WLK57
Connect
A Portmap  203  WLK57_G66  [FSEL= IN]
A Portmap  107  WLK57_F66  [FSEL= OP]
EndConnect
EndNet
Net  WLK43
Connect
A Portmap  223  WLK43_G67  [FSEL= IN]
A Portmap  98  WLK43_F67  [FSEL= OP]
EndConnect
EndNet
Net  WLK89
Connect
A Portmap  237  WLK89_G68  [FSEL= IN]
A Portmap  76  WLK89_F68  [FSEL= OP]
EndConnect
EndNet
Net  WLK91
Connect
A Portmap  250  WLK91_G69  [FSEL= IN]
I Portmap  63  WLK91_F69  [FSEL= OP]
EndConnect
EndNet
Net  WLK6
Connect
A Portmap  204  WLK6_G70  [FSEL= IN]
A Portmap  110  WLK6_F70  [FSEL= OP]
EndConnect
EndNet
Net  WLK5
Connect
A Portmap  200  WLK5_G71  [FSEL= IN]
A Portmap  112  WLK5_F71  [FSEL= OP]
EndConnect
EndNet
Net  WLK32
Connect
A Portmap  268  WLK32_G72  [FSEL= IN]
A Portmap  44  WLK32_F72  [FSEL= OP]
EndConnect
EndNet
Net  WLK33
Connect
A Portmap  270  WLK33_G73  [FSEL= IN]
A Portmap  40  WLK33_F73  [FSEL= OP]
EndConnect
EndNet
Net  WLK70
Connect
A Portmap  201  WLK70_G74  [FSEL= IN]
A Portmap  109  WLK70_F74  [FSEL= OP]
EndConnect
EndNet
Net  WLK69
Connect
A Portmap  193  WLK69_G75  [FSEL= IN]
A Portmap  117  WLK69_F75  [FSEL= OP]
EndConnect
EndNet
Net  WLK34
Connect
A Portmap  272  WLK34_G76  [FSEL= IN]
A Portmap  38  WLK34_F76  [FSEL= OP]
EndConnect
EndNet
Net  WLK35
Connect
A Portmap  276  WLK35_G77  [FSEL= IN]
A Portmap  36  WLK35_F77  [FSEL= OP]
EndConnect
EndNet
Net  WLK42
Connect
A Portmap  215  WLK42_G78  [FSEL= IN]
A Portmap  103  WLK42_F78  [FSEL= OP]
EndConnect
EndNet
Net  WLK4
Connect
A Portmap  196  WLK4_G79  [FSEL= IN]
A Portmap  114  WLK4_F79  [FSEL= OP]
EndConnect
EndNet
Net  WLK92
Connect
A Portmap  257  WLK92_G80  [FSEL= IN]
A Portmap  55  WLK92_F80  [FSEL= OP]
EndConnect
EndNet
Net  WLK94
Connect
A Portmap  271  WLK94_G81  [FSEL= IN]
A Portmap  39  WLK94_F81  [FSEL= OP]
EndConnect
EndNet
Net  WLK41
Connect
A Portmap  207  WLK41_G82  [FSEL= IN]
A Portmap  108  WLK41_F82  [FSEL= OP]
```

```
EndConnect
EndNet
Net WLK3
Connect
A Portmap 198 WLK3_G83 [FSEL= IN]
A Portmap 116 WLK3_F83 [FSEL= OP]
EndConnect
EndNet
Net WLK93
Connect
A Portmap 263 WLK93_G84 [FSEL= IN]
A Portmap 47 WLK93_F84 [FSEL= OP]
EndConnect
EndNet
Net WLK36
Connect
A Portmap 278 WLK36_G85 [FSEL= IN]
A Portmap 34 WLK36_F85 [FSEL= OP]
EndConnect
EndNet
Net WLK56
Connect
A Portmap 195 WLK56_G86 [FSEL= IN]
A Portmap 111 WLK56_F86 [FSEL= OP]
EndConnect
EndNet
Net WLK40
Connect
A Portmap 199 WLK40_G87 [FSEL= IN]
A Portmap 115 WLK40_F87 [FSEL= OP]
EndConnect
EndNet
Net WLK38
Connect
A Portmap 282 WLK38_G88 [FSEL= IN]
A Portmap 32 WLK38_F88 [FSEL= OP]
EndConnect
EndNet
Net WLK2
Connect
A Portmap 194 WLK2_G89 [FSEL= IN]
A Portmap 118 WLK2_F89 [FSEL= OP]
EndConnect
EndNet
Net WLK1
Connect
A Portmap 188 WLK1_G90 [FSEL= IN]
A Portmap 120 WLK1_F90 [FSEL= OP]
EndConnect
EndNet
Net WLK37
Connect
A Portmap 280 WLK37_G91 [FSEL= IN]
A Portmap 30 WLK37_F91 [FSEL= OP]
EndConnect
EndNet
Net WLK95
Connect
A Portmap 279 WLK95_G92 [FSEL= IN]
A Portmap 31 WLK95_F92 [FSEL= OP]
EndConnect
EndNet
Net WLK39
Connect
A Portmap 192 WLK39_G93 [FSEL= IN]
A Portmap 123 WLK39_F93 [FSEL= OP]
EndConnect
EndNet
Net WLK55
Connect
A Portmap 191 WLK55_G94 [FSEL= IN]
A Portmap 119 WLK55_F94 [FSEL= OP]
EndConnect
EndNet
Net WLK0
Connect
A Portmap 190 WLK0_G95 [FSEL= IN]
A Portmap 122 WLK0_F95 [FSEL= OP]
EndConnect
EndNet
Net OP82
Connect
B Portmap 283 OP82_E0 [FSEL= IN]
B Portmap 5 OP82_B0 [FSEL= OP]
EndConnect
Connect
B Portmap 283 OP82_E0 [FSEL= IN]
B Portmap 17 OP82_B0 [FSEL= OP]
EndConnect
EndNet
Net OP85
Connect
B Portmap 205 OP85_E1 [FSEL= IN]
B Portmap 12 OP85_B1 [FSEL= OP]
EndConnect
Connect
B Portmap 205 OP85_E1 [FSEL= IN]
B Portmap 318 OP85_B1 [FSEL= OP]
EndConnect
EndNet
Net OP81
Connect
B Portmap 275 OP81_E2 [FSEL= IN]
B Portmap 2 OP81_B2 [FSEL= OP]
EndConnect
Connect
B Portmap 275 OP81_E2 [FSEL= IN]
B Portmap 25 OP81_B2 [FSEL= OP]
EndConnect
EndNet
Net OP80
Connect
B Portmap 267 OP80_E3 [FSEL= IN]
B Portmap 1 OP80_B3 [FSEL= OP]
EndConnect
Connect
B Portmap 267 OP80_E3 [FSEL= IN]
B Portmap 291 OP80_B3 [FSEL= OP]
EndConnect
EndNet
Net OP24
Connect
C Portmap 267 OP24_E'4 [FSEL= IN]
C Portmap 251 OP24_A0 [FSEL= OP]
```

```
EndConnect.
EndNet
Net  OP79
Connect
B Portmap  261  OP79_E5  [FSEL= IN]
B Portmap  317  OP79_B4  [FSEL= OP]
EndConnect
Connect
B Portmap  261  OP79_E5  [FSEL= IN]
B Portmap  299  OP79_B4  [FSEL= OP]
EndConnect
EndNet
Net  OP78
Connect
B Portmap  255  OP78_E6  [FSEL= IN]
B Portmap  314  OP78_B5  [FSEL= OP]
EndConnect
Connect
B Portmap  255  OP78_E6  [FSEL= IN]
B Portmap  307  OP78_B5  [FSEL= OP]
EndConnect
EndNet
Net  OP23
Connect
C Portmap  261  OP23_E'7 [FSEL= IN]
C Portmap  247  OP23_A1  [FSEL= OP]
EndConnect
EndNet
Net  OP22
Connect
C Portmap  255  OP22_E'8 [FSEL= IN]
C Portmap  243  OP22_A2  [FSEL= OP]
EndConnect
EndNet
Net  OP68
Connect
B Portmap  277  OP68_E9  [FSEL= IN]
B Portmap  292  OP68_B6  [FSEL= OP]
EndConnect
Connect
B Portmap  277  OP68_E9  [FSEL= IN]
B Portmap  319  OP68_B6  [FSEL= OP]
EndConnect
EndNet
Net  OP86
Connect
B Portmap  213  OP86_E10  [FSEL= IN]
B Portmap  16  OP86_B7  [FSEL= OP]
EndConnect
Connect
B Portmap  213  OP86_E10  [FSEL= IN]
B Portmap  309  OP86_B7  [FSEL= OP]
EndConnect
EndNet
Net  OP84
Connect
B Portmap  197  OP84_E11  [FSEL= IN]
B Portmap  10  OP84_B8  [FSEL= OP]
EndConnect
Connect
B Portmap  197  OP84_E11  [FSEL= IN]
B Portmap  4  OP84_B8  [FSEL= OP]
EndConnect
EndNet
Net  OP21
Connect
C Portmap  244  OP21_E'12 [FSEL= IN]
C Portmap  236  OP21_A3  [FSEL= OP]
EndConnect
EndNet
Net  OP20
Connect
C Portmap  277  OP20_E'13 [FSEL= IN]
C Portmap  234  OP20_A4  [FSEL= OP]
EndConnect
EndNet
Net  OP67
Connect
B Portmap  273  OP67_E14  [FSEL= IN]
B Portmap  290  OP67_B9  [FSEL= OP]
EndConnect
Connect
B Portmap  273  OP67_E14  [FSEL= IN]
B Portmap  3  OP67_B9  [FSEL= OP]
EndConnect
EndNet
Net  OP19
Connect
C Portmap  273  OP19_E'15 [FSEL= IN]
C Portmap  230  OP19_A5  [FSEL= OP]
EndConnect
EndNet
Net  OP66
Connect
B Portmap  265  OP66_E16  [FSEL= IN]
B Portmap  288  OP66_B10  [FSEL= OP]
EndConnect
Connect
B Portmap  265  OP66_E16  [FSEL= IN]
B Portmap  9  OP66_B10  [FSEL= OP]
EndConnect
EndNet
Net  OP54
Connect
B Portmap  281  OP54_E17  [FSEL= IN]
B Portmap  186  OP54_M0  [FSEL= OP]
C Portmap  286  OP54_M0  [FSEL= IN]
C Portmap  203  OP54_A6  [FSEL= OP]
EndConnect
EndNet
Net  OP83
Connect
B Portmap  189  OP83_E18  [FSEL= IN]
B Portmap  6  OP83_B11  [FSEL= OP]
EndConnect
Connect
B Portmap  189  OP83_E18  [FSEL= IN]
B Portmap  11  OP83_B11  [FSEL= OP]
EndConnect
EndNet
Net  OP77
Connect
B Portmap  244  OP77_E19  [FSEL= IN]
B Portmap  313  OP77_B12  [FSEL= OP]
```

```
EndConnect
Connect
B Portmap  244  OP77_E19  [FSEL= IN]
B Portmap  316  OP77_B12  [FSEL= OP]
EndConnect
EndNet
Net  OP18
Connect
C Portmap  265  OP18_E'20 [FSEL= IN]
C Portmap  226  OP18_A7  [FSEL= OP]
EndConnect
EndNet
Net  OP17
Connect
C Portmap  259  OP17_E'21 [FSEL= IN]
C Portmap  222  OP17_A8  [FSEL= OP]
EndConnect
EndNet
Net  OP53
Connect
B Portmap  274  OP53_E22   [FSEL= IN]
B Portmap  184  OP53_M1   [FSEL= OP]
C Portmap  284  OP53_M1   [FSEL= IN]
C Portmap  211  OP53_A9   [FSEL= OP]
EndConnect
EndNet
Net  OP65
Connect
B Portmap  259  OP65_E23  [FSEL= IN]
B Portmap  284  OP65_B13  [FSEL= OP]
EndConnect
Connect
B Portmap  259  OP65_E23  [FSEL= IN]
B Portmap  15  OP65_B13  [FSEL= OP]
EndConnect
EndNet
Net  OP64
Connect
B Portmap  252  OP64_E24  [FSEL= IN]
B Portmap  286  OP64_B14  [FSEL= OP]
EndConnect
Connect
B Portmap  252  OP64_E24  [FSEL= IN]
B Portmap  23  OP64_B14  [FSEL= OP]
EndConnect
EndNet
Net  OP52
Connect
B Portmap  269  OP52_E25   [FSEL= IN]
B Portmap  182  OP52_M2   [FSEL= OP]
C Portmap  288  OP52_M2   [FSEL= IN]
C Portmap  219  OP52_A10   [FSEL= OP]
EndConnect
EndNet
Net  OP16
Connect
C Portmap  252  OP16_E'26 [FSEL= IN]
C Portmap  224  OP16_A11  [FSEL= OP]
EndConnect
EndNet
Net  OP15
Connect
C Portmap  281  OP15_E'27 [FSEL= IN]
C Portmap  218  OP15_A12  [FSEL= OP]
EndConnect
EndNet
Net  OP51
Connect
B Portmap  264  OP51_E28   [FSEL= IN]
B Portmap  178  OP51_M3   [FSEL= OP]
C Portmap  290  OP51_M3   [FSEL= IN]
C Portmap  227  OP51_A13   [FSEL= OP]
EndConnect
EndNet
Net  OP63
Connect
B Portmap  238  OP63_E29   [FSEL= IN]
B Portmap  176  OP63_M4   [FSEL= OP]
C Portmap  292  OP63_M4   [FSEL= IN]
C Portmap  192  OP63_A14   [FSEL= OP]
EndConnect
EndNet
Net  OP76
Connect
B Portmap  240  OP76_E30  [FSEL= IN]
B Portmap  310  OP76_B15  [FSEL= OP]
EndConnect
Connect
B Portmap  240  OP76_E30  [FSEL= IN]
B Portmap  0  OP76_B15  [FSEL= OP]
EndConnect
EndNet
Net  OP75
Connect
B Portmap  239  OP75_E31  [FSEL= IN]
B Portmap  306  OP75_B16  [FSEL= OP]
EndConnect
Connect
B Portmap  239  OP75_E31  [FSEL= IN]
B Portmap  7  OP75_B16  [FSEL= OP]
EndConnect
EndNet
Net  OP14
Connect
C Portmap  274  OP14_E'32 [FSEL= IN]
C Portmap  220  OP14_A15  [FSEL= OP]
EndConnect
EndNet
Net  OP50
Connect
B Portmap  258  OP50_E33   [FSEL= IN]
B Portmap  172  OP50_M5   [FSEL= OP]
C Portmap  296  OP50_M5   [FSEL= IN]
C Portmap  235  OP50_A16   [FSEL= OP]
EndConnect
EndNet
Net  OP13
Connect
C Portmap  269  OP13_E'34 [FSEL= IN]
C Portmap  214  OP13_A17  [FSEL= OP]
EndConnect
EndNet
Net  OP12
Connect
```

WALK_ROU.DOC 30742 December 23, 1995 10:56:51 AM
NET ROUTING NETLIST CREATED BY CON2ROU.EXE
INPUT FILES: WALK.CON OUTPUT FILES: WALK.ROU

```
C Portmap  264  OP12_E'35 [FSEL= IN]
C Portmap  216  OP12_A18  [FSEL= OP]
EndConnect
EndNet
Net  OP74
Connect
B Portmap  233  OP74_E36  [FSEL= IN]
B Portmap  302  OP74_B17  [FSEL= OP]
EndConnect
Connect
B Portmap  233  OP74_E36  [FSEL= IN]
B Portmap  13  OP74_B17  [FSEL= OP]
EndConnect
EndNet
Net  OP73
Connect
B Portmap  225  OP73_E37  [FSEL= IN]
B Portmap  304  OP73_B18  [FSEL= OP]
EndConnect
Connect
B Portmap  225  OP73_E37  [FSEL= IN]
B Portmap  19  OP73_B18  [FSEL= OP]
EndConnect
EndNet
Net  OP49
Connect
B Portmap  253  OP49_E38   [FSEL= IN]
B Portmap  170  OP49_M6   [FSEL= OP]
C Portmap  294  OP49_M6   [FSEL= IN]
C Portmap  241  OP49_A19   [FSEL= OP]
EndConnect
EndNet
Net  OP11
Connect
C Portmap  258  OP11_E'39 [FSEL= IN]
C Portmap  212  OP11_A20  [FSEL= OP]
EndConnect
EndNet
Net  OP10
Connect
C Portmap  282  OP10_E'40 [FSEL= IN]
C Portmap  210  OP10_A21  [FSEL= OP]
EndConnect
EndNet
Net  OP48
Connect
B Portmap  249  OP48_E41   [FSEL= IN]
B Portmap  164  OP48_M7   [FSEL= OP]
C Portmap  300  OP48_M7   [FSEL= IN]
C Portmap  238  OP48_A22   [FSEL= OP]
EndConnect
EndNet
Net  OP62
Connect
B Portmap  241  OP62_E42   [FSEL= IN]
B Portmap  162  OP62_M8   [FSEL= OP]
C Portmap  298  OP62_M8   [FSEL= IN]
C Portmap  199  OP62_A23   [FSEL= OP]
EndConnect
EndNet
Net  OP46
Connect
B Portmap  232  OP46_E43   [FSEL= IN]
B Portmap  169  OP46_M9   [FSEL= OP]
C Portmap  304  OP46_M9   [FSEL= IN]
C Portmap  242  OP46_A24   [FSEL= OP]
EndConnect
EndNet
Net  OP47
Connect
B Portmap  245  OP47_E44   [FSEL= IN]
B Portmap  165  OP47_M10   [FSEL= OP]
C Portmap  302  OP47_M10   [FSEL= IN]
C Portmap  240  OP47_A25   [FSEL= OP]
EndConnect
EndNet
Net  OP61
Connect
B Portmap  235  OP61_E45   [FSEL= IN]
B Portmap  156  OP61_M11   [FSEL= OP]
C Portmap  306  OP61_M11   [FSEL= IN]
C Portmap  207  OP61_A26   [FSEL= OP]
EndConnect
EndNet
Net  OP9
Connect
C Portmap  278  OP9_E'46 [FSEL= IN]
C Portmap  208  OP9_A27  [FSEL= OP]
EndConnect
EndNet
Net  OP60
Connect
B Portmap  227  OP60_E47   [FSEL= IN]
B Portmap  161  OP60_M12   [FSEL= OP]
C Portmap  310  OP60_M12   [FSEL= IN]
C Portmap  215  OP60_A28   [FSEL= OP]
EndConnect
EndNet
Net  OP87
Connect
B Portmap  221  OP87_E48  [FSEL= IN]
B Portmap  18  OP87_B19  [FSEL= OP]
EndConnect
Connect
B Portmap  221  OP87_E48  [FSEL= IN]
B Portmap  301  OP87_B19  [FSEL= OP]
EndConnect
EndNet
Net  OP25
Connect
C Portmap  275  OP25_E'49 [FSEL= IN]
C Portmap  246  OP25_A29  [FSEL= OP]
EndConnect
EndNet
Net  OP59
Connect
B Portmap  219  OP59_E50   [FSEL= IN]
B Portmap  163  OP59_M13   [FSEL= OP]
C Portmap  313  OP59_M13   [FSEL= IN]
C Portmap  223  OP59_A30   [FSEL= OP]
EndConnect
EndNet
Net  OP45
Connect
```

```
B Portmap  228  OP45_E51   [FSEL= IN]
B Portmap  150  OP45_M14   [FSEL= OP]
C Portmap  314  OP45_M14   [FSEL= IN]
C Portmap  193  OP45_A31   [FSEL= OP]
EndConnect
EndNet
Net  OP88
Connect
B Portmap  229  OP88_E52  [FSEL= IN]
B Portmap  20  OP88_B20  [FSEL= OP]
EndConnect
Connect
B Portmap  229  OP88_E52  [FSEL= IN]
B Portmap  293  OP88_B20  [FSEL= OP]
EndConnect
EndNet
Net  OP27
Connect
C Portmap  250  OP27_E'53 [FSEL= IN]
C Portmap  254  OP27_A32  [FSEL= OP]
EndConnect
EndNet
Net  OP58
Connect
B Portmap  211  OP58_E54   [FSEL= IN]
B Portmap  148  OP58_M15   [FSEL= OP]
C Portmap  317  OP58_M15   [FSEL= IN]
C Portmap  231  OP58_A33   [FSEL= OP]
EndConnect
EndNet
Net  OP44
Connect
B Portmap  231  OP44_E55   [FSEL= IN]
B Portmap  144  OP44_M16   [FSEL= OP]
C Portmap  1  OP44_M16   [FSEL= IN]
C Portmap  201  OP44_A34   [FSEL= OP]
EndConnect
EndNet
Net  OP26
Connect
C Portmap  283  OP26_E'56 [FSEL= IN]
C Portmap  248  OP26_A35  [FSEL= OP]
EndConnect
EndNet
Net  OP29
Connect
C Portmap  263  OP29_E'57 [FSEL= IN]
C Portmap  249  OP29_A36  [FSEL= OP]
EndConnect
EndNet
Net  OP72
Connect
B Portmap  217  OP72_E58  [FSEL= IN]
B Portmap  298  OP72_B21  [FSEL= OP]
EndConnect
Connect
B Portmap  217  OP72_E58  [FSEL= IN]
B Portmap  287  OP72_B21  [FSEL= OP]
EndConnect
EndNet
Net  OP71
Connect
B Portmap  209  OP71_E59  [FSEL= IN]
B Portmap  300  OP71_B22  [FSEL= OP]
EndConnect
Connect
B Portmap  209  OP71_E59  [FSEL= IN]
B Portmap  295  OP71_B22  [FSEL= OP]
EndConnect
EndNet
Net  OP90
Connect
B Portmap  242  OP90_E60  [FSEL= IN]
B Portmap  22  OP90_B23  [FSEL= OP]
EndConnect
Connect
B Portmap  242  OP90_E60  [FSEL= IN]
B Portmap  26  OP90_B23  [FSEL= OP]
EndConnect
EndNet
Net  OP28
Connect
C Portmap  257  OP28_E'61 [FSEL= IN]
C Portmap  253  OP28_A37  [FSEL= OP]
EndConnect
EndNet
Net  OP8
Connect
C Portmap  280  OP8_E'62 [FSEL= IN]
C Portmap  206  OP8_A38  [FSEL= OP]
EndConnect
EndNet
Net  OP7
Connect
C Portmap  276  OP7_E'63 [FSEL= IN]
C Portmap  202  OP7_A39  [FSEL= OP]
EndConnect
EndNet
Net  OP30
Connect
C Portmap  271  OP30_E'64 [FSEL= IN]
C Portmap  245  OP30_A40  [FSEL= OP]
EndConnect
EndNet
Net  OP31
Connect
C Portmap  279  OP31_E'65 [FSEL= IN]
C Portmap  232  OP31_A41  [FSEL= OP]
EndConnect
EndNet
Net  OP57
Connect
B Portmap  203  OP57_E66   [FSEL= IN]
B Portmap  140  OP57_M17   [FSEL= OP]
C Portmap  2  OP57_M17   [FSEL= IN]
C Portmap  228  OP57_A42   [FSEL= OP]
EndConnect
EndNet
Net  OP43
Connect
B Portmap  223  OP43_E67   [FSEL= IN]
B Portmap  136  OP43_M18   [FSEL= OP]
C Portmap  5  OP43_M18   [FSEL= IN]
C Portmap  209  OP43_A43   [FSEL= OP]
```

```
EndConnect
EndNet
Net  OP89
Connect
B Portmap  237  OP89_E68  [FSEL= IN]
B Portmap  24  OP89_B24  [FSEL= OP]
EndConnect
Connect
B Portmap  237  OP89_E68  [FSEL= IN]
B Portmap  285  OP89_B24  [FSEL= OP]
EndConnect
EndNet
Net  OP91
Connect
B Portmap  250  OP91_E69  [FSEL= IN]
B Portmap  28  OP91_B25  [FSEL= OP]
EndConnect
Connect
B Portmap  250  OP91_E69  [FSEL= IN]
B Portmap  21  OP91_B25  [FSEL= OP]
EndConnect
EndNet
Net  OP6
Connect
C Portmap  272  OP6_E'70 [FSEL= IN]
C Portmap  204  OP6_A44  [FSEL= OP]
EndConnect
EndNet
Net  OP5
Connect
C Portmap  270  OP5_E'71 [FSEL= IN]
C Portmap  200  OP5_A45  [FSEL= OP]
EndConnect
EndNet
Net  OP32
Connect
B Portmap  268  OP32_E72  [FSEL= IN]
B Portmap  138  OP32_M19  [FSEL= OP]
C Portmap  6  OP32_M19  [FSEL= IN]
C Portmap  237  OP32_A46  [FSEL= OP]
EndConnect
EndNet
Net  OP33
Connect
B Portmap  270  OP33_E73  [FSEL= IN]
B Portmap  132  OP33_M20  [FSEL= OP]
C Portmap  10  OP33_M20  [FSEL= IN]
C Portmap  229  OP33_A47  [FSEL= OP]
EndConnect
EndNet
Net  OP70
Connect
B Portmap  201  OP70_E74  [FSEL= IN]
B Portmap  294  OP70_B26  [FSEL= OP]
EndConnect
Connect
B Portmap  201  OP70_E74  [FSEL= IN]
B Portmap  303  OP70_B26  [FSEL= OP]
EndConnect
EndNet
Net  OP69
Connect
B Portmap  193  OP69_E75  [FSEL= IN]
B Portmap  296  OP69_B27  [FSEL= OP]
EndConnect
Connect
B Portmap  193  OP69_E75  [FSEL= IN]
B Portmap  311  OP69_B27  [FSEL= OP]
EndConnect
EndNet
Net  OP34
Connect
B Portmap  272  OP34_E76  [FSEL= IN]
B Portmap  134  OP34_M21  [FSEL= OP]
C Portmap  12  OP34_M21  [FSEL= IN]
C Portmap  221  OP34_A48  [FSEL= OP]
EndConnect
EndNet
Net  OP35
Connect
B Portmap  276  OP35_E77  [FSEL= IN]
B Portmap  128  OP35_M22  [FSEL= OP]
C Portmap  16  OP35_M22  [FSEL= IN]
C Portmap  213  OP35_A49  [FSEL= OP]
EndConnect
EndNet
Net  OP42
Connect
B Portmap  215  OP42_E78  [FSEL= IN]
B Portmap  130  OP42_M23  [FSEL= OP]
C Portmap  18  OP42_M23  [FSEL= IN]
C Portmap  217  OP42_A50  [FSEL= OP]
EndConnect
EndNet
Net  OP4
Connect
C Portmap  268  OP4_E'79 [FSEL= IN]
C Portmap  196  OP4_A51  [FSEL= OP]
EndConnect
EndNet
Net  OP92
Connect
B Portmap  257  OP92_E80  [FSEL= IN]
B Portmap  289  OP92_B28  [FSEL= OP]
EndConnect
Connect
B Portmap  257  OP92_E80  [FSEL= IN]
B Portmap  14  OP92_B28  [FSEL= OP]
EndConnect
EndNet
Net  OP94
Connect
B Portmap  271  OP94_E81  [FSEL= IN]
B Portmap  305  OP94_B29  [FSEL= OP]
EndConnect
Connect
B Portmap  271  OP94_E81  [FSEL= IN]
B Portmap  315  OP94_B29  [FSEL= OP]
EndConnect
EndNet
Net  OP41
Connect
B Portmap  207  OP41_E82  [FSEL= IN]
B Portmap  126  OP41_M24  [FSEL= OP]
```

WALK_ROU.DOC 30742 December 23, 1995 10:56:51 AM
NET ROUTING NETLIST CREATED BY CON2ROU.EXE
INPUT FILES: WALK.CON OUTPUT FILES: WALK.ROU

```
C Portmap  20  OP41_M24   [FSEL= IN]
C Portmap  225  OP41_A52   [FSEL= OP]
EndConnect
EndNet
Net  OP3
Connect
C Portmap  266  OP3_E'83 [FSEL= IN]
C Portmap  198  OP3_A53  [FSEL= OP]
EndConnect
EndNet
Net  OP93
Connect
B Portmap  263  OP93_E84  [FSEL= IN]
B Portmap  297  OP93_B30  [FSEL= OP]
EndConnect
Connect
B Portmap  263  OP93_E84  [FSEL= IN]
B Portmap  8  OP93_B30  [FSEL= OP]
EndConnect
EndNet
Net  OP36
Connect
B Portmap  280  OP36_E85   [FSEL= IN]
B Portmap  124  OP36_M25   [FSEL= OP]
C Portmap  22  OP36_M25   [FSEL= IN]
C Portmap  205  OP36_A54   [FSEL= OP]
EndConnect
EndNet
Net  OP56
Connect
B Portmap  195  OP56_E86   [FSEL= IN]
B Portmap  187  OP56_M26   [FSEL= OP]
C Portmap  289  OP56_M26   [FSEL= IN]
C Portmap  191  OP56_A55   [FSEL= OP]
EndConnect
EndNet
Net  OP40
Connect
B Portmap  199  OP40_E87   [FSEL= IN]
B Portmap  180  OP40_M27   [FSEL= OP]
C Portmap  297  OP40_M27   [FSEL= IN]
C Portmap  233  OP40_A56   [FSEL= OP]
EndConnect
EndNet
Net  OP38
Connect
B Portmap  282  OP38_E88   [FSEL= IN]
B Portmap  174  OP38_M28   [FSEL= OP]
C Portmap  305  OP38_M28   [FSEL= IN]
C Portmap  189  OP38_A57   [FSEL= OP]
EndConnect
EndNet
Net  OP2
Connect
C Portmap  262  OP2_E'89 [FSEL= IN]
C Portmap  194  OP2_A58  [FSEL= OP]
EndConnect
EndNet
Net  OP1
Connect
C Portmap  260  OP1_E'90 [FSEL= IN]
C Portmap  188  OP1_A59  [FSEL= OP]
EndConnect
EndNet
Net  OP37
Connect
B Portmap  278  OP37_E91   [FSEL= IN]
B Portmap  171  OP37_M29   [FSEL= OP]
C Portmap  308  OP37_M29   [FSEL= IN]
C Portmap  197  OP37_A60   [FSEL= OP]
EndConnect
EndNet
Net  OP95
Connect
B Portmap  279  OP95_E92  [FSEL= IN]
B Portmap  308  OP95_B31  [FSEL= OP]
EndConnect
Connect
B Portmap  279  OP95_E92  [FSEL= IN]
B Portmap  312  OP95_B31  [FSEL= OP]
EndConnect
EndNet
Net  OP39
Connect
B Portmap  192  OP39_E93   [FSEL= IN]
B Portmap  167  OP39_M30   [FSEL= OP]
C Portmap  312  OP39_M30   [FSEL= IN]
C Portmap  239  OP39_A61   [FSEL= OP]
EndConnect
EndNet
Net  OP55
Connect
B Portmap  191  OP55_E94   [FSEL= IN]
B Portmap  146  OP55_M31   [FSEL= OP]
C Portmap  8  OP55_M31   [FSEL= IN]
C Portmap  195  OP55_A62   [FSEL= OP]
EndConnect
EndNet
Net  OP0
Connect
C Portmap  256  OP0_E'95 [FSEL= IN]
C Portmap  190  OP0_A63  [FSEL= OP]
EndConnect
EndNet
eof
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
Net  WLK82
A  Portmap  283  WLK82_G0  [FSEL= IN]
A  Portmap  27  WLK82_F0  [FSEL= OP]
EndNet
Net  WLK85
A  Portmap  205  WLK85_G1  [FSEL= IN]
A  Portmap  105  WLK85_F1  [FSEL= OP]
EndNet
Net  WLK81
A  Portmap  275  WLK81_G2  [FSEL= IN]
A  Portmap  35  WLK81_F2  [FSEL= OP]
EndNet
Net  WLK80
A  Portmap  267  WLK80_G3  [FSEL= IN]
A  Portmap  43  WLK80_F3  [FSEL= OP]
EndNet
Net  WLK24
A  Portmap  251  WLK24_G4  [FSEL= IN]
A  Portmap  56  WLK24_F4  [FSEL= OP]
EndNet
Net  WLK79
A  Portmap  261  WLK79_G5  [FSEL= IN]
A  Portmap  51  WLK79_F5  [FSEL= OP]
EndNet
Net  WLK78
A  Portmap  255  WLK78_G6  [FSEL= IN]
A  Portmap  59  WLK78_F6  [FSEL= OP]
EndNet
Net  WLK23
A  Portmap  247  WLK23_G7  [FSEL= IN]
A  Portmap  60  WLK23_F7  [FSEL= OP]
EndNet
Net  WLK22
A  Portmap  243  WLK22_G8  [FSEL= IN]
A  Portmap  64  WLK22_F8  [FSEL= OP]
EndNet
Net  WLK68
A  Portmap  277  WLK68_G9  [FSEL= IN]
A  Portmap  29  WLK68_F9  [FSEL= OP]
EndNet
Net  WLK86
A  Portmap  213  WLK86_G10  [FSEL= IN]
A  Portmap  97  WLK86_F10  [FSEL= OP]
EndNet
Net  WLK84
A  Portmap  197  WLK84_G11  [FSEL= IN]
A  Portmap  113  WLK84_F11  [FSEL= OP]
EndNet
Net  WLK21
A  Portmap  236  WLK21_G12  [FSEL= IN]
A  Portmap  68  WLK21_F12  [FSEL= OP]
EndNet
Net  WLK20
A  Portmap  234  WLK20_G13  [FSEL= IN]
A  Portmap  70  WLK20_F13  [FSEL= OP]
EndNet
Net  WLK67
A  Portmap  273  WLK67_G14  [FSEL= IN]
A  Portmap  37  WLK67_F14  [FSEL= OP]
EndNet
Net  WLK19
A  Portmap  230  WLK19_G15  [FSEL= IN]
A  Portmap  77  WLK19_F15  [FSEL= OP]
EndNet

Net  WLK66
A  Portmap  265  WLK66_G16  [FSEL= IN]
A  Portmap  45  WLK66_F16  [FSEL= OP]
EndNet
Net  WLK54
A  Portmap  281  WLK54_G17  [FSEL= IN]
A  Portmap  33  WLK54_F17  [FSEL= OP]
EndNet
Net  WLK83
A  Portmap  189  WLK83_G18  [FSEL= IN]
A  Portmap  121  WLK83_F18  [FSEL= OP]
EndNet
Net  WLK77
A  Portmap  244  WLK77_G19  [FSEL= IN]
A  Portmap  67  WLK77_F19  [FSEL= OP]
EndNet
Net  WLK18
A  Portmap  226  WLK18_G20  [FSEL= IN]
A  Portmap  81  WLK18_F20  [FSEL= OP]
EndNet
Net  WLK17
A  Portmap  222  WLK17_G21  [FSEL= IN]
A  Portmap  85  WLK17_F21  [FSEL= OP]
EndNet
Net  WLK53
A  Portmap  274  WLK53_G22  [FSEL= IN]
A  Portmap  41  WLK53_F22  [FSEL= OP]
EndNet
Net  WLK65
A  Portmap  259  WLK65_G23  [FSEL= IN]
A  Portmap  53  WLK65_F23  [FSEL= OP]
EndNet
Net  WLK64
A  Portmap  252  WLK64_G24  [FSEL= IN]
A  Portmap  61  WLK64_F24  [FSEL= OP]
EndNet
Net  WLK52
A  Portmap  269  WLK52_G25  [FSEL= IN]
A  Portmap  49  WLK52_F25  [FSEL= OP]
EndNet
Net  WLK16
A  Portmap  224  WLK16_G26  [FSEL= IN]
A  Portmap  80  WLK16_F26  [FSEL= OP]
EndNet
Net  WLK15
A  Portmap  218  WLK15_G27  [FSEL= IN]
A  Portmap  82  WLK15_F27  [FSEL= OP]
EndNet
Net  WLK51
A  Portmap  264  WLK51_G28  [FSEL= IN]
A  Portmap  57  WLK51_F28  [FSEL= OP]
EndNet
Net  WLK63
A  Portmap  238  WLK63_G29  [FSEL= IN]
A  Portmap  69  WLK63_F29  [FSEL= OP]
EndNet
Net  WLK76
A  Portmap  240  WLK76_G30  [FSEL= IN]
A  Portmap  73  WLK76_F30  [FSEL= OP]
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
EndNet
Net WLK75
A Portmap 239 WLK75_G31 [FSEL= IN]

A Portmap 74 WLK75_F31 [FSEL= OP]
EndNet
Net WLK14
A Portmap 220 WLK14_G32 [FSEL= IN]
A Portmap 88 WLK14_F32 [FSEL= OP]
EndNet
Net WLK50
A Portmap 258 WLK50_G33 [FSEL= IN]
A Portmap 65 WLK50_F33 [FSEL= OP]
EndNet
Net WLK13
A Portmap 214 WLK13_G34 [FSEL= IN]
A Portmap 90 WLK13_F34 [FSEL= OP]
EndNet
Net WLK12
A Portmap 216 WLK12_G35 [FSEL= IN]
A Portmap 94 WLK12_F35 [FSEL= OP]
EndNet
Net WLK74
A Portmap 233 WLK74_G36 [FSEL= IN]
A Portmap 78 WLK74_F36 [FSEL= OP]
EndNet
Net WLK73
A Portmap 225 WLK73_G37 [FSEL= IN]
A Portmap 89 WLK73_F37 [FSEL= OP]
EndNet
Net WLK49
A Portmap 253 WLK49_G38 [FSEL= IN]
A Portmap 62 WLK49_F38 [FSEL= OP]
EndNet
Net WLK11
A Portmap 212 WLK11_G39 [FSEL= IN]
A Portmap 96 WLK11_F39 [FSEL= OP]
EndNet
Net WLK10
A Portmap 210 WLK10_G40 [FSEL= IN]
A Portmap 100 WLK10_F40 [FSEL= OP]
EndNet
Net WLK48
A Portmap 249 WLK48_G41 [FSEL= IN]
A Portmap 66 WLK48_F41 [FSEL= OP]
EndNet
Net WLK62
A Portmap 241 WLK62_G42 [FSEL= IN]
A Portmap 75 WLK62_F42 [FSEL= OP]
EndNet
Net WLK46
A Portmap 232 WLK46_G43 [FSEL= IN]
A Portmap 83 WLK46_F43 [FSEL= OP]
EndNet
Net WLK47
A Portmap 245 WLK47_G44 [FSEL= IN]
A Portmap 79 WLK47_F44 [FSEL= OP]
EndNet
Net WLK61
A Portmap 235 WLK61_G45 [FSEL= IN]
A Portmap 72 WLK61_F45 [FSEL= OP]
EndNet
Net WLK9
A Portmap 208 WLK9_G46 [FSEL= IN]
A Portmap 102 WLK9_F46 [FSEL= OP]
EndNet
Net WLK60

A Portmap 227 WLK60_G47 [FSEL= IN]
A Portmap 86 WLK60_F47 [FSEL= OP]
EndNet
Net WLK87
A Portmap 221 WLK87_G48 [FSEL= IN]
A Portmap 91 WLK87_F48 [FSEL= OP]
EndNet
Net WLK25
A Portmap 246 WLK25_G49 [FSEL= IN]
A Portmap 58 WLK25_F49 [FSEL= OP]
EndNet
Net WLK59
A Portmap 219 WLK59_G50 [FSEL= IN]
A Portmap 93 WLK59_F50 [FSEL= OP]
EndNet
Net WLK45
A Portmap 228 WLK45_G51 [FSEL= IN]
A Portmap 87 WLK45_F51 [FSEL= OP]
EndNet
Net WLK88
A Portmap 229 WLK88_G52 [FSEL= IN]
A Portmap 84 WLK88_F52 [FSEL= OP]
EndNet
Net WLK27
A Portmap 254 WLK27_G53 [FSEL= IN]
A Portmap 54 WLK27_F53 [FSEL= OP]
EndNet
Net WLK58
A Portmap 211 WLK58_G54 [FSEL= IN]
A Portmap 99 WLK58_F54 [FSEL= OP]
EndNet
Net WLK44
A Portmap 231 WLK44_G55 [FSEL= IN]
A Portmap 92 WLK44_F55 [FSEL= OP]
EndNet
Net WLK26
A Portmap 248 WLK26_G56 [FSEL= IN]
A Portmap 52 WLK26_F56 [FSEL= OP]
EndNet
Net WLK29
A Portmap 260 WLK29_G57 [FSEL= IN]
A Portmap 50 WLK29_F57 [FSEL= OP]
EndNet
Net WLK72
A Portmap 217 WLK72_G58 [FSEL= IN]
A Portmap 95 WLK72_F58 [FSEL= OP]
EndNet
Net WLK71
A Portmap 209 WLK71_G59 [FSEL= IN]
A Portmap 101 WLK71_F59 [FSEL= OP]
EndNet
Net WLK90
A Portmap 242 WLK90_G60 [FSEL= IN]
A Portmap 71 WLK90_F60 [FSEL= OP]
EndNet
Net WLK28
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
A  Portmap  256  WLK28_G61  [FSEL= IN]
A  Portmap  48  WLK28_F61  [FSEL= OP]
EndNet
Net  WLK8
A  Portmap  206  WLK8_G62  [FSEL= IN]
A  Portmap  104  WLK8_F62  [FSEL= OP]

EndNet
Net  WLK7
A  Portmap  202  WLK7_G63  [FSEL= IN]
A  Portmap  106  WLK7_F63  [FSEL= OP]
EndNet
Net  WLK30
A  Portmap  262  WLK30_G64  [FSEL= IN]
A  Portmap  46  WLK30_F64  [FSEL= OP]
EndNet
Net  WLK31
A  Portmap  266  WLK31_G65  [FSEL= IN]
A  Portmap  42  WLK31_F65  [FSEL= OP]
EndNet
Net  WLK57
A  Portmap  203  WLK57_G66  [FSEL= IN]
A  Portmap  107  WLK57_F66  [FSEL= OP]
EndNet
Net  WLK43
A  Portmap  223  WLK43_G67  [FSEL= IN]
A  Portmap  98  WLK43_F67  [FSEL= OP]
EndNet
Net  WLK89
A  Portmap  237  WLK89_G68  [FSEL= IN]
A  Portmap  76  WLK89_F68  [FSEL= OP]
EndNet
Net  WLK91
A  Portmap  250  WLK91_G69  [FSEL= IN]
A  Portmap  63  WLK91_F69  [FSEL= OP]
EndNet
Net  WLK6
A  Portmap  204  WLK6_G70  [FSEL= IN]
A  Portmap  110  WLK6_F70  [FSEL= OP]
EndNet
Net  WLK5
A  Portmap  200  WLK5_G71  [FSEL= IN]
A  Portmap  112  WLK5_F71  [FSEL= OP]
EndNet
Net  WLK32
A  Portmap  268  WLK32_G72  [FSEL= IN]
A  Portmap  44  WLK32_F72  [FSEL= OP]
EndNet
Net  WLK33
A  Portmap  270  WLK33_G73  [FSEL= IN]
A  Portmap  40  WLK33_F73  [FSEL= OP]
EndNet
Net  WLK70
A  Portmap  201  WLK70_G74  [FSEL= IN]
A  Portmap  109  WLK70_F74  [FSEL= OP]
EndNet
Net  WLK69
A  Portmap  193  WLK69_G75  [FSEL= IN]
A  Portmap  117  WLK69_F75  [FSEL= OP]
EndNet
Net  WLK34
A  Portmap  272  WLK34_G76  [FSEL= IN]
A  Portmap  38  WLK34_F76  [FSEL= OP]
EndNet
Net  WLK35
A  Portmap  276  WLK35_G77  [FSEL= IN]
A  Portmap  36  WLK35_F77  [FSEL= OP]
EndNet
Net  WLK42
A  Portmap  215  WLK42_G78  [FSEL= IN]

A  Portmap  103  WLK42_F78  [FSEL= OP]
EndNet
Net  WLK4
A  Portmap  196  WLK4_G79  [FSEL= IN]
A  Portmap  114  WLK4_F79  [FSEL= OP]
EndNet
Net  WLK92
A  Portmap  257  WLK92_G80  [FSEL= IN]
A  Portmap  55  WLK92_F80  [FSEL= OP]
EndNet
Net  WLK94
A  Portmap  271  WLK94_G81  [FSEL= IN]
A  Portmap  39  WLK94_F81  [FSEL= OP]
EndNet
Net  WLK41
A  Portmap  207  WLK41_G82  [FSEL= IN]
A  Portmap  108  WLK41_F82  [FSEL= OP]
EndNet
Net  WLK3
A  Portmap  198  WLK3_G83  [FSEL= IN]
A  Portmap  116  WLK3_F83  [FSEL= OP]
EndNet
Net  WLK93
A  Portmap  263  WLK93_G84  [FSEL= IN]
A  Portmap  47  WLK93_F84  [FSEL= OP]
EndNet
Net  WLK36
A  Portmap  278  WLK36_G85  [FSEL= IN]
A  Portmap  34  WLK36_F85  [FSEL= OP]
EndNet
Net  WLK56
A  Portmap  195  WLK56_G86  [FSEL= IN]
A  Portmap  111  WLK56_F86  [FSEL= OP]
EndNet
Net  WLK40
A  Portmap  199  WLK40_G87  [FSEL= IN]
A  Portmap  115  WLK40_F87  [FSEL= OP]
EndNet
Net  WLK38
A  Portmap  282  WLK38_G88  [FSEL= IN]
A  Portmap  32  WLK38_F88  [FSEL= OP]
EndNet
Net  WLK2
A  Portmap  194  WLK2_G89  [FSEL= IN]
A  Portmap  118  WLK2_F89  [FSEL= OP]
EndNet
Net  WLK1
A  Portmap  188  WLK1_G90  [FSEL= IN]
A  Portmap  120  WLK1_F90  [FSEL= OP]
EndNet
Net  WLK37
A  Portmap  280  WLK37_G91  [FSEL= IN]
A  Portmap  30  WLK37_F91  [FSEL= OP]
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
EndNet
Net WLK95
A Portmap 279 WLK95_G92 [FSEL= IN]
A Portmap 31 WLK95_F92 [FSEL= OP]
EndNet
Net WLK39
A Portmap 192 WLK39_G93 [FSEL= IN]
A Portmap 123 WLK39_F93 [FSEL= OP]
EndNet

Net WLK55
A Portmap 191 WLK55_G94 [FSEL= IN]
A Portmap 119 WLK55_F94 [FSEL= OP]
EndNet
Net WLK0
A Portmap 190 WLK0_G95 [FSEL= IN]
A Portmap 122 WLK0_F95 [FSEL= OP]
EndNet
Net OP82
B Portmap 283 OP82_E0 [FSEL= IN]
B Portmap 5 OP82_B0 [FSEL= OP]
B Portmap 17 OP82_B0 [FSEL= OP]
EndNet
Net OP85
B Portmap 205 OP85_E1 [FSEL= IN]
B Portmap 12 OP85_B1 [FSEL= OP]
B Portmap 318 OP85_B1 [FSEL= OP]
EndNet
Net OP81
B Portmap 275 OP81_E2 [FSEL= IN]
B Portmap 2 OP81_B2 [FSEL= OP]
B Portmap 25 OP81_B2 [FSEL= OP]
EndNet
Net OP80
B Portmap 267 OP80_E3 [FSEL= IN]
B Portmap 1 OP80_B3 [FSEL= OP]
B Portmap 291 OP80_B3 [FSEL= OP]
EndNet
Net OP24
C Portmap 267 OP24_E'4 [FSEL= IN]
C Portmap 251 OP24_A0 [FSEL= OP]
EndNet
Net OP79
B Portmap 261 OP79_E5 [FSEL= IN]
B Portmap 317 OP79_B4 [FSEL= OP]
B Portmap 299 OP79_B4 [FSEL= OP]
EndNet
Net OP78
B Portmap 255 OP78_E6 [FSEL= IN]
B Portmap 314 OP78_B5 [FSEL= OP]
B Portmap 307 OP78_B5 [FSEL= OP]
EndNet
Net OP23
C Portmap 261 OP23_E'7 [FSEL= IN]
C Portmap 247 OP23_A1 [FSEL= OP]
EndNet
Net OP22
C Portmap 255 OP22_E'8 [FSEL= IN]
C Portmap 243 OP22_A2 [FSEL= OP]
EndNet
Net OP68
B Portmap 277 OP68_E9 [FSEL= IN]
B Portmap 292 OP68_B6 [FSEL= OP]
B Portmap 319 OP68_B6 [FSEL= OP]
EndNet
Net OP86
B Portmap 213 OP86_E10 [FSEL= IN]
B Portmap 16 OP86_B7 [FSEL= OP]
B Portmap 309 OP86_B7 [FSEL= OP]
EndNet
Net OP84
B Portmap 197 OP84_E11 [FSEL= IN]
B Portmap 10 OP84_B8 [FSEL= OP]

B Portmap 4 OP84_B8 [FSEL= OP]
EndNet
Net OP21
C Portmap 244 OP21_E'12 [FSEL= IN]
C Portmap 236 OP21_A3 [FSEL= OP]
EndNet
Net OP20
C Portmap 277 OP20_E'13 [FSEL= IN]
C Portmap 234 OP20_A4 [FSEL= OP]
EndNet
Net OP67
B Portmap 273 OP67_E14 [FSEL= IN]
B Portmap 290 OP67_B9 [FSEL= OP]
B Portmap 3 OP67_B9 [FSEL= OP]
EndNet
Net OP19
C Portmap 273 OP19_E'15 [FSEL= IN]
C Portmap 230 OP19_A5 [FSEL= OP]
EndNet
Net OP66
B Portmap 265 OP66_E16 [FSEL= IN]
B Portmap 288 OP66_B10 [FSEL= OP]
B Portmap 9 OP66_B10 [FSEL= OP]
EndNet
Net OP54
B Portmap 281 OP54_E17 [FSEL= IN]
B Portmap 186 OP54_M0 [FSEL= OP]
C Portmap 286 OP54_M0 [FSEL= IN]
C Portmap 203 OP54_A6 [FSEL= OP]
EndNet
Net OP83
B Portmap 189 OP83_E18 [FSEL= IN]
B Portmap 6 OP83_B11 [FSEL= OP]
B Portmap 11 OP83_B11 [FSEL= OP]
EndNet
Net OP77
B Portmap 244 OP77_E19 [FSEL= IN]
B Portmap 313 OP77_B12 [FSEL= OP]
B Portmap 316 OP77_B12 [FSEL= OP]
EndNet
Net OP18
C Portmap 265 OP18_E'20 [FSEL= IN]
C Portmap 226 OP18_A7 [FSEL= OP]
EndNet
Net OP17
C Portmap 259 OP17_E'21 [FSEL= IN]
C Portmap 222 OP17_A8 [FSEL= OP]
EndNet
Net OP53
B Portmap 274 OP53_E22 [FSEL= IN]
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
B  Portmap  184  OP53_M1  [FSEL= OP]
C  Portmap  284  OP53_M1  [FSEL= IN]
C  Portmap  211  OP53_A9  [FSEL= OP]
EndNet
Net  OP65
B  Portmap  259  OP65_E23  [FSEL= IN]
B  Portmap  284  OP65_B13  [FSEL= OP]
B  Portmap  15  OP65_B13  [FSEL= OP]
EndNet
Net  OP64
B  Portmap  252  OP64_E24  [FSEL= IN]
B  Portmap  286  OP64_B14  [FSEL= OP]

B  Portmap  23  OP64_B14  [FSEL= OP]
EndNet
Net  OP52
B  Portmap  269  OP52_E25  [FSEL= IN]
B  Portmap  182  OP52_M2  [FSEL= OP]
C  Portmap  288  OP52_M2  [FSEL= IN]
C  Portmap  219  OP52_A10  [FSEL= OP]
EndNet
Net  OP16
C  Portmap  252  OP16_E'26  [FSEL= IN]
C  Portmap  224  OP16_A11  [FSEL= OP]
EndNet
Net  OP15
C  Portmap  281  OP15_E'27  [FSEL= IN]
C  Portmap  218  OP15_A12  [FSEL= OP]
EndNet
Net  OP51
B  Portmap  264  OP51_E28  [FSEL= IN]
B  Portmap  178  OP51_M3  [FSEL= OP]
C  Portmap  290  OP51_M3  [FSEL= IN]
C  Portmap  227  OP51_A13  [FSEL= OP]
EndNet
Net  OP63
B  Portmap  238  OP63_E29  [FSEL= IN]
B  Portmap  176  OP63_M4  [FSEL= OP]
C  Portmap  292  OP63_M4  [FSEL= IN]
C  Portmap  192  OP63_A14  [FSEL= OP]
EndNet
Net  OP76
B  Portmap  240  OP76_E30  [FSEL= IN]
B  Portmap  310  OP76_B15  [FSEL= OP]
B  Portmap  0  OP76_B15  [FSEL= OP]
EndNet
Net  OP75
B  Portmap  239  OP75_E31  [FSEL= IN]
B  Portmap  306  OP75_B16  [FSEL= OP]
B  Portmap  7  OP75_B16  [FSEL= OP]
EndNet
Net  OP14
C  Portmap  274  OP14_E'32  [FSEL= IN]
C  Portmap  220  OP14_A15  [FSEL= OP]
EndNet
Net  OP50
B  Portmap  258  OP50_E33  [FSEL= IN]
B  Portmap  172  OP50_M5  [FSEL= OP]
C  Portmap  296  OP50_M5  [FSEL= IN]
C  Portmap  235  OP50_A16  [FSEL= OP]
EndNet
Net  OP13
C  Portmap  269  OP13_E'34  [FSEL= IN]
C  Portmap  214  OP13_A17  [FSEL= OP]
EndNet
Net  OP12
C  Portmap  264  OP12_E'35  [FSEL= IN]
C  Portmap  216  OP12_A18  [FSEL= OP]
EndNet
Net  OP74
B  Portmap  233  OP74_E36  [FSEL= IN]
B  Portmap  302  OP74_B17  [FSEL= OP]
B  Portmap  13  OP74_B17  [FSEL= OP]
EndNet
Net  OP73
B  Portmap  225  OP73_E37  [FSEL= IN]

B  Portmap  304  OP73_B18  [FSEL= OP]
B  Portmap  19  OP73_B18  [FSEL= OP]
EndNet
Net  OP49
B  Portmap  253  OP49_E38  [FSEL= IN]
B  Portmap  170  OP49_M6  [FSEL= OP]
C  Portmap  294  OP49_M6  [FSEL= IN]
C  Portmap  241  OP49_A19  [FSEL= OP]
EndNet
Net  OP11
C  Portmap  258  OP11_E'39  [FSEL= IN]
C  Portmap  212  OP11_A20  [FSEL= OP]
EndNet
Net  OP10
C  Portmap  282  OP10_E'40  [FSEL= IN]
C  Portmap  210  OP10_A21  [FSEL= OP]
EndNet
Net  OP48
B  Portmap  249  OP48_E41  [FSEL= IN]
B  Portmap  164  OP48_M7  [FSEL= OP]
C  Portmap  300  OP48_M7  [FSEL= IN]
C  Portmap  238  OP48_A22  [FSEL= OP]
EndNet
Net  OP62
B  Portmap  241  OP62_E42  [FSEL= IN]
B  Portmap  162  OP62_M8  [FSEL= OP]
C  Portmap  298  OP62_M8  [FSEL= IN]
C  Portmap  199  OP62_A23  [FSEL= OP]
EndNet
Net  OP46
B  Portmap  232  OP46_E43  [FSEL= IN]
B  Portmap  169  OP46_M9  [FSEL= OP]
C  Portmap  304  OP46_M9  [FSEL= IN]
C  Portmap  242  OP46_A24  [FSEL= OP]
EndNet
Net  OP47
B  Portmap  245  OP47_E44  [FSEL= IN]
B  Portmap  165  OP47_M10  [FSEL= OP]
C  Portmap  302  OP47_M10  [FSEL= IN]
C  Portmap  240  OP47_A25  [FSEL= OP]
EndNet
Net  OP61
B  Portmap  235  OP61_E45  [FSEL= IN]
B  Portmap  156  OP61_M11  [FSEL= OP]
C  Portmap  306  OP61_M11  [FSEL= IN]
C  Portmap  207  OP61_A26  [FSEL= OP]
EndNet
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
Net  OP9
C  Portmap  278  OP9_E'46  [FSEL= IN]
C  Portmap  208  OP9_A27  [FSEL= OP]
EndNet
Net  OP60
B  Portmap  227  OP60_E47  [FSEL= IN]
B  Portmap  161  OP60_M12  [FSEL= OP]
C  Portmap  310  OP60_M12  [FSEL= IN]
C  Portmap  215  OP60_A28  [FSEL= OP]
EndNet
Net  OP87
B  Portmap  221  OP87_E48  [FSEL= IN]
B  Portmap  18  OP87_B19  [FSEL= OP]
B  Portmap  301  OP87_B19  [FSEL= OP]
EndNet
Net  OP25

C  Portmap  275  OP25_E'49  [FSEL= IN]
C  Portmap  246  OP25_A29  [FSEL= OP]
EndNet
Net  OP59
B  Portmap  219  OP59_E50  [FSEL= IN]
B  Portmap  163  OP59_M13  [FSEL= OP]
C  Portmap  313  OP59_M13  [FSEL= IN]
C  Portmap  223  OP59_A30  [FSEL= OP]
EndNet
Net  OP45
B  Portmap  228  OP45_E51  [FSEL= IN]
B  Portmap  150  OP45_M14  [FSEL= OP]
C  Portmap  314  OP45_M14  [FSEL= IN]
C  Portmap  193  OP45_A31  [FSEL= OP]
EndNet
Net  OP88
B  Portmap  229  OP88_E52  [FSEL= IN]
B  Portmap  20  OP88_B20  [FSEL= OP]
B  Portmap  293  OP88_B20  [FSEL= OP]
EndNet
Net  OP27
C  Portmap  250  OP27_E'53  [FSEL= IN]
C  Portmap  254  OP27_A32  [FSEL= OP]
EndNet
Net  OP58
B  Portmap  211  OP58_E54  [FSEL= IN]
B  Portmap  148  OP58_M15  [FSEL= OP]
C  Portmap  317  OP58_M15  [FSEL= IN]
C  Portmap  231  OP58_A33  [FSEL= OP]
EndNet
Net  OP44
B  Portmap  231  OP44_E55  [FSEL= IN]
B  Portmap  144  OP44_M16  [FSEL= OP]
C  Portmap  1  OP44_M16  [FSEL= IN]
C  Portmap  201  OP44_A34  [FSEL= OP]
EndNet
Net  OP26
C  Portmap  283  OP26_E'56  [FSEL= IN]
C  Portmap  248  OP26_A35  [FSEL= OP]
EndNet
Net  OP29
C  Portmap  263  OP29_E'57  [FSEL= IN]
C  Portmap  249  OP29_A36  [FSEL= OP]
EndNet
Net  OP72
B  Portmap  217  OP72_E58  [FSEL= IN]
B  Portmap  298  OP72_B21  [FSEL= OP]
B  Portmap  287  OP72_B21  [FSEL= OP]
EndNet
Net  OP71
B  Portmap  209  OP71_E59  [FSEL= IN]
B  Portmap  300  OP71_B22  [FSEL= OP]
B  Portmap  295  OP71_B22  [FSEL= OP]
EndNet
Net  OP90
B  Portmap  242  OP90_E60  [FSEL= IN]
B  Portmap  22  OP90_B23  [FSEL= OP]
B  Portmap  26  OP90_B23  [FSEL= OP]
EndNet
Net  OP28
C  Portmap  257  OP28_E'61  [FSEL= IN]
C  Portmap  253  OP28_A37  [FSEL= OP]
EndNet

Net  OP8
C  Portmap  280  OP8_E'62  [FSEL= IN]
C  Portmap  206  OP8_A38  [FSEL= OP]
EndNet
Net  OP7
C  Portmap  276  OP7_E'63  [FSEL= IN]
C  Portmap  202  OP7_A39  [FSEL= OP]
EndNet
Net  OP30
C  Portmap  271  OP30_E'64  [FSEL= IN]
C  Portmap  245  OP30_A40  [FSEL= OP]
EndNet
Net  OP31
C  Portmap  279  OP31_E'65  [FSEL= IN]
C  Portmap  232  OP31_A41  [FSEL= OP]
EndNet
Net  OP57
B  Portmap  203  OP57_E66  [FSEL= IN]
B  Portmap  140  OP57_M17  [FSEL= OP]
C  Portmap  2  OP57_M17  [FSEL= IN]
C  Portmap  228  OP57_A42  [FSEL= OP]
EndNet
Net  OP43
B  Portmap  223  OP43_E67  [FSEL= IN]
B  Portmap  136  OP43_M18  [FSEL= OP]
C  Portmap  5  OP43_M18  [FSEL= IN]
C  Portmap  209  OP43_A43  [FSEL= OP]
EndNet
Net  OP89
B  Portmap  237  OP89_E68  [FSEL= IN]
B  Portmap  24  OP89_B24  [FSEL= OP]
B  Portmap  285  OP89_B24  [FSEL= OP]
EndNet
Net  OP91
B  Portmap  250  OP91_E69  [FSEL= IN]
B  Portmap  28  OP91_B25  [FSEL= OP]
B  Portmap  21  OP91_B25  [FSEL= OP]
EndNet
Net  OP6
C  Portmap  272  OP6_E'70  [FSEL= IN]
C  Portmap  204  OP6_A44  [FSEL= OP]
EndNet
Net  OP5
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
C  Portmap  270  OP5_E'71  [FSEL= IN]
C  Portmap  200  OP5_A45  [FSEL= OP]
EndNet
Net  OP32
B  Portmap  268  OP32_E72  [FSEL= IN]
B  Portmap  138  OP32_M19  [FSEL= OP]
C  Portmap  6  OP32_M19  [FSEL= IN]
C  Portmap  237  OP32_A46  [FSEL= OP]
EndNet
Net  OP33
B  Portmap  270  OP33_E73  [FSEL= IN]
B  Portmap  132  OP33_M20  [FSEL= OP]
C  Portmap  10  OP33_M20  [FSEL= IN]
C  Portmap  229  OP33_A47  [FSEL= OP]
EndNet
Net  OP70
B  Portmap  201  OP70_E74  [FSEL= IN]
B  Portmap  294  OP70_B26  [FSEL= OP]
B  Portmap  303  OP70_B26  [FSEL= OP]
EndNet

Net  OP69
B  Portmap  193  OP69_E75  [FSEL= IN]
B  Portmap  296  OP69_B27  [FSEL= OP]
B  Portmap  311  OP69_B27  [FSEL= OP]
EndNet
Net  OP34
B  Portmap  272  OP34_E76  [FSEL= IN]
B  Portmap  134  OP34_M21  [FSEL= OP]
C  Portmap  12  OP34_M21  [FSEL= IN]
C  Portmap  221  OP34_A48  [FSEL= OP]
EndNet
Net  OP35
B  Portmap  276  OP35_E77  [FSEL= IN]
B  Portmap  128  OP35_M22  [FSEL= OP]
C  Portmap  16  OP35_M22  [FSEL= IN]
C  Portmap  213  OP35_A49  [FSEL= OP]
EndNet
Net  OP42
B  Portmap  215  OP42_E78  [FSEL= IN]
B  Portmap  130  OP42_M23  [FSEL= OP]
C  Portmap  18  OP42_M23  [FSEL= IN]
C  Portmap  217  OP42_A50  [FSEL= OP]
EndNet
Net  OP4
C  Portmap  268  OP4_E'79  [FSEL= IN]
C  Portmap  196  OP4_A51  [FSEL= OP]
EndNet
Net  OP92
B  Portmap  257  OP92_E80  [FSEL= IN]
B  Portmap  289  OP92_B28  [FSEL= OP]
B  Portmap  14  OP92_B28  [FSEL= OP]
EndNet
Net  OP94
B  Portmap  271  OP94_E81  [FSEL= IN]
B  Portmap  305  OP94_B29  [FSEL= OP]
B  Portmap  315  OP94_B29  [FSEL= OP]
EndNet
Net  OP41
B  Portmap  207  OP41_E82  [FSEL= IN]
B  Portmap  126  OP41_M24  [FSEL= OP]
C  Portmap  20  OP41_M24  [FSEL= IN]
C  Portmap  225  OP41_A52  [FSEL= OP]
EndNet
Net  OP3
C  Portmap  266  OP3_E'83  [FSEL= IN]
C  Portmap  198  OP3_A53  [FSEL= OP]
EndNet
Net  OP93
B  Portmap  263  OP93_E84  [FSEL= IN]
B  Portmap  297  OP93_B30  [FSEL= OP]
B  Portmap  8  OP93_B30  [FSEL= OP]
EndNet
Net  OP36
B  Portmap  280  OP36_E85  [FSEL= IN]
B  Portmap  124  OP36_M25  [FSEL= OP]
C  Portmap  22  OP36_M25  [FSEL= IN]
C  Portmap  205  OP36_A54  [FSEL= OP]
EndNet
Net  OP56
B  Portmap  195  OP56_E86  [FSEL= IN]
B  Portmap  187  OP56_M26  [FSEL= OP]
C  Portmap  289  OP56_M26  [FSEL= IN]
C  Portmap  191  OP56_A55  [FSEL= OP]

EndNet
Net  OP40
B  Portmap  199  OP40_E87  [FSEL= IN]
B  Portmap  180  OP40_M27  [FSEL= OP]
C  Portmap  297  OP40_M27  [FSEL= IN]
C  Portmap  233  OP40_A56  [FSEL= OP]
EndNet
Net  OP38
B  Portmap  282  OP38_E88  [FSEL= IN]
B  Portmap  174  OP38_M28  [FSEL= OP]
C  Portmap  305  OP38_M28  [FSEL= IN]
C  Portmap  189  OP38_A57  [FSEL= OP]
EndNet
Net  OP2
C  Portmap  262  OP2_E'89  [FSEL= IN]
C  Portmap  194  OP2_A58  [FSEL= OP]
EndNet
Net  OP1
C  Portmap  260  OP1_E'90  [FSEL= IN]
C  Portmap  188  OP1_A59  [FSEL= OP]
EndNet
Net  OP37
B  Portmap  278  OP37_E91  [FSEL= IN]
B  Portmap  171  OP37_M29  [FSEL= OP]
C  Portmap  308  OP37_M29  [FSEL= IN]
C  Portmap  197  OP37_A60  [FSEL= OP]
EndNet
Net  OP95
B  Portmap  279  OP95_E92  [FSEL= IN]
B  Portmap  308  OP95_B31  [FSEL= OP]
B  Portmap  312  OP95_B31  [FSEL= OP]
EndNet
Net  OP39
B  Portmap  192  OP39_E93  [FSEL= IN]
B  Portmap  167  OP39_M30  [FSEL= OP]
C  Portmap  312  OP39_M30  [FSEL= IN]
C  Portmap  239  OP39_A61  [FSEL= OP]
EndNet
Net  OP55
```

WALK_PCA.DOC 24733 December 23, 1995 10:55:01 AM
PORTMAP CONFIGURATION ATTRIBUTE NETLIST CREATED BY ROU2IQB.EXE
INPUT FILES: WALK.ROU OUTPUT FILES: WALK.PCA

```
B  Portmap  191  OP55_E94  [FSEL= IN]
B  Portmap  146  OP55_M31  [FSEL= OP]
C  Portmap  8  OP55_M31  [FSEL= IN]
C  Portmap  195  OP55_A62  [FSEL= OP]
EndNet
Net  OP0
C  Portmap  256  OP0_E'95  [FSEL= IN]
C  Portmap  190  OP0_A63  [FSEL= OP]
EndNet

eof
```

WALK_NLT.DOC 18530 December 23, 1995 10:54:09 AM
I-CUBE FORMAT NETLIST CREATED BY IQROUTEB.EXE
INPUT FILES: WALK.PCA OUTPUT FILES: WALK.NLT

```
Device  IQ320   U1   U2   U3;

JtagChain   JC1  U1  U2   U3;

DevInst   IQ320   U1;


Portmap  283  [FSEL= IN];
Portmap  27  [FSEL= OP];
Portmap  205  [FSEL= IN];
Portmap  105  [FSEL= OP];
Portmap  275  [FSEL= IN];
Portmap  35  [FSEL= OP];
Portmap  267  [FSEL= IN];
Portmap  43  [FSEL= OP];
Portmap  251  [FSEL= IN];
Portmap  56  [FSEL= OP];
Portmap  261  [FSEL= IN];
Portmap  51  [FSEL= OP];
Portmap  255  [FSEL= IN];
Portmap  59  [FSEL= OP];
Portmap  247  [FSEL= IN];
Portmap  60  [FSEL= OP];
Portmap  243  [FSEL= IN];
Portmap  64  [FSEL= OP];
Portmap  277  [FSEL= IN];
Portmap  29  [FSEL= OP];
Portmap  213  [FSEL= IN];
Portmap  97  [FSEL= OP];
Portmap  197  [FSEL= IN];
Portmap  113  [FSEL= OP];
Portmap  236  [FSEL= IN];
Portmap  68  [FSEL= OP];
Portmap  234  [FSEL= IN];
Portmap  70  [FSEL= OP];
Portmap  273  [FSEL= IN];
Portmap  37  [FSEL= OP];
Portmap  230  [FSEL= IN];
Portmap  77  [FSEL= OP];
Portmap  265  [FSEL= IN];
Portmap  45  [FSEL= OP];
Portmap  281  [FSEL= IN];
Portmap  33  [FSEL= OP];
Portmap  189  [FSEL= IN];
Portmap  121  [FSEL= OP];
Portmap  244  [FSEL= IN];
Portmap  67  [FSEL= OP];
Portmap  226  [FSEL= IN];
Portmap  81  [FSEL= OP];
Portmap  222  [FSEL= IN];
Portmap  85  [FSEL= OP];
Portmap  274  [FSEL= IN];
Portmap  41  [FSEL= OP];
Portmap  259  [FSEL= IN];
Portmap  53  [FSEL= OP];
Portmap  252  [FSEL= IN];
Portmap  61  [FSEL= OP];
Portmap  269  [FSEL= IN];
Portmap  49  [FSEL= OP];
Portmap  224  [FSEL= IN];
Portmap  80  [FSEL= OP];
Portmap  218  [FSEL= IN];
Portmap  82  [FSEL= OP];

Portmap  264  [FSEL= IN];
Portmap  57  [FSEL= OP];
Portmap  238  [FSEL= IN];
Portmap  69  [FSEL= OP];
Portmap  240  [FSEL= IN];
Portmap  73  [FSEL= OP];
Portmap  239  [FSEL= IN];
Portmap  74  [FSEL= OP];
Portmap  220  [FSEL= IN];
Portmap  88  [FSEL= OP];
Portmap  258  [FSEL= IN];
Portmap  65  [FSEL= OP];
Portmap  214  [FSEL= IN];
Portmap  90  [FSEL= OP];
Portmap  216  [FSEL= IN];
Portmap  94  [FSEL= OP];
Portmap  233  [FSEL= IN];
Portmap  78  [FSEL= OP];
Portmap  225  [FSEL= IN];
Portmap  89  [FSEL= OP];
Portmap  253  [FSEL= IN];
Portmap  62  [FSEL= OP];
Portmap  212  [FSEL= IN];
Portmap  96  [FSEL= OP];
Portmap  210  [FSEL= IN];
Portmap  100  [FSEL= OP];
Portmap  249  [FSEL= IN];
Portmap  66  [FSEL= OP];
Portmap  241  [FSEL= IN];
Portmap  75  [FSEL= OP];
Portmap  232  [FSEL= IN];
Portmap  83  [FSEL= OP];
Portmap  245  [FSEL= IN];
Portmap  79  [FSEL= OP];
Portmap  235  [FSEL= IN];
Portmap  72  [FSEL= OP];
Portmap  208  [FSEL= IN];
Portmap  102  [FSEL= OP];
Portmap  227  [FSEL= IN];
Portmap  86  [FSEL= OP];
Portmap  221  [FSEL= IN];
Portmap  91  [FSEL= OP];
Portmap  246  [FSEL= IN];
Portmap  58  [FSEL= OP];
Portmap  219  [FSEL= IN];
Portmap  93  [FSEL= OP];
Portmap  228  [FSEL= IN];
Portmap  87  [FSEL= OP];
Portmap  229  [FSEL= IN];
Portmap  84  [FSEL= OP];
Portmap  254  [FSEL= IN];
Portmap  54  [FSEL= OP];
Portmap  211  [FSEL= IN];
Portmap  99  [FSEL= OP];
Portmap  231  [FSEL= IN];
Portmap  92  [FSEL= OP];
Portmap  248  [FSEL= IN];
Portmap  52  [FSEL= OP];
Portmap  260  [FSEL= IN];
```

WALK_NLT.DOC 18530 December 23, 1995 10:54:09 AM
I-CUBE FORMAT NETLIST CREATED BY IQROUTEB.EXE
INPUT FILES: WALK.PCA OUTPUT FILES: WALK.NLT

```
Portmap  50  [FSEL= OP];
Portmap  217  [FSEL= IN];
Portmap  95  [FSEL= OP];
Portmap  209  [FSEL= IN];
Portmap  101  [FSEL= OP];
Portmap  242  [FSEL= IN];
Portmap  71  [FSEL= OP];
Portmap  256  [FSEL= IN];
Portmap  48  [FSEL= OP];
Portmap  206  [FSEL= IN];
Portmap  104  [FSEL= OP];
Portmap  202  [FSEL= IN];
Portmap  106  [FSEL= OP];
Portmap  262  [FSEL= IN];
Portmap  46  [FSEL= OP];
Portmap  266  [FSEL= IN];
Portmap  42  [FSEL= OP];
Portmap  203  [FSEL= IN];
Portmap  107  [FSEL= OP];
Portmap  223  [FSEL= IN];
Portmap  98  [FSEL= OP];
Portmap  237  [FSEL= IN];
Portmap  76  [FSEL= OP];
Portmap  250  [FSEL= IN];
Portmap  63  [FSEL= OP];
Portmap  204  [FSEL= IN];
Portmap  110  [FSEL= OP];
Portmap  200  [FSEL= IN];
Portmap  112  [FSEL= OP];
Portmap  268  [FSEL= IN];
Portmap  44  [FSEL= OP];
Portmap  270  [FSEL= IN];
Portmap  40  [FSEL= OP];
Portmap  201  [FSEL= IN];
Portmap  109  [FSEL= OP];
Portmap  193  [FSEL= IN];
Portmap  117  [FSEL= OP];
Portmap  272  [FSEL= IN];
Portmap  38  [FSEL= OP];
Portmap  276  [FSEL= IN];
Portmap  36  [FSEL= OP];
Portmap  215  [FSEL= IN];
Portmap  103  [FSEL= OP];
Portmap  196  [FSEL= IN];
Portmap  114  [FSEL= OP];
Portmap  257  [FSEL= IN];
Portmap  55  [FSEL= OP];
Portmap  271  [FSEL= IN];
Portmap  39  [FSEL= OP];
Portmap  207  [FSEL= IN];
Portmap  108  [FSEL= OP];
Portmap  198  [FSEL= IN];
Portmap  116  [FSEL= OP];
Portmap  263  [FSEL= IN];
Portmap  47  [FSEL= OP];
Portmap  278  [FSEL= IN];
Portmap  34  [FSEL= OP];
Portmap  195  [FSEL= IN];
Portmap  111  [FSEL= OP];
Portmap  199  [FSEL= IN];
Portmap  115  [FSEL= OP];
Portmap  282  [FSEL= IN];
Portmap  32  [FSEL= OP];
Portmap  194  [FSEL= IN];
Portmap  118  [FSEL= OP];
Portmap  188  [FSEL= IN];
Portmap  120  [FSEL= OP];
Portmap  280  [FSEL= IN];
Portmap  30  [FSEL= OP];
Portmap  279  [FSEL= IN];
Portmap  31  [FSEL= OP];
Portmap  192  [FSEL= IN];
Portmap  123  [FSEL= OP];
Portmap  191  [FSEL= IN];
Portmap  119  [FSEL= OP];
Portmap  190  [FSEL= IN];
Portmap  122  [FSEL= OP];

Connect  283  27;
Connect  205  105;
Connect  275  35;
Connect  267  43;
Connect  251  56;
Connect  261  51;
Connect  255  59;
Connect  247  60;
Connect  243  64;
Connect  277  29;
Connect  213  97;
Connect  197  113;
Connect  236  68;
Connect  234  70;
Connect  273  37;
Connect  230  77;
Connect  265  45;
Connect  281  33;
Connect  189  121;
Connect  244  67;
Connect  226  81;
Connect  222  85;
Connect  274  41;
Connect  259  53;
Connect  252  61;
Connect  269  49;
Connect  224  80;
Connect  218  82;
Connect  264  57;
Connect  238  69;
Connect  240  73;
Connect  239  74;
Connect  220  88;
Connect  258  65;
Connect  214  90;
Connect  216  94;
Connect  233  78;
Connect  225  89;
Connect  253  62;
Connect  212  96;
Connect  210  100;
Connect  249  66;
Connect  241  75;
Connect  232  83;
```

```
Connect  245  79;
Connect  235  72;
Connect  208  102;
Connect  227  86;
Connect  221  91;
Connect  246  58;
Connect  219  93;
Connect  228  87;
Connect  229  84;
Connect  254  54;
Connect  211  99;
Connect  231  92;
Connect  248  52;
Connect  260  50;
Connect  217  95;
Connect  209  101;
Connect  242  71;
Connect  256  48;
Connect  206  104;
Connect  202  106;
Connect  262  46;
Connect  266  42;
Connect  203  107;
Connect  223  98;
Connect  237  76;
Connect  250  63;
Connect  204  110;
Connect  200  112;
Connect  268  44;
Connect  270  40;
Connect  201  109;
Connect  193  117;
Connect  272  38;
Connect  276  36;
Connect  215  103;
Connect  196  114;
Connect  257  55;
Connect  271  39;
Connect  207  108;
Connect  198  116;
Connect  263  47;
Connect  278  34;
Connect  195  111;
Connect  199  115;
Connect  282  32;
Connect  194  118;
Connect  188  120;
Connect  280  30;
Connect  279  31;
Connect  192  123;
Connect  191  119;
Connect  190  122;

EndDevInst;

DevInst   IQ320   U2;

Portmap  283  [FSEL= IN];
Portmap  5  [FSEL= OP];
Portmap  17  [FSEL= OP];
Portmap  205  [FSEL= IN];
Portmap  12  [FSEL= OP];
Portmap  318  [FSEL= OP];
Portmap  275  [FSEL= IN];
Portmap  2  [FSEL= OP];
Portmap  25  [FSEL= OP];
Portmap  267  [FSEL= IN];
Portmap  1  [FSEL= OP];
Portmap  291  [FSEL= OP];
Portmap  261  [FSEL= IN];
Portmap  317  [FSEL= OP];
Portmap  299  [FSEL= OP];
Portmap  255  [FSEL= IN];
Portmap  314  [FSEL= OP];
Portmap  307  [FSEL= OP];
Portmap  277  [FSEL= IN];
Portmap  292  [FSEL= OP];
Portmap  319  [FSEL= OP];
Portmap  213  [FSEL= IN];
Portmap  16  [FSEL= OP];
Portmap  309  [FSEL= OP];
Portmap  197  [FSEL= IN];
Portmap  10  [FSEL= OP];
Portmap  4  [FSEL= OP];
Portmap  273  [FSEL= IN];
Portmap  290  [FSEL= OP];
Portmap  3  [FSEL= OP];
Portmap  265  [FSEL= IN];
Portmap  288  [FSEL= OP];
Portmap  9  [FSEL= OP];
Portmap  281  [FSEL= IN];
Portmap  186  [FSEL= OP];
Portmap  189  [FSEL= IN];
Portmap  6  [FSEL= OP];
Portmap  11  [FSEL= OP];
Portmap  244  [FSEL= IN];
Portmap  313  [FSEL= OP];
Portmap  316  [FSEL= OP];
Portmap  274  [FSEL= IN];
Portmap  184  [FSEL= OP];
Portmap  259  [FSEL= IN];
Portmap  284  [FSEL= OP];
Portmap  15  [FSEL= OP];
Portmap  252  [FSEL= IN];
Portmap  286  [FSEL= OP];
Portmap  23  [FSEL= OP];
Portmap  269  [FSEL= IN];
Portmap  182  [FSEL= OP];
Portmap  264  [FSEL= IN];
Portmap  178  [FSEL= OP];
Portmap  238  [FSEL= IN];
Portmap  176  [FSEL= OP];
Portmap  240  [FSEL= IN];
Portmap  310  [FSEL= OP];
Portmap  0  [FSEL= OP];
Portmap  239  [FSEL= IN];
Portmap  306  [FSEL= OP];
Portmap  7  [FSEL= OP];
```

```
Portmap  258  [FSEL= IN];
Portmap  172  [FSEL= OP];
Portmap  233  [FSEL= IN];
Portmap  302  [FSEL= OP];
Portmap  13  [FSEL= OP];
Portmap  225  [FSEL= IN];
Portmap  304  [FSEL= OP];
Portmap  19  [FSEL= OP];
Portmap  253  [FSEL= IN];
Portmap  170  [FSEL= OP];
Portmap  249  [FSEL= IN];
Portmap  164  [FSEL= OP];
Portmap  241  [FSEL= IN];
Portmap  162  [FSEL= OP];
Portmap  232  [FSEL= IN];
Portmap  169  [FSEL= OP];
Portmap  245  [FSEL= IN];
Portmap  165  [FSEL= OP];
Portmap  235  [FSEL= IN];
Portmap  156  [FSEL= OP];
Portmap  227  [FSEL= IN];
Portmap  161  [FSEL= OP];
Portmap  221  [FSEL= IN];
Portmap  18  [FSEL= OP];
Portmap  301  [FSEL= OP];
Portmap  219  [FSEL= IN];
Portmap  163  [FSEL= OP];
Portmap  228  [FSEL= IN];
Portmap  150  [FSEL= OP];
Portmap  229  [FSEL= IN];
Portmap  20  [FSEL= OP];
Portmap  293  [FSEL= OP];
Portmap  211  [FSEL= IN];
Portmap  148  [FSEL= OP];
Portmap  231  [FSEL= IN];
Portmap  144  [FSEL= OP];
Portmap  217  [FSEL= IN];
Portmap  298  [FSEL= OP];
Portmap  287  [FSEL= OP];
Portmap  209  [FSEL= IN];
Portmap  300  [FSEL= OP];
Portmap  295  [FSEL= OP];
Portmap  242  [FSEL= IN];
Portmap  22  [FSEL= OP];
Portmap  26  [FSEL= OP];
Portmap  203  [FSEL= IN];
Portmap  140  [FSEL= OP];
Portmap  223  [FSEL= IN];
Portmap  136  [FSEL= OP];
Portmap  237  [FSEL= IN];
Portmap  24  [FSEL= OP];
Portmap  285  [FSEL= OP];
Portmap  250  [FSEL= IN];
Portmap  28  [FSEL= OP];
Portmap  21  [FSEL= OP];
Portmap  268  [FSEL= IN];
Portmap  138  [FSEL= OP];
Portmap  270  [FSEL= IN];
Portmap  132  [FSEL= OP];
Portmap  201  [FSEL= IN];
Portmap  294  [FSEL= OP];
Portmap  303  [FSEL= OP];
Portmap  193  [FSEL= IN];
Portmap  296  [FSEL= OP];
Portmap  311  [FSEL= OP];
Portmap  272  [FSEL= IN];
Portmap  134  [FSEL= OP];
Portmap  276  [FSEL= IN];
Portmap  128  [FSEL= OP];
Portmap  215  [FSEL= IN];
Portmap  130  [FSEL= OP];
Portmap  257  [FSEL= IN];
Portmap  289  [FSEL= OP];
Portmap  14  [FSEL= OP];
Portmap  271  [FSEL= IN];
Portmap  305  [FSEL= OP];
Portmap  315  [FSEL= OP];
Portmap  207  [FSEL= IN];
Portmap  126  [FSEL= OP];
Portmap  263  [FSEL= IN];
Portmap  297  [FSEL= OP];
Portmap  8  [FSEL= OP];
Portmap  280  [FSEL= IN];
Portmap  124  [FSEL= OP];
Portmap  195  [FSEL= IN];
Portmap  187  [FSEL= OP];
Portmap  199  [FSEL= IN];
Portmap  180  [FSEL= OP];
Portmap  282  [FSEL= IN];
Portmap  174  [FSEL= OP];
Portmap  278  [FSEL= IN];
Portmap  171  [FSEL= OP];
Portmap  279  [FSEL= IN];
Portmap  308  [FSEL= OP];
Portmap  312  [FSEL= OP];
Portmap  192  [FSEL= IN];
Portmap  167  [FSEL= OP];
Portmap  191  [FSEL= IN];
Portmap  146  [FSEL= OP];

Connect  283  5  17;
Connect  205  12  318;
Connect  275  2  25;
Connect  267  1  291;
Connect  261  317  299;
Connect  255  314  307;
Connect  277  292  319;
Connect  213  16  309;
Connect  197  10  4;
Connect  273  290  3;
Connect  265  288  9;
Connect  281  186;
Connect  189  6  11;
Connect  244  313  316;
Connect  274  184;
Connect  259  284  15;
Connect  252  286  23;
Connect  269  182;
Connect  264  178;
Connect  238  176;
Connect  240  310  0;
Connect  239  306  7;
Connect  258  172;
```

```
Connect  233  302  13;
Connect  225  304  19;
Connect  253  170;
Connect  249  164;
Connect  241  162;
Connect  232  169;
Connect  245  165;
Connect  235  156;
Connect  227  161;
Connect  221  18  301;
Connect  219  163;
Connect  228  150;
Connect  229  20  293;
Connect  211  148;
Connect  231  144;
Connect  217  298  287;
Connect  209  300  295;
Connect  242  22  26;
Connect  203  140;
Connect  223  136;
Connect  237  24  285;
Connect  250  28  21;
Connect  268  138;
Connect  270  132;
Connect  201  294  303;
Connect  193  296  311;
Connect  272  134;
Connect  276  128;
Connect  215  130;
Connect  257  289  14;
Connect  271  305  315;
Connect  207  126;
Connect  263  297  8;
Connect  280  124;
Connect  195  187;
Connect  199  180;
Connect  282  174;
Connect  278  171;
Connect  279  308  312;
Connect  192  167;
Connect  191  146;

EndDevInst;

DevInst   IQ320   U3;

Portmap  267  [FSEL= IN];
Portmap  251  [FSEL= OP];
Portmap  261  [FSEL= IN];
Portmap  247  [FSEL= OP];
Portmap  255  [FSEL= IN];
Portmap  243  [FSEL= OP];
Portmap  244  [FSEL= IN];
Portmap  236  [FSEL= OP];
Portmap  277  [FSEL= IN];
Portmap  234  [FSEL= OP];
Portmap  273  [FSEL= IN];
Portmap  230  [FSEL= OP];
Portmap  286  [FSEL= IN];
Portmap  203  [FSEL= OP];
Portmap  265  [FSEL= IN];
Portmap  226  [FSEL= OP];
Portmap  259  [FSEL= IN];
Portmap  222  [FSEL= OP];
Portmap  284  [FSEL= IN];
Portmap  211  [FSEL= OP];
Portmap  288  [FSEL= IN];
Portmap  219  [FSEL= OP];
Portmap  252  [FSEL= IN];
Portmap  224  [FSEL= OP];
Portmap  281  [FSEL= IN];
Portmap  218  [FSEL= OP];
Portmap  290  [FSEL= IN];
Portmap  227  [FSEL= OP];
Portmap  292  [FSEL= IN];
Portmap  192  [FSEL= OP];
Portmap  274  [FSEL= IN];
Portmap  220  [FSEL= OP];
Portmap  296  [FSEL= IN];
Portmap  235  [FSEL= OP];
Portmap  269  [FSEL= IN];
Portmap  214  [FSEL= OP];
Portmap  264  [FSEL= IN];
Portmap  216  [FSEL= OP];
Portmap  294  [FSEL= IN];
Portmap  241  [FSEL= OP];
Portmap  258  [FSEL= IN];
Portmap  212  [FSEL= OP];
Portmap  282  [FSEL= IN];
Portmap  210  [FSEL= OP];
Portmap  300  [FSEL= IN];
Portmap  238  [FSEL= OP];
Portmap  298  [FSEL= IN];
Portmap  199  [FSEL= OP];
Portmap  304  [FSEL= IN];
Portmap  242  [FSEL= OP];
Portmap  302  [FSEL= IN];
Portmap  240  [FSEL= OP];
Portmap  306  [FSEL= IN];
Portmap  207  [FSEL= OP];
Portmap  278  [FSEL= IN];
Portmap  208  [FSEL= OP];
Portmap  310  [FSEL= IN];
Portmap  215  [FSEL= OP];
Portmap  275  [FSEL= IN];
Portmap  246  [FSEL= OP];
Portmap  313  [FSEL= IN];
Portmap  223  [FSEL= OP];
Portmap  314  [FSEL= IN];
Portmap  193  [FSEL= OP];
Portmap  250  [FSEL= IN];
Portmap  254  [FSEL= OP];
Portmap  317  [FSEL= IN];
Portmap  231  [FSEL= OP];
Portmap  1  [FSEL= IN];
Portmap  201  [FSEL= OP];
Portmap  283  [FSEL= IN];
Portmap  248  [FSEL= OP];
Portmap  263  [FSEL= IN];
Portmap  249  [FSEL= OP];
Portmap  257  [FSEL= IN];
Portmap  253  [FSEL= OP];
Portmap  280  [FSEL= IN];
```

WALK_NLT.DOC 18530 December 23, 1995 10:54:09 AM
I-CUBE FORMAT NETLIST CREATED BY IQROUTEB.EXE
INPUT FILES: WALK.PCA OUTPUT FILES: WALK.NLT

```
Portmap  206  [FSEL= OP];
Portmap  276  [FSEL= IN];
Portmap  202  [FSEL= OP];
Portmap  271  [FSEL= IN];
Portmap  245  [FSEL= OP];
Portmap  279  [FSEL= IN];
Portmap  232  [FSEL= OP];
Portmap  2  [FSEL= IN];
Portmap  228  [FSEL= OP];
Portmap  5  [FSEL= IN];
Portmap  209  [FSEL= OP];
Portmap  272  [FSEL= IN];
Portmap  204  [FSEL= OP];
Portmap  270  [FSEL= IN];
Portmap  200  [FSEL= OP];
Portmap  6  [FSEL= IN];
Portmap  237  [FSEL= OP];
Portmap  10  [FSEL= IN];
Portmap  229  [FSEL= OP];
Portmap  12  [FSEL= IN];
Portmap  221  [FSEL= OP];
Portmap  16  [FSEL= IN];
Portmap  213  [FSEL= OP];
Portmap  18  [FSEL= IN];
Portmap  217  [FSEL= OP];
Portmap  268  [FSEL= IN];
Portmap  196  [FSEL= OP];
Portmap  20  [FSEL= IN];
Portmap  225  [FSEL= OP];
Portmap  266  [FSEL= IN];
Portmap  198  [FSEL= OP];
Portmap  22  [FSEL= IN];
Portmap  205  [FSEL= OP];
Portmap  289  [FSEL= IN];
Portmap  191  [FSEL= OP];
Portmap  297  [FSEL= IN];
Portmap  233  [FSEL= OP];
Portmap  305  [FSEL= IN];
Portmap  189  [FSEL= OP];
Portmap  262  [FSEL= IN];
Portmap  194  [FSEL= OP];
Portmap  260  [FSEL= IN];
Portmap  188  [FSEL= OP];
Portmap  308  [FSEL= IN];
Portmap  197  [FSEL= OP];
Portmap  312  [FSEL= IN];
Portmap  239  [FSEL= OP];
Portmap  8  [FSEL= IN];
Portmap  195  [FSEL= OP];
Portmap  256  [FSEL= IN];
Portmap  190  [FSEL= OP];

Connect  267  251;
Connect  261  247;
Connect  255  243;
Connect  244  236;
Connect  277  234;
Connect  273  230;
Connect  286  203;
Connect  265  226;
Connect  259  222;
Connect  284  211;
Connect  288  219;
Connect  252  224;
Connect  281  218;
Connect  290  227;
Connect  292  192;
Connect  274  220;
Connect  296  235;
Connect  269  214;
Connect  264  216;
Connect  294  241;
Connect  258  212;
Connect  282  210;
Connect  300  238;
Connect  298  199;
Connect  304  242;
Connect  302  240;
Connect  306  207;
Connect  278  208;
Connect  310  215;
Connect  275  246;
Connect  313  223;
Connect  314  193;
Connect  250  254;
Connect  317  231;
Connect  1  201;
Connect  283  248;
Connect  263  249;
Connect  257  253;
Connect  280  206;
Connect  276  202;
Connect  271  245;
Connect  279  232;
Connect  2  228;
Connect  5  209;
Connect  272  204;
Connect  270  200;
Connect  6  237;
Connect  10  229;
Connect  12  221;
Connect  16  213;
Connect  18  217;
Connect  268  196;
Connect  20  225;
Connect  266  198;
Connect  22  205;
Connect  289  191;
Connect  297  233;
Connect  305  189;
Connect  262  194;
Connect  260  188;
Connect  308  197;
Connect  312  239;
Connect  8  195;
Connect  256  190;

EndDevInst;
```