CPU Design HOW-TO

Al (Alavoor Vasudevan) Dev

alavoor (at) yahoo.com

도운이

박종수

clhitter (at) bawi.org

송민기

tempest (at) bawi.org

v10.0, 9/Feb/2001

CPU는 컴퓨터의 "두뇌"이고, 컴퓨터 시스템의 핵심 부품이며, Linux나 Windows 등의 운영체제들과는 "사촌 형제"와 같은 존재이다. CPU를 설계하고 제조할 여러 사람들에게 좋은 참고자료가 될 것이라고 생각한다. 컴퓨터 공학을 공부하는 대학생들에게도 유용하게 쓰일 수 있다. CPU가 어떻게 설계되고 제조되는지를 학생들이 이해하는데 많은 도움을 줄 수 있는 URL 링크들을 포함하고 있다. 아마도 빠른 시일내에 Linux, Unix, Microsoft Windows, Apple Mac, BeOS 등의 운영체제들이 구동되는 GNU/GPLed CPU가 나올 것이다!!

고친 과정

고침 번역 1.12001년 3월 20일고친이 박 종수, 송 민기

고침 v10.0 2001년 2월 9일 고친이 Al Dev

1. 소개

이 문서에는 CPU 설계와 제조에 관한 방대한 URL 리스트들이 포함되어 있다. 여러 사람들이 Linux나 Unix 운영체제를 구동할 수 있는 새로운 CPU를 만드는데 좋은 참고자료가 될 것이라고 생각한다.

예전에는, 칩 제조회사들이 IP 개발자이이면서 EDA tool 개발자 역할까지 하였다. 하지만, 최근에 들어, 우리는 전문화된 회사들 (TSMC http://www.tsmc.com), IP 전문 회사들 (ARM http://www.arm.com, MIPS http://www.mips.com, Gray Research LLC http://cnets.sourceforge.net/grllc.html), tool 전문 개발사들 (Mentor http://www.mentor.com, Cadence http://www.cadence.com, etc.), 그리고 이들을 모두 가지고 있는 회사(Intel)를 발견할 수 있다. 여러분은 IP를 하드웨어에 포함된 형태로 살 수도 있고 (Intel), 툴과 같이 살 수도 있고 (EDA companies), IP 만을 개별적으로 구입할 수도 있다 (IP providers).

FPGA [1] 제조 회사들 홈페이지 (Xilinx http://www.xilinx.com, Altera [2] http://www.altera.com). 이 사이트들을 방문해보는 것은 독특한 사업 모델을 포착할 수 있는 좋은 기회가 될 것이다.

VA Linux systems http://www.valinux.com 은 시스템 전체를 구축하고 있으며, 가까운 시일내에 Linux 용의 CPU를 설계하고 제조할 수 있을 것으로 보인다.

아래의 CPU 설계 사이트들을 방문해 보라:

- FPGA CPU 링크들 http://www.fpgacpu.org/links.html
- FPGA 메인 사이트 http://www.fpgacpu.org
- OpenRISC 1000 (시장을 독점하고 있는 ARM과 MIPS와 경쟁중인 공짜 오픈 소스 32-bit RISC [3] processor) IP core http://www.opencores.org
- 오픈 IP 단체 http://www.openip.org
- Free IP 단체 - 대중을 위한 ASIC과 FPGA core들 http://www.free-ip.com

2. IP란 무엇인가?

IP란 무엇인가? IP는 **Intellectual Property**의 약자이다. 좀더 구체적으로 말하면, ASIC [4] 이나 FPGA를 만들 때 사용될 수 있는 논리 회로 블럭을 의미한다. "IP Cores"의 예로는, UART, CPU, Ethernet 콘트롤러, PCI 인터페이스 등이 있다. 예전에는, 이들 제품의 질좋은 core들은 가격이 미화 5000 달러에서 350000 달러에 달했다. 이것은 아무리 신중하게 계획된 용도일지라도 일반적인 사업체나 개인들에게는 너무나 큰 부담이었다 -- 그리하여, Free-IP 프로젝트가 시작되었다.

초창기 Free-IP 프로젝트는 CPU나 Ethernet 콘트롤러와 같은 복잡한 용도의 core 들에 초점을 맞추었다. 좀더 간단한 core들에 대한 개발은 뒤따라 이루어질 것이다.

Free-IP 프로젝트는 누구나가 사용할 수 있는 수준 높은 IP를 만들고자 하는 노력이다.

IP core들에 관한 다음 사이트들을 방문해 보라 -

- 오픈 IP 단체 http://www.openip.org
- Free IP 단체 - 대중을 위한 ASIC과 FPGA core들 http://www.free-ip.com
- FPGA 메인 사이트 http://www.fpgacpu.org

2.1. Free CPU List

사용가능하거나 현재 개발중인 Free CPU List이다 -

- F-CPU 64-bit Freedom CPU http://www.f-cpu.org , 미러 사이트 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.atme1-wm.com/products (http://www.atmel-wm.com/products) click on Aerospace=>Space=>Processors
- Sayuri http://www.morphyplanning.co.jp/Products/FreeCPU/freecpu-e.html , manufactured by Morphy Planning Ltd http://www.morphyone.org , 특징 http://ds.dial.pipex.com/town/plaza/aj93/waggy/hp/features/morphyone.htm
- OpenRISC 1000 (시장을 독점하고 있는 ARM과 MIPS와 경쟁중인 공짜 오픈 소스 32-bit RISC processor) IP core http://www.opencores.org/cores/or1k
- OpenRISC 2000 http://www.opencores.orgnews:comp.lang.vhdl
- Green Mountain - GM HC11 CPU Core http://www.gmvhdl.com/hc11core.html

2.2. 상업용 CPU 리스트

- 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
- Hitachi SH4,3,2,1 CPUs http://semiconductor.hitachi.com/superh
- NVAX CPUs http://www.research.compaq.com/wr1/DECarchives/DTJ/DTJ700 (http://www.research.compaq.com/wrl/DECarchives/DTJ/DTJ700) and at mirror-site (http://www.digital.com/info/DTJ700)
- Univ. of Mich High-perf. GaAs Microprocessor 프로젝트 http://www.eecs.umich.edu/UMichMP
- Hyperstone E1-32 RISC/DSP [5] 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
- CPU Info center - SPARC과 ARM과 같은 CPU 리스트 http://bwrc.eecs.berkeley.edu/CIC/tech

3. CPU 박물관과 실리콘 동물원

CPU 박물관

- Intel CPU 박물관 http://www.intecl.com/intel/intelis/museum (http://www.intel.com/intel/intelis/museum)
- Intel - 마이크로 프로세서의 역사 http://www.intel.com/intel/museum/25anniv
- 컴퓨팅에 관한 가상 박물관 http://vlmp.museophile.com/computing.html
- 실리콘 동물원 http://micro.magnet.fsu.edu/creatures/index.html
- Intel - 마이크로 프로세서가 어떻게 동작하는가? http://www.intel.com/education/mpuworks
- 마이크로 프로세서에 관한 간단한 교육과정 http://www.hkrmicro.com/course/micro.html

3.1. 어떻게 트랜지스터가 동작하는가?

마이크로 프로세서(Microprocessor)는 우리들이 매일 사용하는 TV, 자동차, 라디오, 각종 가전제품들 (컴퓨터는 말할 것도 없다)과 같은 많은 제품의 핵심 부품이다. 트랜지스터(Transistor)는 이러한 마이크로 프로세서의 중심 부품이다. 트랜지스터는 아주 간단하게 보일지 모른다. 하지만, 트랜지스터 개발에는 수십년의 고통스런 연구과정이 필요했다. 트랜지스터가 개발되기 전에는, 컴퓨터에서의 정보처리를 위해 비효율적인 진공관과 기계적인 스위치들이 사용되었다. 1958년, 몇몇 공학자들 (그들 중 한명은 Intel의 설립자 Robert Noyce였다)은 두개의 트랜지스터를 하나의 실리콘 결정(silicon crystal)에 넣는데 성공하였고, 마이크로 프로세서로의 길을 열어준 최초의 집적 회로(integrated circuit)를 창조하였다.

트랜지스터는 아주 작은 전자 스위치이다. 이들이 컴퓨터의 두뇌 역할을 하는 마이크로 프로세서를 구성하는 단위가 된다. 일반적인 스위치들과 마찬가지로, 트랜지스터는 On/Off의 두가지 동작 상태를 갖는다. 트랜지스터의 On/Off, binary 동작 특성이 컴퓨터의 정보 처리를 가능케 한다.

어떻게 조그만 전자 스위치들이 동작하는가?:

컴퓨터가 이해할 수 있는 유일한 정보는 On/Off로 스위치 되는 전기 신호뿐이다. 트랜지스터를 알기 위해서는, 전자 회로들(electronic circuit)이 어떻게 스위칭이 되는지를 이해하는 것이 필요하다. 스위칭 전자회로는 두 부분으로 나누어 볼 수 있다. 하나는 전류가 흐를 수 있는 회로상의 길 (일반적으로 전선이 된다)이다. 다른 하나는 회로상의 길을 개방하거나 폐쇄함으로써 전류의 흐름을 시작시키거나 중단시킬 수 있는 스위치 device이다. 트랜지스터는 기계적인 움직임 없이 전기 신호에 따라 켜지고 꺼진다. 이런 트랜지스터의 특성은 마이크로 프로세서의 동작에 큰 이점을 제공한다.

3.2. 어떻게 트랜지스터가 정보를 처리하는가?

트랜지스터와 같이 두개의 상태만을 가지고 있는 것들을 binary라고 부른다. 트랜지스터의 켜진 상태를 1이라 표현하고 꺼진 상태를 0이라 표현한다. 여러개의 트랜지스터에 의해 생성된 1과 0의 연속된 패턴으로 문자, 숫자, 색과 그래픽 정보들이 표현될 수 있다. 이것이 binary 표현법이다.

3.3. binary 정보 표현

이름을 **binary**로 표기해 보라**:**

각 알파벳의 문자들을 binary로 표현할 수 있다. 아래는 JOHN이라는 이름과 그에 대응되는 binary 표현을 나타내고 있다.

```
J  0100 1010
O  0100 1111
H  0100 1000
N  0100 1110
```

그래픽, 오디오, 비디오와 같은 더 복잡한 정보들도 트랜지스터의 동작에 의해 binary 혹은 On/Off로 만들어질 수 있다.

아래의 알파벳과 그에 대응되는 binary 표현 도표를 참고하라.

표 **1.** 알파벳과 그에 대응되는 **binary** 표현 도표

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

3.4. 반도체(Semi-conductor)란 무엇인가?

도체(Conductor)와 부도체(insulator) :

금속과 같은 많은 물질들은 전류(electrical current)를 흘릴 수 있다. 이들이 도체(Conductor)이다. 전류를 흘리지 못하는 물질들은 부도체(insulator)라 불린다. 대부분 반도체(Semi-conductor)의 기본 재료가 되는 순수한 실리콘(pure silicon)은 불순물(impurity)의 주입 정도에 따라 전도율(conductivity)이 변조(modulation)될 수 있기 때문에 반도체라 불린다.

3.4.1. 트랜지스터의 해부학

반도체와 전류

실리콘에 적절한 종류의 불순물(impurity)을 주입함으로써 결정 구조를 변경시킬 수 있고, 전도율을 강화시킬 수 있다. 붕소(boron)계열의 불순물이 주입된 반도체를 p형 반도체라 한다 (p는 양전하나 정공(hole)을 뜻한다). 인(phosphorus)계열의 불순물이 주입된 반도체를 n형 반도체라 한다 (n은 음전하나 다수 자유전자(major free electron)를 뜻한다).

3.4.2. 트랜지스터의 작동

트랜지스터의 작동 - 트랜지스터의 On/Off 상태

트랜지스터는 source, gate, drain의 세 단자로 이루어진다.

n형 트랜지스터는 양으로 대전된 p형 실리콘 우물위에 음으로 대전된 source와 drain이 떠있는 구조로 되어있다. [6]

gate에 양전압이 가해지면, p형 실리콘의 전자들이 gate 아래의 영역으로 이끌려서 source와 drain 사이에 전자 채널이 형성한다.

drain에 양전압이 가해지면, 전자들이 source에서 부터 drain으로 휩쓸려가게 되어 트랜지스터가 켜진다.

gate의 전압이 제거되면, source와 drain 사이의 영역에 전자가 이끌리지 않게 되어 길이 끊어지고, 트랜지스터가 꺼진다.

3.4.3. 트랜지스터 효과

트랜지스터 효과 - 마이크로 프로세서가 우리의 삶을 어떻게 변모시키는가?

트랜지스터의 binary 특성은 마이크로 프로세서에게 워드 프로세싱이나 비디오 편집등의 다양한 작업을 수행할 수 있는 능력을 부여한다. 마이크로 프로세서는 하나의 칩에서 1초에 수십억개의 인스트럭션을 수행할 수 있을 정도로 발전하였다. 자동차, 의료 장비, 텔레비전, 컴퓨터에 우주 왕복선까지 마이크로 프로세서를 사용한다. 이것은 binary 정보를 처리하는 트랜지스터 덕분이다.

4. CPU 설계와 구조

4.1. CPU 설계

아래 사이트들에는 CPU 설계에 관한 정보들이 있다.

- Hamburg University VHDL [7] archive http://tech-www.informatik.uni-hamburg.de/vhdl
- Kachina Design tools http://SAL.KachinaTech.COM/Z/1/index.shtml
- FPGA-based 컴퓨터 리스트 http://www.io.com/~guccione/HW_list.html
- SPARC International http://www.sparc.com
- 자신만의 CPU 설계하기 http://www.spacetimepro.com
- FPGA를 사용한 컴퓨터 설계 교육 http://www.fpgacpu.org
- 컴퓨터 조직론에 관한 기술 위원회 http://www.computer.org/tab/tcca
- VHDL에 관한 FAQ http://www.vhdl.org/vi/comp.lang.vhdl 또는 여기서 http://www.vhdl.org/comp.lang.vhdl
- 컴퓨터 조직론 FAQ http://www.esacademy.com/automation/faq.htm
- 컴퓨터 조직론 FAQ ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/arch
- VME Bus FAQ http://www.hitex.com/automation/FAQ/vmefaq
- SPEC 홈페이지 http://performance.netlib.org/performance/html/spec.html
- 리눅스 벤치마크 http://www.silkroad.com/linux-bm.html

4.2. 컴퓨터 조직론에 관한 온라인 교과서들

- 온라인 HTML 책 http://odin.ee.uwa.edu.au/~morris/CA406/CA_ToC.html
- Univ of Texas의 컴퓨터 조직론 : http://www.cs.panam.edu/~meng/Course/CS4335/Notes/master/master.html
- 수 체계와 논리 회로 : http://www.tpub.com/neets/book13/index.htm
- 디지탈 논리 : http://www.play-hookey.com/digital
- 플립플롭 : http://www.ece.utexas.edu/~cjackson/FlipFlops/web_pages/Publish/FlipFlops.html
- 인스트럭션 실행 사이클 : http://cq-pan.cqu.edu.au/students/timp1/exec.html
- 진리표 작성기 : http://pirate.shu.edu/~borrowsbr/Truth/Truth.html (http://pirate.shu.edu/~borowsbr/Truth/Truth.html)

- 공유 메모리에 관한 Overview : http://www.sics.se/cna/mp_overview.html
- 프로세서에서의 동시 진행 멀티 스레딩 : 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
- 조언 : 적응과 확장 가능한 분산 가상 메모리 구조 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
- 분산 컴퓨팅 : http://www.geocities.com/SiliconValley/Vista/4015/pdcindex.html
- Pisma 메모리 구조 : http://aiolos.cti.gr/en/pisma/pisma.html
- 공유 메모리 구조 : http://www.ncsa.uiuc.edu/General/Exemplar/ARPA
- 컴퓨터 조직론 교과서들 : http://www.rdrop.com/~cary/html/computer_architecture.html#book VLSI 설계 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. 컴퓨터 조직론 대학 강의 노트

- 고급 컴퓨터 조직론 http://www.cs.utexas.edu/users/dahlin/Classes/GradArch
- 컴퓨터 조직론 - 코스 레벨 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 - 컴퓨터 구조의 원리 : http://www.cs.rutgers.edu/~murdocca/POCA/Chapter02.html
- Brown Univ - http://www.engin.brown.edu/faculty/daniels/DDZO/cmparc.html
- Univ of Sydney - 디지탈 시스템 입문 : http://www.eelab.usyd.edu.au/digital_tutorial/part3
- Bournemouth Univ, UK 컴퓨터 시스템의 원리 : http://ncca.bournemouth.ac.uk/CourseInfo/BAVisAn/Year1/CompSys
- 가상 병렬 처리 기계 : 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 구조

아래의 사이트들에는 CPU 구조에 관한 정보들이 있다.

- 컴퓨터 조직론 : http://www.rdrop.com/~cary/html/computer_architecture.html VLSI 설계 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 CPUhttp://www.dcs.gla.ac.uk/~ian/project3/node1.html
- 어셈블리어 개념 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. CPU 설계에 관한 유스넷 뉴스 그룹들

- 컴퓨터 조직론 뉴스 그룹 news:comp.arch
- FPGA 뉴스그룹 news:comp.arch.fpga
- Arithmetic 뉴스 그룹 news:comp.arch.arithmetic
- Bus 뉴스 그룹 news:comp.arch.bus
- VME Bus 뉴스 그룹 news:comp.arch.vmebus
- embedded 뉴스 그룹 news:comp.arch.embedded

- Embedded piclist 뉴스 그룹 news:comp.arch.embedded.piclist
- Storage 뉴스 그룹 news:comp.arch.storage
- VHDL 뉴스 그룹 news:comp.lang.vhdl
- Computer Benchmarks 뉴스 그룹 news:comp.benchmarks

5. CPU 제조

CPU를 설계하고 테스트한 뒤에, CPU를 대량 생산하길 원할 수 있다. 이럴 때, 세계 곳곳의 반도체 공장들이 만족할 만한 경쟁력있는 가격을 제공해 줄 것이다. 미국, 독일, 영국, 일본, 한국, 중국 등에 이런 회사들이 있다.

TMSC (대만) 은 세계에서 "가장 큰 반도체 전문 제조 공장"이다. 여러 생산 방법들을 훑어 본뒤에, 10만개 단위 이상의 엄청난 대량 생산을 위해 가장 좋은 것을 선택할 수 있다.

5.1. 반도체 제조업의 방대한 규모!

반도체 제조 회사들은 기반 시설에 아주 집중적으로 투자를 하며, 수백만 달러가 들어가는 공장을 짓는다. 반도체 제조업은 2004년까지 70조 달러에서 360조 달러로 성장할 것으로 예상된다. (414% 증가율이다!!) 더 많은 집적 장치 제조사들(IDMs)들이 웨이퍼 가공 수용 능력을 늘리는 대신, 칩 생산 외주를 주는 방법을 채택하고 있다.

반도체 전문 제조 공장들은 현재 세계 반도체 생산량의 12%를 담당하고 있으며, 이 점유율은 2004년 까지 26%로 두 배 이상의 증가세를 보일 것으로 예상된다.

"Big Three"의 전문 반도체 제조 공장들(Taiwan Semiconductor Manufacturing Co. (TSMC), United Microelectronics Copr. (UMC), Chartered Semiconductor Manufacturing Ltd. Pte.)은 현재 반도체 제조 공장 생산 용량의 69%를 담당하고 있지만, 2004년 까지는 그 점유율이 88%까지 증가할 것으로 보인다.

5.2. CPU 제조

전 세계 적으로 수백개의 반도체 제조 공장들이 있다 (너무 많아서 열겨하기 힘들다). 다음은 그들중 일부 리스트이다 -

- TSMC (Taiwan Semi-conductor Manufacturing Co) http://www.tsmc.com, 회사 소개 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 [8] 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, (반도체 생산부) 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 Semi-conductor Co, Japan
- Mitsubhishi Semi-conductor Co, Japan
- Hyandai Semi-conductor, Korea http://www.hea.com
- Samsumg Semi-conductor, Korea
- Atmel, France http://www.atmel-wm.com

이외의 반도체 제조 주요 기업들을 아신다면, 리스트에 추가할 수 있도록 알려달라.

칩 제조 회사 리스트

- 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

요즘 Super computer를 위한 system의 방식은 대부분, RISC 기반의 Symmetric Multi-Processing(SMP) node들을 fast network로 연결하는 것에서 크게 벗어나지 않는데 이것은 구조상으로 자연스러운 발전 모습인 것 같다. 비교적 값싼 RISC processor와 규격화된 통신 software로 processor들을 연결하는 network product들 덕분에, vendor가 제공하는 complete system 대신 자체적으로 cluster computer를 구성할 수 있게 되었다.

다음 site에 Super Computer에 대한 정보가 더 있다.

- 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

Machine 자체에 대한 얘기를 하기 전에, 성능을 향상시키기 위해서 고안된 mechanism들을 살펴보는 것이 좋을 것 같다. 첫 번째로 hardware의 구조는 하나의 CPU 성능을 넘어서 computer system의 속도를 높이는 것의 가능성 여부까지도 결정하는 주된 요소이다. Hardware와 더불어 중요한 또 하나의 요소는, 주어진 hardware platform에서 사용될 효율적인 code를 생성하는 compiler의 성능이다. 대부분의 경우 hardware의 영향과 software의 영향을 분리하기는 힘들기 때문에 각각의 영향으로 인한 결과를 분석할 때는 주의해야 한다. 이 chapter에서는 주로 hardware의 구조에 대해서 살펴본다.

그동안 Flynn의 분류체계는 high-performance computer를 분류하는데 유용했다. 이 체계는 instruction과 data의 흐름을 어떻게 조작하는가에 따라서 4가지로 분류한다. 우선은 각 class를 간략하게 설명하고 나중에 좀더 자세히 살펴보도록 한다.

6.2. SISD machines

SISD machine은 하나의 CPU로 하나의 instuction 흐름을 serial하게 처리하는 system이다. 요즘 많은 대형 mainframe들이 두 개 이상의 CPU를 갖긴 하지만 각각의 instruction 처리 흐름은 서로 연관이 없다. 따라서 그런 것들도 (여러개가) 각자의 data space에서 동작하는 SISD machine으로 분류되는 것이 맞다. DEC, Hewlett-Packard, Sun Microsystems의 workstation 대부분이 SISD machine의 예라고 할 수 있다. 이 문서에서는 SISD machine의 정의만 알아두도록 하고 자세히 살펴보지는 않겠다.

6.3. SIMD machines

SIMD machine에서는 1,024에서 16,384개 까지의 많은 processing unit이 똑같은 instruction을 서로 다른 data에 대해서 lock-step으로 수행해서 한 개의 instruction이 많은 data를 병렬적으로 처리하게된다. SIMD machine의 예로는 CPP DAP Gamma II나 Alenia Quadircs 같은 것이 있다.

SIMD system의 하위 분류로 vector processor가 있다. Vector processor는 특별한 구조의 CPU들을 이용해서 하나의 data가 아니라 비슷한 data들의 array를 처리한다. data가 이렇게 vector 단위로 처리되면 결과를 clock cycle당 한 배나 두 배 -- 특별한 경우에는 -- 세 배의 비율로 얻을 수 있다. (system에서는 clock cycle을 내부 시간의 기본 단위로 정의한다.) 따라서 vector processor는 data를 거의 병렬적으로 처리할 수 있는데, 단 이것은 vector mode로 동작할 때 만이다. 이렇게 하면 일반적인 scalar mode 일 때보다 몇 배 빠르게 처리할 수 있다. 실제로는 vector processor를 대부분 SIMD machine으로 인식하며, 예로는 Hitachi S3600이 있다.

6.4. MISD machines

이론적으로 이런 유형의 machine은 하나의 data 흐름에 대해서 여러개의 instruction을 수행한다. 그렇지만 아직까지 이런 종류의 machine이 실제적으로 구현된 적이 없을뿐더러 이런 system은 생각하기도 쉽지 않다. 앞으로의 내용에서 이 유형은 제외하도록 한다.

6.5. MIMD machines

이런 종류의 machine은 여러개의 instruction을 여러개의 data에 대해서 병렬적으로 수행한다. 앞에서 나왔던 여러 개의 processor를 가진 SISD machine과의 차이점은, instruction이나 data가 수행할 작업의 서로 다른 부분을 나타낼 뿐 실제로는 하나의 작업으로 연관이 되어 있다는 점이다. 따라서 MIMD system에서는 여러 개의 sub-task를 병렬적으로 처리해서 결국 수행할 하나의 main task 결과를 얻는데 까지의 시간을 단축할 수 있다. 한편 MIMD system의 종류는 다양하기 때문에 Flynn의 분류체계가 적절하지 않은 면이 있다. 동작하는 방식이 서로 많이 다른, four-processor NEC SX-5와 thousand processor SGI/Cray T3E가 같은 종류로 분류되는 것이다. 그러므로 추가적인 구분을 두고 그것에 따라서 살펴보도록 하겠다.

6.5.1. Shared memory systems

Shared memory system에서는 여러개의 CPU가 하나의 address space를 공유한다. 이 방식에서는 모든 CPU들이 사용할 수 있는 memory가 한 개뿐이기 때문에 data가 저장된 곳에 대한 정보를 굳이 알고 있을 필요가 없다. Shared memory system은 SIMD일 수도 있고 MIMD 일 수도 있는데, Single-CPU vector processor는 전자의 예이고 multi-CPU model은 후자의 예이다. 앞으로는 SM-SIMD나 SM-MIMD라는 약어로 표현하는 경우도 있을 것이다.

6.5.2. Distributed memory systems

이 형태는 각자의 memory를 갖는 CPU들이 서로 network로 연결되어 필요할 때마다 각각의 memory로부터 data를 교환한다. Shared memory machine의 경우와는 달리 user는 원하는 data가 존재하는 위치를 알아야 하고 필요할 때마다 data를 옮겨 주어야 한다. Distributed memory system도 SIMD나 MIMD가 모두 가능하다. lock step으로 동작하는 SIMD system경우에는 processor마다 각자의 분산된 memory를 갖는다. Distributed memory MIMD system의 경우에는 연결되는 network의 topology가 다양한데, topology의 구체적인 정보를 user가 알 필요까지는 없기 때문에 이식성이 좋다. 앞으로는 distributed memory system에 대해서 DM-SIMD나 DM-MIMD라는 용어로 각각을 가리키도록 한다. Shared memory와 distributed memory는 분명히 구분되는 것처럼 보이지만 항상 그런 것은 아니다. 예를 들어, Kendall Square Research system에서는 hardware level에서 "virtual shared memory"라는 개념을 도입했다. Virtual shared memory는 programming level에서도 simulation이 가능해서, processor들에 data를 분산시키는 compiler의 지시 방법으로 High Performance Fortran(HPF) specification이 1993년에 발표되기도 했는데, HPF가 구현된 system은 user 입장에서는 shared memory machine으로 보이게 된다. HP나 SGI/Cray 같은 Massively Parallel Processor system(MPP system이라고도 한다)의 vendor 들도, 물리적으로 분산된 memory를 collective address space로 addressing 함으로써 virtual shared-memory programming model을 지원한다. 그렇게 해서 system의 모든 memory를 하나의

global address space로 사용할 수 있다. 그런 종류의 system에 대해서는 ccNUMA 부분에서 더 다루도록 한다. 참고로 TreadMarks 같은 package에도 workstation들의 network를 위해서 virtual shared memory environment를 제공한다.

6.6. Distributed Processing Systems

최근의 경향 중 하나는 분산 처리(Distributed Processing)인데 이것은 DM-MIMD에서 한 단계 더 나아간 개념이다. 여러 개의 intergrated processor를 몇 개의 box나 workstation, 또는 mainframe 등에 넣는 대신 (Gigabit) Ehternet이나 FDDI 등으로 연결해서 하나의 program 안에서 concurrent하게 수행되도록 하는 것이다. 개념적으로는 DM-MIMD computing과 별로 다를 것이 없지만 processor간의 통신 속도가 좀 느리다. 많은 package에서 distributed computing이 가능한데, 예를 들면 PVM(Parallel Virtual Machine)이나 MPI(Message Passing Interface)가 있다. 이런 "message passing" model이라고 불리는 방식의 programming이 널리 받아들여졌기 때문에 PVM이나 MPI는 대부분 major vendor의 distributed-memory MIMD에 채택되었고, 심지어 호환성을 위해서 shared-memory MIMD에도 적용되었다. 뿐만 아니라 high computational power를 얻기 위해서 HiPPI channel 등으로 shared-memory system을 cluster하는 경향도 있다. 예를 들면 NEC SX-5나 SGI/Cray SV1 등이 이런 구조를 갖는다. 따라서 clustered node 내에서 shared-memory programming style을 사용할 수 있다.

6.7. ccNUMA machines

앞의 내용에서 나왔던 것처럼, 비교적 적은 수의(16개 정도까지) RISC processor를 Symmetric Multi-Processing(SMP) node에 집적시키는 경향을 볼 수 있다. 그런 node 안의 processor들은 실제로는 값싼 network로 연결되어 있지만 가상적으로는 항상 1-stage crossbar로 연결되어 있는 것이다.

이것은 앞에서 나왔던 vector processor와 비슷한 방식이지만 모든 processor들이 전체 address space에 접근할 수 있다는 점에서 다르다. 따라서 그런 system은 SM-MIMD machine이라고 볼 수 있다. 그렇지만 memory가 실제로는 분산되어 있기 때문에 data access operation이 항상 일정한 시간내에 수행된다는 보장이 없다. 이런 system을 ccNUMA라고 하는데 ccNUMA는 Cache Coherent Non-Uniform Memory Access를 뜻한다. 여기서 "Cache Coherent"라는 것은 모든 CPU에 대해서 사용되는 변수가 일관된 값을 가져야 함을 말한다. 즉 이 변수들을 제공하는 cache가 서로 같은 변수값을 일관되게 가져야 하는데 CPU들의 cache가 consistent하도록 만드는 방법에는 여러 가지가 있다. 그 중 하나는 다른 CPU에서 전송중인 변수들을 유심히 관찰하고 있다가 자신이 가지고 있는 변수이면 그 값을 update하도록 하는 snoopy bus protocol을 사용하는 것이다. 그 외에 각 변수의 유효하도록 해주는 특별한 memory인 directory memory를 갖는 방법도 있다.

물리적으로는 memory가 분산되어 있지만 특별한 효과를 위해서 (directory memory처럼) hardware/software가 통합되어 있기 때문에 실제로는 이런 system을 SM-MIMD라고 볼 수도 있다.

7. Neural Network Processors

NN(Neural Network)은 생물학적인 신경망의 model이고, 물론 안 그런 부분도 있긴 하지만 역사적으로 볼 때 NN 분야 영감중의 많은 부분이, 인간의 두뇌가 일상적으로 수행하는 정교하고 지능적인 연산들이 가능한 인공 system을 만들어서 인간 두뇌에 대한 이해를 가능케 하려는 시도에서 왔다.

대부분의 NN은 어떠한 "training" 규칙에 따라서, data를 기반으로 하는 각 connection의 weight를 조절한다. 다시 말하면, NN은 예를 통해서 "배우고" (마치 아이들이 강아지의 예로부터 강아지를 인식하게 되는 것과 비슷하다.), 훈련된 data 이상의 일반화 능력을 보인다.

NN에서는 각 부분의 연산이 서로 독립적이기 때문에 뛰어난 병렬성을 잠재적으로 가진다. 그래서 어떤 사람들은 NN를 정의하는 특징으로 대규모의 병렬성과 높은 긴밀성을 들기도 하지만, 그런 요건들은 NN의 특별한 경우로 알려져있는 단순선형회귀(겨우 두 개의 unit과 bias로 이루어진 feedforward net)와 같은 simple한 model들을 배재시키는 면이 있다.

Neural Network(NN)의 정의로는 다음과 같은 것들이 있다.:

- DARPA에 의한 정의: Neural Network는 병렬적으로 동작하는 processing element들로 구성된 system이며 각 element의 기능은 network의 구조와 연결의 강도 그리고 각 node에서 수행되는 process로 정의된다.
- Haykin에 의한 정의: Neural Network는, 경험적인 지식을 저장하고 사용하는 경향을 지니는 거대한 병렬분산처리기이며 다음과 같은 두 가지 면에서 인간의 두뇌와 닮았다.:
  - Network는 learning process를 통해서 지식을 얻는다.
  - 지식을 저장하는 것은 synaptic weight라고 하는 neuron 간의 연결이다.
- Nigrin에 의한 정의: Neural Network는 신경을 기반으로하는 단순한 processing element들이 상당히 많이 모여서 이루는 회로이다. 각 element는 지역적인 정보만을 가지고 동작한다. 게다가 각 element는 비동기로 동작하기 때문에 system clock이 필요하지 않다.
- Zurada에 의한 정의: 인공 신경 system이라고도 하는 Neural Network는 물리적인 cellular system으로서 경험적인 지식을 습득하고 저장하며 이용할 수 있다.

다음의 site들에서 Neural Network Processor에 대한 정보를 더 얻을 수 있다.

- 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 (http://www.accurate-automation.com/Products/NNP.HTM) hardware

8. 관련된 곳의 URL들

다음은 이 문서와 관련된 곳의 주소이다.

- 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 와 mirror site인 webjump (http://aldev.webjump.com), angelfire (http://www.angelfire.com/nv/aldev), geocities (http://www.geocities.com/alavoor/index.html), virtualave (http://aldev.virtualave.net), bizland (http://aldev.bizland.com), theglobe (http://members.theglobe.com/aldev/index.html), spree (http://members.spree.com/technology/aldev), infoseek (http://homepages.infoseek.com/~aldev1/index.html), bcity (http://www3.bcity.com/aldev), 50megs (http://aldev.50megs.com), NBCi (http://members.nbci.com/alavoor), Terrashare (http://aldev.terrashare.com), Fortunecity (http://members.fortunecity.com/aldev), Freewebsites (http://aldev.freewebsites.com), Tripod (http://members.tripod.lycos.com/aldev)

9. 이 문서가 제공 되는 형식

이 문서는 다음과 같은 12가지의 다른 형식으로 제공된다. - DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU-info, HTML, RTF(Rich Text Format), Plain-text, Unix man pages, single HTML file, SGML.

- 다음 주소로부터 이 HOWTO 문서의 HTML, DVI, Postscript나 SGML 형식을 얻을 수 있다. - ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/ 와http://www.linuxdoc.org/docs.html#howto
- Plain text 형식의 문서는 다음 주소로부터 얻을 수 있다: ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO 와http://www.linuxdoc.org/docs.html#howto
- HTML file 형식의 문서는 다음 주소에서 얻을 수 있다: http://www.linuxdoc.org/docs.html#howto
- 프랑스어, 독일어, 스페인어, 중국어, 일본어로 번역된 문서를 다음에서 얻을 수 있다. ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO 와http://www.linuxdoc.org/docs.html#howto 다른 언어로 번역하는데 도움을 주는 것은 언제라도 환영이다.

이 문서는 "SGML-Tools"을 이용해서 작성되었고 그 tool은 다음 주소에서 얻을 수 있다 - http://www.sgmltools.org source를 compile하려면 다음의 명령을 준다.

- sgml2html CPU-Design-HOWTO.sgml (html file로 만들때)
- sgml2rtf CPU-Design-HOWTO.sgml (RTF file로 만들때)
- sgml2latex CPU-Design-HOWTO.sgml (latex file로 만들때)

sgml2latex (그리고 dvips) 명령을 사용해서 LaTeX 형식의 문서를 Postscript로 출력한 다음 Acrobat의 **distill** (http://www.adobe.com)을 사용하면 PDF 형식의 문서로 변환 할 수 있다. 그 과정은 다음과 같다:

```
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 &
```

아니면 Ghostscript의 **ps2pdf** 명령을 사용해도 된다. ps2pdf 는 Adobe사의 Acrobat Distiller의 기능을 거의 모두 지원해서 PostScript를 Portable Document Format(PDF)로 변환해준다. **ps2pdf**는 간단한 command script(batch file)로 구현되어 있고 **pdfwrite**를 특별한 "output device"로 선택해서 Ghostscript를 불러낸다. ps2pdf를 사용하려면 Ghostscript를 compile할 때 makefile에 pdfwrite device를 포함시켜야 한다. Ghostscript building에 대한 자세한 내용은 관련 문서를 참고하기 바란다.

관련된 howto 문서는 다음 주소에서 얻을 수 있다. -

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

mirror site의 주소는 다음과 같다. -

- 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
- 다른 mirror site에 대한 정보는 다음 주소에서 볼 수 있다. http://sunsite.unc.edu/LDP/hmirrors.html site를 선택하고 /LDP/HOWTO/CPU-Design-HOWTO.html를 보면 된다.

dvi 형식의 문서를 보려면 xdvi program을 사용하면 된다. xdvi program은 Redhat Linux에서는 tetex-xdvi*.rpm package에 있고 ControlPanel | Applications | Publishing | TeX menu에서 실행할 수 있다. xdvi에 대한 문서를 보려면 다음의 명령을 실행한다.-

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

window의 크기는 마우스로 조정할 수 있다. 화살표, Page Up, Page Down 등의 key를 사용해서 이동할 수 있고 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n'를 이용해서도 move up, down, center, next page, previous page 등의 기능을 사용할 수 있다. expert menu를 끄려면 'x'를 누른다.

postscript 형식의 file은 'gv'(ghostview)나 'ghostscript' 같은 program을 이용해서 볼 수 있다. ghostscript와 gv는 Redhat Linux에서 각각 ghostscript*.rpm과 gv*.rpm package에 있고 ControlPanel | Applications | Graphics menu에서 실행시킬 수 있다. gv가 ghostscript보다 사용하기 쉽다. 그리고 ghostscript나 gv 모두 OS/2, Windows 95, Windows NT 등의 다른 OS에서도 사용할 수 있다.

- Windows95나 OS/2, 그리고 그 외의 OS를 위한 ghostscript를 다음 주소에서 구할 수 있다. http://www.cs.wisc.edu/~ghost

postscript 문서를 보려면 다음의 명령을 입력한다. -

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

HTML 문서는 Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser 등 10 종류 이상의 web browser에서 읽을 수 있다.

X-Windows에서의 latex front end인 LyX를 사용하여 latex나 LyX 출력물을 읽을 수 있다.

10. Copyright

이 문서의 저작권은 LDP(Linux Documentation Project)로서 GNU/GPL을 따른다. LDP는 GNU/GPL project이다. 그 외에도 저자의 이름과 email 주소 그리고 이 저작권에 대한 안내문을 문서에 첨부해야 한다. 그리고 수정이나 보완을 하게되면 이 문서의 저자들에게 알려야 한다.

주석

1. Field Programmable Gate Array, Xilinx에서 만드는 프로그램 가능한 논리회로
2. MAX, FLEX series등의 PLD(Programmable Logic Device)를 만드는 회사
3. Reduced Instruction Set Computing. Instruction 수를 줄이고 register의 수를 늘린 CPU design 방식
4. Application Specific Integrated Circuit Circuits. 주문제작형 반도체
5. Digital Signal Processor : 디지탈 신호 처리용으로 특화된 CPU. TI의 TMS320 series가 대표 제품
6. 이는 FET(Field Effect Transistor) 트랜지스터의 구조에 대한 설명이다. 트랜지스터의 종류에는 크게 BJT(Bipolar Junction Transistor)와 FET가 있으며, 현재의 집적회로에는 FET 중에서도 MOSFET(Metal-Oxide-Semiconductor FET)가 주로 사용되고 있다.
7. Very high speed integrated circuit Hardware Description Language
8. Ball Grid Array, 칩의 아랫면에 반구 모양으로 핀들을 배치하여 칩의 크기를 줄일 수 있게 하는 칩 패키징 방식