

# System Design Frontier

## Editorial Board

Carl Pixley Senior Scientist, Synopsys Inc

Masahiro Fujita, Professor, The University of Tokyo

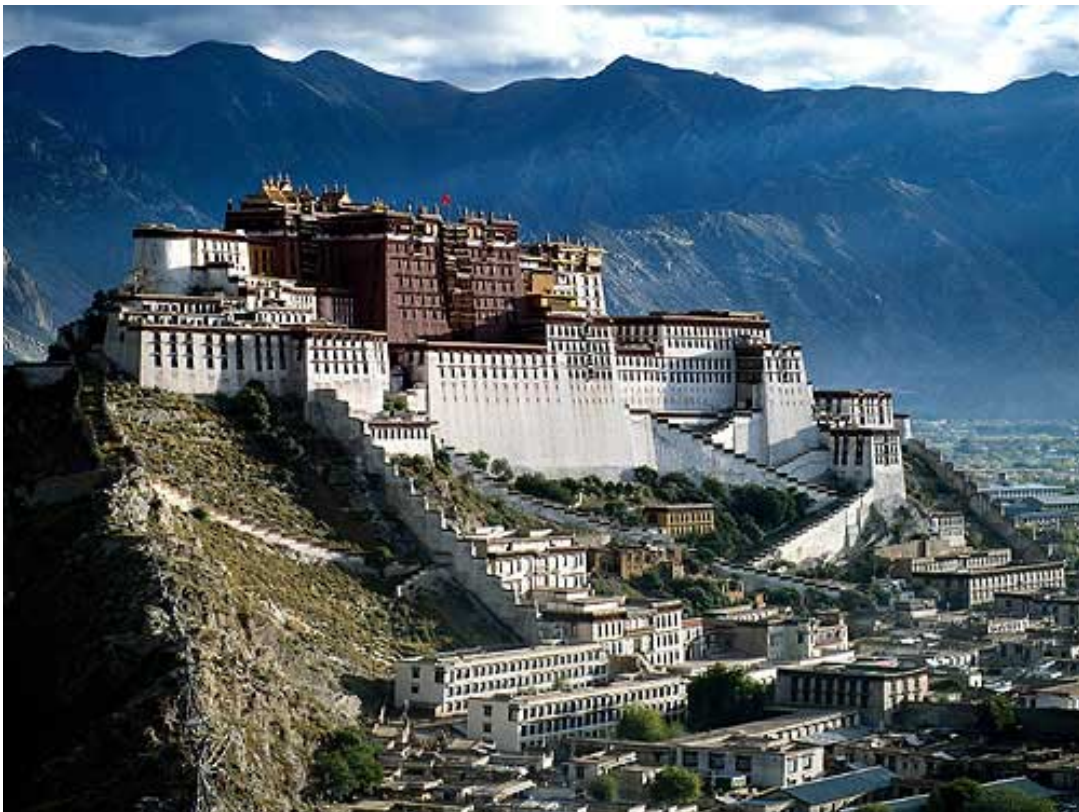
Michael McNamara, Senior VP of Technology, Verisity Ltd

Gabriele Saucier, Chairman of Design and Reuse Inc, IEEE Fellow

Wayne Wolf, Professor, Princeton University, IEEE Fellow

## *Hometown Outsourcing and BPO*

*Off-shore EDA Software Development & Technical Support  
Search Engine Optimization & Pay Per Click for Online Marketing*



The Advancing Ubiquitous Society – Revealing  
What's Next, Changing Tomorrow

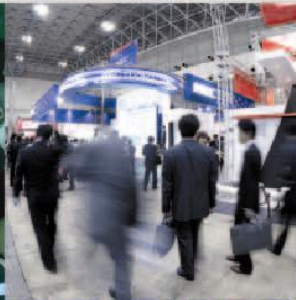
Providing Image, Information and Communications

**CEATEC<sup>®</sup>**  
**JAPAN**

Tuesday, October 4 - Saturday, October 8, 2005 (5 days)  
Makuhari Messe (Nippon Convention Center)



**Combined Exhibition of  
Advanced Technologies**



**SPONSORSHIP**  
CEATEC JAPAN Organizing Committee

**JEITA**

Japan Electronics and Information Technology Industries Association (JEITA)

**CIAJ**

Communications and Information network Association of Japan (CIAJ)

**JPSA**

Japan Personal Computer Software Association (JPSA)

For more information, Please visit our web site!

**[www.ceatec.com](http://www.ceatec.com)**



# 上海國際工業博覽會

## SHANGHAI INTERNATIONAL INDUSTRY FAIR

2005信息化与工业化(现代装备)  
INFORMATIZATION & INDUSTRIALIZATION 2005 (Modern Equipment)

### 电子信息展 Information Technology Equipment Show



#### Organizers

National Development and Reform Commission  
Ministry of Commerce  
Ministry of Science and Technology  
Ministry of Information Industry  
Ministry of Education  
Chinese Academy of Science  
China Council for the Promotion of International Trade  
Shanghai Municipal People's Government

#### Co-organizer

China Machinery Industry Federation

#### Show Management

Shanghai World Expo (Group) Co., Ltd.

#### Media Promotion

VOGEL EUROTRADE

**Date** November 4-9, 2005

**Venue** Shanghai New International Expo Centre

#### Exhibit Profile

- Telecommunication and Network
- Electronic Components and Devices
- Computers and Control Systems

#### Contact Us

Shanghai World Expo (Group) Co., Ltd.  
Department of SIF  
Tel: +86-21-63539977 Fax: +86-21-63542582  
E-mail: ivylu@itpc.com.cn Lucylu@itpc.com.cn

**Web:** <http://www.sif-expo.com>



**Oct. 19-22 2005**

**The Leading Electronic IT Exhibition in China**

**Registration starts now !**

**[www.goemex.com](http://www.goemex.com)**

- **Directed by:**  
Ministry of Commerce, PRC  
Ministry of Information Industry, PRC  
Taiwan Affairs Office, State Council  
Jiang Su Provincial Government
- **Organized by:**  
Suzhou Municipal Government, JiangSu Foreign Trade & Economic Relations Commission  
JiangSu Information Industry Department, JiangSu Provincial Government Taiwan Affairs Office
- **Co-Organized by:**  
Taipei Computer Association (TCA)
- **Executed by:**  
Bridge Exhibition Services (BES)
- **Supported by:**  
China Chamber of Commerce for Import and Export of Machinery and Electronic Products  
Center of Strait Economy and Science Tech Cooperation  
Taipei Electronic Components Suppliers' Association (TECSA), Taiwan Electronic Connectors Association (TECA)  
Taiwan Mold Die Industry Association (TMDIA), Photonics Industry Technology Development Association (PIDA)
- **Date:**  
Oct. 19th ~ 22nd, 2005
- **Time:**  
9:00am ~ 17:00pm (Last day open till 15:00pm only)
- **Venue:**  
Suzhou International Expo Center (Expo Plaza, Suzhou Industrial Park)
- **Scale:**  
Approximately 500 exhibitors occupying 1,500 booths
- **Exhibitors:**  
Electronic related enterprises from Mainland China, Taiwan, Hong Kong, Japan, Korea, Singapore, European and American electronic enterprises
- **Exhibition Product Categories:**  
1. 3C Products and Peripherals; 2. Electronic Components; 3. Technical Equipment and Services

**eMEX CHINA** SUZHOU  
electronic Manufacturer EXpo

Bridge Exhibition Services (BES)  
Taipei Computer Association (TCA)

[www.goemex.com](http://www.goemex.com)  
[www.tca.org.tw](http://www.tca.org.tw)

Tel: +86-512-6808 3188  
Tel: +886-2-2577 4249

Fax: +86-512-6808 3288  
Fax: +886-2-2578 5392



2005第九届中国国际软件博览会  
**INT'L SOFT CHINA 2005**

**Registration: Dec. 2004 - April. 2005**

**Exhibition, Forums & Business Meetings**

**Come Here - We Introduce China Software Market to the World**

**Join Hands - Let's Explore Software Markets together**

**We Invite Over 20 Countries and Regions.**

**We Invite Global TOP 500.**

**We Invite China TOP 500.**

**We Invite Government & Industrial Buyers.**

**Sponsor: Ministry of Information Industry**

**Organizers: China Software Industry Association**

**China Information Industry Trade Association**

**China Information World**

**China International Trade Center**

**Co-organizers: [www.sohu.net](http://www.sohu.net) [chanye.sina.net](http://chanye.sina.net) [www.ccidnet.com](http://www.ccidnet.com) [www.blogchina.com](http://www.blogchina.com)**

**Time: June 23-25, 2005**

**Venue: China International Exhibition Center, Beijing, P.R.China**

**Enquiry: 8610-62143871, 8610-51527167, 8610-62186579(F)**

**Visit : [www.csia.org.cn](http://www.csia.org.cn)**

**E-mail : [csia@css.com.cn](mailto:csia@css.com.cn)**

# Content

<i>Editorial formation of INformation</i> <i>Editor-in-Chief</i> .....	7
<i>Power Optimization Within Nanometer Designs</i> <i>Dr. Danny Rittman</i> .....	9
<i>Epistemologically Multiple Actor-Centred System:</i> <i>Or, EMACS at work!</i> <i>Yuwei Lin</i> .....	25
<i>Single-Chip Digitally Controlled Data-Acquisition as Core of</i> <i>Reliable DWDM Communication Systems</i> <i>Mark Malae</i> .....	35
<i>Towards Next Generation URLs</i> <i>Thomas A. Powell and Joe Lima</i> .....	41
<i>Versatile Mixed-Signal Front Ends Speed Customized Design of</i> <i>Wireline Broadband Modems and Home Networks</i> <i>Joseph DiPilato, Iuri Mehr, Brian Harrington</i> .....	49
<i>A Natural Model for EDA Sensible IP Behavior</i> <i>Melanie Yunk, Silicon Integration Initiative</i> .....	59
<i>The daemon, the gnu, and the Penguin</i> <i>A History of Free and Open Source</i> <i>Peter H. Salus</i> .....	70

**Copyrights** Copyright © 2004 ~ 2005, Shanghai Hometown Microsystems Inc (“Publisher”). All rights reserved. No part of System Design Frontier (“Publication”) may be reproduced, transmitted, or translated, in any form or by any means, electronics, mechanical, manual, optical, or otherwise, without prior written permission of Publisher.

**Disclaimer** Publisher makes no warranty or any kind, express or implied, with regard to this Publication

## The formation of INformation Editor-in-Chief

In my one of previous editorials I recalled that, the first time I used the Internet was using the only email account in my colleague campus to contact with my first advisor abroad, it was not free, and that was in Beijing in later 1993. A year later, while I was attending my first graduate school, I got to know the first Internet browser Mosaic as well as its commercial version Netscape, all a sudden, everyone was eager to create their own homepages. In less than two years, while I was attending another graduate school, yahoo! went to public, and all the people surrounding me were discussing the Internet mania, at that time both Amazon and eBay were still startups, and google was not even born yet. As a graduate student, searching relevant cited papers was my daily routine since it is part of research business, unfortunately, both search engines and directories at that time were not that as efficient as that of the-state-of-the-art. In other words, the information was not as that rich as that of today, and for the available scarce information online, they were neither easy to locate, nor quick to retrieve.

Ten years passed, where are we here now and where can we get from here? Information now can be retrieved by anyone at anytime, anywhere in anyway. You can access the Internet, not only from either office or home, but also either at the airport or even on the road. The Internet now is not only rich in images, but also audios and videos beyond plain texts. PC is not the only Internet access device any more, as the merging of computation, communication and consumer electronics, Internet ready mobile phones, power-line, even the freezer at your kitchen are becoming more and more popular. Let's take China as an example, in China, the number of Netizen is 87 million as of June of 2004 according to an official report released by CNNIC; on the instant messaging side, the market leader NASDAQ listed Tencent owns 320 million registered members through its QQ platform, an IM system similar to MSN and yahoo! Messenger; on the mobile communication side, two hundred million Chinese possesses at least one mobile phone, and the number of short messages sent by Chinese in the first quarter of 2005 is 217 billion, according the Minister of Information Industry Ministry in China. We can see in the beginning of this century, now the key problem is not how to deal with information deficiency problem, instead it is how to tackle the challenge of information overflow

According to my observation, the Internet is just like a jet, which can make your information journey faster and cheaper, now without jet, can people still travel? Of course, except you are one of the few highly busy tycoons in the Silicon Valley, you do not need a jet (suppose you can afford that) to get to work. The key concern for you is how to choose most suitable car and find a most efficient route to reduce your commute time. The choice of such a car and such a route, here at Hometown Microsystems we call it the *formation of INformation*

HITACHI  
Inspire the Next

您投入巨资的IT系统发挥应有作用了吗？

您的企业是否已投入巨资构建了庞大的IT系统,并把它当作高水准服务、高效业务运作和企业优良发展的基础?

但您是否察觉到随着IT系统的日益庞大和复杂,传统的人员维护已无法满足,而导致IT系统出现非良好运作,使得它的作用没有得到充分的发挥,产生了巨大的损失?

这个时候您就需要划时代的日立JP1系统运行管理软件,来帮助您有效管理整个IT系统。

JP1系统运行管理软件由7个模块组成,具备高可靠性,易导入性,易操作性等特点,从四大方面支持企业的IT系统:

- (1) 提高IT系统的运行质量,提供24小时×365天的高可靠性服务;
- (2) 减少IT系统的维护工作量和维护费用;
- (3) 提高IT环境的性能和实用性;
- (4) 减少IT环境潜在的风险。

追求高效无止境

- 在日本拥有20,000多家用户,具有丰富的IT系统运行管理经验,能够根据企业特点进行综合和提供特定解决方案。
- 在日本销售量(97-02年连续六年)及客户满意度(99年)名列前茅。
- 被广泛应用于金融、制造、电子商务等众多行业。



JP1

系统运行管理软件

日立信息系统(上海)有限公司 021-64728068

如果您需要了解更多详细资料请登陆<http://www.hitachi.com/jp1> 北京分公司 010-64670211 杭州分公司 020-38703586

认证业务伙伴: ePRO 鼎宝电脑系统(北京)有限公司 电话:010-64689391 线213 或512

Shi ech 北京宇信鸿泰科技发展有限公司 电话:010-85268708



# Power Optimization Within Nanometer Designs

Dr. Danny Rittman  
danny@tayden.com

## Abstract

As the integrated circuits (ICs) are scaled into deep nanometer dimensions and operate in giga-hertz frequencies, power optimization is becoming critical in determining system performance and reliability. Power dissipation is becoming one of the most challenging design constraints in nanometer technologies. For example, among various design implementation schemes, standard cell ASICs offers one of the best power efficiency for high-performance applications. The flexibility of ASICs allow for the use of multiple voltages and multiple thresholds to match the performance of critical regions to their timing constraints, and minimize the power everywhere else. Typically, implementing nanometer-scale ICs begins and ends with wires. The ability of an IC to perform its function is dependent upon the transformation of that function into a specific configuration of wires and their connections to cell-level and, ultimately, to transistor-level behaviors. This paper discusses nanometer power aware design, optimization and management that operate upon a behavioral design description.

## Introduction

The steady downscaling of transistor dimensions over the past two decades has been the main reason to the growth of silicon integrated circuits (ICs) and the electronic industry. The more an IC is scaled, the higher becomes its packing density, the higher its circuit speed, and the lower its power dissipation. These have been the key in the evolutionary progress leading to today's computers and communication systems that offer superior performance, dramatically reduced cost per function, and much-reduced physical size compared to their predecessors. As ICs marched down the technology path from 0.13 to 90 nm, designers and their customers enjoyed faster and more complex IC, without a significant increase in power consumption. But at 90 nm and below, the cost of developing nanometer designs significantly increased, particularly in masks and NRE costs. In addition, more designs are targeted for mobile applications running off batteries resulting in new design constraints for both dynamic and static power. Finally, the physics of the silicon (power related) started to work against engineering. However, the increase in leakage current was most critical. Thinner gate oxides and lower threshold voltages result in a significant increase in leakage current and static power consumption. There are, in total, seven additional secondary sources of leakage current, all of which get worse at 130 nm and even more so at 90 nm and 65 nm. We begin with an overview of a power fundamentals and behavioral design methodology. We will continue with a description of how power issues interact with the nanometer design space and describes how some of these issues can be influenced early in the design cycle. The industry direction is towards behavioral design that is becoming an absolute necessity in order to address the many nearly intractable issues that rise in nanometer design. In 2004,

consumer electronics, mostly mobile, battery-powered applications is one of the major driving forces in the growth of fabless IC companies. Gartner/Dataquest forecasts that global revenue from semiconductors used in consumer electronics will increase by 20% in 2004 (\$34.1 billion) and even more in 2005. Many of these devices will go into mobile and low power applications. No Doubt, the market is demanding an urgent solution to the nanometer power crisis.

### Power Fundamentals

When discussing power fundamentals the two subjects to explore are two main effects that contribute to the total power dissipation on a chip. These effects are dynamic power and leakage power. Dynamic (also called active) power occurs when a transistor switches state and is due to capacitive charging and discharging. Leakage power arises due to leakage (also called static) current flowing through the transistor. Leakage current can even consume power in standby or sleep modes of operation. With each step down in process geometry, leakage power has more or less doubled in magnitude. For example, 90nm process technology leakage power contributes 40 to 50 percent of the total power budget, with active power dissipation from transistor switching making up the rest. The management of power effects in the design process should also be concerned with power integrity. This includes analysis of IR drop, which degrades performance and reduces noise immunity, and electromigration (EM). EM is due to high current density causing metal electrons migration, resulting in open or short circuits. EM also causes performance and reliability issues over time. The key power management challenges are to reduce device power consumption by controlling dynamic and leakage effects, and eliminate power-related failures by addressing IR drop and EM.

### Current approaches

The VLSI technology today comprises devices because of their unique characteristic of negligible standby power, which allows the integration of tens of millions of transistors on a processor chip with only a very small fraction (<1%) of them switching at any given instant. As the CMOS dimension, in particular the channel length, is scaled to the nanometer regime (<100 nm), however, the electrical barriers in the device begin to lose their insulating properties because of thermal injection and quantum-mechanical tunneling. This results in a rapid rise of the standby power of the chip, placing a limit on the integration level as well as on the switching speed. IC design companies are mainly attacking the issue of nanometer power consumption. (Fig. 1) While individually effective in their own target areas, the current solutions really only provide a partial solution to the problem.

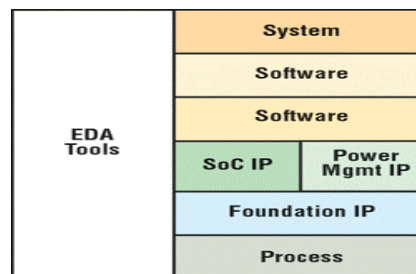


Fig. 1 – Nanometer power consumption issue.

Batteries have made significant strides in providing more power in a smaller space and with less weight. Yet these gains only offset the demand for more power as these products (laptops, PDAs, MP3 players, etc.) incorporate faster processors, advanced graphics and now micro hard drives in MP3 devices. Better batteries will not do it alone. EDA tools are used extensively throughout the design process. Analysis tools are doing excellent job of problem identification at the silicon level, but often do not provide a solution to the designer. The RTL optimization provides improvement in power, but as it is not tied to the actual silicon performance, it is at a fairly high level. Synthesis optimization for power is constantly improving, yet not enough for deep nanometer regime.

The wafer fabs have addressed the issue of power as well. Most fabs and independent foundries offer several process variations at 130 and 90 nm, each optimized for speed, power, or voltage. But all this really offers the designer is the classic power/speed tradeoff. They can have a high-speed high-leakage process or a low-speed low-leakage process. By raising the voltage threshold of the transistors, the leakage can be reduced, but this comes at a corresponding decrease in the speed of the transistor. The industry direction is to incorporate power management considerations throughout the design levels and not only at high level.

### **Efficient Power Management**

In order to achieve effective power management an entire range of design techniques and tools must be applied, throughout the entire design process, to progressively diminish the main sources of power dissipation. Considering the system as a whole, applying power reduction measures through the process technology, circuit design, architecture, system and software, is the best strategy for effective power management. At the system level, it is clear that excessive clock speeds will dissipate power unnecessarily. The system must be clocked at a sufficient rate to meet the application software needs, and not higher. Along with determining the slowest possible clock speed, applying the lowest possible supply voltage also helps to reduce dynamic power. Conveniently, supply voltage scales with feature size. However, lowering supply voltage requires that device thresholds are also lowered, which can increase leakage currents.

Partitioning the architecture to enable multi-voltage design can help to ensure that different parts of the chip operate at the optimum clock rates and supply voltages, ensuring both dynamic and leakage power are minimized. Due to the fact that many applications demand processing performance that varies over time between compute intensive operations, background processing and system idle, an optimized system would adjust voltage and frequency to meet each demand. In general, the decisions taken at the system level will have the greatest impact on overall power consumption. Such decisions include the functionality (for example, whether floating-point arithmetic is really necessary), hardware-software partitioning, bus strategy, clocking strategy, physical partitioning and selection of intellectual property (IP). Choice of memory architecture and implementation can have a significant effect on power budget. Applying well-known techniques during implementation such as RTL clock-gating, gate-level optimization and operand isolation can help to reduce dynamic power considerably. Clock-gating can be applied globally, or at a block level. It is important that application of clock-gating techniques does not impair testability or the ability to perform timing and formal verification. Leakage reduction techniques such as multi-Vt

optimization, active well bias and state retention power gating can help to reduce leakage power considerably. Drawn feature sizes less than 100 nm (0.10 micron), dynamic power scaling trends can also lead to major packaging problems. To alleviate these concerns, techniques like thermal monitoring and feedback mechanisms can limit worst-case dissipation and reduce costs.

### **Low-power IP Methods**

Low-power IP methods offer savings in power through the use of higher threshold transistors and by designing circuits for reduced power, but these methods come with a speed penalty. For low-speed applications, this technique produces acceptable results. The faster speed of the 130-nm process will offset the loss of performance of the low-power IP. For current-generation wireless baseband applications (cellular, Bluetooth) this works well. But for next-generation wireless products with greater graphics and video capabilities, and with high-speed wireless Internet connections, the performance trade-offs are unacceptable. What mobile electronic SoC designers require is a way to have their cake and eat it too.

They cannot make any sacrifices to the performance of their designs and still meet the hard cost requirements that are dictated by the consumer. We all want our cell phones to have a camera and Internet access, but we still want them to fit in our pockets. As more complex and high-performance applications go mobile, SoC designers have no choice but to adopt the use of power management IP to control the increasingly severe dynamic power and leakage power issues.

### **Nanometer Power Obstacles**

**Design Size and Complexity** – Larger and more complex circuitry requires more power – for many reasons. Typically, it is a good idea to reduce the number of transistors required to implement a design, which reduces the total switching activity and the nanometer main issue, leakage current. It also helps overall if the design is organized hierarchically. When a design is organized hierarchically, it can be treated as a modular structure of cells. The largest possible collection of cells can be optimized, with the lowest level of cells instantiated from a library having a selection of cells matching the optimization function: low power, specific voltage, specific clock speed, and so on. This provides a basis for planning and estimating wiring, and the opportunity to revise the architecture or the wiring plan to deal with wire-related issues.

**Timing (Signal Integrity dependent)** – Signal Integrity has major impact on power design. Optimizing the architecture prior to establishing the RTL description, can ensure optimum power and thus minimum opportunity for these effects. This is accomplished through simultaneously optimizing power and clock speed within appropriately segmented domains of the design; a library of high level (complex) cells characterized for various speed and voltage combinations supports this optimization process. If need be, the power constraints can then be selectively relaxed to improve performance or area considerations. An essential need in the entire nanometer-scale optimization process is access to cells characterized for signal integrity, switching power and other characterizations.

**IR Drop** – This issue is typically embedded within automated power optimization of

architecture in cooperation with a characterized library of macro cells, or even of large IP blocks. EDA tools provide a successful solution predicting IR drop effects based on power grid requirements derived from an architectural design.

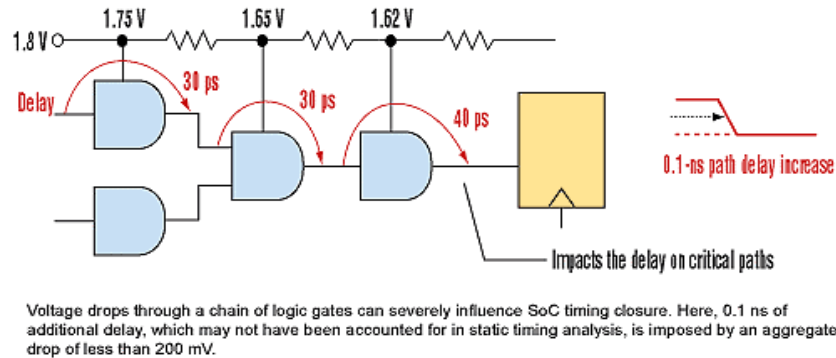
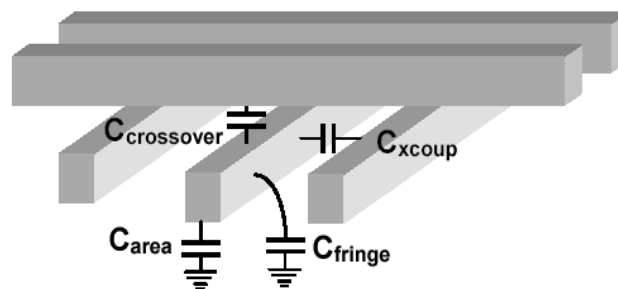


Image: Magma Design Automation

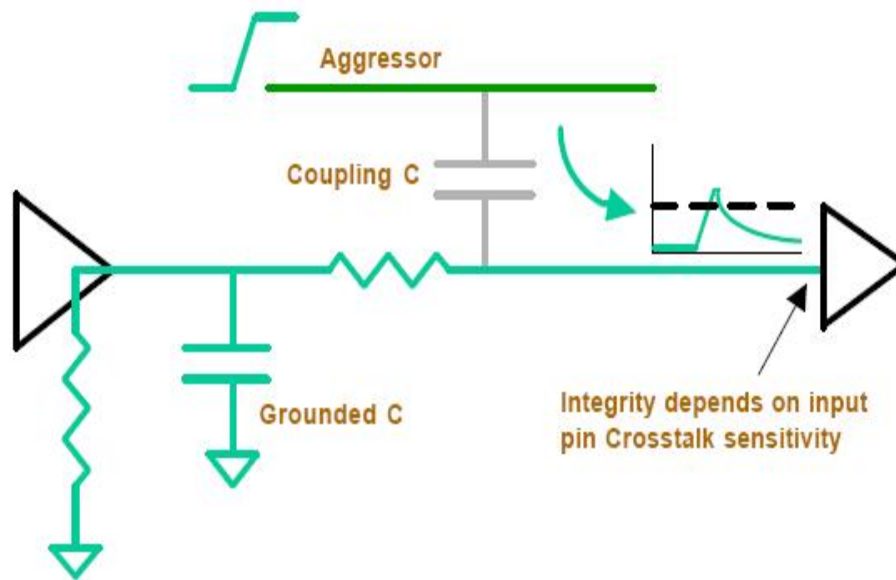
**Crosstalk and Inductance** – With the scaling of the horizontal dimensions of wires, the aspect ratio of the horizontal to vertical dimensions is reduced, resulting in increased ratios of coupling capacitance to ground capacitance (over or under crossovers or to substrate). A significant crosstalk noise may occur due to the relative rate of switching (rise and fall times of the signals) and the amount of mutual capacitance. Crosstalk noise, depending on its amplitude and its timing may cause false switching or delays. The ability of a physical design environment concurrently to analyze and correct for these various signal integrity problems during a physical implementation flow is highly dependent on the design system architecture. An integrated design system is necessary to address VDSM phenomenon efficiently and to provide design closure in a timely manner.

This issue is also typically embedded within automated power optimization of architecture, performing global wiring. However, the optimization process can observe top-down constraints while growing a bottom-up design through the use of characterized cells. Behavioral design provides the means for minimizing the opportunity for crosstalk and other SI issues by managing voltage levels and clock speeds.

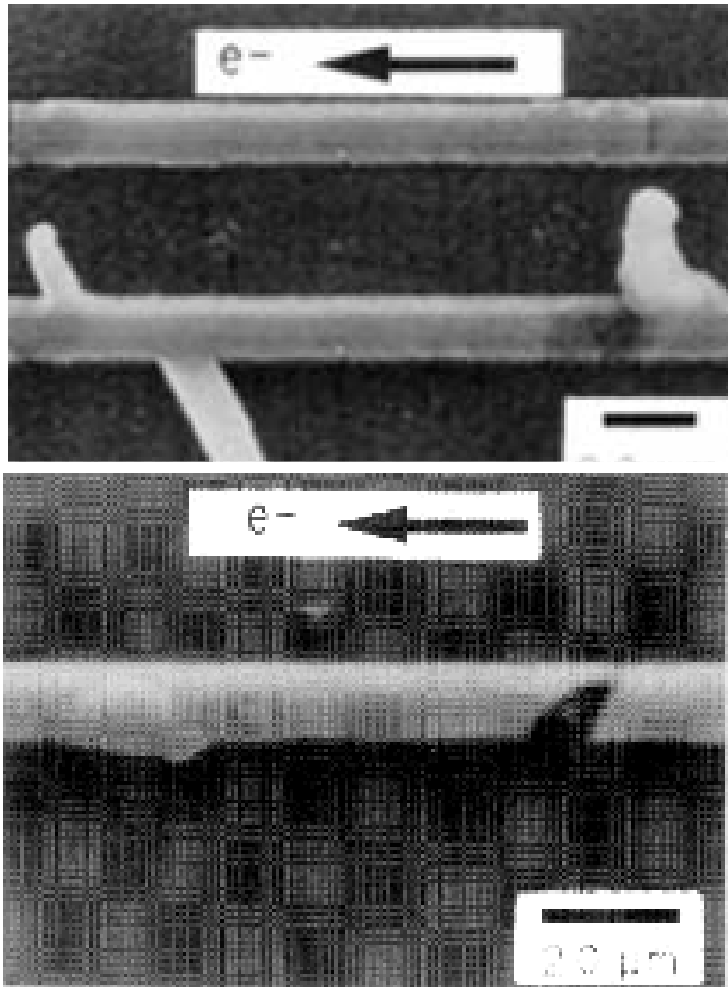


Crosstalk Noise

Image: Magma Design Automation



**Electromigration** – Very Deep Sub Micron designs contain millions of devices and operate at very high frequencies. The current densities (current per cross-sectional area) in the signal lines and power are consequently high and can result in either signal or power electromigration problems. The electron movement induced by the current in the metal power lines causes metal ions to migrate. That phenomenon of transport of mass in the path of a DC flow, as in the metal power lines in the design, is termed power electromigration. There are two types of electromigration. Uni-Directional, for example power and static signals and Bi-Directional, for example clocks and other switching signals. The most critical is the Uni-Directional electromigration type since the electron ‘erosion’ move constantly in one direction and can cause signal line failure. The power electromigration effect is harmful from the point of view of design reliability, since the transport of mass can cause open circuits, or shorts, to neighboring wires. Determination of voltage levels within an architecture or segment of same has a major influence on EM. Especially at nanometer scales, EM can be minimized through selection of connectivity and cells. Performing this at the behavioral level provides the maximum flexibility in meeting other constraints, such as performance. In power-driven design, the goal is to minimize clock speeds, voltage levels, transistor switching and capacitive loading. This is accomplished step-wise, by proposing architecture and then improving it iteratively. Each instance of architecture is marked by specific scheduling of computations, by a controller design, and by assignment of leaf cells. The leaf cells are the key. They are usually large blocks implementing multi-bit functions or even greater functionality. The cells are (ideally) available over a range of voltage levels and clock speeds to meet the possible needs of the architecture under design. It is even possible that such cells can be synthesized in response to need. However, the cells must be characterized. The result of the architecture synthesis then is an RTL design containing leaf cells of known activity characterization, which activity can be further applied in the context of a predicted physical design to support estimation of IR, EM and crosstalk effects, among others, which can then be optimized through iteration.



Electromigration Effect – Open Circuit  
Image: Computer Simulation Laboratory

**Digital-Analog Integration** – These types of power issues typically are pre calculated within EDA tools results. Architectural optimization can account for voltages and clocks in order to minimize the effects of transitions in the area of analog blocks inside digital optimization cycle. It is unlikely to be practical to optimize it outside the cycle in real time.

**Power Consumption** – Power consumption is directly related to wiring concerns, for all the reasons already discussed. Addressing the magnitude of supply voltage, switching activity in the circuit, switching capacitive loads and clock frequency in a top-down, behavioral, iterative manner, can strongly affect many of the determinants of signal behavior, and in many cases optimize their effects achieving best results.

**Power Integrity** - Power integrity is a significant challenge as chip designs move to 90nm and below process technology. With the increasing of number of device, technology scaling, increasing device integration, decreasing supply voltages, increasing leakage currents and

the use of low-power techniques negatively impact current distribution gradients ( $Ldi/dt$ ) and compound power integrity issues. At 90nm, static (or average) IR-drop analysis is insufficient to capture the power-ground noise-related chip failure issues. Static IR-drop takes into account only the power grid resistance ( $R$ ), which can be used to identify gross design violations. At 90nm, it is important to consider the complete dynamic nature of the power-ground noise, including capacitive and/or inductive effects, simultaneous switching noise, and the effect of decoupling capacitors (de-caps). Undetected and uncorrected, these effects can lead to design failure. Power integrity analysis has become a must as part of the design sign-off process.

### **Power management IP**

Power management and power rail design are critical issues for very deep submicron (VDSM) designs. With decreasing supply voltages, increasing demand for low power applications and increasing device densities, the challenge to minimize power consumption, minimize voltage drop effects, and maximize product reliability cannot be handled by traditional physical design techniques. Traditional back-end verification of power and rail design are necessary for sign-off, but the cost of detecting and repairing such problems at the end is extremely high. The IC layout need to be analyzed and optimized to provide a layout that meets the designer's power and reliability constraints. The entire circuit power consumption should be optimized and reduced if possible. Voltage drop and electromigration violations should be correct. Accurate timing for the full design is needed including each cell's timing to reflect the local voltage condition. With the physics of the semiconductor working against engineering at nanometer process technologies, the designer must turn to the voltage and clock frequency of the IC to find power savings. In the simplest sense, designers want to be able to treat individual blocks of an IC in much the same way they did when the different parts of the IC were separate chips on a pc board.

These chips could run at different clock rates and different voltages and could be turned off when not in use in order to save power. The designer needs to be able to partition the IC into electrically autonomous areas, called power islands, based on functionality (CPU, MPEG4, memory, analog, etc.) and power requirements. They then need to be able to dynamically scale the voltage and clock frequency of the power islands, so that each island is only running at the performance required at that time. By enabling designers to dynamically manage the voltage and frequency of individual power islands, designers can significantly lower both the dynamic power, and the quiescent or leakage power, of their IC's designs.

Dynamic voltage and frequency scaling can be additive to many of the other current power optimization techniques. But full implementation of this technique requires a vertically integrated approach within IC design supply chain, and power management IP. Power management IP is more than just low-power IP, it is IP that allows the designer to actively control and manage the power consumption of the IC during operation. Power management IP allows the CPU and software to issue commands to dynamically scale the voltage, frequency, and leakage current of the power islands of an IC. It allows the designer to leverage the enabling elements of the IC design supply chain—low-power IP (standard cells, memory, etc.), software, SoC IP (CPU, DSP, etc.), process technology and power supply IC—to dynamically manage the power of an IC while it is operating.



## Dynamic Power Analysis

### 1. Power Heat Dissipation (Packaging)

As the semiconductor industry adopting the very deep-submicron (VDSM) technology, it creates a new demands on the packaging industry. Especially with high performance IC's the packaging is becoming a significant factor. Increased functionality, faster performance, lower operating voltages and reduced size are leading to increases in die density and I/Os, boosting package pin count and complexity. This has created the need for a new breed of high-density, multilayer, custom-designed packages. We will mention few; flip-chip, ball-grid-array (BGA), and pin-grid-array (PGA).

With chip's power rising, packaging technology must improve to meet heat dissipation demands. The reduction of thermal junction resistance requires advanced cooling techniques such as larger, more powerful fans, liquid/gas vapor cooling, etc'. Packaging experts believe cooled systems are the best solution for packaging high power density VDSM designs.

The advantages of cooling the ambient and junction temperatures are well known: improved voltage scalability due to reduced current leakage, higher carrier motilities, lower interconnect resistances, and improved reliability. Advanced cooling techniques like vapor compression are expensive and predicted to be used only for large IC's. Desktop applications are expected to use low power cooling methods.

Another approach to the packaging heat-constrain is dynamic thermal management. This concept involved thermal management technique that can be achieved in few ways. An example is Transmeta's approach to dynamically varies the supply voltage when the CPU is not heavily loaded. Another example is the thermal monitor in Intel's Pentium IV design which has an on-chip temperature sensor (The temperature sensor is a diode with a fixed voltage across it) along with a reference current source and current comparator to determine when the on-die temperature exceeds a given value. When the temperature (and power consumption) is exceeded the permitted level, the internal clock frequency is reduced, limiting power, throughput and performance. The immediate effect is a reduction in the chip thermal level to bring it to the permitted range.

The importance of dynamic thermal management techniques lies in their ability to reduce the chip power (Wattage) to the effective worst-case power dissipation rather than the theoretical worst-case. The effective worst-case power consumption, as found by running power-hungry applications, is about 75% of the theoretical worst-case, which is determined using synthetic input code sequences that are not realized in practice. This difference has major implications for packaging costs and design flexibility. Small increases in the maximum power can lead to significantly, expensive cooling techniques.

No Doubt, packaging will become more and more in the critical path of the high performance VDSM designs. The increase demand for larger and more powerful chips creates new challenges for the IC's packaging industry.

### 2. Global Signaling

Global signaling within high performance VDSM designs is one of the serious challenges in the nanometer arena. The propagation of global signals across a large die in a shrinking clock period creates an entire series of electrical phenomenon. It appears likely that global

signaling will use a slower clock than localized logic such as datapaths (although multi-cycles nets can be broken up using latches). Even with relaxed timing constraints on global communication, significant power is consumed to achieve the desired global clock speeds. Based on the current signaling paradigm of inserting large CMOS buffers along an RC line, this requires over 50 W of power in the nanometer arena. The proliferation of repeaters (nearly 106 required at 50-nm compared to about 104 in a large 180nm microprocessor and controllers) heightens difficulties in power distribution and floorplanning<sup>2</sup>. One solution is to use advanced signaling strategies such as differential and/or low-swing drivers and receivers for global communication.

In many cases, these approaches can lead to power and tpd (time propagation delay) savings due to smaller voltage transitions as well as major reductions in the scale of power grid current transients. For instance, the Alpha chip uses differential low-swing buses to communicate between functional units. Worst-case power for these buses was reduced considerably by limiting the voltage swing to 10% of V<sub>dd</sub>. Differential signaling increases routing area, but the increase may be less than the expected factor of 2 due to the use of shielding in global signaling to limit coupling from neighboring signals on long lines. In addition, shielding may be insufficient to limit inductively coupled noise, whereas low-swing differential signaling creates less noise and is more noise immune than single-ended full-swing CMOS. With the industry trends indicating rising power consumption for global communication, the use of alternative signaling strategies will most likely increase. Further study is necessary to provide an efficient solution to the global signaling concern.

### **3. Library Optimization**

Silicon-proven libraries, give designers and fabless companies very high performance solutions, using some of the most advanced processes available. While most high performance microprocessors rely heavily on custom design, library optimization can significantly enhance performance in these applications. Advances in library generation, and synthesis tools that take advantage of improved libraries, can together yield more automated, less expensive design flows. Libraries are one important reason that custom designs are significantly faster (6-10X) than counterpart ASIC designs. For instance, asserts that the lowest performance level (smallest) gates in modern libraries are nearly 10X larger than minimum-sized gates, leading to major power increases due to overdriving small loads. However, most current libraries contain a large number of drive strengths, including some very near minimum size. As evidence, we refer to the same 180 nm library as the smallest standard cell inverter has an input capacitance of just 1.5fF and the smallest inverter with balanced rise/fall delays has an input capacitance of 6.6fF. Other leading-edge libraries contain a rich set of drive strengths (e.g. 11 2-input NANDs, 16 inverter sizes), dual output polarities, and single pin inverted inputs on NAND/NOR's. This recent increase in library complexity seems to be closing the gap slightly between custom designed cells and those from libraries.

Today EDA vendors provide an entire line of optimized VDSM libraries in order to help customers to achieve efficient designs. For example like Synopsis (using Avant! Tools) provide today an entire set of optimized libraries including standard cells, IO's and memory compilers. The prediction is that in the near future the entire industry will use an optimized libraries provided by the foundries.

#### 4. Multiple Powers Vdd

One of the most efficient methods to rise of dynamic power in VDSM designs is to use multiple power supply lines. (Vdd's) The general idea called clustered voltage scaling (CVS). With two Vdd levels (Vdd\_h and Vdd\_l), the circuit is partitioned so that non-critical gates run at Vdd\_l and only critical gates use Vdd\_h. Level conversions, performed when gates running at Vdd\_l fan-out to gates at Vdd\_h, are reduced by clustering Vdd\_l and Vdd\_h gates together to minimize the number of such interactions. Analysis indicates that Vdd\_l should be around 0.6 to 0.7 times Vdd\_h to maximize power savings. The dynamic power reduction by using two Vdd levels is readily calculated if one can estimate the fraction of cells that can be assigned to Vdd\_l. Existing media processor designs that use CVS report that ~75% of all gates can tolerate Vdd\_l without altering the critical path delay.

The key challenges to the use of multiple supplies on a chip lie in minimizing area overhead and providing EDA tool support for Vdd cell selection, placement given new clustering constraints, dual power grid routing, and enhanced library generation capabilities. Using this system within EDA tools provides a powerful capability for VDSM high performance designs.

#### Static Power Analysis

##### 1. Multiple Vth Approaches

In order to reduce CMOS static power consumption several approaches have been developed. In this section we will briefly discuss several of these techniques that use multiple thresholds on a single chip to limit I<sub>off</sub>.

##### A. MTCMOS Method

Multi-Threshold CMOS (MTCMOS) gates a high-V<sub>th</sub> transistor with a sleep mode signal to virtually eliminate leakage current in idle states. The sleep transistor is placed between ground and fast low-V<sub>th</sub> CMOS logic. As it is in series, it adds delay, which can be reduced by increasing its area. Disadvantages include no leakage reduction in active mode, increased device area, and additional overhead for routing sleep signals. Other similar techniques include dual-V<sub>th</sub> domino logic, substrate biasing to modify V<sub>th</sub> in standby, and using negative NMOS gate voltages to bias the devices further into cut-off. A single threshold leakage reduction technique combines the concepts of sleep transistors and state dependent leakage. All these techniques trade off area to limit static power and most only reduce leakage in standby mode. In fact, they are currently limited to portable applications such as notebook processors. Also, some of the proposed methods do not scale well – the use of domino logic for example, and substrate bias controlled V<sub>th</sub> (body bias is less effective at controlling V<sub>th</sub> in scaled devices). Dual V<sub>th</sub> insertion, described next, is the only technique used in current high-end MPUs.

##### B. Dual-Vth Method

Today, circuit designers have access to multiple threshold voltages on a single IC to select between gates that use high or low thresholds. The impact of V<sub>th</sub> on the delay and power of gates such as inverters and NANDs is significant. A reduction in V<sub>th</sub> (with constant Vdd) exponentially increases off current and roughly linearly reduces the propagation delay. An additional threshold adjustment implantation step allows designers to choose from a wider range within the power-performance design envelope. Gates located on critical paths can be

assigned fast low  $V_{th}$ , while gates that are not timing critical can tolerate high  $V_{th}$  and slower response times. Typical results show leakage power reductions of 40-80% with minimal penalty in critical path delay compared to all low- $V_{th}$  implementations. Figure 2 shows the increase in  $I_{on}$  for the low- $V_{th}$  device. The relative difference in  $I_{off}$  between the two devices will remain constant throughout the roadmap (at about a 15X increase in  $I_{off}$  for 100 mV reduction in  $V_{th}$ ). Given that the off current change is constant, the steady improvement in  $I_{on}$  with scaling demonstrates that the dual- $V_{th}$  (or multi- $V_{th}$ ) approach to leakage reduction is inherently scalable. Figure 2 also shows the resulting  $I_{off}$  increase for  $I_{on}$  to rise 20% beyond the high- $V_{th}$  case. At 35 nm, just a 7X rise in  $I_{off}$  is required to yield 20% drive current improvement, compared with a factor of 54X today. In Figure 3 we can see the  $I_{off}$  decrement with the process shrink and the reduction of  $V_{dd}$ .

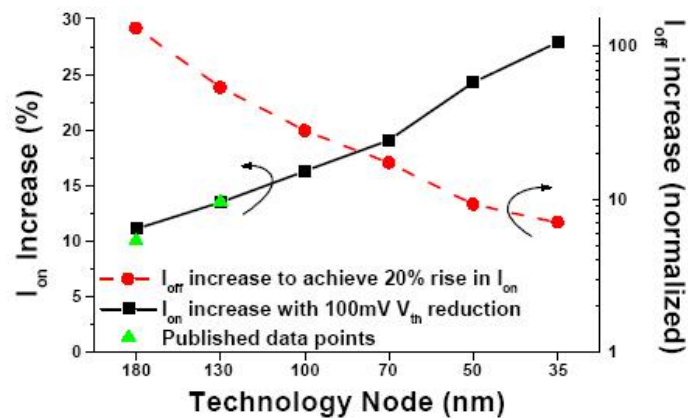


Figure 2.  $I_{on}$  increases more rapidly with a 100mV change in  $V_{th}$  for scaled technologies.  $I_{off}$  penalty for 20%  $I_{on}$  gain reduces with scaling.

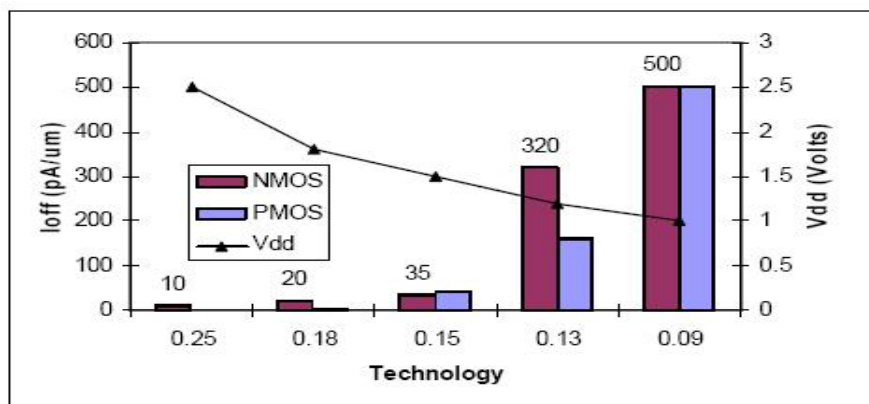


Figure 3 -  $I_{off}$  reduction with the process and power

## 2. Scalable Dynamic/Static Power Approach

One of the most appealing approaches, in order to achieve scalable, flexible and cost effective design is a combination of multiple Vdd's and multiple Vth's. The combination of multiple Vdd's, multiple Vth's, and intra-cell size and Vth assignments points to a highly flexible, scalable, cost effective design approach to dynamic and static power minimization. With two voltage supply values available, different Vth's will allow designers or EDA tools to choose to emphasize speed, standby power, or dynamic power.

### Summary

The EDA industry continues to face new challenges as process continues to shrink into nanometer geometries. With each successive advancement of semiconductor technology a new VDSM challenge is born. Especially with high performance reliable designs the industry has to face a wide variety of phenomenon such as heat dissipation, electromigration, interconnect coupling and more. Many EDA tools have been enhanced to deal with these issues. The keys to a successful nanometer design are:

1. Efficient and reliable power management techniques such as on-chip temperature monitors and multiple voltage supplies will reduce dynamic power, enabling cheaper packaging and higher integration densities.
2. Power distribution will be manageable from the standpoint of IR drop – given changes in the ITRS to take advantage of technological advancements in flip-chip packaging. However, large current transients may be exacerbated by the use of sleep/standby modes.
3. Alternative techniques to CMOS repeaters for global signaling need to be studied and implemented within EDA tools to minimize power consumed in global communications.
4. A multi-layered approach to power reduction (both dynamic and static), combining multiple threshold and supply voltages with flexible gate layouts using different thresholds and device sizes within a gate. Non-critical gates are first assigned to a reduced Vdd, followed by sizing and Vth selection to reduce power most efficiently.

Taken together, the various aspects described in this document form a methodology that provides comprehensive power optimization. The power optimization aspect needs to be performed throughout the entire design stages particularly during architectural synthesis phase in order to solve critical nanometer issues.

While the power crisis in the nanometer arena is clearly a challenge, designers are beginning to see the introduction of a new breed of power management IP integrate with other elements of the design supply chain. In the coming years, we can expect more innovative EDA products that will deliver powerful solutions for the nanometer power crisis.

### References

- [1] J. Cong, L. He, C.-K. Koh, D. Z. Pan, and X. Yuan, "TRIO: Tree, repeater and interconnect optimization package." [http://cadlab.cs.ucla.edu/\\_trio](http://cadlab.cs.ucla.edu/_trio).
- [2] J. Cong, L. He, A. B. Kahng, D. Noice, N. Shirali, and S. H.-C. Yen, "Analysis and justification of a simple, practical 2 1/2-d capacitance extraction methodology," in *Proc. ACM/IEEE Design Automation Conf.*, pp. 40.1.1–40.1.6, June, 1997.

- [3] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integration, the VLSI Journal*, vol. 21, pp. 1–94, 1996.
- [4] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh, and D. Z. Pan, "Interconnect design for deep submicron ICs," in *Proc. Int. Conf. on Computer Aided Design*, pp. 478–485, 1997.
- [5] Semiconductor Industry Association, *National Technology Roadmap for emiconductors*, 1997.
- [6] J. Cong and D. Z. Pan, "Interconnect delay estimation models for synthesis and design planning," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 97–100, Jan., 1999.
- [7] J. Cong and D. Z. Pan, "Interconnect estimation and planning for deep submicron designs," in *Proc. Design Automation Conf.*, June, 1999.
- [8] C.-P. Chen and D. F. Wong, "Optimal wire sizing function with fringing capacitance consideration," in *Proc. Design Automation Conf.*, pp. 604–607, 1997.
- [9] J. Cong, "Challenges and opportunities for design innovations in nanometer technologies," Dec. 1997. <http://www.src.org/research/frontier.dgw>.
- [10] J. Cong, L. He, C.-K. Koh, and D. Z. Pan, "Global interconnect sizing and spacing with consideration of coupling capacitance," in *Proc. Int. Conf. on Computer Aided Design*, pp. 628–633, 1997.
- [11] K. Usami and M. Horowitz, Clustered voltage scaling techniques for low-power Design, ISLPED 1995.
- [12] C. Chen, A. Srivastava, and M. Sarrafzadeh, On gate level power optimization using dual-supply voltages, *IEEE Trans. on VLSI Systems*, vol.9, p.616-629, Oct. 2001.
- [13] N. Rohrer, et al., A 480MHz RISC microprocessor in a 0.12\_μm Le\_1\_CMOS technology with copper interconnects, *ISSCC 1998*, p.240-241.
- [14] S. Sirichotiyakul, et al., Standby power minimization through simultaneous threshold voltage selection and circuit sizing, *DAC 1999*, p.436-441.
- [15] Q. Wang and S.Vrudhula, Algorithms for minimizing standby power in deep submicron, dual-Vt CMOS circuits, *IEEE Transactions on CAD*, vol.21, p.306-318, 2002.
- [16] M. Hamada, Y. Ootaguro, and T. Kuroda, Utilizing surplus timing for power reduction, *CICC 2001*, p.89-92.
- [17] K. Usami, et al., Automated Low-Power Technique Exploiting Multiple Supply Voltages Applied to a Media Processor, *IEEE JSSC*, Vol.33, No.3, 1998.
- [18] M. Hamada, et al., A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme, *CICC 1998*, p.495-498.
- [19] D. Sylvester and H. Kaul, Future performance challenges in nanometer design, *DAC 2001*, p.3-8.
- [20] A. Srivastava and D. Sylvester, Minimizing total power by simultaneous Vdd/Vth assignment, *Proc. Asia-South Paci\_c DAC 2003*, p.400-403.
- [21] C. Yeh, et al., Layout Techniques supporting the use of Dual Supply Voltages for Cell-based Designs, *DAC 1999*.
- [22] D. Lackey, et al., Managing Power and Performance for SOC Designs using voltage islands, *ICCAD 2002*.
- [23] S. Kosonocky, et al., Low Power Circuits and Technology for wireless digital Systems, *IBM Journal of R&D*, Vol. 47, No. 2/3, 2003.
- [24] A. Correale, D. Pan, D. Lamb, D. Wallach, D. Kung, R. Puri, Generic Voltage Island: CAD Flow and Design Experience, *Austin Conference on Energy E\_cient Design*, March 2003 (IBM Research Report)

- [25] W. Donath, et al., Transformational Placement and Synthesis, DATE, 2000.
- [26] R. Puri, E. D'souza, L. Reddy, W. Scarpero, B. Wilson, Optimizing Power-Performance with Multi-Threshold Cu11-Cu08 ASIC Libraries, Austin Conference on Energy Efficient Design, March 2003 (IBM Research Report).
- [27] R. Puri, D. Pan, D. Kung, A Flexible Design Approach for the Use of Dual Supply Voltages and Level Conversion for Low-Power ASIC Design, Austin Conference on Energy Efficient Design, March 2003 (IBM Research Report).
- [28] Y. Taur, CMOS Design near the limit of scaling, IBM Journal of R&D, Vol. 46, No. 2/3, 2002.
- [29] J. P. Fishburn, Clock Skew Optimization," IEEE Transactions on Computers C-39, pp 945-951, 1990.
- [30] T. G. Szymanski, Computing Optimal Clock Schedules," DAC 1992, p.399-404.
- [31] C. Albrecht, B. Korte, J. Schietke and J. Vygen, Cycle Time and Slack Optimization for VLSI-Chips, ICCAD 1999, p.232-238.
- [32] S. Bhattacharya, J. Cohn, R. Puri, L. Stok and D. Sunderland, Power reduction of Hardwired DSPs in standard ASIC methodology, Submitted to CICC, 2003.
- [33] L. Stok, et al., BooleDozer Logic Synthesis for ASICs, IBM Journal of R&D, Volume 40, no. 3/4, 1996.
- [34] F. Beeffink, P. Kudva, D. Kung, R. Puri, L. Stok, Combinatorial cell design for CMOS libraries, Integration: the VLSI Journal, V.29, p.67, 2000
- [35] P. Kudva, D. Kung, R. Puri, L. Stok, Gain based Synthesis, ICCAD Tutorial, 2000.

## Europe Union FP6 Program



[http://www.delchn.cec.eu.int/en/Science\\_Technology/FP6.htm](http://www.delchn.cec.eu.int/en/Science_Technology/FP6.htm)

**Dr Jurgen SANDER**  
**Counsellor for Sci. & Tech., EU Delegation Beijing**  
**[jurgen.sanders@cec.eu.in](mailto:jurgen.sanders@cec.eu.in)**



# Epistemologically Multiple Actor-Centred System: Or, EMACS at work!

Yuwei Lin  
University of York, Uk

## Abstract

The paper begins with the story of EMACS (short for Editing MACroS), an editor programme originally written for TECO (Text Editor and Corrector) language and PDP-10 machines in the MIT AI Lab by Richard Stallman, from which various more sophisticated versions have been developed. I analyse how the innovation of EMACS took place over time as a socio-technical process. The EMACS story serves to illustrate how the innovation process in the FLOSS (Free/Libre Open Source Software) community occurred, but one that is then adopted and deployed in other social contexts, including the commercial sector. The analysis of EMACS is especially useful since it spans the period that saw the origins of the free software movement and the subsequent development of a broader FLOSS social world. I will talk about how a variety of EMACSen (a plural form of EMACS) (e.g. GNU Emacs, XEmacs, MulticsEmacs etc.) is created, developed and employed/deployed in mundane programming within an actor-centred network. Actors from different backgrounds contribute multiple ways of knowing, understanding and resolving problems that arise in the innovation process. A socio-technical perspective is employed to analyse how EMACSen are shaped by diverse actors, and at the same time also shape these actors and their practices.

To widen the scope of the paper in terms of its implication in a wider societal dimension, anchored in sociology of intellectual/knowledge, this paper also contributes to our understanding of the formation of knowledge in the Internet era, where information and knowledge flow fluidly and rapidly. The EMACS case denotes various key factors of forming cosmopolitan knowledge: how actors network together (e.g. shared interests), how they interact with one another (e.g. problem-solving process), and how local epistemologies and tacit knowledge being translated into cosmopolitan expertise in an in/tangible form (e.g. materiality of software). I believe this empirical enquiry will provide us with a means of retaining the holistic and meaningful characteristics of real-life events. Methodologically speaking, the contextual thickness makes a case study appropriate for "how" and "why" research questions because answering these questions deals with operational links needing to be traced over time. The detailed investigation of FLOSS phenomenon with attention to its context by using multiple sources of evidence and various methods of data collection will help to examine the innovation process by which new FLOSS technologies are created, arguing that this is ongoing and involves diverse groups who give the technology different meanings.

## 1.Introduction

The paper begins with the story of EMACS (short for Editing MACroS), an editor programme originally written for TECO (Text Editor and Corrector) language and PDP-10 machines in the MIT AI Lab by Richard Stallman, from which various more sophisticated versions have been developed. I analyse how the innovation of EMACS took place over time as a socio-technical process. The EMACS story serves to illustrate how the innovation process in the FLOSS (Free/Libre Open Source Software) community occurred, but one that is then adopted and deployed in other social contexts, including the commercial sector. The analysis of EMACS is especially useful since it spans the period that saw the origins of the free software movement and the subsequent development of a broader FLOSS social world.

## 2. EMACS

In 1976 Richard Stallman, an employee at MIT AI Lab, and his colleagues, wrote the editor EMACS to upgrade the previous editor TECO on an ITS (Incompatible Time-Sharing System), which was the software running on the AI Lab's Digital PDP-10 mini-computer. In the text-based pre-graphical world that existed before the Apple Macintosh and Microsoft Windows, the editor was a programme crucially important for creating and manipulating text (Moody 2001: 16). Instead of typing commands when editing texts, the TECO editor enabled users to employ macros, command strings for a cluster of TECO programmes, which provided a more immediate onscreen feedback for users. TECO had already had the 'WYSIWYG' (What You See Is What You Get) feature named Control-R, written by Carl Mikkelsen, which allowed users to enter macros (command strings) and discarded them after entering them. Borrowing the idea from another WYSIWYG editor named 'E', Stallman then brought additional functionality to TECO to make it possible to save macro shortcuts on file and call them up at will. It is said that this improvement was subtle but significant in that this raised TECO to the level of a user-programmable WYSIWYG editor, which later on enabled innovation at another meta level that became the progenitor of FLOSS (Williams 2002: 82). The amended macro function in TECO permitted users to redefine their own screen-editor commands, pass them around and improve them, make them more powerful and more general, and then the collections of user-redefinitions gradually became system programmes in their own right (ibid.). In so doing, users extended the original TECO system by adding or replacing functions based on their self-defined definitions of 'the problem'. Users were not, then, limited by the decisions made or problem-solving approaches taken by the original innovators (Stallman 1998: 2). The extensibility made TECO more flexible for use and in turn attracted a larger number of users to incorporate the macro function into their TECO programmes.

However, a new problem emerged along with this new feature. While the new full-screen capabilities were embraced vigorously, various customised visions of TECO also led to over-complexity. The most obvious example was that one had to spend more time than before figuring out what macro commands did what in terms of an individual's self-definition of 'the problem' when improving each other's work. Guy Steele, a colleague of Stallman's at the AI Lab, recognised this problem and sought to address it. He firstly gathered together four different macro packages and began assembling a chart that he believed identified and organised the most useful macro commands (Williams 2002: 83). In the course of implementing the design specified by the chart, Steele's work attracted Stallman's attention and led to the latter's participation in this renovation project. Together with David Moon and Dan Weinreid, the four tried on the one hand, to develop a standard

system of macro commands, and on the other hand, to still keep the command set open-ended to enable ongoing programmability/extensibility by others. The programme was named EMACS.

The distribution of EMACS marked another milestone in the software history. In response to the prevalence and technical opacity associated with the practice of entirely self-defined commands, and to endorse the hacker tenet of sharing information, Stallman set the terms of on which EMACS could be used in the statement linked to the source code when distributing the editor. EMACS, as noted in Stallman's biography, served as a social contract that rendered communal sharing the basis of its distribution. Users, on the one hand, were able to modify and redistribute the code; on the other hand, they were asked to report back the extensions they might have made to Stallman so that he could incorporate and distribute those again. In so doing, Stallman strengthened the functionality of EMACS, making programming with macros more standardised through creating a reciprocal understanding of the written codes through sharing problem solutions, as well keeping the extensibility that macros afforded. Consequently, a library was created for users to load new or redefined functions and to publish and share their extensions. "By this route, many people can contribute to the development of the system, for the most part without interfering with each other. This has led the EMACS system to become more powerful than any previous editor." (Stallman 1998: 2). Since then, an archetype of EMACS had been established.

### **3. Affordance and EMACS: Extensibility and Customisation**

The earlier generation of EMACS had been successful because it provided flexible use with an embedded programming language, TECO. Because of this feature, editing commands could be written in that programming language and users could load new commands into her/his editor while s/he was editing. EMACS resembled a system that was useful for things other than programming, and yet one could program it while s/he was using it. It was claimed to be the first editor that could operate in this way (Stallman 2002: 1). Consequently, the socio-technical networks of EMACSen have been expanded. GNU Emacs for example, is now available for Unix, VMS, GNU/Linux, FreeBSD, NetBSD, OpenBSD, MS Windows, MS-DOS, and other systems. GNU Emacs has been re-configured more than 30 times as part of other systems. Other variants include GOSMACS, CCA Emacs, UniPress Emacs, Montgomery Emacs, and XEmacs. Jove, Epsilon, and MicroEmacs are limited look-alikes. These systems on the other hand also have requirements, needs, and visions that differ from the original GNU Emacs. This phenomenon widens the range of what we might see as "digital epistemologies" (ways of ordering and knowing software) and their expression through software artefacts in the FLOSS social world.

### **4. Problems and solutions: the sine qua non of the software innovation process**

In light of the EMACS account above, problems play a crucial role in the innovation process. Triggering a problem or perceiving a problem and dealing with it are important tasks for scientists and engineers, and through their resolution help generate innovation. A problem thus can be seen as the inauguration of an innovation. The confronting of a problem provides an opportunity of coming up with something new or different. A problem de facto denotes one's perception of the situation, one's knowledge, and skills. Accordingly a problem signifies one's identity as an expert or a novice, for example.

Efficiency was an important parameter in the process of defining software problems. A problem would not exist if users did not recognise the existing practice (e.g. composing code with printing terminal editors in the 70s) as inefficient. As efficiency is regarded as key within the field of engineering; engineers were and are still taught to design efficient technologies. Efficiency is not however, self-evident: Stallman reports that he did not have a strong sense of the need for a real-time display editor until he encountered the *E* programme when he visited the Stanford Artificial Intelligence Lab in 1976. He was inspired by the function *E* afforded and sought to expand TECO's functionality in the same way and helped form the group that was to work on a real-time display system. Meanwhile, there were other parallel groups providing solutions for real-time display systems, and their work could become complementary to the work that Stallman's group were undertaking. Stallman's macros improvement to TECO enhanced Mikkelson's earlier WYSIWYG feature for TECO. As a result, with this greater affordance and functionality, TECO became more popular. The TECO socio-technical network expanded when more people accepted the macro innovations and incorporated them into their own versions of the TECO programme.

It is worth noting that Stallman did not sit down and write the editor system programme immediately after his encounter with *E*. Instead, he looked up the database and found that Mikkelson had made a WYSIWYG feature for TECO. He then integrated his idea into that. If Mikkelson's work had not existed, we may have seen a different technical option taken, as the *problem* may have been defined differently. This points to the contingency of the innovation process. This process is reflected in many FLOSS developers' own reflections on their systems as seen in Stallman's and Torvald's biographies (e.g. Torvald said he probably would not have started the Linux kernel project if the GNU Hurd had already existed; Stallman said he would not have started to write the GCC compiler if Tanenbaum had agreed to share his work). Hence, looking for existing material and tools is another common practice in solving problems in software innovation. Software engineering, as in other fields, is built on existing technologies. Programmers typically explore existing databases and see whether any tool is available; if not, they are likely to try to create one to solve the problem.

Access to problems is another issue that needs to be discussed in light of the data from my fieldwork. The accessibility of problems measures the relative ease with which problems can be understood. If one problem entails an opportunity for innovation, an active innovation field should welcome more problems. In a less open innovation system, problems are less accessible, and the boundary is relatively impermeable to new entrants and new ways through which the system can be enhanced. In such an innovation system, problems are less likely to be seen to appear, innovation options likely to be more pre-defined, and innovators sharing a consensus on *what needs to be done*. On the contrary, I want to argue that in a heterogeneous field where diverse actors are found, more problems arise or are triggered. If the boundary of the social group centring on the problem is soft, more diverse actors will be included in the circle. There is a positive correlation between the elasticity of the boundary of an innovation field and the momentum behind the pace of innovation because the more accessible the problem is, the higher the level of multivocality existing in the innovation system.

The elasticity of the boundary can be manipulated through a range of educational, legal, political, economic, social and technical means. In the 1970s, software problems were more

accessible in the sense that fewer regulations were applied to restrict programmers to access key materials (codes peculiarly). There was a so-called “collaborative hacker” approach, sharing knowledge and improving each other’s work, that encouraged programmers to continually to redefine the boundaries of the problem (e.g. conducting reverse engineering to deconstruct a software to understand how a code was written). As a result, a wide range of software programming tools (languages, editors, compilers etc.) was created in the 1970s (Ceruzzi, 2003).

However, the generation of too many problems may become counterproductive to innovation. The ability to solve problems is key to innovation. The more a problem is accessible, as noted above, the more diverse actors will be invited to participate in the innovation group centring on the problem. As there is no single perfect solution for a problem, multiple voices and silences should always be welcome (Bowker & Star, 1999: 41). If a problem is presented in an intelligible/perceivable/accessible way, it will encourage more participants to craft solutions (though there may be nothing wrong with asking a “dumb” question, as the dumbest question can sometimes produce the best answers). As noted by a number of commentaries, well-defined problems in which the given information, operations, and goal are clearly specified will more likely to have solutions than ill-defined problems (Glass et al., 1979; Borgman, 2003). Furthermore, such sources argue that an expert can articulate the queries more specifically and completely than could someone new to the domain. This ability to articulate problems becomes one of the parameters that defines expertise, which will be discussed later in this paper.

As I noted above, while Stallman’s innovation was celebrated because of its extensibility and flexibility, it did however create new problems. One that many saw was the sense of a growing confusion derived from the plethora of self-defined macro commands, which was seen to eventually work against a more efficient process of programming. In response, another member of the AI group, Steele, tried to provide a solution by charting the macros. Steele’s solution encouraged Stallman, Moon and Weinreid to participate in a solution-crafting innovation group. These four who shared the same interest in this project formed a social network of expertise and began to fashion the digital tools that would be seen to provide the solution they were looking for: in this sense the path they took and tools they developed reflected a shared conceptual frame that was “not accidental, but constitutive.” (Clarke, 1998; Bowker & Star, 1999: 36). What made the process more intriguing is the relationships and interactions between the actors themselves and the actants (materials), and the transitions – the boundary crossings – that a problem so identified enabled. Boundary crossings can require both getting in (e.g. gaining access to the problems) and getting out (e.g. forging an alternative solutions different from the original one). These boundaries are interpreted or constructed through the problem-solving process of software design. In the process, actors move across boundaries and shift their identities as outsiders or insiders to the core innovation group.

### **5. Shared interests and translation of interests**

As seen in the story of EMACS, engaging actors in a network is the key to effective innovation in that the range of expertise in the network will affect a group's abilities to solve problems. Since everyone is an expert with regard to some things and a novice with regard to others, a problem/question in an open environment will be answered sooner or later. It is

worth noting, however, actors' involvement in a network is not randomly assembled, but determined by shared interests. As Latour articulates,

The first and easiest way to find people who will immediately believe the statement, invest in the project, or buy the prototype is to tailor the object in such a way that it caters to these people's explicit interests. As the name "inter-esse" indicates, "interests" are what lie in between actors and their goals, thus creating a tension that will make actors select only what, in their own eyes, helps them reach these goals amongst many possibilities. (Latour 1987: 108-9).

After Stallman left the MIT AI Lab and planned to write an Unix-like operating system for non-commercial distribution, a range of ICTs facilitated his individual action. Stallman's GNU project, as if many other FLOSS projects, precisely took advantage of ICTs to target peers to join his innovation network. These artefacts enabled him to maintain communication with other programmers and even helped to secure some innovation resources when he worked alone. On 27 September 1983, Stallman posted the message below onto the Usenet newsgroup net.unix-wizards in order to invite people who shared the same interest or parallel knowledge to join the discussion. Stallman posted the following message:

Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for GNU's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed. (Williams 2002: 89)

In this message, Stallman revealed his defined problem and the proposition for possible solutions a complete Unix-compatible software system. Because he had lost institutional support (financial and intellectual) from the MIT AI Lab, he was more likely to find peers interested in joining the social network of developing a new operating system and to engage their attention by posting messages onto the Unix newsgroup, where specific users/programmers share the same interest inhabit. Without knowing who was going to get in touch with, the message posted entailed uncertainty and risk in Stallman's project. While Stallman posted this message, he created a social network of crafting a new operating system. If Stallman was able to invite many programmers to join this project, the network would grow and the project would take off. Here, making decisions of which listserv or newsgroup a message should be posted to in order to attract as many actors as possible is another form of classification shaping the innovation process. Stallman reckoned the Unix group was the one where he would be most likely to find his target peers. This tendency of looking for peers who share the same interests echoes my previous argument that a shared interest among peers is crucial for the continuation of collaboration. The common interests engage actors to work together, share knowledge and exchange information. The teamwork gets more complex with higher peer participations. If the management style stays in a democratic/open way, the boundary of the team will remain soft. This type of innovation is more accessible because open debate within the team is more likely to produce multiple topics to attract actors. Here, the construction of a shared interest or a common topic is resonant with Latour's notion of inter-essant (1987), through which actors are enrolled to mobilise innovation networks. The working environment at MIT AI Lab in the 70s was

similar to this way. Most of the FLOSS projects that rely on virtual collaboration also meet the criteria of Latourian inter-esse/sant. In contrast, a closed or centralised direction of a project would reinforce innovation boundaries and restrict accessibility to the innovation process. In so doing, a project can be kept under control to eliminate risks and uncertainties generated by multivocality. Following this route, a project will approach closure eventually. Proprietary software is mostly managed in this way.

## **6. EMACS as a Boundary Object**

Hitherto, I have investigated how a software innovation network is created and developed in terms of classifications of problems, identifications of solutions and common interests. Actors and actants are brought together, interact and negotiate with each other to solve problems. Boundaries of networks, which determine access to innovation systems, vary in different contexts. A successful innovation, as discussed, is the one that manages to bring in as many actors and actants as possible by translating their interests to extend and mobilise the network as well as handles the uncertainties and risks emerging during the process.

In reviewing the innovation story of EMACS, a variety of EMACSen have been innovated/renovated by diverse actors for different purposes. The functions of EMACS have been expanded and are still expanding. In other words, the affordance of EMACS is sustained. While diverse innovation repertoires are brought into the social network to tackle a joint problem, they also complicate the situation by defining and redefining EMACS' innovation concept over and over again. If different definitions of innovation concepts cannot be reconciled, a project diverges, as the development of many versions of EMACS show. The reasons for the divergences vary in social scope (e.g. disagreement on Stallman's social contract), technical/material scope (e.g. original EMACS did not run on other programming language than TECO, so new version of EMACS was designed for other programming language such as Lisp, such as EINE and ZWEI, developed by Weinreb, a fellow colleague working on the original version of EMACS as well), or other contingent factors (e.g. experimental projects-- just for fun, perhaps). These parallel processes demonstrate the dynamic in the innovation network of EMACS. Facing the challenge of the heterogeneity, authors and maintainers of EMACSen try to enhance their legitimacy and uniqueness in providing greater socio-technical functions.

In this account, EMACS, as an extensible editor that can be used flexibly, has served as a boundary object for diverse users. They interpret the role of EMACS in different ways according to their situated practices. A number of common practices of EMACS can be summarised. Some take EMACS as a pure tool for editing texts, programming, gaming, web browsing. Others contribute to the FLOSS community by reporting bugs of EMACS while using it, and still the others residing in the core of EMACS innovation system take EMACS as an artefact or even art work. Because EMACS denotes so many different meanings, the diverse interpretations and manipulations of EMACS can prevent it becoming a stereotyped editor, even though its material effects (as affordances) do give it a specific technical character. This is in contrast to other proprietary software products. For example, Microsoft Office Word software has enrolled a huge number of users for processing their Word documents. When mentioning "Word" people take Microsoft Office Word for granted. Accordingly, Microsoft Office Word is used not more than for typing and word processing. But for EMACS, its high affordance enables many roles to be played in mundane computing

practices. Once EMACS is mentioned, people would like to know which version of EMACS is indicated (e.g. GNU Emacs, XEmacs – each indicates different usage habits, particular interests in programming languages, or even political views), and for what purpose it is used. The potentiality of EMACS for different functions is resilient. Unlike Microsoft Office Word, the open trait of EMACS gives users more space to configure and customise their own versions which meets individual requirements.

Since EMACS are used in many different ways and denotes various socio-cultural meanings, diverse projects have symbolised users' habits and preferences (socially and technically). In adopting specific tools and participating in specific projects, users are attached to the artefacts. These artefacts grow to be norms to demarcate boundaries. For example, the users of GNU Emacs see themselves different from those of XEmacs, while the broad range of users of Emacs distinguish themselves from other users of other editor programmes, such as vi. Holy wars happen often because these users want to fortify their boundaries against each other to show their identities. Accordingly, software programmes containing symbolic contents should be seen not merely as algorithm codes but also as social codes. It would be interesting to see how these projects are symbolised as norms, how they are interpreted and used. That is, in the course of explaining the heterogeneous FLOSS social world, one can study the socio-technical meanings given to various projects to understand "FLOSS". This actor-centred view may provide a distinctive research result from the prevailing structure-centred or essentialist approaches in the FLOSS studies.

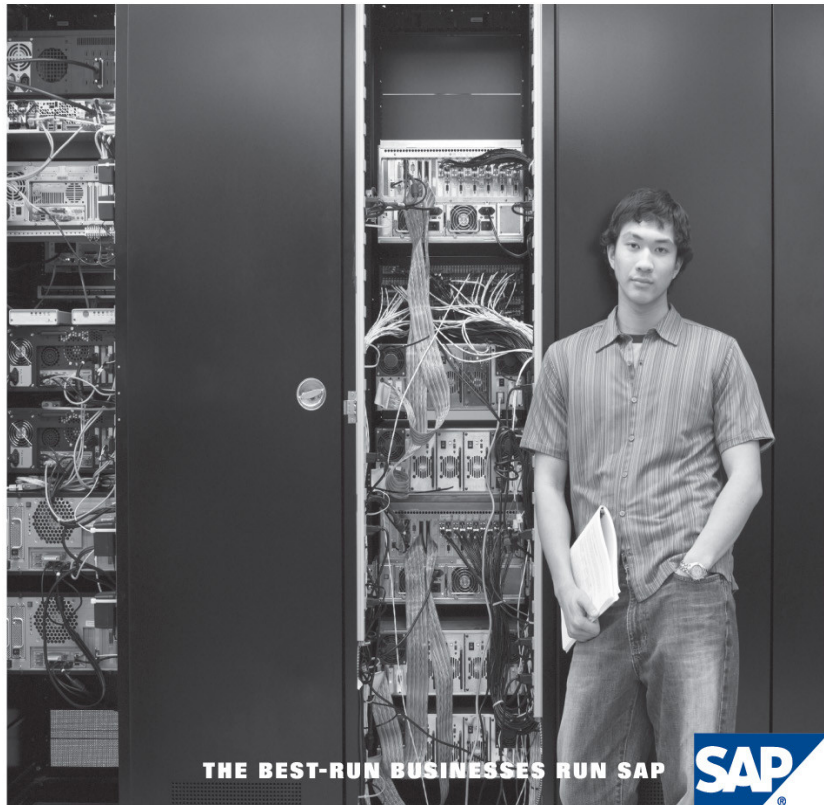
Life in EMACS will continue to change, but the crucial presumption will be whether the boundary of the innovation network can be maintained to sustain the innovation. For instance, Gosling did not continue to share his codes, rather, he sold his EMACS version to a commercial company. His version of EMACS therefore did not continue to grow. This is not to say that selling software to a commercial company will kill the product. There are other reasons for the elimination of a software product and sometimes the involvement of commercial companies does provide other sources for sustaining or developing a product. But in the case of Gosling Emacs: 1) He thought selling the product to a firm might broaden the network. But the transaction symbolised Gosling's failure to continue the network expansion. 2) The commercial product was not successful. The firm failed to engage actors' interest in it. The commercialised software forms a firmer boundary than FLOSS community projects to exclude outsiders of the developing team from the innovation. In that case, problems will not be triggered that easily. If problems are the initiatives of innovation, a less diverse character in the team will not help the innovation at that point. When GNU Emacs was just invented and still in an unstable stage, it did not put users off, instead, the existing problems invited peers to tackle them. GNU Emacs was able to engage actors in its social network of innovation. The network expanded and a variety of functions were developed. These functions become cornerstones to attract more actors/users as if a snowball effect. Unlike some proprietary software firms try to lock users in by using propriety document formats (and critics say they do this in order to dominate the market), EMACS engage users by presenting greater shared interests in socio-cultural or technical aspects. The story of EMACS sheds some light on the FLOSS innovation, albeit the commercial impetus should be taken into account in order to understand the situation completely.

## Reference



- [1] Borgman, C. L. 2003. "Designing Digital Libraries for Usability". In Ann P. Bishop, Nancy A. Van House & Barbara P. Buitendijk (Eds) *Digital Library Use: Social Practice in Design and Evaluation*. London: The MIT Press.
- [2] Bowker G. C. & Star, S. L. 1999. *Sorting Things Out: Classification and its Consequences*. London: The MIT Press.
- [3] Ceruzzi, P. 2003. *A History of Modern Computing*. 2nd edition. The MIT press.
- [4] Debian Documentation Team. 2003. *A Brief history of Debian*. URL (consulted 27 November 2003): <http://www.debian.org/doc/manuals/project-history/project-history.en.txt>
- [5] Glass, A. L., K. J. Holyoak, and J. L. Santa. 1979. *Cognition*. Reading, MA: Addison-Wesley.
- [6] Latour, B. 1987. *Science in action: How to follow scientists and engineers through society*. Cambridge, MA: Harvard University Press.
- [7] Lave, J. 1988. *Cognition in Practice: Mind, Mathematics, and Culture in Everyday Life*. Cambridge: Cambridge University Press.
- [8] Liff, S. and Steward, F. 2003. "Shaping e-access in the cybercafe: networks, boundaries and heterotopian innovation. *New Media & Society*, vol 5(3): 313-334.
- [9] McKelvey, M. 2001. "Internet Entrepreneurship: Linux and the dynamics of open source software " , CRIC Discussion Paper no. 44. Centre for Research on Innovation and Competition, The University of Manchester & UMIST.
- [10] Moody, G. 2002. *Rebel Code: Linux and the Open Source Revolution*. London: Penguin Books.
- Reitman, W. 1964. *Heuristic Decision Procedures, Open Constraints, and the Structure of Ill-Defined Problems*. In M. W. Shelley and G. L. Bryan, (eds.), *Human Judgment and Optimality*. New York: Wiley.
- [11] Simon, H. 1973. *The Structures of Ill Structured Problems*. *Artificial Intelligence*, 4, 181-201.
- [12] Stallman, R. "The Emacs Full-Screen Editor " . URL (consulted 27 November 2003): <http://www.lysator.liu.se/history/garb/txt/87-1-emacs.txt>
- [13] Stallman, R. 1995. *GNU Emacs Manual*. 11th Edition, Version 19.29. Boston: Free Software Foundation.
- [14] Stallman, R. 1998. "EMACS: The Extensible, Customizable Display Editor", 11 February, URL (consulted November 2003): <http://www.gnu.org/software/emacs/emacs-paper.html>
- [15] Stallman, R. 2002. *My Lisp Experiences and the Development of GNU Emacs* (transcript of Richard Stallman's Speech, 28 Oct 2002, at the International Lisp Conference). URL (consulted December 2003): <http://www.gnu.org/gnu/rms-lisp.html>
- [16] Star, S. L., Bowker, G. C., Neumann L. J. 2003. "Transparency beyond the Individual Level of Scale: Convergence between Information Artifacts and Communities of Practice " , in the book *Digital library use: social practice in design and evaluation*, edited by Ann Peterson Bishop, Nancy A. Van House, and Barbara P. Buitendijk. The MIT press.
- [17] Williams, S. 2002. *Free as in freedom: Richard Stallman's crusade for free software*. Sebastopol, CA: O'Reilly.
- [18] Zawinski, J. 2003. *Emacs Timeline*. URL (consulted 16 December 2003): <http://www.jwz.org/doc/emacs-timeline.html>

# 希望轻松集成的企业 首选SAP商务解决方案



THE BEST-RUN BUSINESSES RUN SAP



每一个领导团队都希望获得节节攀升的收益。促进信息共享，优化管理模式，并降低运营成本是提升企业效率和盈利能力的关键所在。让完全了解您需求的SAP商务解决方案，以实力助您获益丰厚。

现在，无论何种规模和业务形态的企业，都能轻松拥有功能强大、全面集成、无缝升级且实施快捷的世界级解决方案。30多年来，基于真正开放的集成化技术平台，SAP凭借服务于全球众多领先企业的丰富实践经验，发展出一系列融合了行业成功经验，并可满足您不同阶段发展需要的商务解决方案，能够全面覆盖财务管理、供应链、客户关系、人力资源等各个领域，助您整合核心业务，优化客户服务，赢得利润增长。

世界财富500强80%以上的优秀企业正从SAP商务解决方案中获益。欢迎您加入成功者的行列，请即访问 [www.sap.com/china](http://www.sap.com/china)，或拨打免费咨询热线800-820-0727。

# Single-Chip Digitally Controlled Data-Acquisition as Core of Reliable DWDM Communication Systems

Mark Malaeb  
Analog Devices Inc

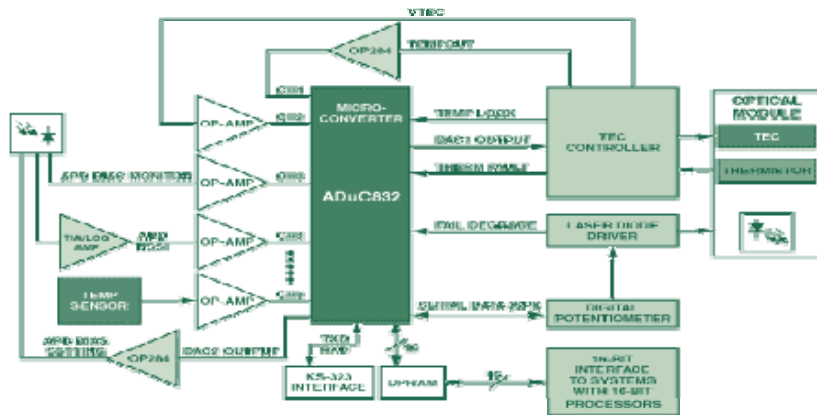
The recent lifting of government regulations on the communications industry has fueled an explosion of new ideas and inventions, especially in the optical space. Many players, from startups to Fortune 500 companies, participated in the 1998 to 2000 growth period in an attempt to implement these ideas. Today, although the industry is going through a painful period and many of the startups are gone, the need for implementing these ideas and inventions is still alive and well.

Optical fiber is the transport medium that has emerged to accommodate the expected long-term growth. In order to make the most of the wide bandwidth, which is the principal advantage of fiber transport, a method called wave-division multiplexing (WDM) and later, dense WDM (DWDM) as implemented. This method of transmission allows the transport of multiple wavelengths through a single fiber, but imposes stringent requirements on lasers.

One major requirement is that the laser temperature be held constant so that its wavelength will not drift and interfere with other lasers. This usually involves a thermoelectric-cooler (TEC) controller (addressed later in this article). Controlling the laser power level and its modulation mechanism over time and temperature is another system requirement. This job is handled by the laser-diode driver (LDD). For long-haul applications, optical amplifiers are needed for signal reconditioning and retransmitting. Erbium-doped fiber amplifiers (EDFAs) and Raman amplifier types predominate. They recondition the signal without having to convert it from optical to electrical, and then back to optical which was the widely used method (and only option) in the past. The amplification is now done solely in the optical domain. However, to control their parameters, optical amplifiers, need high-end A/D and D/A converters (ADCs and DACs), logarithmic amplifiers (log amps), and transimpedance amplifiers (TIAs), plus a controller.

## **Optical Communications Systems**

Most optical communication systems (Figure 1) use ADCs and DACs in the control loops used for the thermoelectric cooler (TEC), laser diode driver (LDD), and avalanche-photodiode (APD) monitoring and biasing circuitry. Dedicated control loops are used for driving the pump laser and reading its power level; adjusting the extinction ratio and average power in the transmit laser diode; and maintaining the laser diode at a stable temperature in WDM and DWDM systems. Loop signals are digitally processed by a microcontroller. Containing all these tasks in its portfolio, the ADuC832, a member of the MicroConverter<sup>™</sup> family integrated data-acquisition chip, combined with a CPU core and standard peripherals a strong candidate to serve as the heart of such control systems.

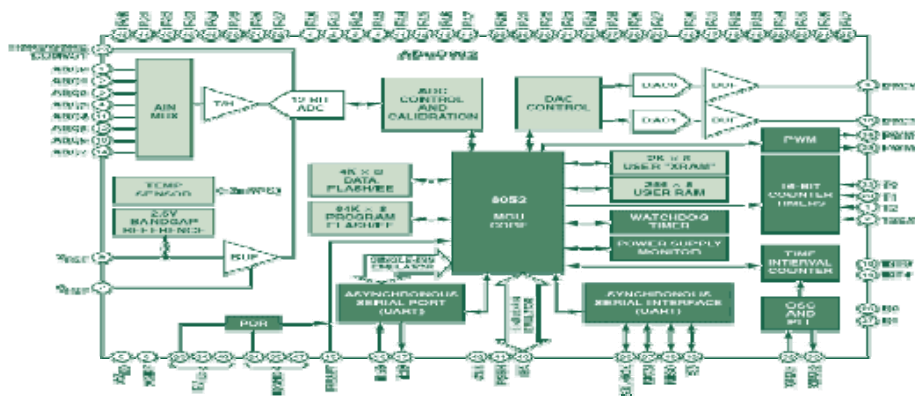


**Figure 1. Overall system block diagram.**

The ADuC832 (Figure 2) includes: an 8-channel, 12-bit, 5- $\mu$ s self-calibrating ADC; a 2-channel, 12-bit DAC with rail-to-rail outputs; an industry-standard 8052 microcontroller; 62 Kbytes of Flash program-RAM, and a host of peripheral functions. All of these features, plus a temperature monitor, programmable PLL clock, voltage reference, synchronous and asynchronous serial ports integrated into a space-saving 56-lead chip-scale package (CSP), allowing the entire control system to fit within the housing of an optical module.

We describe here the possible practical implementation of the various portions of an optical communication system, as shown in Figure 1, including a thermoelectric-cooler controller, laser-diode drivers (LDDs), photodetector-diode biasing, received optical signal-strength indicator (RSSI), and temperature sensors.

Complete, fully tested software modules are available for each of these applications. These modules are written and commented especially for analog/optical designers who are not software savvy and never want to be. This will help cut design time and overall time to market for both experienced and novice programmers.

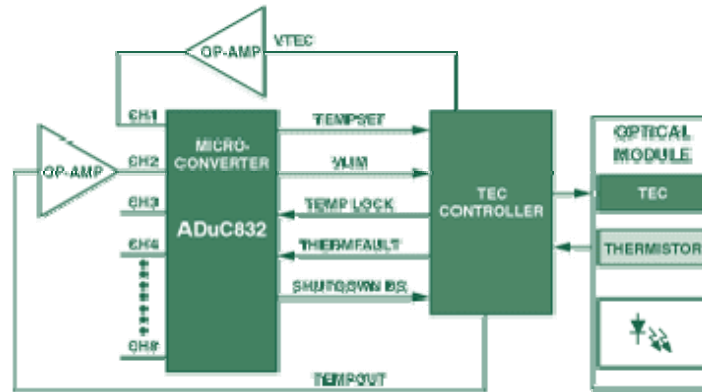


**Figure 2. MicroConverter block diagram.**

**TEC Controller**

To ensure wavelength stability, the thermoelectric-cooler (TEC) controller is designed to maintain a laser diode or optical module at a specific temperature, with precision typically to

within 0.01. This device relies on a negative-temperature-coefficient (NTC) thermistor to sense the temperature of the object attached to the TEC. The target temperature is set with an analog input voltage from a DAC, as shown in Figure 3, or with an external resistor divider. A positive or negative current is applied to the TEC, either heating or cooling the object as required. An internal proportional-integral-derivative (PID) compensation amplifier stabilizes the temperature. An external PID network can be adjusted to optimize the settling time.

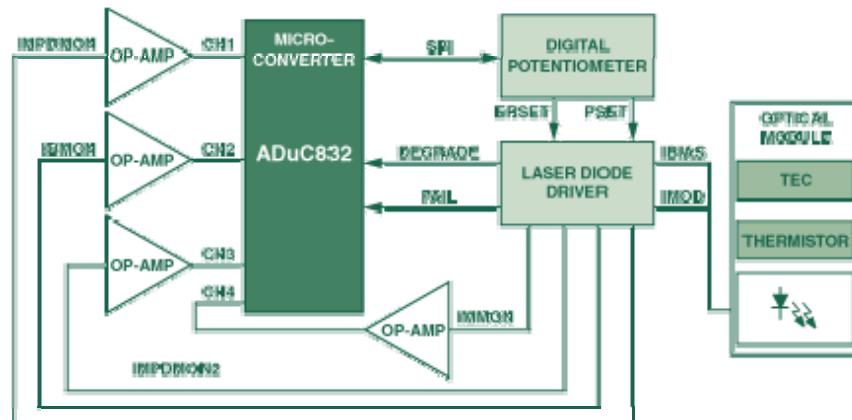


**Figure 3. The MicroConverter sets the TEC target temperature and monitors its performance.**

One of the 12-bit DACs, connected to the TEMPSET pin, sets the target temperature. The voltage corresponds to a specific target temperature (typically 25 mV for most laser diode applications). The TEC, through its PID loop, maintains the target temperature to within 0.01 (naturally, this value is dependent on the thermistor quality and suitable care). To protect the thermoelectric cooler from overvoltage, a maximum voltage is specified at the VLIM pin. This voltage is supplied and set by the second 12-bit DAC (typical range 0-to-1.5 V) or by a simple resistance-divider network. The VTEC pin is connected to one of the eight ADC channels, allowing the actual voltage across the TEC to be monitored. The TEMPOUT pin is connected to the second ADC channel, allowing the TEC temperature to be dynamically monitored (typical range 0-3 V). An op amp (OP184, OP162, etc.) is used ahead of each of the ADC inputs for signal scaling, buffering and filtering, if needed. Two digital inputs are used: TEMPLOCK indicates that the desired TEC temperature has been reached; THERMFAULT is used to indicate a thermistor malfunction. One digital output is connected to the SD pin, allowing the device to be put into a low-current shutdown mode.

### Laser Diode Driver

As part of the system, a laser-diode driver (LDD), the ADN2841, is used to control both average power and extinction ratio of the laser diode using a dual-loop control scheme. The extinction ratio is controlled as a function of modulation current, while the average power is controlled by the bias current. This dual loop configuration compensates for the variation in laser diode slope efficiency due to temperature, aging and diode-to-diode production tolerances, and enhances the designer's ability to source laser diodes from multiple manufacturers.



**Figure 4. The MicroConverter sets the average power and extinction ratio of the laser diode, and monitors alarms.**

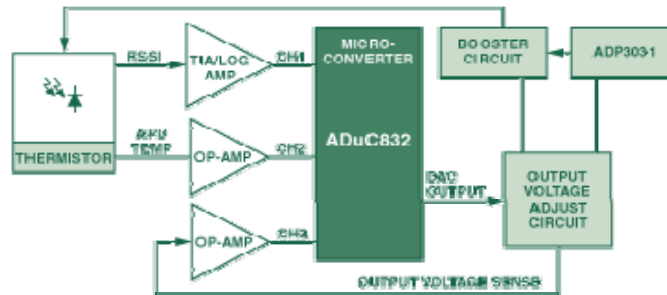
The MicroConverter is used for both control and monitoring. Figure 4 shows that the extinction ratio (at the ERSET pin) and the average power (at the PSET pin) are set using an ADN2850 dual 10-bit digital pot. This pot is controlled through the serial peripheral interface (SPI) port. The monitor photodiode currents, IMPDMON and IMPDMON2, flow to ground through 1.5-k  $\Omega$  resistors to provide voltages proportional to the monitor currents. The diode's modulation and bias currents, IMON and IBMON, flow to ground in the same way. The voltages developed across these resistors are connected to the ADC channels, making them available in a digital format. The ADN2841 has active high monitoring alarms to identify degraded-diode and failed diode conditions. These two digital outputs are connected to two general-purpose I/O pins.

### APD Monitoring and Biasing

Avalanche photodiodes are known for their extreme sensitivity and high internal gain. These characteristics make them ideal for optical receiver applications that require optimum sensitivity. Unfortunately, such high gains necessitate bias voltages ranging from 40 to 60 volts. The designer must also deal with the fact that the APD gain function is temperature dependent. The gain can be held constant by adjusting the APD bias voltage as the temperature changes. Figure 5 shows how an ideal solution to this problem can be implemented with a DAC, a switching regulator, and an adjustable voltage booster circuit.

The gain can be held constant by increasing the APD bias voltage as the temperature increases, as specified by the APD manufacturer.

The ADP3031 switching regulator can provide output voltages of up to 12 volts. Several ADP3031 boost stages can be cascaded to achieve the desired final voltage.

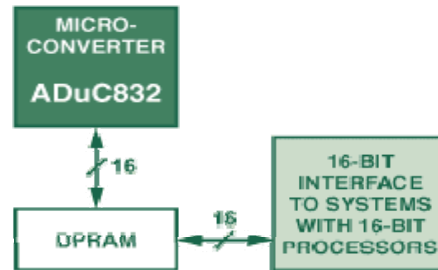


**Figure 5. The ADuC832 controls the APD biasing and gain, and reads the received signal strength indicator.**

A DAC is used in the range of 0-to-2 V to vary the voltage across the diode with temperature. The actual voltage across the diode can be monitored with the A/D, thus providing complete closed-loop control. With the diode gain maintained at its target, the received optical signal strength can then be accurately monitored with a transimpedance amplifier (TIA) or a log amp plus another channel of the ADC. Calibration coefficients can be conveniently stored in Flash data memory, enabling adjustments to be made as needed.

### The 16-Bit Interface

This interface is accomplished using general-purpose I/O ports 0 and 2, plus an external latch. It serves two main functions: as a memory interface for the ADC when it is run in direct memory access (DMA) mode; and as a general-purpose 16-bit peripheral interface through the use of a dual-port RAM or other type of memory. Figure 6 shows the details of such a configuration.



**Figure 6. The MicroConverter and its 16-bit interface for data exchange.**

The ADC runs at a maximum update rate of one conversion per 5  $\mu$ s. At this rate, the microcontroller has 5  $\mu$ s to read the ADC result and store it in memory for further processing. It has to be done within this time interval or the next sample will be lost especially time-consuming when using an interrupt routine for the ADC. Thus, in applications where the MicroConverter cannot service a very fast interrupt rate, DMA mode should be used. In DMA mode, ADC results are written directly to external memory.

Like any standard 8051-compatible controller, this 16-bit interface can be used to exchange data with systems running a 16-bit processor. The dual port memory helps prevent bus alteration and contention and provides a somewhat independent interface system. Because "real estate" (circuit-board area) is critical in most optical modules, integration of ADCs, DACs, Flash memory, and an 8052 MCU in a 56-lead CSP package provides the designer with a compact and powerful solution for optical communication systems.



大唐微电子技术有限公司  
Datang Microelectronics Technology Co.,Ltd.

我们创造"芯"的世纪

大唐微电子拥有完全自主知识产权的智能卡芯片及软件，  
近年来连续推出自主创新的OTA技术、  
超级号簿、短信群发、动态分配、一卡多号及名片册等多项卡端业务。

作为中国国内唯一一家能够同时在芯片级、模块级、卡级向客户提供全方位产品及增值业务服务的集成电路设计企业，大唐微电子将凭借其强大的卡类芯片设计实力，与合作伙伴一道，携手共创未来！



# Towards Next Generation URLs

## Port80 Software

Thomas A. Powell and Joe Lima

**F**or many years we have heard about the impending death of URLs that are difficult to type, remember and preserve. The use of URLs has actually improved little thus far, but changes are afoot in both development practices and Web server technology that should help advance URLs to the next generation.

### **Dirty URLs**

Complex, hard-to-read URLs are often dubbed dirty URLs because they tend to be littered with punctuation and identifiers that are at best irrelevant to the ordinary user. URLs such as `http://www.example.com/cgi-bin/gen.pl?id=4&view=basic` are commonplace in today's dynamic Web. Unfortunately, dirty URLs have a variety of troubling aspects, including:

- **Dirty URLs are difficult to type.**

The length, use of punctuation, and complexity of these URLs makes typos commonplace.

- **Dirty URLs do not promote usability.**

Because dirty URLs are long and complex, they are difficult to repeat or remember and provide few clues for average users as to what a particular resource actually contains or the function it performs.

- **Dirty URLs are a security risk.**

The query string which follows the question mark (?) in a dirty URL is often modified by hackers in an attempt to perform a front door attack into a Web application. The very file extensions used in complex URLs such as `.asp`, `.jsp`, `.pl`, and so on also give away valuable information about the implementation of a dynamic Web site that a potential hacker may utilize.

- **Dirty URLs impede abstraction and maintainability.**

Because dirty URLs generally expose the technology used (via the file extension) and the parameters used (via the query string), they do not promote abstraction. Instead of hiding such implementation details, dirty URLs expose the underlying "wiring" of a site. As a result, changing from one technology to another is a difficult and painful process filled with the potential for broken links and numerous required redirects.

### **Why Use Dirty URLs?**

Given the numerous problems with dirty URLs, one might wonder why they are used at all. The most obvious reason is simply convention -- using them has been, and so far still is, an accepted practice in Web development. This fact aside, dirty URLs do have a few real benefits, including:

- **They are portable.**

A dirty URL generally contains all the information necessary to reconstruct a particular dynamic query. For example, consider how a query for "web server software" appears in Google -- `http://www.google.com/search?hl=en&ie=UTF-8&oe=UTF-8&q=Web+server+software`. Given this URL, you can rerun the query at any time in the future. Though difficult to type, it is easily bookmarked.

- **They can discourage unwanted reuse.**

The negative aspects of a dirty URL can be regarded as positive when the intent is to discourage the user from typing a URL, remembering it, or saving it as a bookmark. The intimidating look and length of a dirty URL can be a signal to both user and search engine to stay away from a page that is bound to change. This is often simply a welcome side effect, rather than a conscious access control policy -- frequently nothing is done to prevent actual use of the URL by means of session variables or referring URL checks.

### **Cleaning URLs**

The disadvantages of dirty URLs far outweigh their advantages in most situations. If the last 30 or 40 years of software development history are any indication of where development for the Web is headed, abstraction and data hiding will inevitably increase as Web sites and applications continue to grow in complexity. Thus, Web developers should work toward cleaner URLs by using the following techniques:

- **Keep them short and sweet.**

The first path to better URLs is to design them properly from the start. Try to make the site directories and file names short but meaningful. Obviously, `/products` is better than `/p`, but resist the urge to get too descriptive. Having `www.xyz.com/productcatalog` doesn't add much meaning (if a user looks for a product catalog, they might well expect to find it at or near the top-level products page), but it does needlessly restrict what the page can reasonably contain in the future. It's also harder to remember or guess at. Shoot for the shortest identifiers consistent with a general description of the page's (or directory's) contents or function.

- **Avoid punctuation in file names.**

Often designers use names like `product_spec_sheet.html` or `product-spec-sheet.html`. The underscore is often difficult to notice and type, and these connectors are usually a sign of a carelessly designed site structure. They are only required because the last rule wasn't followed.

- **Use lower case and try to address case sensitivity issues.**

Given the last tip, you might instead name a file `ProductSpecSheet.html`. However, casing in URLs is troubling because depending on the Web server's operating system, file names and directories may or may not be case sensitive. For example, `http://ww.xyz.com/Products.html` and `http://www.xyz.com/products.html` are two different files on a UNIX system but the same file on a Windows system. Add to this the fact that `www.xyz.com` and `WWW.XYZ.COM` are always the same domain, and the potential for confusion becomes apparent. The best solution is to make all file and

directory names lowercase by default and, in a case sensitive server operating environment, to ensure that URLs will be correctly processed no matter what casing is used. This is not easy to do under Apache on Unix/Linux systems (related info), although URL rewriting and spellchecking can help (discussed below).

- **Do not expose technology via directory names.**

Directory names commonly or easily associated with a given server-side technology unnecessarily disclose implementation details and discourage permanent URLs. More generic paths should be used. For example, instead of /cgi-bin, use a /scripts directory, instead of /css, use /styles, instead of /javascript, use /scripts, and so on.

- **Plan for host name typos.**

The reality of end user navigation is that around half of all site traffic is from direct type or bookmarked access. If users want to go to Amazon's web site, they know to type in www.amazon.com. However, accidentally typing ww.amazon.com or wwwwww.amazon.com is fairly easy if a user is in a hurry. Adding a few entries to a site's domain name service to map w, ww, and www to the main site, as well as the common www.site.com and site.com, is well worth the few minutes required to set them up.

- **Plan for domain name typos.**

If possible, secure common "fat finger" typos of domain names. Given the proximity of the "z" and "x" keys on a standard computer QWERTY keyboard, it is no wonder Amazon also has contingency domains like amaxon.com. Google allows for such variations as gooogle.com and gogle.com. Unfortunately, many Web traffic aggregators will purchase the typo domains for common sites, but most organizations should find some of their typo domains readily available. Organizations with names that are difficult to spell, like "Ximed," might want to have related domains like "Zimed" or "Zymed" for users who know the name of the organization but not the correct spelling. The particular domains needed for a company should reveal themselves during the course of regular offline correspondence with customers.

- **Support multiple domain forms.**

If an organization has many forms to its name, such as International Business Machines and IBM, it is wise to register both forms. Some companies will register their legal form as well, so XYZ, LLC or ABC, Inc. might register xyzllc.com and abcinc.com as well as primary domains. While it seems like a significant investment, if you use one of the new breed of low-cost registrars (like itsyourdomain.com), the price per year for numerous domains for a site is quite reasonable. Given alternate domain extensions like .net, .org, .biz and so on, the question begs -- where to stop? Anecdotally, the benefits are significantly reduced with new alternate domain forms (like .biz, .cc, and so on), so it is better to stick with the common domain form (.com) and any regional domains that are appropriate (e.g. co.uk).

- **Add guessable entry point URLs.**

Since users guess domain names, it is not a stretch for users -- particularly power users -- to guess directory paths in URLs. For example, a user trying to find information about Microsoft Word might type http://www.microsoft.com/word. Mapping multiple URLs to

common guessable site entry points is fairly easy to do. Many sites have already begun to create a variety of synonym URLs for sections. For example, to access the careers section of the site, the canonical URL might be <http://www.xyz.com/careers>. However, adding in URLs like <http://www.xyz.com/career>, <http://www.xyz.com/jobs>, or <http://www.xyz.com/hr> is easy and vastly improves the chances that the user will hit the target. You could even go so far as to add hostname remapping so that <http://investor.xyz.com>, <http://ir.xyz.com>, <http://investors.xyz.com>, and so on all go to <http://www.xyz.com/investor>. The effort made to think about URLs in this fashion not only improves their usability, but should also promote long term maintainability by encouraging the modularization of site information.

- **Where possible, remove query strings by pre-generating dynamic pages.**

Often, complex URLs like <http://www.xyz.com/press/releasedetail.asp?pressid=5> result from an inappropriate use of dynamic pages. Many developers use server-side scripting technologies like ASP/ASP.NET, ColdFusion, PHP, and so on to generate "dynamic" pages which are actually static. For example in the previous URL, the ASP script drills press release content out of a database using a primary key of 5 and generates a page. However, in nearly all cases, this type of page is static both in content and presentation. The generation of the page dynamically at user view time wastes precious server resources, slows the page down, and adds unnecessary complexity to the URL. Some dynamic caches and content distribution networks will alleviate the performance penalty here, but the unnecessarily complex URLs remain. It is easy to directly pre-generate a page to its static form and clean its URL. Thus, <http://www.xyz.com/press/releasedetail.asp?pressid=5> might become [www.xyz.com/press/pressrelease5](http://www.xyz.com/press/pressrelease5) or something much more descriptive like <http://www.xyz.com/press/03-02-2003> -- or even better like <http://www.xyz.com/press/newproduct>. The issue of when to generate a page, either at request time or beforehand, is not much different than the question of whether a program should be interpreted or compiled.

- **Rewrite query strings.**

In the cases where pages should be dynamic, it is still possible to clean up their query strings. Simple cleaning usually remaps the ?, &, and + symbols in a URL to more readily typeable characters. Thus, a URL like <http://www.xyz.com/presssearch.asp?key=New+Robot&year=2003&view=print> might become something like <http://www.xyz.com/presssearch.asp/key/New-Robot/year/2003/view/print>. While this makes the page "look" static, it is indeed still dynamic. The look of the URL is a little less intimidating to users and may be more search engine friendly as well (search engines have been known to halt at the ? character). In conjunction with the next tip, this might even discourage URL parameter manipulation by potential site hackers who can't tell the difference between a dynamic page and a static one. The challenge with URL rewriting is that it takes some significant planning to do well, and the primary tools used for these purposes -- rule-based URL rewriters like `mod_rewrite` for Apache and ISAPI Rewrite for IIS -- have daunting rule syntax for developers unseasoned in the use of regular expressions. However, the effort to learn how to use these tools properly is well worth it.

- **Remove extensions from files in URL and source.**

Probably the most interesting URL improvement that can be made involves the concept of content negotiation. Despite being a long-supported HTTP specification, content negotiation is rarely used on the Web today. The basic idea of content negotiation is that the browser transmits information about the resources it wants or can accept (MIME types preferred, language used, character encodings supported, etc.) to the server, and this information is then used, along with server configuration choices, to dynamically determine the actual content and format that should be transmitted back to the browser. Metaphorically, the browser and the server hold a negotiation over which of the available representations of a given resource is the best one to deliver, given the preferences of each side. What this means is that a user can request a URL like `http://www.xyz.com/products`, and the language of the content returned can be determined automatically -- resulting in the content being delivered from either a file like `products-en.html` for English speaking users or one like `products-es.html` for Spanish speakers. Technology choices such as file format (PNG or GIF, xhtml or HTML) can also be determined via content negotiation, allowing a site to support a range of browser capabilities in a manner transparent to the end user.

Content negotiation not only allows developers to present alternate representations of content but has a significant side effect of allowing URLs to be completely abstract. For example, a URL like `http://www.xyz.com/products/robot`, where `robot` is not a directory but an actual file, is completely legal when content negotiation is employed. The actual file used, be it `robot.html`, `robot.cfm`, `robot.asp`, etc., is determined using the negotiation rules. Abstracting away from the file extension details has two significant benefits. First, security is significantly improved as potential hackers can't immediately identify the Web site's underlying technology. Second, by abstracting the extension from the URL, the technology can be changed by the developer at will. If you consider URLs to be effectively function calls to a Web application, cleaned URLs introduce the very basics of data hiding.

URLs can be cleaned server-side using a Web server extension that implements content negotiation, such as `mod_negotiation` for Apache or `PageXchanger` for IIS. However, getting a filter that can do the content negotiation is only half of the job. The underlying URLs present in HTML or other files must have their file extensions removed in order to realize the abstraction and security benefits of content negotiation. Removing the file extensions in source code is easy enough using search and replace in a Web editor like Dreamweaver MX or HomeSite. Some tools like `w3compiler` also are being developed to improve page preparation for negotiation and transmission. One word of assurance: don't jump to the conclusion that your files won't be named `page.html` anymore. Remember that, on your server, the precious extensions are safe and sound. Content negotiation only means that the extensions disappear from source code, markup, and typed URLs.

- **Automatically spell check directory and file names entered by users.**

The last tip is probably the least useful, but it is the easiest to do: spell check your file and directory names. On the off chance that a user spells a file name wrong, makes a typo in extension or path, or encounters a broken link, recovery is easy enough with a spelling check. Given that the typo will start to generate a 404 in the server, a spelling

module can jump in and try to match the file or directory name most likely typed. If file and directory names are relatively unique in a site, this last ditch effort can match correctly for numerous typos. If not, you get the 404 as expected. Creating simple "Did you mean X?"-style URLs requires the simple installation of a server filter like `mod_speling` for Apache or `URLSpellCheck` for IIS. The performance hit is not an issue, given that the correction filter is only called upon a 404 error, and it is better to result in a proper page than serve a 404 to save a minor amount of performance on your error page delivery. In short, there is no reason this shouldn't be done, and it is surprising that this feature is not built-in to all modern Web servers.

## Conclusions

Most of the tips presented here are fairly straightforward, with the partial exception of URL cleaning and rewriting. All of them can be accomplished with a reasonable amount of effort. The result of this effort should be cleaned URLs that are short, understandable, permanent, and devoid of implementation details. This should significantly improve the usability, maintainability and security of a Web site. The potential objections that developers and administrators might have against next generation URLs will probably have to do with any performance problems they might encounter using server filters to implement them or issues involving search engine compatibility. As to the former, many of the required technologies are quite mature in the Apache world, and their newer IIS equivalents are usually explicitly modeled on the Apache exemplars, so that bodes well. As to the search engine concerns, fortunately, Google so far has not shown any issue at all with cleaned URLs. At this point, the main thing standing in the way of the adoption of next generation URLs is the simple fact that so few developers know they are possible, while some who do are too comfortable with the status quo to explore them in earnest. This is a pity, because while these improved URLs may not be the mythical URN-style keyword always promised to be just around the corner, they can substantially improve the Web experience for both users and developers alike in the long run.

## Further Resources

### Articles

Numerous articles have been written about the need for clean URLs. A few of the more prominent ones are cited here.

- Tim Berners Lee's Cool URIs don't change
- Jakob Nielsen on URL as UI
- Bill Humphries @ Zeldman's on URLs! URLs! URLs!
- Clean Up Those URLs with Apache
- Making "clean" URLs with Apache and PHP
- Search Engine Friendly URLs with PHP and Apache: Part I and Part II

### Apache Tools

For Apache, nearly all modules can be found at [modules.apache.org](http://modules.apache.org).

Links to useful information about `mod_rewrite` can be found at [modrewrite.com](http://modrewrite.com).

A good overview of content negotiation on Apache can be found at <http://httpd.apache.org/docs/content-negotiation.html>.

### **Microsoft IIS Tools**

IIS does not quite have the strong module culture Apache does, but the site [iismodules.com](http://iismodules.com) lists many commercially and freely available modules, and [iisanswers.com](http://iisanswers.com), [iisfaq.com](http://iisfaq.com), and [iis-resources.com](http://iis-resources.com) have related links and more detailed information on filter use on IIS.

The specific commercial IIS products mentioned in the article include URLSpellCheck, ISAPI Rewrite, PageXchanger, and w3compiler.

The authors would encourage submission of other tools and articles to improve the article's resource listing.

### **About the Authors**

Thomas Powell is founder of PINT, Inc. and a lecturer in the Computer Science department at University of California San Diego. His articles have appeared in several magazines and sites, including Network World, Internet Week and ZDNet. He has also published numerous books on Web technology and design, including the best-selling Web Design: The Complete Reference. Visit [pint.com](http://pint.com).

Joe Lima is the Director of Product Development for Port80 Software. He has worked for a variety of Internet, wireless and software development companies, specializing in research and development for server-centric technologies. Visit [port80software.com](http://port80software.com).



**上海贝岭股份有限公司**  
Shanghai Beiling Corp. Ltd.

# 测试技术服务

芯片或器件测试验证业务:

用户电话或email联系商务人员提出申请要求进行有关芯片或器件的测试, 并提交测试技术资料, 签订保密协议。

测试部分析用户提供的技术资料, 确定自己是否有能力完成相关测试, 并由商务人员就价格问题与用户展开协商。

NO → 终止

yes  
双方就协商内容填写测试服务申请书, 签订测试技术合同。

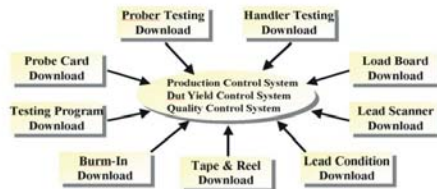
用户直接与测试部指定工程师进行技术交流, 并提供测试详细资料, 包括测试码。

测试部工程师进行程序开发, 10个工作日后进行器件测试, 并协同用户进行样品分析与数据采集。

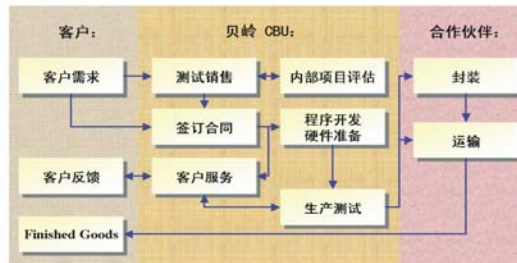
在双方确认测试结果后, 测试部提供测试分析报告。

合同完成

## Testing Foundry Control System



## 测试服务流程:



业务联系电话: 021-64850700 转 253 分机  
Email: test\_foundry@belling.com.cn

021-64954185  
联系人: 顾先生、戴先生

传真: 021-64950375

WEBSITE: www.belling.com.cn

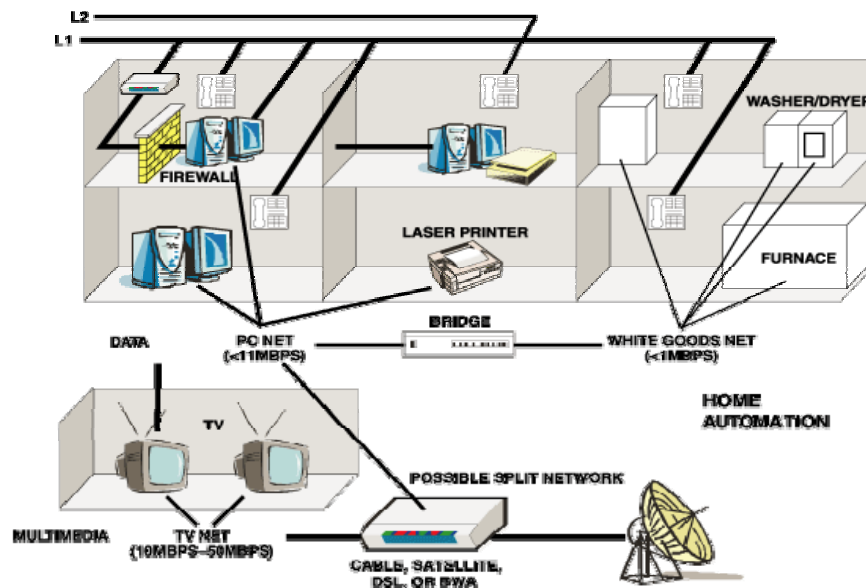


# Versatile Mixed-Signal Front Ends Speed Customized Design of Wireline Broadband Modems and Home Networks

Joseph DiPilato, Iuri Mehr, Brian Harrington  
Analog Devices Inc

## INTRODUCTION

The widespread growth of broadband modems (predominantly based on cable and DSL technology) has made broadband communications available to residential households throughout the world for use in applications such as internet access, interactive gaming, and telecommuting. The number of households with multiple personal computers (PCs) is also on the rise. The ready availability of these facilities has led to the increasing popularity of home networking for sharing Internet access and printer resources for work, academics, and entertainment. Figure 1 shows a typical domestic network connected to the broadband gateway and to various devices inside the residence/office.



**Figure 1. Home-networking/broadband access configuration.**

A variety of technologies have been developed to address high-speed communication between home computers and peripheral devices—and to interface to the hub (or gateway) for broadband access. Until recently, Ethernet has been the most viable method for providing networking within the home. The attractiveness of Ethernet has been the availability of inexpensive network interface cards based on proven technology. However, Ethernet suffers from one main disadvantage—the requirement of having Category 5 (CAT5) cable wired

throughout the home. This almost always means the homeowner must run new wires—a cumbersome and potentially expensive solution.

Wireless LAN provides an alternative to Ethernet that doesn't require installation of new wires. Despite the convenience of a wireless solution, it has achieved limited success due to higher cost, potential insecurity, multiple competing standards, lack of interoperability, and interference/robustness concerns. Recent advancements in the wireless standards within IEEE 802.11 and the emergence of the WiFi consortium have increased the prospects for wireless home networking. Nevertheless, the scarcity of frequency spectrum and the ubiquity of potent interferers—such as microwave ovens, garage door openers, etc.—will continue to pose numerous challenges to a wireless home network, including increased cost.

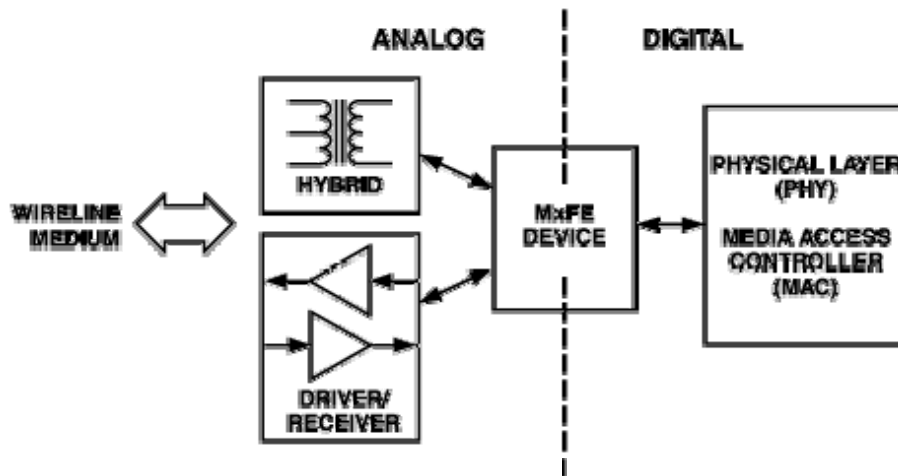
The newer preferred wireline technologies are those that require “no new wire”; they offer the homeowner a potentially better way of establishing a home network by avoiding the burden of installing new cabling, while providing comparable performance to that of Ethernet at an affordable price. The two choices for operating over the installed infrastructure involve either telephone- or the power-line wiring. Until recently, utilizing this existing home wiring for broadband networking was impossible due to very poor channel quality. However, cost-effective broadband home networking is now a reality because of advances in signal-processing techniques, lowered costs resulting from persistent reduction of silicon fabrication geometries, and performance improvements in high-speed mixed-signal circuitry. The Home Phone Networking Alliance (HPNA) standard provides the framework for phone-line networking and allows data rates of up to 10 Mbits/s. And the HomePlug™ standard describes the specifications for implementing a power-line network system with data rates comparable to those of HPNA.

These wireline home networking technologies, together with high-speed access technologies like DSL and cable, require mixed-signal interfacing between the transmission medium (power line, cable, twisted pair) and the digital baseband processors and controllers. Analog Devices has developed a family of monolithic mixed-signal front-end (MxFE™) integrated circuits (ICs) to bridge this gap (see Analog Dialogue Volume 35, Number 1, January-February, 2001 article, “AD9873 Mixed-Signal (MxFE) Front-End for Broadband Digital Set-Top Boxes”).

The recently introduced AD9875 and AD9876 MxFE devices, to be discussed in this article, were developed and aimed specifically at broadband home-networking and broadband-access applications, both of which require high data rates or signal bandwidth up to 25 MHz and impose similar demands in functionality, performance, and cost. Given their flexibility, these parts can be used in modems for both home networking (HomePlug, HPNA) and high-speed data access (VDSL, power line). Developed for large volume, cost-sensitive, consumer-class applications, they offer exceptional value to providers of system solutions.

Figure 2 depicts the versatile role played by the MxFE chip. Its receive (Rx) circuitry accepts the (analog-domain) signals from the transmission medium, once they have been appropriately interfaced, and provides analog signal conditioning and A/D conversion to produce multiplexed digital signals that can be dealt with by the digital physical layer (PHY) and/or media access controller (MAC). It also accepts digital data from these entities,

processes it and converts it to analog—and outputs this transmit (Tx) signal to the media interface. The device design is based on the objective of optimizing system performance—irrespective of the analog or digital nature of its I/O—as implemented in ADI’s “smart-partitioning methodology (see Appendix).



**Figure 2. Typical wireline network node.**

Figure 2 represents a typical wireline network node. The MxFE function may be connected to the wireline medium passively through a hybrid transformer block, or actively, with an amplifier on the receive side and a driver on the transmit side. On the digital side, the MxFE device needs to interface with the PHY and the MAC, which reside on a separate chip.

Critically important factors for the designer of MxFE circuitry for the home networking and high-speed access markets are performance, time to market, and low cost. In this type of emerging broadband communication application, discrete analog components can be prohibitively expensive, power-hungry, and demanding of board space. The combination of cost, performance, size, and power dissipation made possible by integrating the difficult mixed-signal, analog, digital, and signal processing functions on a single chip—such as the AD9875/AD9876—makes complex consumer-market communication products feasible. In fact, by implementing their intellectual property and design expertise in a digital gate array or ASIC, in conjunction with these devices, design engineers can develop new products and prototypes faster than ever to capitalize on rapidly changing markets.

Figure 3 shows a block diagram of the AD9875/AD9876 mixed-signal front-end converter for broadband modems. On the analog side, the AD9875 and AD9876 combine the ADC, DAC, clock generation, programmable-gain amplification, and analog and digital filtering circuitry to provide a level of integration and performance usually implemented with discrete solutions costing significantly more. The ADC uses a pipelined multistage architecture to achieve high sample rates while consuming low power. In the AD9875, the receive path provides 9.5 ENOB at 32 MSPS and 8.6 ENOB at 50 MSPS. Comparable numbers for the AD9876 are 10.2 ENOB @ 32 MSPS and 9.3 ENOB @ 50 MSPS. The DACs in each device are, respectively, 10- and 12-bit interpolating TxDAC circuits.

On the digital side, the DAC inputs and ADC outputs are available on separate ports to accommodate both full-duplex and half-duplex operations. Each converter's port is multiplexed into high- and low nybbles to reduce the number of package pins. [A 10-bit half-duplex device, the AD9875-HD, available in summer 2002, will connect seamlessly to currently available HomePlug-compliant physical-layer digital ASICs that support multiplexed transmit and receive data ports.]

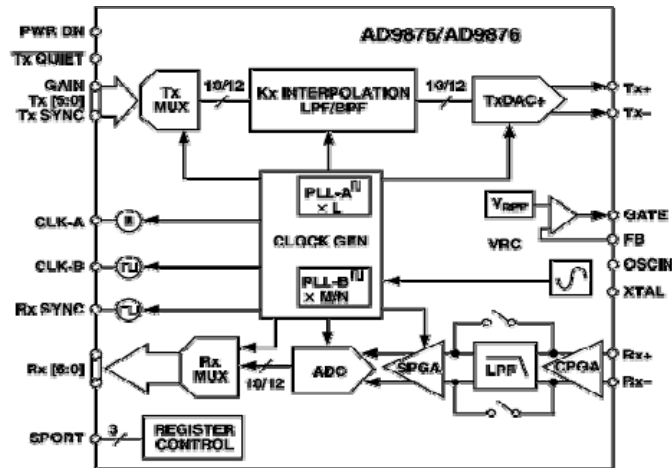


Figure 3. AD9875/6 block diagram.

The availability of low-cost, flexible, high-performance, off-the-shelf mixed-signal front-ends, such as these, with optimized functionality for broadband modem design using a variety of modulation formats—including OFDM—simplifies the ASIC design, specification, and testing process for both vendor and OEM and can substantially cut time to market. The unique features of the AD9875/AD9876 make these parts ideal for phone and power line networking as well as some xDSL applications.

### Broadband wireline modems

Wideband signals with high peak-to-average ratios, commonly found in broadband modems—irrespective of the modulation scheme employed—put great demands on the system's analog- and mixed-signal processing components. Figure 2 and the AD9875/AD9876 block diagram in Figure 3 together show details of a generic wireline broadband modem. Not every modem requires every block shown, although most of them do employ the same or similar functions, while others may even require additional functional blocks like additional analog filtering. The block-level components of a wireline modem can be grouped into three main functions; a transmit path, a line coupler or hybrid, and a receive path. Most of the transmit and receive blocks are addressed directly by the AD9875/AD9876 component.

The main function of the transmit path is to send the signal onto the line with sufficient fidelity to reach the far end of the wire with a high-enough signal-to-noise ratio (SNR) to allow faithful decoding at the receiver end. Invariably, the transmitter must also do this while conforming to a spectral mask to ensure that the modem does not cause excessive noise outside its channel bandwidth. Meeting the requirements of a power spectral-density (PSD) mask usually drives the design and performance requirements of the transmit path

components. The two converter parameters that get the most attention are the number of bits and the sampling rate.

For the DAC, the number of bits required will depend on the desired SNR, the signal's peak-to-average ratio (PAR) and the ratio of the signal bandwidth to the sample rate. For a given SNR requirement, higher-precision converters are required as signal bandwidths and PARs increase. Often the most demanding DAC performance parameter turns out to be its spurious-free dynamic range (SFDR). The spurs generated by the non-ideal DAC transfer function may show up anywhere in the spectrum. If the magnitude of the spurs are high and fall close to the signal band, they may be impossible to filter adequately. The DAC's SFDR performance must be able to meet the system linearity requirements.

The AD9875/AD9876 incorporates a 10-/12-bit DAC and makes use of interpolation filters to oversample the input data. Oversampling, because it moves DAC image frequencies away from the frequency of the desired signal, may result in substantially simpler external analog filter requirements, which translate into lower complexity and cost. Ideally, the driver will be able to provide the transmit-path gain and deliver the required output power while maintaining the DAC linearity performance. In cases where the DAC's available peak output power is not sufficient, AD832x cable drivers or AD8xxx DSL drivers could provide the interface to the wire. To deliver high peak-output signal with low distortion requires high voltage rails and high bias currents in the amplifier's output stage, which conflicts with the need for low power consumption and CMOS integration.

The method of coupling the modem's analog front-end to the line depends on whether the type of modulation/demodulation (modem) is time-domain duplex (TDD) or frequency-domain duplex (FDD). A TDD modem will normally employ a simple transformer coupling to the line and a switch that connects either the transmitter or receiver to the transformer. The main concern is that the switching and settling times, when the connections are made, meet the system requirements. An FDD modem will normally employ a hybrid to connect the modem's analog front-end to the line. A hybrid is required because the modem can be transmitting large signals while the receiver is listening to greatly attenuated signals—which can be orders of magnitude smaller. In order to limit the amount of signal coupling from the transmitter to the receiver, some type of line matching, cancellation circuitry, and filtering is used. The AD9875/AD9876 integrates a low-pass analog receive filter (LPF in Figure 3) with variable cutoff to provide for various signal bandwidth requirements.

The effectiveness with which the receiver maintains the SNR of the incoming signal within the receive channel bandwidth will be the most important determinant of the modem's raw data rate. The SNR of the receive signal over the signal bandwidth, determined by the channel; puts a fundamental limit on the amount of data the channel will carry. Any degradation beyond this in the circuitry between the line interface and the digital output samples is considered implementation noise and is determined by the quality of the receiver. It is the receiver's job to eliminate out-of-band channel noise and compensate for signal attenuation, then digitize the analog signal for further digital signal processing.

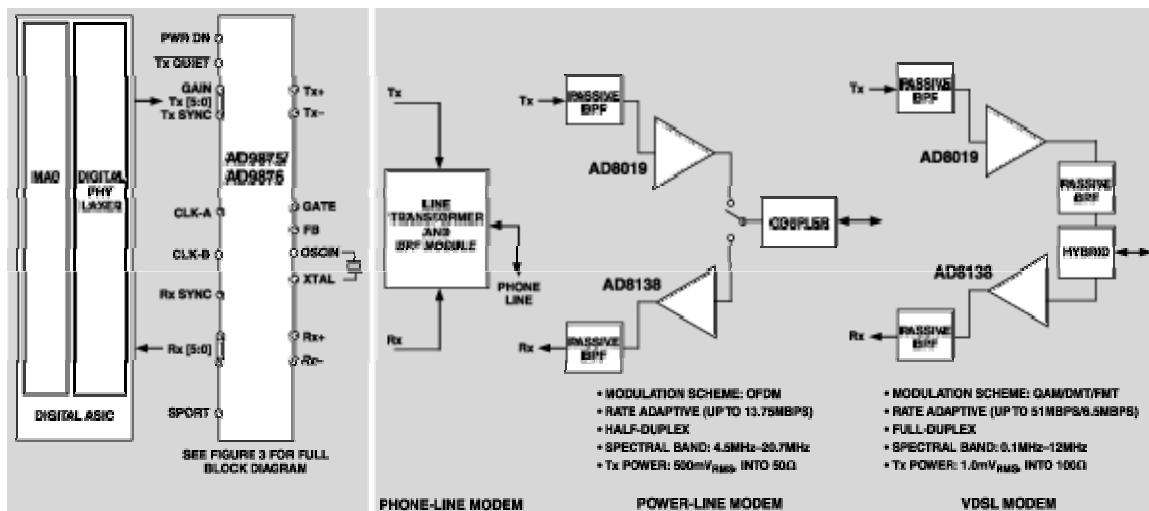
The out-of-band noise and interference on the line reduce the receiver SNR in two ways. First, the noise present may be folded back into the signal band of interest by the sampling

process, raising the existing noise floor. Second, if the noise and interference are orders of magnitude greater than the desired signal, this will reduce the gain that can be applied to the signal to compensate for signal attenuation—again resulting in lower SNR. Effective filtering is essential to reduce these effects. In order to optimize noise- and distortion performance, a variable-gain amplifier is implemented on-chip. This function is shared by a continuous-time programmable gain stage (CPGA in Figure 3) and a discrete-time switched programmable gain stage (SPGA), sandwiching the LPF. Following the SVGA, a 12-/10-bit ADC digitizes the signal with sampling rates up to 50 MHz.

Several auxiliary function are also present on-chip. Two phase-locked-loop (PLL) blocks, a voltage-regulator control circuit, and a serial-port interface help reduce external component count and optimize performance.

### Phone-line networking

Figure 4 and 5a show the AD9875 in a phone-line networking application that is capable of data rates up to 32 Mbps using QAM modulation. The separation of mixed-signal and digital circuitry allows the digital ASIC to be implemented on the most cost effective geometry possible. Because the modulation coding is located in the digital ASIC, designers can maximize their ‘value added’ while they minimize time to market.



**Figure 4. Mixed-signal and digital interface.** (a) (b) (c)  
**Figure 5. Analog Interfaces.**

### Power-line and VDSL modems

The AD9876 provides the optimum partitioning for a power-line or a VDSL modem, as illustrated in Figures 4, 5b, and 5c. With its integrated programmable-gain amplifier, low-pass filter, and 12-bit ADC, in combination with 2x/4x interpolation filter and 12-bit TxDAC® D/A converter, the AD9876 provides nearly the entire analog portion of the signal chain—in a 48-lead LQFP package. By adding analog filters and line drivers, designers can bring a power-line or VDSL modem product to market quickly with an integrated single-chip solution. Compared with other, less-integrated, solutions in this type of application,

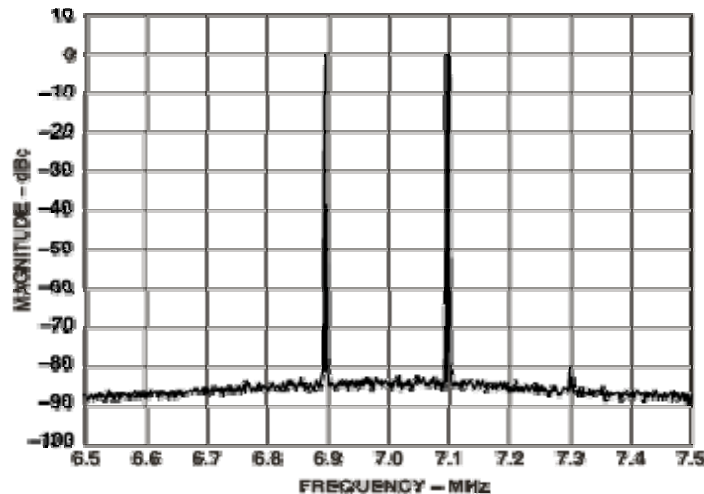
component count can be reduced by as much as 50%; and the overall bill of materials can be reduced by more than the purchase price of the included integrated MxFE component. System costs can be reduced further by using the AD9875/6's integrated 4x PLL clock multiplier and system clock outputs. They allow the entire system clock to be implemented with an inexpensive low-frequency crystal.

## AD9875/AD9876 Key Features, Specifications, and Performance

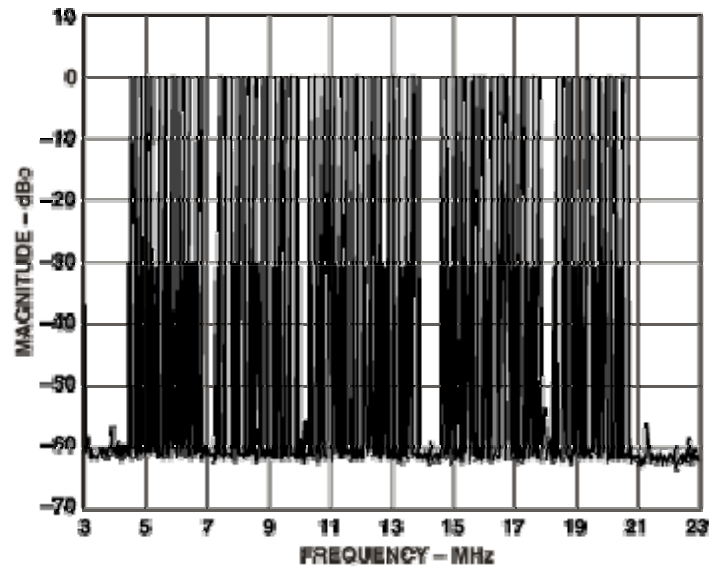
- Low-cost 3.3V-CMOS mixed-signal front-end converter for broadband networking/modems
- 10-/12-bit 128-MSPS TxDAC+ ® D/A converter
- 64-/32-MSPS input word rate
- 2x/4x programmable transmit path interpolation LPF or BPF
- Flexible power-down modes
- 10-/12-bit, 50-MSPS ADC
- 4th-order low-pass filter 12- or 26 MHz with bypass
- -6-dB to 36-dB programmable-gain amplifier
- Internal 4x clock multiplier (PLL) clock outputs
- Voltage regulator controller
- 48-lead LQFP package

AD9875/AD9876 performance was characterized over the  $-40^{\circ}$  to  $+85^{\circ}\text{C}$  extended industrial temperature range.

The following two graphs show the transmit-path performance in multi-tone applications of these parts and illustrate their exceptional linearity. Figure 6 shows intermodulation-distortion of the AD9876; under the indicated conditions, it is less than  $-80$  dB. Figure 7 shows the multi-tone power ratio of the transmit path: about 55 dB when transmitting 70 tones in the 4.5 to 20.7-MHz frequency range, corresponding to the HomePlug frequency band.

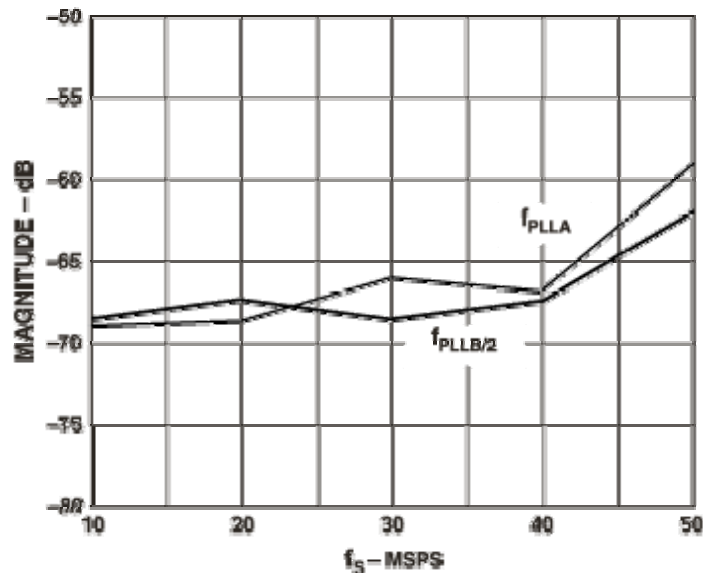


**Figure 6. Dual-tone spectral plot of AD9876's 12-bit DAC @  $f_{\text{DATA}} = 50$  MSPS,  $f_{\text{OUT}} = 6.9$  MHz and 7.1 MHz.**



**Figure 7. “In-band ” multi-tone spectral plot of AD9876's 12-bit DAC @  $f_{DATA} = 50$  MSPS,  $f_{OUT} = k \times 195$  kHz, 2 x LPF.**

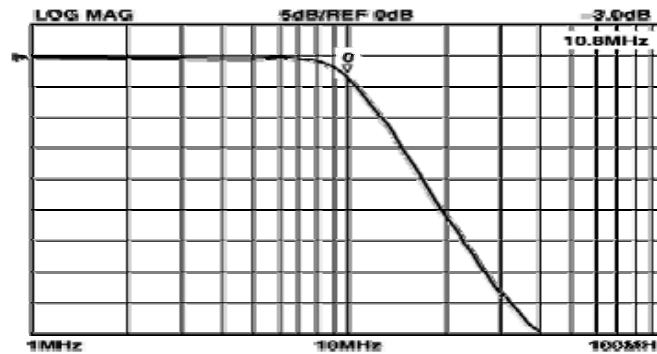
The AD9876 12-bit ADC's performance meets the requirements for two-band VDSL and has been designed into system solutions for that application and for power line access modems. Figure 8 is a plot of THD as a function of ADC sampling rate for a 5-MHz sinusoidal input.



**Figure 8. AD9876 12-bit ADC THD performance vs.  $f_{ADC}$  @  $f_{IN} = 5$  MHz.**

The Rx LPF of the MxFE device has two frequency settings, one with a center frequency of about 10.8 MHz, the other at about 26 MHz. The filter transfer function is similar to that of a fourth-order Butterworth. The filters have a self-tuning feature that corrects for variation in device elements from part to part and for temperature drift. The center frequency can be adjusted over a 20% range if different cutoff frequencies are desired. Figure 9 shows LPF performance with the lower cutoff frequency selected.



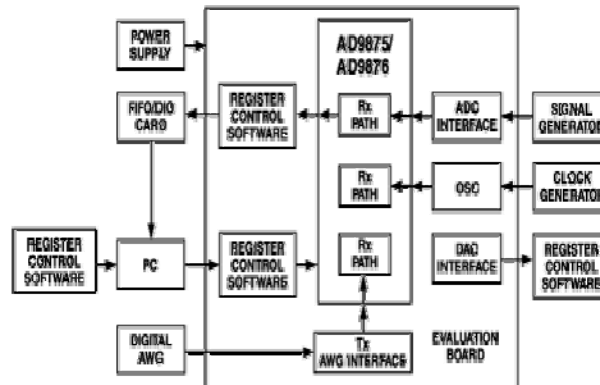


**Figure 9. AD9876 Rx LPF frequency response, low  $f_C$ .**

The AD9875/6 evaluation board (Figure 10), with its software, allows users to easily program and quickly evaluate the device for a specific modem application. The evaluation board provides connector access to the device's digital transmit and receive ports—which are buffered for reliable data transmission. The on-chip configuration registers can be programmed through the use of a PC serial port, which connects directly to a connector on the evaluation board; and the PC-resident evaluation-board software provides for a simple means of configuring the MxFE operation.

The analog output from the DAC, and the analog input to the ADC circuitry can be accessed through SMA connectors or pin headers on the evaluation board. The jumper programmable interface configurations allow several different circuit options to suit different testing methods. An on-board digital loopback feature allows the ADC and DAC to be tested simultaneously with the use of a signal source and spectrum analyzer. Also available is a daughter card, which can provide a phone line compatible interface to the MxFE device.

The AD9875-EB software provides a graphical user interface for easy programming and querying of the AD9875 registers. Three programming windows are available. The direct-register-access window allows AD9875 write- and readback in decimal, binary or hexadecimal data formats. The register-map window provides easy, function oriented programming of the AD9875 registers. This window displays graphically all of the MxFE™ functions on the screen. The advanced-register-access window allows programming of sequences of register accesses.



**Figure 10. AD9875/6 evaluation setup.**

**Availability**

The AD9875 and AD9876 were released to production in summer 2001. They are available in an economical, space-saving 48-lead LQFP and are priced at \$9.24 and \$14.45 (1000s) respectively. The AD9875 is priced at less than \$5.00 (AD9876 <\$7.00) in high volumes.

**APPENDIX**

These members of the Analog Devices MxFE family, the AD9875/6 support smart partitioning, a mixed-signal design technique that partitions the signal path along lines that optimize system performance, rather than at analog/digital, boundaries. This methodology complements the digital PHY (physical layer) and MAC (media access controller), which consist of several hundred thousand to well over a million gates fabricated on state-of-the-art 0.18-  $\mu\text{m}$  or finer geometry CMOS processes to take full advantage of the speed, power and cost advantages that those processes offer. Meanwhile, at the heart of the AD9875/6 are high-performance data-converter cores, fabricated on a well-established and cost-effective 0.35-  $\mu\text{m}$  CMOS process. To these cores are added both analog- and digital signal-processing circuitry, to provide the necessary interfacing to both worlds, thus reducing complexity and cost of circuitry on the system board, without reducing flexibility. This combination of mixed-signal and digital functions provides overall system benefits by reducing component count and footprint without compromising performance.

Besides its performance and cost advantages, “smart partitioning” provides a logical place for the division of labor in developing complete physical-layer solutions. Companies whose core competencies are in the development of communications systems and algorithms can concentrate on those aspects of the digital design and capitalize on their differentiating technology. They can rest assured that their combined analog and mixed-signal solution is at the state of the art by partnering with a leader in the field, such as Analog Devices.

# A Natural Model for EDA Sensible IP Behavior

Mark Fredrickson, IBM Corporation

Dan Notestein, SynaptiCAD

Melanie Yunk, Silicon Integration Initiative, Inc.

## Introduction

The Timing Diagram Markup Language (TDML) is being developed by an Electronic Component Information eXchange (ECIX) Working Group under the management of the Silicon Integration Initiative (SI2). Members of this Working Group include Chronology Corporation, Denali Software, Inc., IBM, Mentor Graphics/Interconnectix Business Unit, and SynaptiCAD.

In November of 1996, SynaptiCAD approached SI2 and asked to be part of the ECIX program in order to develop a timing diagram format as part of the ECIX electronic data sheet standard. Subsequently, SI2 issued a Request for Technology (RFT) in Spring, 1997 to address this issue. Responses were received from both Chronology Corporation and SynaptiCAD. As a result, the TDML Working Group was formed in December 1998, with the goal of developing an interchange standard for timing diagram information that would:

- Allow the customer to choose a timing diagram browser or other support tool,
- Allow the component information (and Intellectual Property) provider to choose a timing diagram editor or other generation tool, and
- At the same time, this new standard would support standalone application use while enabling adoption of timing diagram data as a strategic element of the ECIX Pinnacles Component Information Standard (PCIS) standard.

TDML is an open, industry standard language for the exchange of interactive timing diagrams. TDML allows persons to exchange interactive timing diagram information in a standard form. This standard form promotes the sharing and interchange of information between different organizations and allows interested parties to develop tools for generating, editing, and browsing, these diagrams.

TDML is currently at version 0.9.5, with TDML 1.0 scheduled for release in October 1998.

The standard resulting from this process will be designed for use by ECIX PCIS-compliant datasheets for representation of timing and waveform diagrams that describe component and intellectual property (IP) characteristics. This interchange format will be

promoted as an industry-wide standard first through Si2 membership, then formally through the IEEE/IEC. The standard will be openly available to companies worldwide for adoption and use with both commercial and proprietary software tools.

### **Interactive timing diagrams the intuitive way to work with timing relationships**

#### ***What are interactive timing diagrams?***

Interactive timing diagrams look like traditional timing diagrams, showing signal transitions with respect to time and relationships between signal transitions. However, as the name indicates, interactive timing diagrams interact with the designer in a variety of ways. For example, tools that interact with timing diagrams allow the designer to select a signal transition and move it in time. Signal transitions that are dependent on the moved edge may also move automatically. Setup, hold, and pulse-width tests are automatically recalculated and drawn to show how the timing relationships change when the transitions move.

The timing diagram shown in Figure 1. is not an interactive diagram, but if it were, a user could, for example, select the valid-time edge of the DATA signal and move it to the right, modeling a later data arrival time. This would automatically cause the margin on the setup test to decrease. The data at output Q would become valid at a later time, moving to the right by the same amount as the input DATA.

Timing diagrams are one of the most intuitive ways to visualize and work with any kind of timing relationships. The relationships are easy to see and understand. It is no wonder that databooks, blackboards, and engineer's heads are full of them. Interactive timing diagrams are even better because they make it easy to see and understand how timing relationships change in response to changes in signal arrival-times, path delays, and so forth.

Several EDA companies market timing diagram tools that make use of interactive timing diagrams to help designers understand and control the timing of electronic designs. Some examples of these tools are the "TimingDesigner" family of tools from Chronology (<http://www.chronology.com/>) and the "WaveFormer Pro" set of tools from SynaptiCAD.

#### ***What's inside an interactive timing diagram?***

Timing diagrams contain information about physical cause-and-effect relationships such as delays and constraints. This information is timing model information that is valid no matter how the part is used. This modeling information is contained in the relationships between signals in the diagrams (see Figure 2.). Diagrams also contain information about when signal transitions arrive at the design inputs, and when the dependent signal transitions occur at other places in the design. These signal arrival-times are usage dependent information that is contained in the transition of the signals in the diagrams.

#### ***What are the benefits of interactive timing diagrams?***

Since timing diagrams contain both modeling and usage information, they can be used for both modeling and usage timing tasks. Here are some examples of how interactive timing diagrams are used:

- Using timing diagrams to help designers understand and analyze system timing behavior:

Timing diagrams are a tool used by engineers and designers universally in their attempts to understand and manipulate timing relationships between electronic signals. Timing diagrams are used to show timing relationships, identify timing problems, and analyze potential solutions. Timing diagrams are used because they are easy to understand. They show the timing information in a graphic, intuitive format that allows the user to immediately visualize the important relationships. "Interactive" timing diagrams are especially applicable to timing problem resolution because designers can "try out" different potential solutions and immediately see graphically the effects of those changes. Several popular EDA tools (as noted above) use interactive timing diagrams as a primary interface between person and computer.

- Using timing diagrams to set up timing assertions or simulation patterns:

Timing diagrams are an intuitive way to define the input signal transitions for a static timing analysis or simulation program. In a graphical manner, the user can see the clock and determine the input signals and their transition times and slopes.

- Using timing diagrams to define timing models

As mentioned above, timing diagrams are great for describing input-output relationships and timing constraints between signals. Component databooks contain these types of timing diagrams. Interactive timing diagrams are useful for timing models because the designer can change the input signal transitions and see the results immediately. IP vendors are interested in using interactive timing diagrams as models so that potential customers may test the components or technology right in the intended application! Some vendors are already working with EDA tool providers to offer these models.

- Using timing diagrams to visualize results from static timing analysis tools:

Many powerful timing analysis tools provide information to electronic designers in form of textual reports. These reports are loaded with good information but they are hard to digest. Designers often take information from the reports and manually translate it into timing diagrams in order to visualize the timing relationships being described. It is no exaggeration to say that timing diagrams are the defacto preferred medium for thinking about timing relationships.

#### **TDML - the Timing Diagram Markup Language**

TDML is an industry-backed effort to establish an open, industry standard language for the exchange of interactive timing diagrams. The language is currently under development by the TDML Working Group of the Silicon Integration Initiative (Si2) (<http://www.si2.org/>) as part of the Electronic Component Information Exchange (ECIX) project with members from component and virtual component vendor and EDA companies.

The TDML language allows persons to exchange interactive timing diagram information in a standard form. A standard form for this information promotes sharing between

different organizations and allows interested parties to develop tools for generating, editing, browsing, etc. the diagrams.

### **How will TDML be used?**

Figure 3 and text below describe some of the ways in which TDML will be used.

#### 1. Generating TDML with a timing diagram tool:

Anyone wishing to describe the interface timing behavior of an IP may describe that behavior with timing diagrams by using a timing diagram tool which supports TDML. The diagrams may be edited, printed, etc. as needed, and those diagrams may then be converted into standard TDML for export into other tools.

Semiconductor manufacturers and vendors will exploit this method to display their wares in an easy-to-distribute and easy-to-use method that accurately models the complex timing relationships of their products.

#### 2. Generating TDML from other types of programs:

Companies or individuals that provide IP models of different types will want to see those models represented as interactive timing diagrams, and will want to export those models in TDML format. These persons may use existing programs that generate models from user input, parameter databases, etc. These programs may be interfaced directly to timing diagram tools, or the programs may generate TDML which may then be imported into a timing diagram tool, so that the users may view and graphically manipulate the models, and export those models in TDML format for use elsewhere.

##### A. Browsing IP specifications:

Viewers of the timing/interface behavior of an intellectual property will import TDML describing that behavior into a TDML browser that will display the interactive waveforms. Here is an example of how this scenario will be used: An engineer wants to find out whether or not IP XYZ from company ABC will work in his/her project. So, the engineer surfs over to the company ABC internet website, locates the specifications for the desired IP, and examines them with an online browser which supports TDML.

It is expected that TDML browsers will be available as "plug-ins" to common internet browsers such as Netscape(TM).

##### B. Importing TDML IP interface models into timing tools:

Persons using timing diagram analysis tools will be able to download TDML IP timing models into their tools to test the IP with the rest of their design.

##### C. Converting TDML IP interface models into other forms:

Tools will be available to convert TDML timing models into many other useful forms for importation into other types of timing and analysis programs.

Scenario mixing: These usage scenarios can be mixed in any desirable manner. For example one practical scenario would be "1-A" in which semiconductor vendors export TDML onto the internet describing their products, and their customers examine the IP specifications using an online TDML browser.

### ***What does TDML mean to the IP Provider and Customer?***

If the user is a design engineer, TDML means that in the near future the design engineer will be able to select among competing, off-the-shelf EDA tools which can read and write interactive timing diagrams. The user will be able to import interactive timing models from IP vendors and from colleagues who may be using the same or different EDA programs. The engineer will be able to export timing models which will not only look good to humans, but will also work well with a variety of EDA computer tools. The user may also be able to use graphic diagram tools to define timing assertions for favorite timing analysis program, or define simulation vectors for a simulation engine. And, after analysis or simulation, the design engineer may be able to take the results into a diagram tool and view them graphically.

If the user is an EDA tool provider dealing with timing or simulation tools, the user should consider adding TDML import and export capabilities to existing tools so that they may communicate with timing models and/or signal transition models with other tools.

Static timing tools such as Motive and PrimeTime (from Synopsys) or EinsTimer (from IBM) have traditionally been functionally very powerful, but difficult to use. Two of the most difficult tasks involved in using these static timing engines are setting up the assertions correctly, and interpreting the output (usually textual reports). TDML-based tools could be used to improve the usability of these tools and add additional capabilities. For example, TDML-based tools could be used to graphically set up timing assertions for a static timing engine. Timing analysis results could be exported into a diagram-based tool for intuitive examination.

If the user (or author in this case) is an IP vendor or a provider of IP models, the author should explore the benefits of making models available in TDML form. Some of those benefits include creating models that are good for both human and computer usage, creating models that are interactive and respond to changing input conditions, and creating models that can be used by a variety of EDA tools.

#### **A look at TDML syntax**

Here is an excerpt from a TDML instance created by SynaptiCAD's WaveFormer Pro timing diagram editor (used with permission). The following code describes the waveform and delay parameter displayed in Figure 4.

Notice when looking at TDML code that each object is contained within matching tags. For example the code excerpt below is one of the first code blocks in a TDML file. The block begins with <TDML.ADMIN.INFO> tag and ends with </TDML.ADMIN.INFO> tag. All the information between these particular tags describes the type of program used to generate the TDML file.

```
<TDML.ADMIN.INFO>
```

```
  <TOOL.INFO ID="X1">
```

```
    <TOOL.NAME ID="X2">WaveFormer  
    Pro</TOOL.NAME>
```

```
    <TOOL.TYPE ID="X3">Timing Diagram  
    Editor</TOOL.TYPE>
```

```
  </TOOL.INFO>
```

```
</TDML.ADMIN.INFO>
```

The next code block entitled <SIGNAL> describes SIG0 and its waveform as displayed in the timing diagram. The waveform consists of signal transitions or "edges" that represent a change of state. An edge (E) contains the state (S) of the signal after the transition, and the earliest time and latest time that the transition could happen (TE,TL). Edges can contain an optional ID attribute that enables the edge to be referenced by external objects.

```
<SIGNAL ID="ID2" SHOW="1" SHOW.GRID="0"  
GRID.COLOR="0000FF"
```

```
  GRID.STYLE="Solid" DRAW.MIN="1"  
  EDGES.PER.CYCLE="1">
```

```
  <CONN.PTR  
  CONN.ID="ID1">SIG0</CONN.PTR>
```

```
  <WAVEFORM ID="ID3" LOCKED="0">
```



```
<E ID="ID3V" DRIVEN="1" S="1"  
TE="0" TL="0"></E>
```

```
<E ID="ID4" DRIVEN="1" S="0"  
TE="30000.0" TL="30000.0"></E>
```

```
<E ID="ID5" DRIVEN="1" S="1"  
TE="70000.0" TL="80000.0"></E>
```

```
<E DRIVEN="1" S="0" TE="140000.0"  
TL="140000.0"></E>
```

```
<E DRIVEN="1" S="1" TE="200000.0"  
TL="200000.0"></E>
```

```
<E TE="270000.000000"  
TL="270000.0"></E>
```

```
</WAVEFORM>  
</SIGNAL>
```

The delay parameter in the timing diagram is a little more complicated because it references the edges contained in an external signal. The min and max values of the delay are stored in a separate element called a <TDML.CC> so that timing characteristics can be shared by multiple delay parameters.

```
<EDGE.RELATIONSHIPS>
```

```
  <RELATIONSHIP TYPE="delay"  
  TDML.CC.PTR="ID6" SHOW="1">
```

```
    <TWO.EDGE SOURCE.E="ID4"  
    TARGET.E="ID5">
```

```
      <RELATIONSHIP.LABEL>D0</RELATIONSHIP.LABEL>
```

```
    </RELATIONSHIP>
```

```
  </EDGE.RELATIONSHIPS>
```

```
  <CC.LIST>
```

```
    <TITLE ID="X6">Parameter Data  
    Table</TITLE>
```

```
    <TDML.CC ID="ID6" STATIC="1" SHOW="1">
```

```
    <PARAM ID="X7">
```

```
    <PARAM.SYMBOL
```

```
    ID="X8">D0</PARAM.SYMBOL>
```

```
    <PARAM.DESC ID="X9"></PARAM.DESC>
```

```
    </PARAM>
```

```
    <TDML.VALUE VALUE.TYPE="MIN">
```

```
    <TDML.EXPRESSION>
```

```
    <EXPRESSION
```

```
    ID="X10">40</EXPRESSION>
```

```
</EVALUATED.TO>40000.000000</EVALUATED.TO>
```

```
  </TDML.EXPRESSION>
```

```
</TDML.VALUE>
```

```
<TDML.VALUE VALUE.TYPE="MAX">
```

```
<TDML.EXPRESSION>  
  <EXPRESSION  
  ID="X11">50</EXPRESSION>  
  
<EVALUATED.TO>50000.000000</EVALUATED.TO>  
  </TDML.EXPRESSION>  
  </TDML.VALUE>  
  </TDML.CC>  
</CC.LIST>
```

TDML can be parsed using a SGML/XML parser. Both commercial and free SGML parsers such as James Clark's SP parser (<http://www.jclark.com/sp/index.htm>) are available online for download.

### **TDML and the IP Industry**

One of the major challenges to using IP is to determine a method for communicating with the IP block. IP blocks communicate using transaction protocols that describe the rules that must be followed for successful operation. Transaction protocols have long been described using timing diagrams because they form a concise visual description of cause-effect relationships between transaction events. Timing diagrams may also unambiguously describe design constraints such as setup and hold time requirements that must be met by interacting components.

Another important issue in IP usage is substitution of one IP block with another that is behaviorally equivalent either for higher performance, increased functionality, or simply because a design is being ported to a different process. Since timing diagrams focus on describing interface behavior rather than internal operation, they document the design requirements for IP use without dictating implementation details, simplifying drop-in replacement of IP blocks.

TDML provides an open format for distribution of protocol specifications, enabling IP vendors to release complete interface information without locking their customer base down to a specific EDA tool suite. An open format also encourages tool vendors to find new uses for TDML information, increasing the ultimate usefulness of TDML in the design process.

### **References**

Silicon Integration Initiative, *ECIX*, <http://www.si2.org/ecix>  
Chronology Corporation, *Timing Designer*, <http://www.chronology.com>.  
SynaptiCAD, *WaveFormer Pro*, <http://www.synCAD.com>.

Lewis, Larry, Silver, Kevin, *Virtual components for system-on-a-chip design*, Computer Design, November, 1997, page 52.

Figures

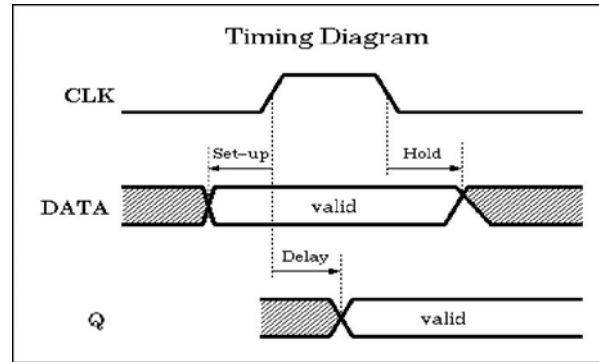


Figure 1. Sample Timing Diagram

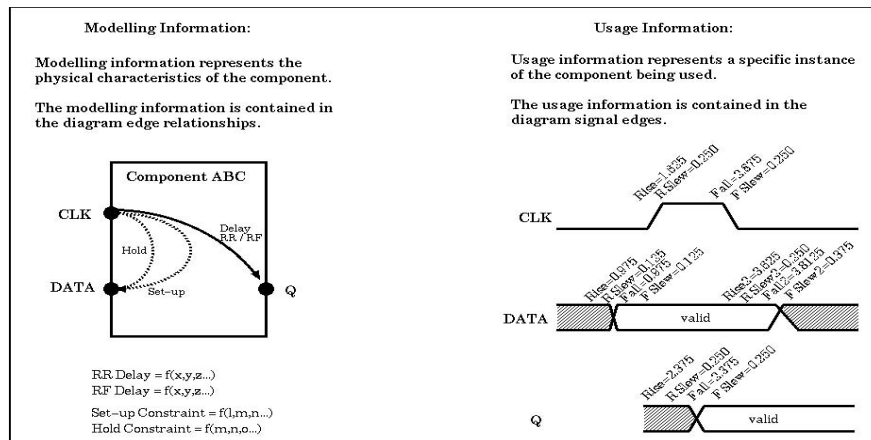
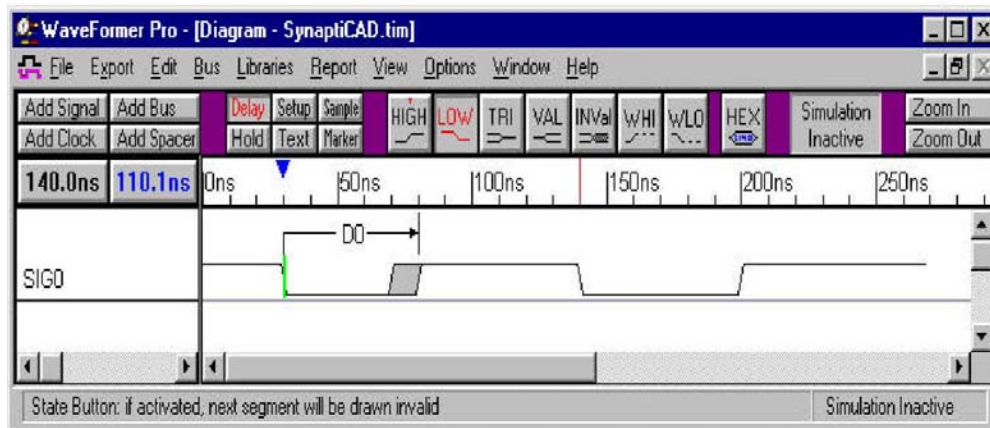
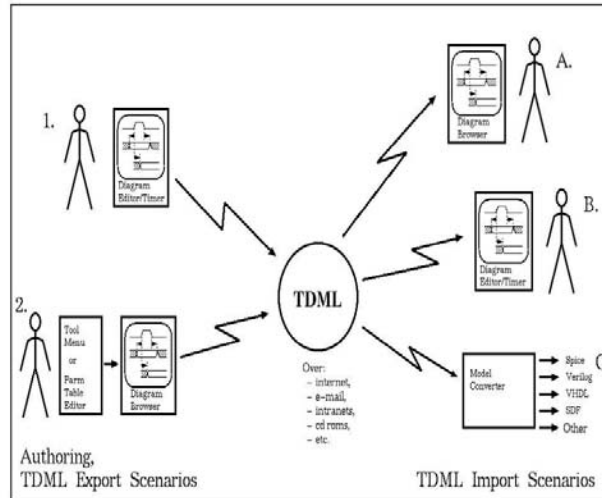


Figure 2. Modeling and Usage Information in a Timing Diagram



# The daemon, the gnu, and the penguin

## A History of Free and Open Source

Peter H. Salus  
CKO of Maxtrix.net

The activities of a distributed and unorganized band of scholars led to the conceptual revolution that produced the modern world. For example, Copernicus (1473-1543) observed the heavens and recorded his measurements. In 1563, Tycho Brahe (1546-1601) noted that Copernicus' figures weren't quite right, so, from 1577 to 1597, Tycho recorded extraordinarily accurate astronomical measurements. In 1599 Tycho moved from Denmark to Prague, where Johannes Kepler (1571-1630) was his assistant, until he succeeded him in 1601, when Tycho died.

Copernicus established heliocentricity. Tycho found that circular orbits just didn't work, and devoted decades to better measurements, which Kepler later used to determine that the orbits were ellipses, not circles. (In 1610, Galileo [1564-1642] pointed out that one could observe phases on Venus, and that therefore Venus must be nearer the Sun than the Earth was.) And, Newton (1643-1727) showed us the force (gravity) that held everything in place.

Poland. Denmark. Austria. Italy. Germany. England. Despite the Papacy, the 30 Years' War, turmoil in the Netherlands, in France, and in England, thought moved in print and in correspondence. Though countries were at war and religions were in conflict, scientific exchange of ideas and sharing of data persisted.

During the Renaissance it could take months for findings to reach those interested in other countries. In the seventeenth and eighteenth centuries lengthy epistles between scholars were distributed to others beyond the addressees. Scientific journals followed. Thanks to the progress of communications media, it now takes seconds where it once took decades for an idea or a discovery to proliferate. The fact is undeniable: Invention and scholarship have been the motor driving the development of civilization and culture.

The revolution of knowledge has led us to exploration and discovery. The computer, the Internet, and the Web have led to a similar revolution. While certainly no computer user, Thomas Jefferson, in a letter to Isaac McPherson (13 August 1813), wrote:

If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it.

My aim is to show how the advent of the computer and the Internet have given rise to the expansion of the academic/scholarly notions of sharing, and how this in turn has brought us free and open software, which will bring about a major change in the way we do business.

This effort is more than a history of Linux, of the Free Software Foundation (FSF), the Internet, software licensing, and myriad other topics. It will contain a number of histories within it, which (I hope) will serve as an antidote to the cloud of FUD stirred up by those who fear that change will mean that their businesses will fail (certainly more a sign of lack of imagination and flexibility than of anything else).

On the contrary: change yields opportunity. But change also requires adaptability. We are embarking on a new business model, which will change the way we do business as much as mass production and global electronic communication did over the 19th and 20th centuries.

Since 1990, there has been an insistent drumbeat of anti-FSF FUD. Since 2000, this has focused on Linux. Some examples of this are:

- On June 1, 2001, Steve Ballmer, CEO of Microsoft, told the Chicago Sun-Times: "Linux is cancer."
- On October 15, 2002, Darl McBride, CEO of The SCO Group, said: "We are more committed to Linux than ever before."
- On March 4, 2003, Blake Stowell, SCO director of Public Relations, said: "C++ is one of the properties SCO owns."
- On May 14, 2004, the Alexis de Tocqueville Institution issued a press release in which it revealed that its Director, Ken Brown, had discovered that Linus Torvalds had not "invented" Linux.
- On August 26, 2004, Kieran O'Shaughnessy, director of SCO Australia and New Zealand, told LinuxWorld: "Linux doesn't exist. Everyone knows Linux is an unlicensed version of Unix."

The remarks are noise. But though ludicrous, statements like these make businessfolk fearful. They then hug Windows the way a different Linus clutches his blanket. My goal here is to show a wider audience just what went into the creation of open source and its worldwide network of contributors and users over the past 50 years.

Over four centuries have passed since our static heliocentric universe was replaced by a dynamic one. Today, the business model that has persisted since the late eighteenth century is being replaced. Here's how it's happening.

1. In June 1968, the Federal Communications Commission's "Carterphone" decision compelled AT&T to allow its customers to connect non-Western Electric equipment to the telephone network. [FCC Docket Number 16942; 13 FCC 2nd, 420].
2. In July 1968, Andrew Grove and Gordon Moore founded Intel.
3. In August 1968, William G. McGowan established Microwave Communications of America [MCI] and the FCC ruled that MCI could compete with AT&T, using microwave transport between Chicago and St. Louis.
4. In December 1968, the Defense Advanced Research Projects Agency let a contract to Bolt, Beranek and Newman of Cambridge, MA, for just over \$1 million. The contract was for a packet-switching network of four nodes.

Four more events of importance followed the next year.

1. In August, humans landed on the moon.
2. Summer saw the invention of UNIX.
3. In the autumn, those first four nodes of the ARPAnet went up.
4. And, in December, Linus Torvalds was born.

Had anyone asked, I would have thought the first of these events was the most important. Outside of his immediate family, I seriously doubt whether anyone even knew about the last of these.

As of the outset of the Twenty-First Century, the moon landing has taken us nowhere. The other items in this list though are the stuff of revolution.

### Ancient History

While mechanical calculation goes back to the seventeenth century, computation is far more recent. Though first conceived by Charles Babbage in 1823, the computer as we know it needed more than a century to come into being. The first true electro-mechanical computer was Harold Aiken's Mark I (conceived in 1937 and put into operation in 1944) and the first fully electronic machine was Maurice Wilkes' EDSAC (1949).

### IBM and SHARE

The first commercial computer, the IBM 701, wasn't completed until late in 1952. The first production machine was shipped from Poughkeepsie to the IBM headquarters building in Manhattan that December. The second machine was destined for Los Alamos, and production continued in IBM's Poughkeepsie facility through June 1954, when



machine 18 was shipped to Lockheed in Burbank. That's rather slow production by our standards, but literally everything was new in the early 1950s.

Prior to the 701, all computers had been one-offs. Aiken's, Wilkes', ENIAC, etc.; each was sui generis. The 701 was a genuine breakthrough. On 7 May 1954, the redesigned 701 was announced as the IBM 704. It was more than merely a redesign. The 704 was incompatible with the 701. It had 4096 words of magnetic core memory. It had three index registers. It employed the full, 36-bit word (as opposed to the 701's 18-bit words). It had floating-point arithmetic. It could perform 40,000 instructions per second. While deliveries began in late 1955, the operators (today we would think of them as system administrators) of the eighteen 701s were already fretful months earlier.

IBM itself had no solution to the problem. Though IBM had hosted a "training class" for customers of the 701 in August 1952, there were no courses, no textbooks. But several of the participants in the training class decided to continue to meet informally and discuss mutual problems. (According to Pugh, their first meeting was "in February 1953 during an AIEE-IRE Computer Conference in Los Angeles.") The participants agreed to hold a second meeting after their own 701s had been installed. The second meeting was hosted by Douglas Aircraft in Santa Monica in August 1953. There were other informal meetings and then, following an IBM Symposium, The RAND Corporation hosted a meeting in Los Angeles in August 1955 of representatives from all seventeen organizations that had ordered 704s. It was at this meeting that the world's first computer user group was formed. It was called SHARE.

IBM encouraged the operators to meet, to discuss their problems, and to share their solutions to those problems. IBM funded the meetings as well as making a library of 300 computer programs available to members. SHARE, 50 years later, is still the place where IBM customers gain information. (A number of the earliest contributed programs are still available.)

The importance of SHARE can be seen in the fact that in December 1955, early purchasers of Remington Rand's ERA1103A formed an organization called USE [= Univac Scientific Exchange]. In 1956, user groups for Burroughs and Bendix computers were formed, as well as IBM's GUIDE, for users of their business computers. Though SHARE was vendor-sponsored at the outset, today it is an independent organization.

User groups are one thread in the complex fabric which we employ today. Another is communication.

### **DARPA and IPTO**

In response to the USSR's launching of Sputnik in October 1957, the US Department of Defense set up the Defense Advanced Research Projects Agency (DARPA). That charge

was "to think independently of the rest of the military and to respond quickly and innovatively to national defense challenges."

In 1962, Jack Ruina, the Director of DARPA, hired J. C. R. Licklider to be the first Director of DARPA's new Information Processing Techniques Office (IPTO).

Originally, the IPTO was to extend research into the computerization of the air defense system. The IPTO funded research into advanced computer (and networking) technologies and funded fifteen groups to do research in human-computer interaction and distributed systems. (Among the research sites were: Carnegie-Mellon University, MIT, the RAND Corporation, the Stanford Research Institute, the System Development Corporation, UC Berkeley, UC Santa Barbara, UCLA, the University of Southern California, and the University of Utah.) In 1963, Lick (as many called him) funded Project MAC at MIT, headed by Robert Fano. Project MAC explored the potential for communities on time-sharing machines. That is, relationships among the uses and the users of shared mainframes.

And this leads directly to the next strand in our narrative: time-sharing.

### **Time-Sharing**

John McCarthy had begun thinking about time-sharing in the mid-1950s. But it was only at MIT in 1961-62 that he, Jack Dennis and Fernando Corbato talked seriously about permitting "each user of a computer to behave as though he were in sole control of a computer."

When McCarthy went to MIT from Dartmouth in 1957, it was clear that time-sharing the IBM 704 would require an interrupt system which didn't exist yet. So McCarthy proposed a hardware solution involving a relay whereby the 704 could be set to "trapping mode" by an external signal. But, like many other brilliant insights, McCarthy's notion went undeveloped for several years.

Four years later, MIT had a transistorized computer, the IBM 7090, and so Corbato wrote CTSS (Compatible Time-Sharing System). While it had bugs, it was a wild success, influencing systems at Dartmouth (DTSS) and the Incompatible Time-Sharing System (ITS) for the PDP-10s at MIT (more about this later).

At the same time, Lick's imagination led him to note how many different multi-million dollar computers he was funding, each of which was a solitude, unable to communicate with others. In early 1963 he sent a memo to "Members and Affiliates of the Intergalactic Computer Network." He consistently asserted that the computer was a communications, not a computation device. Then he returned to MIT.

Lick's successor at the IPTO was Robert Taylor. He was interested in networking and, in 1966, was funding 17 sites with a variety of incompatibilities. He needed help; and he found it in Larry Roberts.

Roberts had been working at the Lincoln Laboratory in Massachusetts since 1963. While there, he and Thomas Marill had conducted a networking experiment connecting the Systems Development Corporation's AN/FSQ-32 in Santa Monica, CA, with the TX-2 at Lincoln via a 1200 bps dedicated phone link. This permitted any program on one machine to dial the other computer, log in and run a program from a server (somewhat like a subroutine call).

While this was quite an achievement, it really did not further the aim of ARPA, except to demonstrate that long-distance data transfer via telephone wires was indeed feasible.

In April 1967, Roberts and Taylor took advantage of the meeting of the IPTO Principal Investigators in Ann Arbor, MI, to talk up their ideas of a network. Some of the PIs were interested in "resource sharing," but the contractors in attendance set up a sub-group, "Communication Group," to work on problems. Among the problems were the conventions to be used in communications and the kinds of communications lines.

It was agreed that work should be begun on the conventions and that the connections would be via dial-up lines. The plan as developed was for the computer sites to be connected via commercial phone lines and data sets, so that each computer could be connected with every other computer via circuit-switching. During the discussion, Wesley Clark (who had moved to Washington University in St. Louis from Lincoln) had an idea. He thought about it and described it to Roberts after the meeting during a shared cab ride between Ann Arbor and the Detroit airport.

Clark's idea was that the problems of working out the many possible connections could be solved by placing a mini-computer on each site. These mini-computers would communicate with each other and each site would only have to concern itself with the task of communicating with its mini. Roberts incorporated the idea into his summary of the meeting, "Message Switching Network Proposal," which he sent on April 27, 1967. He called the mini an "Interface Message Processor." The IMP was born.

Nearly a year later, on March 1, 1968, the IPTO reported to the Director of ARPA that the specifications were "essentially complete." Larry Roberts submitted a "program plan" to the Director on June 3rd and it was approved on June 21st. The ARPA budget for 1968 earmarked \$500,000 for the ARPANET.

ARPA sent out a Request for Quotation to 140 potential bidders. The Defense Supply Service - Washington received twelve proposals. Four of the bidders were deemed to be in contention and, finally, the week before Christmas 1968, the contract was awarded to

BBN in Cambridge, Massachusetts. Work began on January 2, 1969. At the end of December, there were four nodes on the ARPAnet: UCLA, SRI, UCSB, and the University of Utah.

### **Excursus: Law I**

In 1949, the Truman Department of Justice filed suit against AT&T and Western Electric, claiming the companies were acting "in restraint of trade." On 24 January 1956, Judge Thomas F. Meaney entered a "consent decree," in which the companies were enjoined "from commencing ... manufacture for sale or lease any equipment" other than that used in providing telephone or telegraph services; from "engaging ... in any business not of a character or type engaged in by Western or its subsidiaries ..."; and AT&T was enjoined "from engaging ... in any business other than the furnishing of common carrier communications services."

There were a few exceptions. Exception (b) was "experiments for the purpose of testing or developing new common carrier communications services."

AT&T was further required to reveal the patents it held and to license these when asked. No one could have foreseen the problems that this consent decree would entail.

### **UNIX**

In spring 1969, AT&T decided to terminate its involvement in a project called Multics -- Multiplexed Information and Computing Service -- which had been started in 1964 by MIT, GE and Bell Labs. This left those at AT&T Bell Labs who had been working on the project -- notably Doug McIlroy, Dennis Ritchie and Ken Thompson -- at loose ends. Doug immediately got involved with other things in Murray Hill, NJ, but Dennis and Ken had been interested in the project per se and wanted to explore several of its ideas.

Ken has said:

Dennis and [Rudd] Canaday and I were discussing these ideas of the general nature of keeping the files out of each other's hair and the nitty-gritty of expanding, of the real implementation where you put block addresses... We did it in Canaday's office, and, at the end of the discussion, Canaday picked up the phone; there was a new service at Bell Laboratories that took dictation. You call up essentially a tape recorder and you give notes, and then the next morning the notes are typed and sent to you. The next day these notes came back, and all the acronyms were butchered, like 'inode' was 'eyen...'. So we got back these descriptions and they were copied, and we each had copies of them and they became the working document for the file system -- which was just built in a day or two on the PDP-7.

At first ... we used it for other things, you know, the famous Space Travel game, and it was the natural candidate as the place to put the file system. When we hacked out this design, this rough design of the file system on the dictation [machine] that day in Canaday's office, I went off and implemented it on the PDP-7.

I won't go into full detail on the evolution of that file system on the PDP-7 to Unics [Uniplexed Information and Computing Service, a pun on cut-down (emasculated) Multics devised by Peter Neumann, an inveterate punster. Several people told me that Brian Kernighan had changed the spelling to UNIX, but Brian told me that he had not, and that no one recalled who had done it.] For now, it is important to realize that it was the cooperative product of several brilliant minds: Ritchie, Thompson and Canaday, of whom Robert Morris (who joined Bell Labs in 1960 and is now Chief Scientist at the National Computer Security Center in Maryland) said he was the "most underrated" of the original participants.

In August 1969. Ken Thompson's wife Bonnie took their year-old son on a trip to California to show off to their families. As a temporary bachelor, Ken had time to work.

I allocated a week each to the operating system, the shell, the editor and the assembler [he told me]... and during the month she was gone, it was totally rewritten in a form that looked like an operating system, with tools that were sort of known, you know, assembler, editor, and shell -- if not maintaining itself, right on the verge of maintaining itself, to totally sever the GECOS [= General Electric Comprehensive Operating System, a clone of System/360 DOS] connection. ... Yeh, essentially one person for a month.

It didn't exist by itself for very long ... maybe a day or two before we started developing the things we needed.

While Multics certainly influenced UNIX, there were also profound differences.

Dennis Ritchie explained:

We were a bit oppressed by the big system mentality. Ken wanted to do something simple. Presumably, as important as anything was the simple fact that our means were much smaller -- we could get only small machines with none of the fancy Multics hardware.

So UNIX wasn't quite a reaction against Multics, it was more a combination of these things. Multics wasn't there for us any more, but we liked the feel of interactive computing that it offered; Ken had some ideas about how to do a

system that he had to work out; and the hardware available as well as our inclinations tended to trying to build neat small things, instead of grandiose ones.

Thompson "scarfed up" a PDP-7 and "did this neat stuff with it," Ritchie told me, modestly. Thompson created a new toy that would initiate work on a new system all over the world.

Soon a PDP-11 was acquired and UNIX was rewritten and expanded and rewritten. With McIlroy prodding, Dennis and Ken produced a *UNIX Programmer's Manual* (dated "November 3, 1971"). A "Second Edition" was issued June 12, 1972: "the number of Unix installations has grown to 10, with more expected," the Preface told us. Third Edition of the manual appeared "February, 1973," and noted that there were "now 16 installations." That was soon to wax quite rapidly.

All of the first 10 installations were at AT&T in New Jersey. In the late summer of 1972, UNIX leaped across the Hudson River to an office on the 14th floor of 330 Madison Avenue in Manhattan. Neil Groundwater had joined New York Telephone upon graduating from Penn State. He commuted from his apartment in Manhattan to Whippany, NJ., where he worked on programming for the Electronic Switching System. But being in Whippany placed him in proximity to Bell Labs and he began learning about UNIX. It was no easy task. "There was documentation on some parts," he told me. "But as we would come to say years later, 'Use the source, Luke' was the sole answer to many questions."

In October 1973, Dennis Ritchie and Ken Thompson drove up the Hudson Valley to the new IBM Research Center at Yorktown Heights to deliver the first UNIX paper at the Symposium on Operating System Principles.

"It was a beautiful fall day," Dennis remarked. Ken, who delivered the paper, told me: "The audience was several hundred. I was pretty nervous. The response was the normal, polite applause. I don't recall any questions."

Ken was over-modest. The audience was quite enthusiastic. Ken and Dennis were immediately asked for copies of the new system.

This put the AT&T lawyers in a bind: was a computer operating system part of "common carrier communications services"? Was AT&T required to distribute UNIX?

The decision of the corporate lawyers was that Bell Labs should distribute UNIX to academic and research institutions at the cost of the media involved plus a shipping charge. Within a few months, several dozen institutions requested UNIX.

### The Users

Let's go back to the mid-1950s. At the time that Judge Meaney was considering the action against AT&T, IBM was coming out with the 704, an upgrade of the 701. As mentioned earlier, the transitioning from the 701 to the 704 wasn't easy, so some of IBM "operators" formed the organization still known as SHARE.

Soon, many computer manufacturers were sponsoring user organizations. DECUS -- the DEC Users' Society -- first met in 1961. It soon had a British branch (DECUS UK), and rapidly became yet more international. Remington Rand, Bendix and Burroughs had formed user groups. And in but a few years, Prime and Apollo had user organizations as well -- PRIMUS and ADUS.

So, by the beginning of 1974 there were a number of user groups exchanging information and a new operating system that was beginning to get folks excited. No one had thought seriously about licensing. And there were 40 nodes on the ARPAnet.

Early in 1974, Mel Ferentz (then at Brooklyn College) and Lou Katz (then at Columbia's College of Physicians and Surgeons) called a meeting of UNIX users in New York in May. Ken Thompson supplied them with a list of those who had requested a copy of UNIX after the SOSP meeting. Nearly three dozen in under six months. The meeting took place on May 15, 1974. The agenda was a simple one: descriptions of several installations and uses; lunch; "Ken Thompson speaks!"; interchange of UNIX hints; interchange of DEC hints; free-for-all discussion. Lou told me that he thought there were about 20 people in attendance; Mel thought it might have been a few more than that. That's the organization that's now the USENIX Association.

The Ritchie-Thompson paper appeared in the July 1974 issue of *Communications of the ACM*. The editor described it as "elegant." Soon, Ken was awash in requests for UNIX.

Mike O'Dell's reaction to the article is typical. In 1974, Mike was an undergraduate at the University of Oklahoma. He told me:

When the famous 1974 *CACM* issue appeared, I was working at the OU Computer Center. We had this thing called ITF, the Intermittent Terminal Facility, which had the world's worst implementation of BASIC, and one of the guys had written some routines which let you do I/O on terminals -- and this was a non-trivial feat. So a group of us sat down and tried to figure out whether we could do something interesting. ...

The UNIX issue came. I remember going down the hall and getting it out of my mailbox and saying to myself, Oh, ACM's got something on operating systems, maybe it's worth reading. And I started reading through it. I remember reading this paper on the UNIX time-sharing system. It was sort of like being hit in the

head with a rock. And I reread it. And I got up and went out of my office, around the corner to George Maybry who was one of the other guys involved with this. And I threw the issue down on his desk and said: "How could this many people have been so wrong for so long?"

And he said: "What are you talking about?"

And I said: "Read this and then try to tell me that what we've been doing is not just nuts. We've been crazy. This is what we want."

The *CACM* article most definitely had a dramatic impact.

Today, things would be quite different. Lou Katz wouldn't have relied on written notices; Ferentz might not have produced a purple-Dittoed newsletter. O'Dell wouldn't have gleaned the news from *CACM*, but from email and the Internet and the Web.

By 1975, the ARPAnet (with 60 nodes and soon to turn into the Internet) was becoming a way of distributing information. In late 1969, what we would think of as telnet and ftp were all there was. Then, in 1971, Ray Tomlinson invented email (which soon became the principal use of the ARPAnet), and in May 1975, RFC 681, "Network UNIX," appeared. Written by Steve Holmgren, Steve Bunch and Gary Grossman, the RFC began:

The secret, such as it was, was out. Several people have expressed their strong feelings as to just how this "put UNIX on the Net." I feel that the effect was more powerful: over the next few years, the result was that the Internet was run on UNIX. The protocols all were in tune with the "UNIX Philosophy." What we would now call "source" was widely available. Anyone actually running UNIX had accessible source. This meant that there could be true communication and we were approaching interoperability. The direct result was that UNIX was soon in use throughout the world: Japan and Australia; most of Europe; North America.

Just how widespread UNIX was can be seen from Ferentz' first mailing list (July 30, 1975) published in *UNIX NEWS*:

#### **The First Mailing List**

- Bell Telephone Labs
- Brooklyn College
- Carleton College
- Case Western Reserve University
- The Children's Museum
- City University of New York
- Columbia University
- Duke Medical Center



East Brunswick High School  
Harvard University  
Hebrew University of Jerusalem  
Heriot-Watt University  
Johns Hopkins University  
Knox College  
Naval Postgraduate School  
Oregon Museum of Science  
Polytechnic University of NY  
Princeton University  
The Rand Corporation  
St. Olaf College  
Stanford University  
The Spence School  
Univ. Catholique de Louvain  
University of Alberta  
U. of California, Berkeley  
U. of Manitoba  
U. of North Carolina  
U. of Saskatchewan  
U. of Texas at Dallas  
U. of Toronto  
U. of Utah  
U. of Waterloo  
U. of Wisconsin

The US, Scotland, Belgium, and Canada; universities and museums; a public high school and a private girls' school. In one year from publication. But in mid-1975, few of these establishments had electronic connectivity. In a few years, that would change and many (if not all) of the user sites would have some sort of network connection.

Another problem was hardware. In 1975, if you wanted to run UNIX, you needed a PDP-11 from DEC. That, too, was to change.

That change came about first at Princeton and then, simultaneously, at two sites about half the world apart from one another: Bell Labs in Murray Hill, NJ, and the Wollongong satellite campus of the University of New South Wales in Australia.

First, at Princeton, in 1976 and 1977, Tom Lyon enabled some parts of UNIX to run under VM/360 on an IBM 360. It was only the first step.

At the Labs, in 1977-78, Dennis Ritchie and Steve Johnson ported UNIX to the Interdata 8/32; in Australia, Richard Miller and his colleagues were porting UNIX to the Interdata

7/32. Dennis Ritchie has said that porting to the Interdata was both a challenge and an achievement he was most proud of, for it demonstrated that UNIX could be ported to non-DEC hardware. Steve Johnson told me that once one had ported something to an alien architecture, one knew better than to try it again. He referred to the Interdata as the "Intersnail."

Australia? Yes.

John Lions read the *CACM* article in the summer of 1974, when the University of New South Wales was about to get a PDP-11/40, and the University negotiated a license with Western Electric. In 1975-76, UNIX was a real hit on the UNSW campus. But Lions had a problem. He wanted to use UNIX in teaching operating systems. But there was no textbook and there was no explicated version of the code -- v6. So Lions decided to do something about the lack: he wrote a commentary on the code (9073 lines at that time) and received permission from Western Electric to print out the code and commentary for instructional purposes. UNSW duplicated the code in red cardboard covers and the commentary in orange. They were as big a hit as the system.

The March 1977 issue of *UNIX NEWS* (vol. 2, no. 3) announced the availability of the books (to licensees) together with a note by Mel Ferentz: "Ken Thompson has seen the first version of the book and reports that it is a good job" (quite a review). The price, including airmail, was \$A17.70 (under \$20 US, at that time). The *UKUUG Newsletter* announced the availability of the code and commentary, too, but the next issue said that future orders should be placed with Bell Laboratories and by 1978 the volumes were no longer available. (The Labs' reproductions were in a single volume bound in black.) Someone at AT&T/Western Electric had woken up.

Once again, the proverbial cat was out of the bag.

Over the years, over nearly two decades, John Lions' *Code and Commentary* became the most copied work in computing. They carry the appropriate copyright notices and the restriction to licensees, but there was no way that Western Electric could stem their circulation. They were just too valuable. (I admit that I possess an nth-generation photocopy as well as treasured copies, in orange and red covers, inscribed to me by John Lions.)

Why care? Because here we are in the mid-1970s with the users taking control and determining what to distribute where information was concerned. Luckily, Western Electric was no more successful at controlling information than Popes Paul V and Urban VIII were when Galileo wrote of heliocentricity. But note again: In the 1970s, you received Lions' work in hard copy, via airmail from Sydney, Australia.

Similarly, the inability of the AT&T/Western Electric lawyers to decide just what was permissible led an announcement in *UNIX NEWS* (30 April 1976) that Lew Law of the Harvard Science Center was

willing to undertake the task of reproducing and distributing the manuals for UNIX. ... 'The UNIX PROGRAMMER'S MANUAL' Sixth Edition dated May 1975 will be reproduced in its entirety. Most installations will want to remove several pages...

The May-June 1976 issue announced "the first mailing from the Software Exchange." This first software tape contained Harvard software; the duplication and mailing was done by Mike O'Brien, then at the University of Illinois at Chicago Circle. The idea had come to him earlier.

It depends on what you mean by "began". Actually, I was one of the "forty people in a classroom" at the meeting called much earlier than the Urbana meeting by Mel Ferentz [1975]. It was at that meeting that the idea of hosting a "Unix Users' Group tape exchange" hit me. I came home from that meeting, cleared it with my management, and declared myself open for business. By the time Urbana came around [1977], the UNIX Users' Group Software Distribution Center had been a going concern for some time.

The second software tape was announced in November 1976, along with the following note from O'Brien:

I got the "diff" listing of all changes to Bell UNIX system proper from "standard" version 6 ... Anyway, I've itemized some 50 changes, and sent the list to Ken for verification and comments. The changes will be available through the center by special request.

The second distribution tape contained contributions from the RAND Corporation, the Naval Postgraduate School, the University of California at San Diego, Yale, and UIUC. The Third Software Distribution was announced in May 1977. The last USENIX distribution was in 1988 and consisted of two 10-inch reels. The 50-bugs tape has an interesting tale connected to it.

Ken Thompson told me:

The first thing to realize is that the outside world ran on releases of UNIX (V4, V5, V6, V7) but we did not. Our view was a continuum.

After V6, I was preparing to go to Berkeley to teach for a year. I was putting together a system to take. Since it was almost a release, I made a "diff" with V6.

On the way to Berkeley, I stopped by Urbana-Champaign to keep an eye on Greg Chesson who was finishing up his Ph.D. (subtle recruiting). I left the "diff" tape there and told him that I wouldn't mind it if it got around. (I think I gave it to others too, perhaps Katz.)...

Lou Katz' version is a bit different:

A large number of bug fixes was collected, and rather than issue them one at a time, a collection tape was put together by Ken. Some of the fixes were quite important... I suspect that a significant number of the fixes were actually done by non-Bell people. Ken tried to send it out, but the lawyers kept stalling and stalling and stalling.

Finally, in complete disgust, someone "found" a tape on Mountain Avenue [The address of Bell Laboratories was 600 Mountain Avenue, Murray Hill, NJ] which had the fixes.

When the lawyers found out about it, they called every licensee and threatened them with dire consequences if they didn't destroy the tape ... after trying to find out how they got the tape. I would guess that no one would actually tell them how they came by the tape (I didn't). It was the first of many attempts by the AT&T lawyers to justify their existence and to kill UNIX.

At the 1994 USENIX technical meeting, there was a 25th birthday session after which Lou "confessed" that he had received a phone message at Columbia to the effect that if he drove down to Mountain Avenue "around 2pm," he'd "find" something of interest. So he and Reidar Bornholdt drove from Manhattan to Murray Hill and "found" the can with the tape in it. Ken told me he had "no idea" how the tape had gotten there. Dennis suggested that it might have "fallen from a truck." Everyone laughed.

At this time AT&T had a strict policy of

1. no advertising
2. no support
3. no bug fixes
4. payment in advance

This forced the users to band together and compelled them to share what they had learned and what they knew.

### **A Tale of Two Editors**

In Chapter 1, I mentioned CTSS and ITS. At that time, early in the 1960s, TECO (Tape Editor and COrrector; later, Text Editor ...) was the editor everyone used on the PDP-1

and, later, on the PDP-6. Not only was it widely used, but just about everyone modified it. (In RFC 681, quoted earlier, an editor "based on QED" is mentioned. QED was written by Butler Lampson, who wrote the QED text editor for the Berkeley Time-Sharing System on the SDS 940. It was character-oriented and based on TECO. Ken Thompson used this version of QED while a student at Berkeley, prior to going to Bell Labs in 1966.) Indirectly, TECO was the ancestor of vi; directly, it was the parent of Emacs (= Editing macros for TECO).

Interestingly, Bill Joy created vi in 1976 and Richard Stallman (together with Guy Steele and Dave Moon) created Emacs the same year. The original version was based on TECMAC and TMACS, two TECO editors. Stallman and Michael McMahon ported it to the Tenex [for the DEC-10] and TOPS-20 [for the DEC-20] operating systems. [James Gosling, the creator of Oak/Java, wrote the first Emacs for UNIX at Carnegie-Mellon in 1981. RMS began work on GNU EMACS in 1984.]

Joy's creation had a more complex origin.

The editor created by Ken Thompson in August 1969 was called ed. Ken had written a version of QED for CTSS on the IBM 7094 at MIT. He and Ritchie then wrote a version for the GE-635 at Bell Labs. The cut-down version of this for the PDP-7 was ed. While TECO was known for its complex syntax, ed must have been the most user-hostile editor ever created.

Across the Atlantic in London, George Coulouris at Queen Mary College (now Queen Mary and Wakefield College) had gotten UNIX v4 in late 1973. George explained to me how unhappy he had been with ed and how he created em (editor for mortals) so that QMC students could "exploit more effectively some vdu [visual display unit] that we had recently acquired..."

Then I spent the summer of 1976 as a visitor to the CS Department at Berkeley. I worked in a room full of teletype terminals using the departmental UNIX. I had brought em with me on DECTape and installed it there for my own use...

One day, sitting at the next terminal was this fairly frenzied hacker/Ph.D. Student [Bill Joy] who told me he was writing a Pascal compiler. I showed him em, and he said "that's nice, the systems support people might be interested in that." He took me and introduced me to them. They had a couple of PDP-11s ... supporting several rooms full of vdu terminals connected at 9600 baud, an environment in which em could really shine.

I explained that em was an extension of ed that gave key-stroke level interaction for editing within a single line, displaying the up-to-date line on the screen (a sort of single-line screen editor)...

The system support person [Jeff Schriebman] said something like: "That's very nice, but if we made it available to all of our users the overheads associated with running in raw mode would swamp the cpu."

I was rather depressed by this reaction, thinking "I guess I have been unrealistic in developing an editor that is so expensive to run..."

Nevertheless, Bill and the support people took a copy of my source to see if they would use it. I then went to the East Coast for a week or so. When I returned, I found that Bill had taken my code as a starting point and had got a long way towards what was to become ex and subsequently vi, and that the editor was installed on the service machines ...

1976! Created in 1969, ed had travelled east to Australia and west to Vienna. Coulouris had created em in London and brought it to Berkeley. Now the Berkeley editor, ex, would be available on the first UCB tape. But vi, which was available on 2BSD (1979), only made it into a BTL distribution with v8 (1985).

Even in 1976, international communication and access to source meant the distribution of new tools and new programs encouraged and enlivened the user community. Let's look at the landscape for a few minutes.

1. In 1974, Bob Kahn and Vint Cerf published the first paper describing what was to become TCP/IP.
2. In 1975, RFC 681 was published.
3. In January 1976, there were 63 hosts on the ARPAnet, which was on the verge of becoming the Internet.
4. And UNIX was available throughout the world -- but only on hardware that cost well over \$10,000.

### **UUCP and USENET**

Also in 1976, Mike Lesk at AT&T developed UUCP (UNIX-to-UNIX copy). <sup>1</sup> V2 was implemented in 1977.

UUCP meant that information could be directed around the network (as it was). It also meant that one could establish a telephone connection and transmit information across that (relatively expensive) link. Two years later, three graduate students in North Carolina (Tom Truscott, Jim Ellis, and Steve Bellovin) took the next step.

Tom Truscott had an early interest in chess. While a student at Duke in 1974, he devised a chess program (Duchess) and played against Ken Thompson's Belle. Duchess lost on time. (In competitive chess, each side has a given time to make its next move; Duchess exceeded that time due to a core dump.) But Truscott competed in every ACM computer

chess tournament from 1974 through 1980. He also attended the 1976 UNIX Users Group meeting at Harvard (1-2 April) and the 1978 meeting at Columbia (24-28 May), where he met Ken and others. In 1979, Truscott went to the Labs as a summer student and, on his return to Duke, arranged for a UUCP link. (He also attended the USENIX meeting in Toronto [20-23 June], to which we'll return in the next chapter.)

When he returned to Duke, he found that Jim Ellis had installed V7 on the Computer Science PDP 11/70. They employed the auto-dialer capacity to dial up two other Duke computers and one at the University of North Carolina. Ellis and Truscott then called a meeting to discuss their idea -- to have something like the mailing lists on the ARPAnet for computer sites that weren't on the ARPAnet. Steve Bellovin, then a graduate student at the University of North Carolina, attended and then wrote the first Netnews program -- three pages of shell script (later rewritten in C). The first implementation was between the Duke and UNC Computer Science departments; the Duke Medical Center Department of Physiology was added at the beginning of 1980. In January 1980, Ellis and Truscott went to Boulder, CO, and announced their Netnews design at the USENIX meeting (January 29 - February 1).

The first version of Netnews was fairly simple and efficient. It periodically checked the "last saved" time-stamp of each file in a specified directory, and then sent any file updated since the last check to another computer using UUCP across a modem connection.

Tom Truscott and Steve Daniel (also a graduate student at Duke) then rewrote the program to create what was called Netnews Version A. Since Netnews was designed for UNIX at a university, it was automatically categorized as public domain software under the conditions of the AT&T UNIX license, which greatly facilitated its subsequent use and adoption. This implementation appeared on the 1980 USENIX distribution tape, which was distributed at the Newark, DE, meeting (June 17-20). Duke University then invited other sites to join the network, which was made easier by the fact that the software was free, starting the first Usenet expansion -- to 15 sites. But one of those was Berkeley, which resulted in an explosive growth spurt.

That connection was the responsibility of Armando Stettner, then with DEC. Someone at the Delaware meeting complained about the inordinate cost of the long-distance telephone connections needed to get news to the West Coast. Armando spoke to Bill Shannon and they said that if they could get a news feed to decvax (in New Hampshire), they'd pick up the Berkeley phone bill. [Armando later supplied the first news feeds to Europe, Japan, and Australia, too.] The network soon spread to universities across North America, quickly establishing a critical mass of useful information, which made it even more popular.

In under a year, Usenet grew to 100 sites and over 25 articles per day (the original protocol had been optimized for 1-2 messages per day). Mike Lesk had never contemplated such uses of uucp. Truscott, Ellis and Bellovin had never imagined such popularity. The system collapsed.

In 1982, Netnews Version B was developed by Mark Horton (a graduate student at Berkeley) and Matt Glickman (a high school student) to increase efficiency so that USENET could cope with the increasing loads on the growing network. Horton continued to maintain the system till 1984, when Rick Adams at the Center for Seismic Studies took over maintenance of Version B, releasing 2.10.2. There were 135 news groups at the end of March. Version 2.11 of Netnews was released in late 1986.

In 1989, Netnews Version C was developed by Henry Spencer and Geoff Collyer at the University of Toronto again increasing efficiency.

By the late 1980s there were 20,000 newsgroups. Looking at what was current became nearly impossible. The remedy was "searching." As Mike O'Dell put it: "The Internet is now a rich fabric of resources and capabilities, and it is no longer possible to simply *know* all the places where interesting stuff is available. We now need tools with which to discover and navigate this world-wide treasure trove" (*Computing Systems* 5.4 [Fall 1992] p. 373).

### **Searching in a Distributed Environment**

By the beginning of 1992, there were a number of "resource discovery systems" (much of this section is drawn from the special issue of *Computing Systems* cited above).

The earliest of these were search engines which pointed at specific, dedicated servers: whois, which responded to queries about people, network numbers and domains across the Internet; and X.500 (ISO DIS 9594-1 [1988]), a distributed directory look-up service.

The next tool was archie, developed by Alan Emtage and Peter Deutsch when they were at McGill University in Montreal. archie maintained a directory of materials available on about 1100 UNIX archives accessible by anonymous FTP. In 1992 it contained about 2.6 million files with 150 gigabytes of information.

Gopher (from the University of Minnesota's "Golden Gophers") provided a simple menu-driven interface to data searching.

Prospero (Neuman, 1992) was an "enabling technology," allowing users to create their own views of information in a distributed file system.



WAIS, wide-area information service, developed by Brewster Kahle and colleagues at Thinking Machines, was a network of over 70 servers world-wide, offering access to over 300 databases by natural language keyword search.

There were also knowbots (developed at the CNRI) and Netfind and a host of others.

But the "winner" was the World Wide Web (WWW), invented by Tim Berners-Lee at CERN, and first published in *Electronic Networking* in Spring 1992. We will return to the Web in about a dozen years.

### **1979**

In Chapter 3, UNIX V6, John Lions, and the ports of UNIX to the Interdata 7 and the Interdata 8 were mentioned. We're about to move on to V7. But first, a note about names and dates.

**Names.** By and large, Unix users refer to "Sixth Edition" and "V6" interchangeably. At Bell Labs, there was a continually changing version of Unix running. Only when Doug McIlroy caused the first "UNIX PROGRAMMER'S MANUAL" to be written, did there appear to be a fixed form. So, the manuals were listed by "Edition," and the system referred to was the "Version."

**Dates.** Every AT&T manual carried a date. This is the set.

- First Edition - November 3, 1971
- Second Edition - June 12, 1972
- Third Edition - February 1973
- Fourth Edition - November 1973
- Fifth Edition - June 1974
- Sixth Edition - May 1975
- Seventh Edition - January 1979
- Eighth Edition - February 1985
- Ninth Edition - September 1986
- Tenth Edition - October 1989

**V7.** The wonder of V6 was that it was under 10,000 lines of code. (But it had no full-screen editor, no windowing system, etc.) V7 was much larger. And it contained much more.

V7 accommodated large filesystems; it did not restrict the number of user accounts; it had improved reliability. Steve Johnson referred to V7 as "the first portable Unix. It also had a large number of new commands. Among these were:

**COMMANDS:** at, awk, calendar, cb, cd, cpio, cu, deroff, expr, f77, find, lex, lint, m4, make, refer, sed, tail, tar, touch, uucp, uux

**SYSTEM CALLS:** ioctl

**SUBROUTINES:** malloc, stdio, string

**GAMES:** backgammon

awk (=Aho-Weinberger-Kernighan; a pattern scanning and processing language; gawk [GNU AWK] is the most employed version), lint (Johnson, a C program verifier), make (Feldman, a utility to simplify the maintenance of other programs), and uucp (Lesk) would have been enough, but there was much more. The V7 manual had grown to over 400 pages, with two 400-page supplementary volumes. V7 contained a full Kernighan-Ritchie C compiler; a far more sophisticated shell (sh), the Bourne shell; Dick Haight's find, cpio and expr; and a large number of include files. Dated "January 1979," the title page of the Seventh Edition *UNIX PROGRAMMER'S MANUAL* bore neither Dennis' nor Ken's name. It was headed: **UNIX™ TIME-SHARING SYSTEM.**

Along with all this, V7 came with a major drawback: its performance was poorer than most V6 systems, especially those that had been "tuned." The users went to work.

Bill Joy (at Berkeley) changed the size of the data blocks on the VAX 11/780. (Jeff Schriebman ported that to the PDP-11/70 in April 1980, having gone to UniSoft from Berkeley.) In December 1979, Ed Gould moved the buffers out of kernel address space. Joy changed the stdio library on the VAX and Tom Ferrin (at UCSF) ported those changes to the PDP-11. Tom London (in Holmdel, NJ) improved the movement of output characters from user space to kernel space. John Lions (at UNSW in Australia) proposed a new procedure for directory pathnames. UNSW also provided new code for process table matches. Bruce Borden (at the RAND Corporation) provided the symorder program. Ferrin also rewrote parts of copyseg() and clearseg(). (The entire set of improvements was made available to the community on a PDP-11 distribution: 2.8.1BSD -- it was announced by Ferrin at the USENIX Conference in January 1982 in Santa Monica, CA.

The users had enhanced V7's performance dramatically.

But I've gone too far ahead.

**USENIX.** Ron Baecker of the University of Toronto hosted the USENIX Association meeting in Toronto, June 20-23. There were about 400 attendees.

Al Arms, an AT&T lawyer, announced the new licensing terms for V7. In addition to a fee schedule that increased the costs of the various licenses, the academic/research license no longer automatically permitted classroom use. As Greg Rose (who was one of John Lions' students at UNSW and is now with Qualcomm) told me:

It seems that in so many stages of Unix's evolution, an action that AT&T took in order to stifle something actually caused the opposite to happen.

Suppressing Lions' commentary led to wholesale Xeroxing. Here it led to something truly unforeseen.

Andrew S. Tanenbaum of the Vrije Universiteit in Amsterdam had been using V6 in his classes on operating systems. It now appeared that he would not be able to employ V7. "It was the new licensing restrictions," he told me.

When AT&T released Version 7, it began to realize that UNIX was a valuable commercial product, so it issued Version 7 with a license that prohibited the source code from being studied in courses, in order to avoid endangering its status as a trade secret. Many universities complied by simply dropping the study of UNIX, and teaching only theory. [*Operating Systems* (1987), p. 13]

Andy was not to be deterred: he created an operating system like Unix for use on Intel's new x86 architecture: Minix.

**MIT.** In 1974, Robert Greenblatt at the MIT AI Lab began the Lisp machine project. His first machine was called CONS (1975). This was improved into a version called CADR (1977). CADR was the direct ancestor of both Lisp Machines, Inc., of which Greenblatt was the founder and president, and Symbolics.

And Symbolics, in several ways, forced Richard Stallman to form the Free Software Foundation and the GNU Project. I'll discuss these in Chapter 8.

**Berkeley.** Though I will go into detail concerning the Computing Systems Research Group (CSRG) at Berkeley in the next chapter, I think it important to note here that 3BSD, the first Berkeley release for the VAX, came out at the end of 1979. But this was not based on V7, but on 32V. Berkeley utilities and modifications from 2BSD (including the C shell) had been added, as well as a virtual memory system done by a number of Berkeley graduate students, including Bill Joy.

It was an exciting year.

**32V.** Before going on to Berkeley, I should say something about 32V, the UNIX port to the VAX.

Nearly four years passed between V6 and V7. But, as I noted, things didn't stand still at the Labs. In fact, what we think of as V7 was available internally nearly a year prior to the publication of Seventh Edition.

Things hadn't remained static at DEC, either, and the first VAX (Virtual Address eXtension), a 32-bit computer, was pre-announced in 1977 and went on sale in 1978. Dennis, Ken and Steve Johnson felt alienated by DEC, for a variety of reasons. And Dennis and Steve were working on the port to the Interdata 8. So when DEC offered them a VAX, they just said "no." DEC then turned to Bell Labs in Holmdel. I spoke to Charlie Roberts, who was to manage the project.

DEC came to us in Holmdel. We were clearly the second-string. Tom London and John Reiser were interested and so was Ken Swanson, and we got the VAX in early '78. I didn't do any of the technical work. In fact, I devoted a lot of energy and time to getting the management to let us do it. It wasn't research you see. However, they let us take the time. And in about three months my small group ported Version 7 to the VAX. We got the machine in January; they had it running in April; and by August it really worked. By then folks knew what we were doing. I had had calls from about six universities -- Brown, UCLA, Berkeley, Waterloo. I can't recall the others. So I went to Roy Lipton and Al Arms in Patents and Licensing about getting it out. After a lot of back-and-forth, they decided that we could give it to one university for research purposes and that Al would set up a "special research agreement" with that institution.

I had met [Bob] Fabry at a couple of conferences and I had been out in Berkeley and given a paper and talked to [Domenico] Ferrari as well as Emmanuel Blum and Bill Joy. So, with the blessings of BTL area 11 management, we sent 32V to Berkeley. It was in October or November 1978.

With that background, let's go cross country to Berkeley.