# CPU Design HOW-TO

# Table of Contents

# CPU Design HOW-TO

## Al Dev (Alavoor Vasudevan) alavoor[AT]yahoo.com

v12.5, 17 Feb 2002

---

*CPU is the "brain" of computer and is a very vital component of computer system and is like a "cousin brother" of operating system (Linux or Unix). This document helps companies, businesses, universities and research institutes to design, build and manufacture CPUs. Also the information will be useful for university students of U.S.A and Canada who are studying computer science/engineering. The document has URL links which helps students understand how a CPU is designed and manufactured. Perhaps in near future there will be a GNU/GPLed CPU running Linux, Unix, Microsoft Windows, Apple Mac and BeOS operating systems!!*

---

# 1. Introduction

**(The latest version of this document is at http://www.milkywaygalaxy.freeservers.com. You may want to check there for changes).**

This document provides you comprehensive list of URLs for CPU Design and fabrication. Using this information students, companies, universities or businesses can make new CPUs which can run Linux/Unix operating systems.

In olden days, chip vendors were also the IP developers and the EDA tools developers. Nowadays, we have specialized fab companies (TSMC http://www.tsmc.com), IP companies (ARM http://www.arm.com, MIPS http://www.mips.com, Gray Research LLC http://cnets.sourceforge.net/grllc.html ), and tools companies ( Mentor http://www.mentor.com, Cadence http://www.cadence.com, etc.), and combinations of these (Intel). You can buy IP bundled with hardware (Intel), bundled with your tools (EDA companies), or separately (IP providers).

Enter the FPGA vendors (Xilinx http://www.xilinx.com, Altera http://www.altera.com). They have an opportunity to seize upon a unique business model.

VA Linux systems http://www.valinux.com builds the entire system and perhaps in future will design and build CPUs for Linux.

Visit the following CPU design sites:

- FPGA CPU Links http://www.fpgacpu.org/links.html
- FPGA Main site http://www.fpgacpu.org
- OpenRISC 1000 Free Open-source 32-bit RISC processor IP core competing with proprietary ARM and MIPS is at http://www.opencores.org
- Open IP org http://www.openip.org
- Free IP org - ASIC and FPGA cores for masses http://www.free-ip.com

# 2. What is IP ?

What is IP ? IP is short for **Intellectual Property**. More specifically, it is a block of logic that can be used in making ASIC's and FPGA's. Examples of "IP Cores" are, UART's, CPU's, Ethernet Controllers, PCI

Interfaces, etc. In the past, quality cores of this nature could cost anywhere from US$5,000 to more than US$350,000. This is way too high for the average company or individual to even contemplate using -- Hence, the Free-IP project.

Initially the Free-IP project will focus on the more complex cores, like CPU's and Ethernet controllers. Less complex cores might follow.

The Free-IP project is an effort to make quality IP available to anyone.

Visit the following sites for IP cores -

- Open IP org http://www.openip.org
- Free IP org - ASIC and FPGA cores for masses http://www.free-ip.com
- FPGA Main site http://www.fpgacpu.org

# 2.1 Free CPU List

Here is the list of Free CPUs available or curently under development -

- F-CPU 64-bit Freedom CPU http://www.f-cpu.org mirror site at http://www.f-cpu.de
- SPARC Organisation http://www.sparc.org
- SPARC International http://www.sparc.com
- European Space Agency - SPARC architecture LEON CPU http://www.estec.esa.nl/wsmwww/leon
- European Space Agency - ERC32 SPARC V7 CPU http://www.estec.esa.nl/wsmwww/erc32
- Atmel ERC32 SPARC part # TSC695E http://www.atmel-wm.com/products click on Aerospace=>Space=>Processors
- Sayuri at http://www.morphyplanning.co.jp/Products/FreeCPU/freecpu-e.html and manufactured by Morphy Planning Ltd at http://www.morphyone.org and feature list at http://ds.dial.pipex.com/town/plaza/aj93/waggy/hp/features/morphyone.htm and in Japanese language at http://www.morphyplanning.or.jp
- OpenRISC 1000 Free 32-bit processor IP core competing with proprietary ARM and MIPS is at http://www.opencores.org/cores/or1k
- OpenRISC 2000 is at http://www.opencores.org
- STM 32-bit, 2-way superscalar RISC CPU http://www.asahi-net.or.jp/~uf8e-itu
- Green Mountain - GM HC11 CPU Core is at http://www.gmvhdl.com/hc11core.html
- Open-source CPU site - Google Search "Computers>Hardware>Open Source" http://directory.google.com/Top/Computers/Hardware/Open_Source
- Free microprocessor and DSP IP cores written in Verilog or VHDL http://www.cmosexod.com
- Free hardware cores to speed development http://www.scrap.de/html/opencore.htm
- Linux open hardware and free EDA systems http://opencollector.org

# 2.2 Commercial CPU List

- **Russian E2K 64-bit CPU (Very fast CPU !!!)** website : http://www.elbrus.ru/roadmap/e2k.html. **ELBRUS** is now partnered (alliance) with Sun Microsystems of USA.
- Korean CPU from Samsung 64-bit CPU original from DEC Alpha http://www.samsungsemi.com Alpha-64bit CPU is at http://www.alpha-processor.com Now there is collaboration between Samsumg, Compaq of USA on Alpha CPU
- Intel IA 64 http://developer.intel.com/design/ia-64
- Transmeta crusoe CPU and in near future Transmeta's 64-bit CPU http://www.transmeta.com

- Sun Ultra-sparc 64-bit CPU http://www.sun.com or http://www.sunmicrosystems.com
- HAL-Fujitsu (California) Super-Sparc 64-bit processor http://www.hal.com also compatible to Sun's sparc architecture.
- SPARC Organisation http://www.sparc.org
- SPARC International http://www.sparc.com
- MIPS RISC CPUs http://www.mips.com
- Silicon Graphics MIPS Architecture CPUs http://www.sgi.com/processors
- IDT MIPS Architecture CPUs http://www.idt.com
- IBM Power PC (motorola) http://www.motorola.com/SPS/PowerPC/index.html
- Motorola embedded processors. SPS processor based on PowerPC, M-CORE, ColdFire, M68k, or M68HC cores http://www.mot-sps.com
- Hitachi SuperH 64-bit RISC processor SH7750 http://www.hitachi.com sold at $40 per cpu in quantities of 10,000. Hitachi SH4,3,2,1 CPUs http://semiconductor.hitachi.com/superh
- Fujitsu 64-bit processor http://www.fujitsu.com
- Seimens Pyramid CPU from Pyramid Technologies
- Intel X86 series 32-bit CPUs Pentiums, Celeron etc..
- AMDs X86 series 32-bit CPUs K-6, Athlon etc..
- National's Cyrix X86 series 32-bit CPUs Cyrix etc..
- ARC CPUs : http://www.arccores.com
- QED RISC 64-bit and MIPS cpus : http://www.qedinc.com/about.htm
- Origin 2000 CPU - http://techpubs.sgi.com/library/manuals/3000/007-3511-001/html/O2000Tuning.1.html
- NVAX CPUs http://www.research.compaq.com/wrl/DECarchives/DTJ/DTJ700 and at mirror-site
- Univ. of Mich High-perf. GaAs Microprocessor Project http://www.eecs.umich.edu/UMichMP
- Hyperstone E1-32 RISC/DSP processor http://bwrc.eecs.berkeley.edu/CIC/tech/hyperstone
- PSC1000 32-bit RISC processor http://www.ptsc.com/psc1000/index.html
- IDT R/RV4640 and R/RV4650 64-bit CPU w/DSP Capability http://www.idt.com/products/pages/Processors-PL100_Sub205_Dev128.html
- ARM CPU http://www.arm.com/Documentation
- Cogent CPUs http://www.cogcomp.com
- CPU Info center - List of CPUs sparc, arm etc.. http://bwrc.eecs.berkeley.edu/CIC/tech
- Main CPU site is : Google Search engine CPU site "Computers>Hardware>Components>Microprocessors" http://directory.google.com/Top/Computers/Hardware/Components/Microprocessors

Other important CPU sites are at -

- World-wide 24-hour news on CPUs http://www.newsnow.co.uk/cgi/NewsNow/NewsLink.htm?Theme=Processors
- The computer architecture site is at http://www.cs.wisc.edu/~arch/www
- ARM CPU http://www.arm.com/Documentation
- Great CPUs http://www.cs.uregina.ca/~bayko/cpu.html
- Microdesign resources http://www.mdronline.com

# 3. CPU Museum and Silicon Zoo

This chapter gives very basics of CPU technology. If you have good technical background then you can skip this entire chapter.

## 3.1 CPU Museum

CPU Museum is at

- Intel CPU Museum http://www.intel.com/intel/intelis/museum
- Intel - History of Microprocessors http://www.intel.com/intel/museum/25anniv
- Virtual Museum of Computing http://www.museums.reading.ac.uk/vmoc
- Silicon Zoo http://micro.magnet.fsu.edu/creatures/index.html
- Intel - How the Microprocessors work http://www.intel.com/education/mpuworks
- Simple course in Microprocessors http://www.hkrmicro.com/course/micro.html

## 3.2 How Transistors work

Microprocessors are essential to many of the products we use every day such as TVs, cars, radios, home appliances and of course, computers. Transistors are the main components of microprocessors. At their most basic level, transistors may seem simple. But their development actually required many years of painstaking research. Before transistors, computers relied on slow, inefficient vacuum tubes and mechanical switches to process information. In 1958, engineers (one of them Intel founder Robert Noyce) managed to put two transistors onto a silicon crystal and create the first integrated circuit that led to the microprocessor.

Transistors are miniature electronic switches. They are the building blocks of the microprocessor which is the brain of the computer. Similar to a basic light switch, transistors have two operating positions, on and off. This on/off, or binary functionality of transistors enables the processing of information in a computer.

**How a simple electronic switch works:**

The only information computers understand are electrical signals that are switched on and off. To comprehend transistors, it is necessary to have an understanding of how a switched electronic circuit works. Switched electronic circuits consist of several parts. One is the circuit pathway where the electrical current flows - typically through a wire. Another is the switch, a device that starts and stops the flow of electrical current by either completing or breaking the circuit's pathway. Transistors have no moving parts and are turned on and off by electrical signals. The on/off switching of transistors facilitates the work performed by microprocessors.

## 3.3 How a Transistors handles information

Something that has only two states, like a transistor, can be referred to as binary. The transistor's on state is represented by a 1 and the off state is represented by a 0. Specific sequences and patterns of 1's and 0's generated by multiple transistors can represent letters, numbers, colors and graphics. This is known as binary notation

## 3.4 Displaying binary information

**Spell your name in Binary:**

Each character of the alphabet has a binary equivalent. Below is the name JOHN and its equivalent in binary.

```
        J   0100 1010
        O   0100 1111
```

```
H  0100 1000
N  0100 1110
```

More complex information can be created such as graphics, audio and video using the binary, or on/off action of transistors.

Scroll down to the Binary Chart below to see the complete alphabet in binary.

| Character | Binary | Character | Binary |
|---|---|---|---|
| A | 0100 0001 | N | 0100 1110 |
| B | 0100 0010 | O | 0100 1111 |
| C | 0100 0011 | P | 0101 0000 |
| D | 0100 0100 | Q | 0101 0001 |
| E | 0100 0101 | R | 0101 0010 |
| F | 0100 0110 | S | 0101 0011 |
| G | 0100 0111 | T | 0101 0100 |
| H | 0100 1000 | U | 0101 0101 |
| I | 0100 1001 | V | 0101 0110 |
| J | 0100 1010 | W | 0101 0111 |
| K | 0100 1011 | X | 0101 1000 |
| L | 0100 1100 | Y | 0101 1001 |
| M | 0100 1101 | Z | 0101 1010 |

**Binary Chart for Alphabets**

# 3.5 What is a Semi-conductor?

Conductors and insulators :

Many materials, such as most metals, allow electrical current to flow through them. These are known as conductors. Materials that do not allow electrical current to flow through them are called insulators. Pure silicon, the base material of most transistors, is considered a semiconductor because its conductivity can be modulated by the introduction of impurities.

## Anatomy of Transistor

Semiconductors and flow of electricity

Adding certain types of impurities to the silicon in a transistor changes its crystalline structure and enhances its ability to conduct electricity. Silicon containing boron impurities is called p-type silicon - p for positive or lacking electrons. Silicon containing phosphorus impurities is called n-type silicon - n for negative or having a majority of free electrons

## A Working Transistor

A Working transistor - The On/Off state of Transistor

Transistors consist of three terminals; the source, the gate and the drain.

In the n-type transistor, both the source and the drain are negatively-charged and sit on a positively-charged well of p-silicon.

When positive voltage is applied to the gate, electrons in the p-silicon are attracted to the area under the gate forming an electron channel between the source and the drain.

When positive voltage is applied to the drain, the electrons are pulled from the source to the drain. In this state the transistor is on.

If the voltage at the gate is removed, electrons aren't attracted to the area between the source and drain. The pathway is broken and the transistor is turned off.

## Impact of Transistors

The Impact of Transistors - How microprocessors affect our lives.

The binary function of transistors gives micro- processors the ability to perform many tasks; from simple word processing to video editing. Micro- processors have evolved to a point where transistors can execute hundreds of millions of instructions per second on a single chip. Automobiles, medical devices, televisions, computers and even the Space Shuttle use microprocessors. They all rely on the flow of binary information made possible by the transistor.

# 4. CPU Design and Architecture

# 4.1 CPU Design

Visit the following links for information on CPU Design.

- Hamburg University VHDL archive http://tech-www.informatik.uni-hamburg.de/vhdl
- Kachina Design tools http://SAL.KachinaTech.COM/Z/1/index.shtml
- List of FPGA-based Computing Machines http://www.io.com/~guccione/HW_list.html
- SPARC Organisation http://www.sparc.org
- SPARC International http://www.sparc.com
- Design your own processor http://www.spacetimepro.com
- Teaching Computer Design with FPGAs http://www.fpgacpu.org
- Technical Committee on Computer Architecture http://www.computer.org/tab/tcca
- Frequently Asked Questions FAQ on VHDL http://www.vhdl.org/vi/comp.lang.vhdl or it is at http://www.vhdl.org/comp.lang.vhdl
- Comp arch FAQ http://www.esacademy.com/automation/faq.htm
- Comp arch FAQ ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/arch
- VME Bus FAQ http://www.hitex.com/automation/FAQ/vmefaq
- Homepage of SPEC http://performance.netlib.org/performance/html/spec.html
- Linux benchmarks http://www.silkroad.com/linux-bm.html

## 4.2 Online Textbooks on CPU Architecture

- Online HTML book http://odin.ee.uwa.edu.au/~morris/CA406/CA_ToC.html
- Univ of Texas Comp arch :
  http://www.cs.panam.edu/~meng/Course/CS4335/Notes/master/master.html
- Number systems and Logic circuits : http://www.tpub.com/neets/book13/index.htm
- Digital Logic: http://www.play-hookey.com/digital
- FlipFlops: http://www.ece.utexas.edu/~cjackson/FlipFlops/web_pages/Publish/FlipFlops.html
- Instruction Execution cycle: http://cq-pan.cqu.edu.au/students/timp1/exec.html
- Truth Table constructor: http://pirate.shu.edu/~borowsbr/Truth/Truth.html
- Overview of Shared Memory: http://www.sics.se/cna/mp_overview.html
- Simulaneous Multi-threading in processors : http://www.cs.washington.edu/research/smt
- Study Web : http://www.studyweb.com/links/277.html
- Univ notes: http://www.ece.msstate.edu/~linder/Courses/EE4713/notes
- Advice: An Adaptable and Extensible Distributed Virtual Memory Architecture
  http://www.gsyc.inf.uc3m.es/~nemo/export/adv-pdcs96/adv-pdcs96.html
- Univ of Utah Avalanche Scalable Parallel Processor Project
  http://www.cs.utah.edu/avalanche/avalanche-publications.html
- Distributed computing : http://www.geocities.com/SiliconValley/Vista/4015/pdcindex.html
- Pisma Memory architecture: http://aiolos.cti.gr/en/pisma/pisma.html
- Shared Mem Arch: http://www.ncsa.uiuc.edu/General/Exemplar/ARPA
- Textbooks on Comp Arch: http://www.rdrop.com/~cary/html/computer_architecture.html#book and
  VLSI design http://www.rdrop.com/~cary/html/vlsi.html
- Comp Arch Conference and Journals http://www.handshake.de/user/kroening/conferences.html
- WWW Comp arch page http://www.cs.wisc.edu/~arch/www

## 4.3 University Lecture notes on CPU Architecture

- Advanced Computer Architecture http://www.cs.utexas.edu/users/dahlin/Classes/GradArch
- Computer architecture - Course level 415 http://www.diku.dk/teaching/2000f/f00.415
- MIT: http://www.csg.lcs.mit.edu/6.823
- UBC CPU slides : http://www.cs.ubc.ca/spider/neufeld/courses/cs218/chapter8/index.htm
- Purdue Univ slides: http://www.ece.purdue.edu/~gba/ee565/Sessions/S03HTML/index.htm
- Rutgers Univ - Principles of Comp Arch :
  http://www.cs.rutgers.edu/~murdocca/POCA/Chapter02.html
- Brown Univ - http://www.engin.brown.edu/faculty/daniels/DDZO/cmparc.html
- Univ of Sydney - Intro Digital Systems : http://www.eelab.usyd.edu.au/digital_tutorial/part3
- Bournemouth Univ, UK Principles of Computer Systems :
  http://ncca.bournemouth.ac.uk/CourseInfo/BAVisAn/Year1/CompSys
- Parallel Virtual machine: http://www.netlib.org/pvm3/book/node1.html
- univ center: http://www.eecs.lehigh.edu/~mschulte/ece401-99
- univ course: http://www.cs.utexas.edu/users/fussell/cs352
- Examples of working VLSI circuits(in Greek) http://students.ceid.upatras.gr/~gef/projects/vlsi

## 4.4 CPU Architecture

Visit the following links for information on CPU architecture

- Comp architecture: http://www.rdrop.com/~cary/html/computer_architecture.html and VLSI design http://www.rdrop.com/~cary/html/vlsi.html
- Beyond RISC - The Post-RISC Architecture http://www.cps.msu.edu/~crs/cps920
- Beyond RISC - PostRISC : http://www.ceng.metu.edu.tr/~e106170/postrisc.html
- List of CPUS http://einstein.et.tudelft.nl/~offerman/cl.contents2.html
- PowerPC Arch http://www.mactech.com/articles/mactech/Vol.10/10.08/PowerPcArchitecture
- CPU Info center - List of CPUs sparc, arm etc.. http://bwrc.eecs.berkeley.edu/CIC/tech
- cpu arch intel IA 64 http://developer.intel.com/design/ia-64
- Intel 386 CPU architecture http://www.delorie.com/djgpp/doc/ug/asm/about-386.html
- Freedom CPU architecture http://f-cpu.tux.org/original/Freedom.php3
- Z80 CPU architecture http://www.geocities.com/SiliconValley/Peaks/3938/z80arki.htm
- CRIMSEN OS and teaching-aid CPU http://www.dcs.gla.ac.uk/~ian/project3/node1.html
- Assembly Language concepts http://www.cs.uaf.edu/~cs301/notes/Chapter1/node1.html
- Alpha CPU architecture http://www.linux3d.net/cpu/CPU/alpha/index.shtml
- http://hugsvr.kaist.ac.kr/~exit/cpu.html
- Tron CPU architecture http://tronweb.super-nova.co.jp/tronvlsicpu.html

# 4.5 Usenet Newsgroups for CPU design

- Newsgroup computer architecture news:comp.arch
- Newsgroup FPGA news:comp.arch.fpga
- Newsgroup Arithmetic news:comp.arch.arithmetic
- Newsgroup Bus news:comp.arch.bus
- Newsgroup VME Bus news:comp.arch.vmebus
- Newsgroup embedded news:comp.arch.embedded
- Newsgroup embedded piclist news:comp.arch.embedded.piclist
- Newsgroup storage news:comp.arch.storage
- Newsgroup VHDL news:comp.lang.vhdl
- Newsgroup Computer Benchmarks news:comp.benchmarks

# 5. Fabrication, Manufacturing CPUs

After doing the design and testing of CPU, your company may want to mass produce the CPUs. There are many "semi-conductor foundries" in the world who will do that for you for a nominal competetive cost. There are companies in USA, Germany, UK, Japan, Taiwan, Korea and China.

TMSC (Taiwan) is the **"largest independent foundry"** in the world. You may want to shop around and you will get the best rate for a very high volume production (greater than 100,000 CPU units).

# 5.1 Foundry Business is in Billions of dollars!!

Foundry companies invested very heavily in the infra-structure and building plants runs in several millions of dollars! Silicon foundry business will grow from $7 billion to $36 billion by 2004 (414% increase!!). More integrated device manufacturers (IDMs) opt to outsource chip production verses adding wafer-processing capacity.

Independent foundries currently produce about 12% of the semiconductors in the world, and by 2004, that share will more than double to 26%.

The "Big Three" pure-play foundries in the whole world are:

1. Taiwan Semiconductor Manufacturing Co. (TSMC)
2. United Microelectronics Corp. (UMC)
3. Chartered Semiconductor Manufacturing Ltd. Pte.

These three companies collectively account for 69% of today's silicon foundry volume, but their share is expected to grow to 88% by 2004. These percentages exclude those companies which are not "pure-play foundries" like Intel, IBM and others who have in-house foundries for self-production of wafers.

# 5.2 Fabrication of CPU

There are hundreds of foundries in the world (too numerous to list). Some of them are -

- Fabless Semiconductor Association http://www.fsa.org
- TSMC (Taiwan Semi-conductor Manufacturing Co) http://www.tsmc.com, about co http://www.tsmc.com/about/index.html
- Chartered Semiconductor Manufacturing, Singapore http://www.csminc.com
- United Microelectronics Corp. (UMC) http://www.umc.com/index.html
- Advanced BGA Packing http://www.abpac.com
- Amcor, Arizona http://www.amkor.com
- Elume, USA http://www.elume.com
- X-Fab, Gesellschaft zur Fertigung von Wafern mbH, Erfurt, Germany http://www.xfab.com
- IBM corporation, (Semi-conductor foundry div) http://www.ibm.com
- National Semi-conductor Co, Santa Clara, USA http://www.natioanl.com
- Tower Semiconductor, San Jose, USA http://www.towersemi.com
- Intel corporation (Semi-conductor foundries), USA http://www.intel.com
- Hitachi Semi-conductor Co, Japan http://www.hitachi.com
- FUJITSU limited, Japan has Wafer-foundry-services
- Mitsubhishi Semi-conductor Co, Japan
- Hyandai Semi-conductor, Korea http://www.hea.com
- Samsumg Semi-conductor, Korea
- Atmel, France http://www.atmel-wm.com

If you know any major foundries, let me know I will add to list.

List of CHIP foundry companies

- Chip directory http://www.xs4all.nl/~ganswijk/chipdir/make/foundry.htm
- Chip makers http://www.xs4all.nl/~ganswijk/chipdir/make/index.htm
- IC manufacturers http://www.xs4all.nl/~ganswijk/chipdir/c/a.htm

# 6. Super Computer Architecture

For building Super computers, the trend that seems to emerge is that most new systems look as minor variations on the same theme: clusters of RISC-based Symmetric Multi-Processing (SMP) nodes which in turn are connected by a fast network. Consider this as a natural architectural evolution. The availability of relatively low-cost (RISC) processors and network products to connect these processors together with standardised communication software has stimulated the building of home-brew clusters computers as an alternative to complete systems offered by vendors.

Visit the following sites for Super Computers -

- Top 500 super computers http://www.top500.org/ORSC/2000
- National Computing Facilities Foundation http://www.nwo.nl/ncf/indexeng.htm
- Linux Super Computer Beowulf cluster http://www.tldp.org/HOWTO/Beowulf-HOWTO.html
- Extreme machines - beowulf cluster http://www.xtreme-machines.com
- System architecture description of the Hitachi SR2201
  http://www.hitachi.co.jp/Prod/comp/hpc/eng/sr1.html
- Personal Parallel Supercomputers http://www.checs.net/checs_98/papers/super

# 6.1 Main Architectural Classes

Before going on to the descriptions of the machines themselves, it is important to consider some mechanisms that are or have been used to increase the performance. The hardware structure or architecture determines to a large extent what the possibilities and impossibilities are in speeding up a computer system beyond the performance of a single CPU. Another important factor that is considered in combination with the hardware is the capability of compilers to generate efficient code to be executed on the given hardware platform. In many cases it is hard to distinguish between hardware and software influences and one has to be careful in the interpretation of results when ascribing certain effects to hardware or software peculiarities or both. In this chapter we will give most emphasis to the hardware architecture. For a description of machines that can be considered to be classified as "high-performance".

Since many years the taxonomy of Flynn has proven to be useful for the classification of high-performance computers. This classification is based on the way of manipulating of instruction and data streams and comprises four main architectural classes. We will first briefly sketch these classes and afterwards fill in some details when each of the classes is described.

# 6.2 SISD machines

These are the conventional systems that contain one CPU and hence can accommodate one instruction stream that is executed serially. Nowadays many large mainframes may have more than one CPU but each of these execute instruction streams that are unrelated. Therefore, such systems still should be regarded as (a couple of) SISD machines acting on different data spaces. Examples of SISD machines are for instance most workstations like those of DEC, Hewlett-Packard, and Sun Microsystems. The definition of SISD machines is given here for completeness' sake. We will not discuss this type of machines in this report.

# 6.3 SIMD machines

Such systems often have a large number of processing units, ranging from 1,024 to 16,384 that all may execute the same instruction on different data in lock-step. So, a single instruction manipulates many data items in parallel. Examples of SIMD machines in this class are the CPP DAP Gamma II and the Alenia Quadrics.

Another subclass of the SIMD systems are the vectorprocessors. Vectorprocessors act on arrays of similar data rather than on single data items using specially structured CPUs. When data can be manipulated by these vector units, results can be delivered with a rate of one, two and --- in special cases --- of three per clock cycle (a clock cycle being defined as the basic internal unit of time for the system). So, vector processors execute on their data in an almost parallel way but only when executing in vector mode. In this case they are several times faster than when executing in conventional scalar mode. For practical purposes vectorprocessors are

therefore mostly regarded as SIMD machines. Examples of such systems is for instance the Hitachi S3600.

# 6.4 MISD machines

Theoretically in these type of machines multiple instructions should act on a single stream of data. As yet no practical machine in this class has been constructed nor are such systems easily to conceive. We will disregard them in the following discussions.

# 6.5 MIMD machines

These machines execute several instruction streams in parallel on different data. The difference with the multi-processor SISD machines mentioned above lies in the fact that the instructions and data are related because they represent different parts of the same task to be executed. So, MIMD systems may run many sub-tasks in parallel in order to shorten the time-to-solution for the main task to be executed. There is a large variety of MIMD systems and especially in this class the Flynn taxonomy proves to be not fully adequate for the classification of systems. Systems that behave very differently like a four-processor NEC SX-5 and a thousand processor SGI/Cray T3E fall both in this class. In the following we will make another important distinction between classes of systems and treat them accordingly.

## Shared memory systems

Shared memory systems have multiple CPUs all of which share the same address space. This means that the knowledge of where data is stored is of no concern to the user as there is only one memory accessed by all CPUs on an equal basis. Shared memory systems can be both SIMD or MIMD. Single-CPU vector processors can be regarded as an example of the former, while the multi-CPU models of these machines are examples of the latter. We will sometimes use the abbreviations SM-SIMD and SM-MIMD for the two subclasses.

## Distributed memory systems

In this case each CPU has its own associated memory. The CPUs are connected by some network and may exchange data between their respective memories when required. In contrast to shared memory machines the user must be aware of the location of the data in the local memories and will have to move or distribute these data explicitly when needed. Again, distributed memory systems may be either SIMD or MIMD. The first class of SIMD systems mentioned which operate in lock step, all have distributed memories associated to the processors. As we will see, distributed-memory MIMD systems exhibit a large variety in the topology of their connecting network. The details of this topology are largely hidden from the user which is quite helpful with respect to portability of applications. For the distributed-memory systems we will sometimes use DM-SIMD and DM-MIMD to indicate the two subclasses. Although the difference between shared- and distributed memory machines seems clear cut, this is not always entirely the case from user's point of view. For instance, the late Kendall Square Research systems employed the idea of "virtual shared memory" on a hardware level. Virtual shared memory can also be simulated at the programming level: A specification of High Performance Fortran (HPF) was published in 1993 which by means of compiler directives distributes the data over the available processors. Therefore, the system on which HPF is implemented in this case will look like a shared memory machine to the user. Other vendors of Massively Parallel Processing systems (sometimes called MPP systems), like HP and SGI/Cray, also are able to support proprietary virtual shared-memory programming models due to the fact that these physically distributed memory systems are able to address the whole collective address space. So, for the user such systems have one global address space spanning all of the memory in the system. We will say a little more about the structure of such systems in the ccNUMA section. In addition, packages like TreadMarks provide a virtual shared memory environment for networks of

workstations.

# 6.6 Distributed Processing Systems

Another trend that has came up in the last few years is distributed processing. This takes the DM-MIMD concept one step further: instead of many integrated processors in one or several boxes, workstations, mainframes, etc., are connected by (Gigabit) Ethernet, FDDI, or otherwise and set to work concurrently on tasks in the same program. Conceptually, this is not different from DM-MIMD computing, but the communication between processors is often orders of magnitude slower. Many packages to realise distributed computing are available. Examples of these are PVM (st anding for Parallel Virtual Machine), and MPI (Message Passing Interface). This style of programming, called the "message passing" model has becomes so much accepted that PVM and MPI have been adopted by virtually all major vendors of distributed-memory MIMD systems and even on shared-memory MIMD systems for compatibility reasons. In addition there is a tendency to cluster shared-memory systems, for instance by HiPPI channels, to obtain systems with a very high computational power. E.g., the NEC SX-5, and the SGI/Cray SV1 have this structure. So, within the clustered nodes a shared-memory programming style can be used while between clusters message-passing should be used.

# 6.7 ccNUMA machines

As already mentioned in the introduction, a trend can be observed to build systems that have a rather small (up to 16) number of RISC processors that are tightly integrated in a cluster, a Symmetric Multi-Processing (SMP) node. The processors in such a node are virtually always connected by a 1-stage crossbar while these clusters are connected by a less costly network.

This is similar to the policy mentioned for large vectorprocessor ensembles mentioned above but with the important difference that all of the processors can access all of the address space. Therefore, such systems can be considered as SM-MIMD machines. On the other hand, because the memory is physically distributed, it cannot be guaranteed that a data access operation always will be satisfied within the same time. Therefore such machines are called ccNUMA systems where ccNUMA stands for Cache Coherent Non-Uniform Memory Access. The term "Cache Coherent" refers to the fact that for all CPUs any variable that is to be used must have a consistent value. Therefore, is must be assured that the caches that provide these variables are also consistent in this respect. There are various ways to ensure that the caches of the CPUs are coherent. One is the snoopy bus protocol in which the caches listen in on transport of variables to any of the CPUs and update their own copies of these variables if they have them. Another way is the directory memory, a special part of memory which enables to keep track of the all copies of variables and of their validness.

For all practical purposes we can classify these systems as being SM-MIMD machines also because special assisting hardware/software (such as a directory memory) has been incorporated to establish a single system image although the memory is physically distributed.

# 7. Linux Super Computers

Supercomputers traditionally have been expensive, highly customized designs purchased by a select group of customers, but the industry is being overhauled by comparatively mainstream technologies such as Intel processors, InfiniBand high-speed connections (see also Myricom, and Fibre Channel storage networks that have become fast enough to accomplish many tasks.

The new breed of supercomputers usually involve numerous two-processor servers bolted into racks and joined with special high-speed networks into a <u>cluster</u>.

<u>Linux Networx</u> customers include Los Alamos and Lawrence Livermore national laboratories for nuclear weapons research, Boeing for aeronautic engineering, and Sequenom for genetics research.

About <u>Clusterworx</u> : Clusterworx is the most complete administration tool for monitoring and management of Linux-based cluster systems. Clusterworx increases system uptime, improves cluster efficiency, tracks cluster performance, and removes the hassle from cluster installation and configuration. The primary features of Clusterworx include monitoring of system properties, integrated disk cloning using multicast technology, and event management of node properties through a remotely accessible, easy-to-use graphical user interface (GUI). Some of the system properties monitored include CPU Usage, Memory Usage, Disk I/O, Network Bandwidth, and many more. Additional custom properties can easily be monitored through the use of user-specific plug-ins. Events automate system administration tasks by setting thresholds on these properties and then taking default or custom actions when these values are exceeded.

About <u>Myricom</u>: Myrinet clusters are used for computationally demanding scientific and engineering applications, and for data-intensive web and database applications. All of the major OEM computer companies today offer cluster products. In addition to direct sales, Myricom supplies Myrinet products and software to IBM, HP, Compaq, Sun, NEC, SGI, Cray, and many other OEM and system-integration companies. There are thousands of Myrinet clusters in use world-wide, including several systems with more than 1000 processors.

# 7.1 Little Linux SuperComputer In Your Garage

Imagine your garage filled with dozens of computers all linked together in a super-powerful Linux cluster. You still have to supply your own hardware, but the geek equivalent of a Mustang GT will become easier to set up and maintain, thanks to new software to be demonstrated at LinuxWorld next week.

The Open Source Cluster Applications Resources (OSCAR) software, being developed by the <u>Open Cluster Group</u>, will allow a non-expert Linux user to set up a cluster in a matter of hours, instead of the days of work it now can take an experienced network administrator to piece one together. Developers of OSCAR are saying it'll be as easy as installing most software. Call it a "supercomputer on a CD."

"We've actually taken it to the point where a typical high school kid who has a little bit of experience with Linux and can get their hands on a couple of extra boxes could set up a cluster at home," says Stephen L. Scott, project leader at the Oak Ridge National Laboratory, one of several organizations working on OSCAR. "You can have a little supercomputer in your garage."

Supercomputing in Linux:

From A <u>step-by-step guide</u> on how to set up a cluster of PCQLinux machines for supercomputing

Shekhar Govindarajan, Friday, May 10, 2002

To keep it simple, we start with a cluster of three machines. One will be the server and the other two will be the nodes. However, plugging in additional nodes is easy and we will tell you the modification to accommodate additional nodes. Instead of two nodes, you can have a single node. So, even if you have two PCs, you can build a cluster. We suggest that you go through the article Understanding Clustering, page 42, which explains what a cluster is and what server and nodes mean in a cluster before you get started.

*Set up server hardware *You should have at least a 2 GB or bigger hard disk on the server. It should have a graphics card that is supported by PCQLinux 7.1 and a floppy drive. You also need to plug in two network cards preferably the faster PCI cards instead of ISA supported by PCQLinux.

Why two network cards? Adhering to the standards for cluster setups, if the server node needs to be connected to the outside (external) network? Internet or your private network?the nodes in the cluster must be on a separate network. This is needed if you want to remotely execute programs on the server. If not, you can do away with a second network card for the external network. For example, at PCQ Labs, we have our regular machines plugged in the 192.168.1.0 network. We selected the network 172.16.0.0 for the cluster nodes. Hence, on the server, one network card (called external interface) will be connected to the Labs network and the other network card (internal interface) will be connected to a switch. We used a 100/10 Mbps switch. A 100 Mbps switch is recommended because the faster the speed of the network, the faster is the message passing. All cluster nodes will also be connected to this switch.

*PCQLinux on server *If you already have a machine with PCQLinux 7.1, including the X Window (KDE or GNOME), installed you can use it as a server machine. In this case you may skip the following steps for installation. If this machine has a firewall (ipchains or iptables) setup, remove all strict restrictive rules, as it will hinder communication between the server and the nodes. The 'medium' level of firewall rules in PCQLinux is suitable. After the cluster set up, you may selectively enable the rules, if required.

If you haven't installed PCQLinux on the machine, opt for custom system install and manual partitioning. Create the swap and / (ROOT) partitions. If you are shown the 1024 cylinder limit problem, you may also have to create a /boot partition of about 50 MB. In the network configuration, fill in the hostname (say, server. cluster.net), IP address of the gateway/router on your network, and the IP of a DNS server (if any) running on your network. Leave other field to their defaults. We will set up the IP addresses for network cards after the installation. Select 'Medium' for the firewall configuration. We now come to the package-selection wizard. You don't need to install all the packages. Besides the packages selected by default, select 'Development' and 'Kernel Development' packages. These provide various libraries and header files for writing programs and are useful if you will develop applications on the cluster. You will need the X Window system because we will use a graphical tool for cluster set up and configuration. By default, GNOME is selected as the Window Manager. If you are comfortable using KDE, select it instead. By suggesting that you select only a few packages for install, we aim at a minimal installation. However, if you wish to install other packages like your favorite text editor, network management utilities or a Web server, then you can select them. Make sure that you set up your graphics card and monitor correctly.

After the installation finishes, reboot into PCQLinux. Log in as root.

*Set up OSCAR *Mount this month's CD and copy the file oscar-1.2.1.tar.gz from the directory system/cdrom/ unltdlinux/linux on the CD to /root. Uncompress and extract the archive as:

tar -zxvf oscar-1.2.1.tar.gz

This will extract the files in a directory named oscar-1.2.1 within /root directory.

OSCAR installs Linux on the nodes from the server across the network. For this, it constructs an image file from RPM packages. This image file is in turn picked up by the nodes to install PCQLinux onto them. The OSCAR version we've given on the CD is customized for RedHat 7.1. Though PCQLinux 7.1 is also based on RedHat 7.1, some RPMs with PCQLinux are of more recent versions than the ones required by OSCAR. OSCAR constructs the image out of a list of RPMs specified in sample.rpmlist in the subdirectory oscarsamples in oscar-1.2.1. You have to replace this file with the one customized for PCQLinux RPMs. We have given a file named sample.rpmlist on this month's CD in the directory system/cdrom/unltdlinux /linux.

Overwrite the file sample.rpmlist in the oscarsamples directory with this file.

```
*Copy PCQLinux RPMs to /tftpboot/rpm
*For creating the image, OSCAR will look for the PCQLinux RPMs in the
directory /tftpboot/rpm. Create a directory /tftpboot and a subdirectory
named rpm within it

mkdir /tftpboot
mkdir /tftpboot/rpm

Next, copy all the PCQLinux RPMs from both the CDs to /tftpboot/rpm
directory. Insert CD 1 (PCQLinux CD 1, given with our July 2001 issue)
and issue the following commands:

mount /mnt/cdrom
cd /mnt/cdrom/RedHat/RPMS
cp *.rpm /tftpboot/ rpm
cd
umount /mnt/cdrom

Insert CD 2 (given with the July 2001 issue) and issue the above
commands again.

Note. If you are tight at the disk space, you don't need to copy all the
RPMs to /tftpboot/rpm. You can copy only the RPMs listed in
sample.rpmlist file. Copy only the required RPMs.

*Copy required RPMs
*Type the following in a Linux text editor and save the file as copyrpms.sh

#!/bin/bash
rpms_path="/mnt/cdrom/RedHat/RPMS/"
rpms_list="/root/oscar-1.2.1/oscarsamples/sample.rpmlist"
mount /mnt/cdrom
while read line
do file="$rpms_path$line.i386.rpm"
if [ -f $file ]
then
cp $file /tftpboot/rpm
else file="$rpms_path$line.noarch.rpm"
if [ -f $file ]
then
cp $file /tftpboot/rpm
else file="$rpms_path$line.i586.rpm"
if [ -f $file ]
then
cp $file /tftpboot/rpm
else file="$rpms_path$line.i686.rpm"
if [ -f $file ]
then
cp $file /tftpboot/rpm
fi
fi
fi
fi
done < $rpms_list
eject

Give executable permissions to the file as:

chmod +x copyrpms.sh
```

Assuming that you have created the directory /tftpboot/rpm, insert
PCQLinux CD 1 (don't mount it) and issue:
./copyrpms

When all the RPMs from the CD are copied, the CD drive will eject. Next,
insert CD 2 and issue ./copyrpms again.

*Fix glitch in PCQLinux
*On this month's CD we have carried the zlib
rpm 'zlib-1.1.3-22.i386.rpm' which you can find in the directory
system/cdrom/ unltdlinux/linux on the CD. (We had given this on our July
CD as well, but the file was corrupt.) Install the RPM as:

rpm -ivh zlib-1.1.3-22.i386.rpm

Copy this file to /tftpboot/rpm directory. This will prompt you to
overwrite the corrupted zlib RPM, already in the directory. Go for it.

*Set up networking
*Linux names network cards or interfaces as eth0, eth1, eth2. In our
case eth0 is the internal interface and eth1 is the external interface.
We assign eth0, an IP address of 172.16.0.1. Since we are running a DHCP
server on the PCQ Labs network, we will set eth1 to obtain IP address
from the DHCP server. If you are using a single network card for the
cluster network, skip setting up the second card.

Launch X Window. Launch a terminal window within GNOME or KDE and issue
the command netcfg. This will pop up a graphical network configurator.
Click on the Interfaces tab. To set up the internal interface, click on
eth0 and then on edit. For IP address, enter 172.16.0.1 and for the
netmask enter 255.255.255.0. Click on 'Activate interface at boot time'.
For 'Interface configuration protocol' select 'none' from the drop-down
list.

To set up the external interface, select eth1 and click on edit. If you
are running a DHCP server, select dhcp from the drop down list. Else,
enter a free IP address (say, 192.168.1.23), the associated netmask
(say, 255.255.255.0) and select none from the drop-down list. In either
case, make sure to click on 'Activate interface at boot time'.

Highlight eth0 and click on the button 'Activate'. Do the same for eth1.
Finally, click on save and quit the configurator.

Issue the command, ifconfig to check whether the network interfaces are
up and have been given the correct IP addresses.
You are now ready to start Oscar.

*Run OSCAR
*In the terminal window, change to oscar-1.2.2 directory and issue the
command:

./install_cluster eth0

Replace eth0 with the name of the internal interface in your case. You
will see text flowing in the window. After a couple of minutes, the
graphical wizard of OSCAR will pop up. OSCAR installation calls cluster
nodes as clients

*Build image from RPMs
*Click on 'Build Oscar Client Image'. We assume that all the node
machines will have IDE hard disks. If you are using SCSI hard disk in
the nodes, you need to change the Disk Partition File. Refer to the

OSCAR installation documentation on the CD. When finished, a message
'Successfully created image oscarimage' will pop up.

*Tell OSCAR about the nodes
*Click on the button 'Define OSCAR clients'. Here you should see the
domain name, starting IP and subnet mast, pre-filled with cluster.net,
172.16.0.2 and 255.255. 255.0. With 'Number of hosts' you specify the
number of nodes. As per the OSCAR documentation, OSCAR supports up to
100 nodes or may be more. But it hasn't been experimented with arbitrary
large number of nodes. In our case we fill in two. If you are
experimenting with two machines, one server and the other the node, then
fill in one.

In OSCAR once you define the number of nodes you cannot change it after
the cluster is installed. You need to again start from the beginning,
ie, from the step when we issued 'install_cluster'

Note. If for any reason you need to start again, before issuing
./install_cluster, execute the script named start_over located in the
subdirectory scripts as:

/root/oscar-1.2.1/script/start_over'

Clicking on the 'Add clients' button will show 'Successfully created
clients' after a couple of seconds.

*Set up the nodes *
Before carrying out the subsequent steps in OSCAR installation, connect
the network cards of the node machines to the switch and set them up to
boot from floppy from their BIOS.

*Set up nodes to network
*We come back to OSCAR installation wizard running on the server
machine. Click on the button 'Set up Networking'. In the right frame you
will see a tree-like structure as shown in the screenshot. In our case,
the two nodes are given a hostname of oscarnode1.cluster.net and
oscarnode2. cluster.net. They are assigned IP addresses 172.16.0.2 and
172.16. 0.3 respectively. Next, we assign the MAC (Media Access Control)
address of the nodes to the listed IP addresses. This can be done by
booting the nodes using a floppy created by OSCAR or by networking
booting them. For the latter refer to the OSCAR documentation given on
the CD.

Click on the button 'Build AutoInstall Floppy'. This will pop up a
terminal window. Insert a blank floppy in the server and click 'y' to
continue. After the terminal window disappears, click on the button
'Collect MAC addresses' in the OSCAR window. Insert the floppy in one of
the node machines and power it on. The machine will boot from the
floppy. Press enter at the boot: prompt. After some time, the MAC
address of the node will show up in the left frame. Suppose we want to
assign the IP address 172.16.0.1 to this node. Click on the MAC address
in the left and on the 'osacrnde1.cluster.net' in the right frame. Then,
click on 'Assign MAC to node'.

*Assign IP addresses to the nodes of the cluster
*Switch off the node machine. Now boot the second node machine from the
same floppy. As before, the MAC address of the second node will appear
in the left frame. Assign it to oscarnode2. cluster.net.

If you want to plug in more node machines, repeat the above process for
them. When done, click on the button 'Stop collecting' on the OSCAR window.

After shutting down all the node machines, click on the button
'Configure DHCP Server'. Then click on the close button in the 'MAC
address collection' window.

*PCQLinux on the nodes
*Next, boot the first node machine again from the floppy. This time the
node machine will install PCQLinux 7.1 from the network. When done, a
message, as following, will be shown:

I have done for ' seconds. Reboot me already

Take out the floppy and reboot the node machine. This time it should
boot from the hard disk. If everything has gone well, you will boot into
PCOLinux 7.1. While booting, PCQLinux will detect and prompt you to set
up hardware like mouse, graphics card, sound card etc on the nodes.

*Problem: No active partition
*If you are shown an error during booting which says no active
partition, then boot from a Windows bootable floppy or CD. Launch fdisk
and select option2 (Set active partition). Set partition 1 of type
non-dos and about 31 MB in size as active. This is the /boot partition
from where the kernel boot image resides.

*Test networking of nodes
*On the server, open another terminal window and issue:

/root/oscar-1.2.2/scripts/ping_clients

If there is no problem with the networking, you will be shown 'All
clients responded'. Else check whether all nodes are powered on, defects
in network cables, hub/ switch ports etc. From now on, ideally, you
don't need to work physically on the node machines. Hence you can plug
off the monitor, keyboard, mouse, etc from the node machines. If the
node machines need to be accessed and worked upon, you should use SSH
(Secure Shell), similar to telnet but secure, to access them from the
server.

*All done
*Click on 'Complete Cluster Setup' and then on 'Test cluster Setup'.
This will pop up a terminal window and prompt you to enter a non-root
username. Enter 'shekhar' (say). If the user account does not exist on
the server machine, it will be created. In the latter case, you will be
prompted for a password for the new account. Click on the 'Quit' button
on the OSCAR window. Reboot the server machine.

*Test the cluster
*To test the cluster, log in as the user that you created above (shekhar
in our case) and issue:

cd OSCAR_test
./text_cluster

Enter the number of nodes when prompted (two in our case). For the
number of processors on each client enter 1 (assuming uniprocessor
machines). The test verifies the running of PBS and runs example
programs coded using LAM, MPICH, PVM libraries by dispatching them
through PBS to the nodes. You can see pbs_mom (see Understanding
Clustering, page 42) running on the nodes by issuing the command 'ps 'e
| grep pbs_mom' on the nodes.

If there are no error messages in the output, congratulations, you have
your supercomputer up and running. Our cluster setup qualifies to be

```
called a Beowulf cluster because it has been built using easily
available hardware, free and open-source software, the /home directory
on the server is exported to all the nodes via NFS (you can check this
by issuing the command 'mount' on the nodes), and finally the server and
nodes can execute command and scripts remotely on each other via SSH.
Using the libraries installed on the cluster, you can start developing
or executing cluster-aware applications on the server. The compilers for
them (like, gcc, g++) are same as with PCQLinux.

Shekhar Govindarajan
```

# 8. Neural Network Processors

NNs are models of biological neural networks and some are not, but historically, much of the inspiration for the field of NNs came from the desire to produce artificial systems capable of sophisticated, perhaps "intelligent", computations similar to those that the human brain routinely performs, and thereby possibly to enhance our understanding of the human brain.

Most NNs have some sort of "training" rule whereby the weights of connections are adjusted on the basis of data. In other words, NNs "learn" from examples (as children learn to recognize dogs from examples of dogs) and exhibit some capability for generalization beyond the training data.

NNs normally have great potential for parallelism, since the computations of the components are largely independent of each other. Some people regard massive parallelism and high connectivity to be defining characteristics of NNs, but such requirements rule out various simple models, such as simple linear regression (a minimal feedforward net with only two units plus bias), which are usefully regarded as special cases of NNs.

Some definitions of Neural Network (NN) are as follows:

- According to the DARPA Neural Network Study : A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.
- According to Haykin: A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:
  - ♦ Knowledge is acquired by the network through a learning process.
  - ♦ Interneuron connection strengths known as synaptic weights are used to store the knowledge.
- According to Nigrin: A neural network is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates only on local information. Furthermore each element operates asynchronously; thus there is no overall system clock.
- According to Zurada: Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge.

Visit the following sites for more info on Neural Network Processors

- Omers Neural Network pointers http://www.cs.cf.ac.uk/User/O.F.Rana/neural.html
- FAQ site ftp://ftp.sas.com/pub/neural/FAQ.html
- Automation corp Neural Network Processor hardware

# 9. Related URLs

Visit following locators which are related -

- Color Vim editor <u>http://metalab.unc.edu/LDP/HOWTO/Vim-HOWTO.html</u>
- Source code control system <u>http://metalab.unc.edu/LDP/HOWTO/CVS-HOWTO.html</u>
- Linux goodies main site <u>http://www.milkywaygalaxy.freeservers.com</u> and mirrors at
  <u>http://aldev0.webjump.com</u>, <u>angelfire</u>, <u>geocities</u>, <u>virtualave</u>, <u>50megs</u>, <u>theglobe</u>, <u>NBCi</u>, <u>Terrashare</u>,
  <u>Fortunecity</u>, <u>Freewebsites</u>, <u>Tripod</u>, <u>Spree</u>, <u>Escalix</u>, <u>Httpcity</u>, <u>Freeservers</u>.

# 10. Other Formats of this Document

This document is published in 14 different formats namely - DVI, Postscript, Latex, Adobe Acrobat PDF,
LyX, GNU-info, HTML, RTF(Rich Text Format), Plain-text, Unix man pages, single HTML file, SGML
(Linuxdoc format), SGML (Docbook format), MS WinHelp format.

This howto document is located at -

- <u>http://www.linuxdoc.org</u> and click on HOWTOs and search for howto document name using CTRL+f
  or ALT+f within the web-browser.

You can also find this document at the following mirrors sites -

- <u>http://www.caldera.com/LDP/HOWTO</u>
- <u>http://www.linux.ucla.edu/LDP</u>
- <u>http://www.cc.gatech.edu/linux/LDP</u>
- <u>http://www.redhat.com/mirrors/LDP</u>
- Other mirror sites near you (network-address-wise) can be found at
  <u>http://www.linuxdoc.org/mirrors.html</u> select a site and go to directory
  /LDP/HOWTO/xxxxx-HOWTO.html

- You can get this HOWTO document as a single file tar ball in HTML, DVI, Postscript or SGML
  formats from - <u>ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO/other-formats/</u> and
  <u>http://www.linuxdoc.org/docs.html#howto</u>
- Plain text format is in: <u>ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO</u> and
  <u>http://www.linuxdoc.org/docs.html#howto</u>
- Single HTML file format is in: <u>http://www.linuxdoc.org/docs.html#howto</u>

  Single HTML file can be created with command (see man sgml2html) - sgml2html -split 0
  xxxxhowto.sgml
- Translations to other languages like French, German, Spanish, Chinese, Japanese are in
  <u>ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO</u> and <u>http://www.linuxdoc.org/docs.html#howto</u> Any
  help from you to translate to other languages is welcome.

The document is written using a tool called "SGML-Tools" which can be got from - <u>http://www.sgmltools.org</u>
Compiling the source you will get the following commands like

- sgml2html xxxxhowto.sgml (to generate html file)
- sgml2html -split 0 xxxxhowto.sgml (to generate a single page html file)
- sgml2rtf xxxxhowto.sgml (to generate RTF file)

• sgml2latex xxxxhowto.sgml (to generate latex file)

# 10.1 Acrobat PDF format

PDF file can be generated from postscript file using either acrobat **distill** or **Ghostscript**. And postscript file is generated from DVI which in turn is generated from LaTex file. You can download distill software from http://www.adobe.com. Given below is a sample session:

```
bash$ man sgml2latex
bash$ sgml2latex filename.sgml
bash$ man dvips
bash$ dvips -o filename.ps filename.dvi
bash$ distill filename.ps
bash$ man ghostscript
bash$ man ps2pdf
bash$ ps2pdf input.ps output.pdf
bash$ acroread output.pdf &
```

Or you can use Ghostscript command **ps2pdf**. ps2pdf is a work-alike for nearly all the functionality of Adobe's Acrobat Distiller product: it converts PostScript files to Portable Document Format (PDF) files. **ps2pdf** is implemented as a very small command script (batch file) that invokes Ghostscript, selecting a special "output device" called **pdfwrite**. In order to use ps2pdf, the pdfwrite device must be included in the makefile when Ghostscript was compiled; see the documentation on building Ghostscript for details.

# 10.2 Convert Linuxdoc to Docbook format

This document is written in linuxdoc SGML format. The Docbook SGML format supercedes the linuxdoc format and has lot more features than linuxdoc. The linuxdoc is very simple and is easy to use. To convert linuxdoc SGML file to Docbook SGML use the program **ld2db.sh** and some perl scripts. The ld2db output is not 100% clean and you need to use the **clean_ld2db.pl** perl script. You may need to manually correct few lines in the document.

• Download ld2db program from http://www.dcs.gla.ac.uk/~rrt/docbook.html or from Milkyway Galaxy site
• Download the cleanup_ld2db.pl perl script from from Milkyway Galaxy site

The ld2db.sh is not 100% clean, you will get lots of errors when you run

```
bash$ ld2db.sh file-linuxdoc.sgml db.sgml
bash$ cleanup.pl db.sgml > db_clean.sgml
bash$ gvim db_clean.sgml
bash$ docbook2html db.sgml
```

And you may have to manually edit some of the minor errors after running the perl script. For e.g. you may need to put closing tag < /Para> for each < Listitem>

# 10.3 Convert to MS WinHelp format

You can convert the SGML howto document to Microsoft Windows Help file, first convert the sgml to html using:

```
bash$ sgml2html xxxxhowto.sgml    (to generate html file)
bash$ sgml2html -split 0  xxxxhowto.sgml (to generate a single page html file)
```

Then use the tool HtmlToHlp. You can also use sgml2rtf and then use the RTF files for generating winhelp files.

## 10.4 Reading various formats

In order to view the document in dvi format, use the xdvi program. The xdvi program is located in tetex-xdvi*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Publishing | TeX menu buttons. To read dvi document give the command -

```
xdvi -geometry 80x90 howto.dvi
man xdvi
```

And resize the window with mouse. To navigate use Arrow keys, Page Up, Page Down keys, also you can use 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' letter keys to move up, down, center, next page, previous page etc. To turn off expert menu press 'x'.

You can read postscript file using the program 'gv' (ghostview) or 'ghostscript'. The ghostscript program is in ghostscript*.rpm package and gv program is in gv*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Graphics menu buttons. The gv program is much more user friendly than ghostscript. Also ghostscript and gv are available on other platforms like OS/2, Windows 95 and NT, you view this document even on those platforms.

- Get ghostscript for Windows 95, OS/2, and for all OSes from http://www.cs.wisc.edu/~ghost

To read postscript document give the command -

```
gv howto.ps
ghostscript howto.ps
```

You can read HTML format document using Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser or any of the 10 other web browsers.

You can read the latex, LyX output using LyX a X-Windows front end to latex.

## 11. Copyright

Copyright policy is GNU/GPL as per LDP (Linux Documentation project). LDP is a GNU/GPL project. Additional restrictions are - you must retain the author's name, email address and this copyright notice on all the copies. If you make any changes or additions to this document then you should intimate all the authors of this document.