CPU Design HOW−TO

Table of Contents

CPU Design HOW−TO

Al Dev (Alavoor Vasudevan) alavoor@yahoo.com

v11.0, 26 Feb 2001

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 GNU/GPLed CPU running Linux, Unix, Microsoft Windows, Apple Mac and BeOS operating systems!!

---

1. Introduction

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

- 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
- 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
- HAL−Fujitsu (California) Super−Sparc 64−bit processor http://www.hal.com also compatible to Sun's sparc architecture.
- 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 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 are −− Taiwan Semiconductor Manufacturing Co. (TSMC), United Microelectronics Corp. (UMC), and Chartered Semiconductor Manufacturing Ltd. Pte.−−collectively account for 69% of today's silicon foundry volume, but their share is expected to grow to 88% by 2004.

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
- 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.linuxdoc.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. 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

8. Related URLs

Visit following locators which are related −

- Color Vim editor http://metalab.unc.edu/LDP/HOWTO/Vim−HOWTO.html
- Source code control system http://metalab.unc.edu/LDP/HOWTO/CVS−HOWTO.html

- Linux goodies main site http://www.aldev.8m.com and mirrors at webjump, angelfire, geocities, virtualave, bizland, theglobe, spree, infoseek, bcity, 50megs, NBCi, Terrashare, Fortunecity, Freewebsites, Tripod

9. Other Formats of this Document

This document is published in 12 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 and SGML.

- You can get this HOWTO document as a single file tar ball in HTML, DVI, Postscript or SGML formats from − ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other−formats/ and http://www.linuxdoc.org/docs.html#howto
- Plain text format is in: ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO and http://www.linuxdoc.org/docs.html#howto
- Single HTML file format is in: http://www.linuxdoc.org/docs.html#howto
- Translations to other languages like French, German, Spanish, Chinese, Japanese are in ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO and http://www.linuxdoc.org/docs.html#howto 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 − http://www.sgmltools.org Compiling the source you will get the following commands like

- sgml2html CPU−Design−HOWTO.sgml (to generate html file)
- sgml2rtf CPU−Design−HOWTO.sgml (to generate RTF file)
- sgml2latex CPU−Design−HOWTO.sgml (to generate latex file)

LaTeX documents may be converted into PDF files simply by producing a Postscript output using **sgml2latex** (and dvips) and running the output through the Acrobat **distill** (http://www.adobe.com) command as follows:

```
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.

This howto document is located at −

- http://sunsite.unc.edu/LDP/HOWTO/CPU−Design−HOWTO.html

Also you can find this document at the following mirrors sites −

- http://www.caldera.com/LDP/HOWTO/CPU−Design−HOWTO.html
- http://www.WGS.com/LDP/HOWTO/CPU−Design−HOWTO.html
- http://www.cc.gatech.edu/linux/LDP/HOWTO/CPU−Design−HOWTO.html
- http://www.redhat.com/linux−info/ldp/HOWTO/CPU−Design−HOWTO.html
- Other mirror sites near you (network−address−wise) can be found at http://sunsite.unc.edu/LDP/hmirrors.html select a site and go to directory /LDP/HOWTO/CPU−Design−HOWTO.html

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.

10. 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.