Bell's Law for the Birth and Death of Computer Classes: A theory of the Computer's Evolution[1]

Gordon Bell, Microsoft Research, Silicon Valley

Introduction

In 1951, a person could walk inside a computer and by 2010 a single computer (or "cluster') with millions of processors has expanded to building size. More importantly, computers are beginning to "walk" inside of us. These ends illustrate the vast dynamic range in computing power, size, cost, etc. for early 21st century computer classes.

A computer class is a set of computers in a particular price range with unique or similar programming environments (e.g. Linux, OS/360, Palm, Symbian, Windows) that support a variety of applications that communicate with people and/or other systems. A new computer class forms roughly each decade establishing a new industry. A class may be the consequence and combination of a new platform with a new programming environment, a new network, and new interface with people and/or other information processing systems.

Bell's Law accounts for the *formation, evolution, and death of computer classes* based on logic technology evolution beginning with the invention of the computer and the computer industry in the first generation, *vacuum tube computers* (1950-1960), second generation, *transistor computers* (1958-1970), through the invention and evolutions of the third generation *TTL and ECL bipolar Integrated Circuits* (1965-1985), and the fourth generation *bipolar, MOS and CMOS ICs enabling the microprocessor*, (1971) represents a "break point" in the theory because it eliminated the other early, more slowly evolving technologies. Moore's Law (Moore 1965, revised in 1975) is an observation about integrated circuit semiconductor process improvements or evolution since the first IC chips, and in 2007 Moore extended the prediction for 10-15 more years:

Transistors per chip = $2^{(t-1959)}$ for $1959 \leq t \leq 1975$; $2^{16} \times 2^{(t-1975)/1.5}$ for $t \geq 1975$.

In 2007, Moore predicted another 10-15 years of density evolution. The evolutionary characteristics of disks, networks, display, and other user interface technologies will not be discussed. However for classes to form and evolve, <u>all technologies</u> need to evolve in scale, size, and performance, (Gray, 2000) though at comparable, but their own rates!

In the first period, the mainframe, followed by minimal computers, smaller mainframes, supercomputers, and minicomputers established themselves as classes in the first and second generations and evolved with the 3rd generation integrated circuits c1965-1990. In the second or current period, with the 4th generation, marked by the single processor-on-a-chip, evolving large scale integrated circuits (1971-present) *<u>CMOS became the single, determinant technology for establishing all computer classes</u>*. By 2010, scalable CMOS microprocessors combined into powerful, multiple processor clusters of up to a million independent computing streams will certainly exist. Beginning in the mid 1980s, scalable systems have eliminated and replaced the previously established, more slowly evolving classes of the first period that used interconnected bipolar and ECL ICs. Simultaneously *smaller*, CMOS system-on-a-chip computer evolution has enabled low cost, small form factor or cell phone sized devices; PDA, cell phone, personal audio (and video) device (PAD, PA/VD), GPS and camera convergence into a single platform has become the worldwide personal computer, c2010. Dust sized chips with a relatively small numbers of transistors enable the creation of ubiquitous, radio networked, implantable, sensing platforms to be part of everything and everybody as a wireless sensor network class. Field Programmable Logic Array chips with 10s-100s of million cells exist as truly universal devices for building "anything".

Bell's Law Origin & Motivation—The Computer History Museum, a By-product

In 1966, after six years as a computer engineer at Digital Equipment Corporation, designing the first computers that established the minicomputer industry and the first timesharing computers, I joined the faculty of Carnegie Mellon University. While mentoring me for six years, Allen Newell and I wrote Computer Structures: Readings and Examples (Bell & Newell, 1971) which posited notations to describe computers, their behavior, and a taxonomy of computers including their constituent components. Working with Newell stimulated a deep concern about the origin of computers, classifying them (e.g. size, function, price, performance), and especially their evolution. Several of us wrote a paper (Bell et al, 1972) that showed computers were falling into several different price bands over time, similar to other manufactured goods e.g. cars, planes and in addition, new computers were being introduced in lower price bands afforded by the logic and memory technology.

On returning to Digital in 1972 as its VP of Engineering, I started collecting computer logic and memory technology in my office. Simultaneously, Ken Olsen, acquired two historically important MIT computers: Whirlwind (c1951), and TX-0 (c1956) that

[1] An abridged version of this paper has appeared in the Communications of the ACM, Vol. 51, No. 1, January 2008.

should be preserved for history, and that might be part of some eventual display. In 1975, I curated an exhibit of logic and memory in a converted coat closet of Digital's main office building, Maynard, MA that eventually moved and occupied the lobby of at Marlboro MA. Maurice Wilkes opened the Digital Computer Museum there in 1979.

As head of engineering and curator of a potential Computer Museum, I first spoke at MIT and elsewhere (Bell, 1972) about the future of computing based on logic technology. It also became clear that once established, a class stays roughly constant price. I used this basic idea to look back in time to create early generations: manual (1600-1800), mechanical (1800-1890), electro-mechanical (1890-1930, vacuum tube (1930-1960), transistor (1959-1966), integrated circuit 1966-1990), microprocessor (1971-present). In 1980 I gave a talk at Stanford's First Forsythe Lecture, "Generating Computer Generations" describing my theory on computer classes based on structure, technology, need, and actual use that has since been refined as I describe.

The museum became a public 501c(3) institution when it opened in Boston in 1983. In 1995 the artifacts moved to Silicon Valley, as the Computer History Museum, Mountain View, CA.

Bell's Law

A *computer class* is a set of computers in a particular price range defined by: a programming environment e.g. Linux, Windows to support a variety of applications including embedded apps; a network; and user interface for communication with other information processing systems including people and other information processing systems. A class establishes a horizontally structured industry composed of hardware components through operating systems, languages, application programs and unique content e.g. databases, games, pictures, songs, video that serves a market through various distribution channels.

The universal nature of stored program computers is such that a computer may be programmed to replicate function from another class. Hence, over time, one class may subsume or kill off another class. Computers are generally created for one or more basic information processing functions– storage, computation, communication, or control (see Figure1 Taxonomy). Market demand for a class and among all classes is fairly elastic. In 2010, the number of units sold in classes vary from 10s, for computers costing around $100 million to billions for small form factor devices e.g. cell phones selling for under $100. Costs decline by increasing volume through manufacturing learning curves (i.e. doubling the total number of units produced result in cost reduction of 10-15%). Finally, computing resources including processing, memory, and network are fungible and can be traded off at various levels of a computing hierarchy e.g. data can be held personally or provided globally and held on the web.

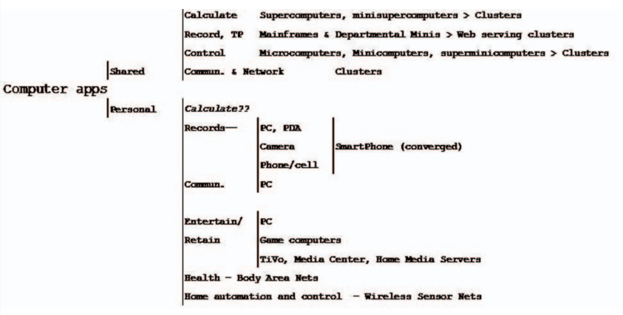
Computer apps
Shared
Calculate
Supercomputers, minisupercomputers > Clusters
Record, TP
Mainframes & Departmental Minis > Web serving clusters
Control
Microcomputers, Minicomputers, superminicomputers > Clusters
Commun. & Network
Clusters
Personal
Calculate??
Records—
PC, PDA
Camera
Phone/cell
SmartPhone (converged)
Commun.
PC
Entertain/
Retain
PC
Game computers
TiVo, Media Center, Home Media Servers
Health – Body Area Nets
Home automation and control – Wireless Sensor Nets

Figure 1. Taxonomy of computer functions (applications) taxonomy divided into personal and non-personal, i.e. institutional infrastructure computers that carry out calculation, record keeping and transaction processing, networking and personal communication (e.g. word processing, email, web), control, personal health, and entertainment functions. Note the convergences: personal media device, PDA, camera, cell phone become the Smart Phone; Entertainment devices of TV, Media Centers & Servers.

The class creation, evolution, and dissolution process can be seen in the three design styles and price trajectories and one resulting performance trajectory that threatens higher priced classes: an established class tends to be re-implemented to maintain its price, providing increasing performance; minis or minimal cost computer designs are created by using the technology improvements to create smaller computers used in more special ways; supercomputer design, i.e. the largest computers at a given time, come into existence by competing and "pushing technology to the limit" to meet the unending demand for capability; and the inherent increases in performance at every class, including just constant price, threaten and often subsume higher priced classes.

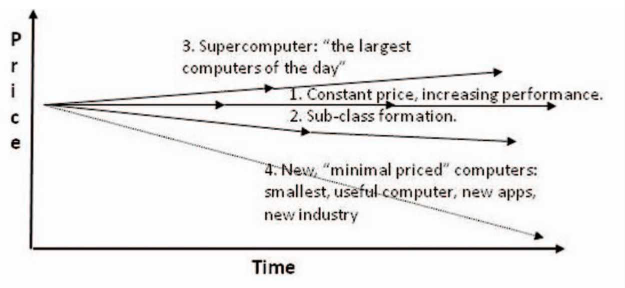

Figure 2. evolving computer classes based on technology and design styles: 1. constant price, INcreasing Performance; 2. sub-class, lower price and performance to extend range; 3. supercomputer – largest computers that can be built that extend performance; and 4. new, minimal, order of magnitude lowe priced class formations every decade.

All of the classes taken together that form the computer and communications industry shown in Figure 2, behave generally as follows:

1. *Computers are born* i.e. classes come into existence through intense, competitive, entrepreneurial action over a period of 2-3 years to occupy a price range, through the confluence of new hardware, programming environments,

networks, interfaces, applications, and distribution channels. During the formation period, 10s to 100s of companies compete to establish a market position. After this formative and rapid growth period, 2 or 3, or a dozen primary companies remain as a class reaches maturity depending on the class volume.

2. *A computer class, determined by a unique price range evolves in functionality and gradually expanding price range of 10* maintains a stable market. This is *followed by a similar lower priced sub-class that expands the range another factor of 5 to 10.* Evolution is similar to Newton's First Law (i.e. bodies maintain their motion and direction unless acted on externally). For example, the "mainframe" class was established in the early 1950s using vacuum tube technology by Univac and IBM and functionally bifurcated into commercial and scientific applications. Constant price evolution follows directly from Moore's Law whereby a given collection of chips provide more transistors and hence more performance.

   A lower entry price, similar characteristics sub-class often follows to increase the class's price range by another factor of 5 to 10, attracting more usage and extending the market. For example, smaller "mainframes" existed within 5 years after the first larger computers as subclasses.

3. *Semiconductor density and packaging inherently enable performance increase to support a trajectory of <u>increasing</u> price and function*

   3.1 *Moore's Law single chip evolution, or microprocessor computer evolution after 1971 enabled new, higher performing and more expensive classes.* The initial introduction of the microprocessor at a substantially lower cost accounted for formation of the initial microcomputer that was programmed to be a calculator. This was followed by more powerful, more expensive classes forming including the home computer, personal computer, workstation, the shared microcomputer, and eventually every higher class.

   3.2 *The supercomputer class c1960 was established as the highest performance computer of the day*— however, since the mid-1990s supercomputers are formed by combining the largest number of high performance computers to form a single, clustered computer system in a single facility. In 2010 over a million processors will likely constitute a cluster. Geographically coupled computers including GRID computing e.g. SETI@home are outside the scope.

4. *Approximately every decade a new computer class forms as a new "minimal" computer either through using fewer components or use of a small fractional part of the state-of-the-art chips.* For example, the 100 fold increase in component density per decade enables smaller chips, disks, screens, etc. at the same functionality of the previous decade especially since powerful microprocessor cores e.g. the ARM use only a few <100,000 transistors versus over a billion for the largest Itanium derivatives.

   Minimal computers design. Building the smallest possible computer accounts for the creation of computers that were used by one person at a time and were forerunners of the workstation e.g. Bendix G-15 and LGP 30 in 1955, but the first truly personal computer was the 1962 Laboratory Instrument Computer (LINC). LINC was a self-contained computer for an individual's sole use with appropriate interfacial hardware (e.g. keyboards, displays), program/data filing system, with interactive program creation and execution software. Digital Equipment's PDP-1 (1961), followed by its more "minimal" PDP-5 & 8 established the minicomputer class that were predominately designed for embedded applications.

   System-on-a-Chip (SOCs) use a fraction of a chip for the microprocessor(s) portion or "cores" to create classes and are the basis of fixed function devices and appliances beginning in the mid 1990s. These include cameras, cell phones, PDAs, PAD (personal audio & video devices) and their convergence into a single cell phone sized device (CPSD) or small form factor (SFF) package. This accounts for the PC's rapidly evolving microprocessor's ability to directly subsume the 1980's workstation class by 1990.

5. *Computer classes die or are overtaken by lower priced, more rapidly evolving general purpose computers* as the less expensive alternatives <u>operating alone, combined into multiple shared memory micro-processors, and multiple computer clusters</u>. Lower priced platforms result in more use and substantially higher volume manufacture thereby decreasing cost while simultaneously increasing performance more rapidly than higher priced classes.

   5.1 Computers can be combined to form a single, shared memory computer. A "multi" or multiple CMOS microprocessor, shared memory computer displaced bipolar minicomputers c1990 and mainframes c1995, and formed a component for supercomputers.

   5.2 Scalable, multiple computers can be networked into arbitrary large computer to form "clusters "that replace custom ECL and CMOS vector supercomputers beginning mid 1990s simply because arbitrarily large computers can be created. Clusters of multiprocessors were called constellations; clusters using low latency and proprietary networks are

MPPs (massively parallel processors).

5.3 *Generality ALWAYS Wins! A computers created for a particular, specialized function e.g. word processing, interpreting a language, used for a particular application is almost certain to be taken over by a faster evolving, general purpose computer.* The computer's universality property allows any computer to take on the function of another, given sufficient memory and interfaces.

5.4 Small form factor devices subsume a personal computing functionality as they take on the communications functions of the PC (e.g. email and web browsing), given sufficient memory and interfaces. Small form factor devices or television sets or kiosks accessing supercomputers with large stores, subsume personal computing functionality. The large central stores retain personal information, photos, music, and video.

The paper will describe how these characteristics of the classes account for the birth, growth, diminution, and demise of various parts of the computers and communications industry.

Overview of the Birth and Death of the Computer Classes 1951-2010

Figure 1 is a computer function taxonomy based: first on buyers/users and second, by application. The information processing elements i.e. application functions are: memory or storage for record keeping that was the province of IBM and other card tabulation equipment makers prior to the computer's invention; computation or calculation characterizing science and engineering use; networking and communication that provide the interconnection infrastructure; control of other systems (e.g. process control); and interface with humans and other information processing entities.

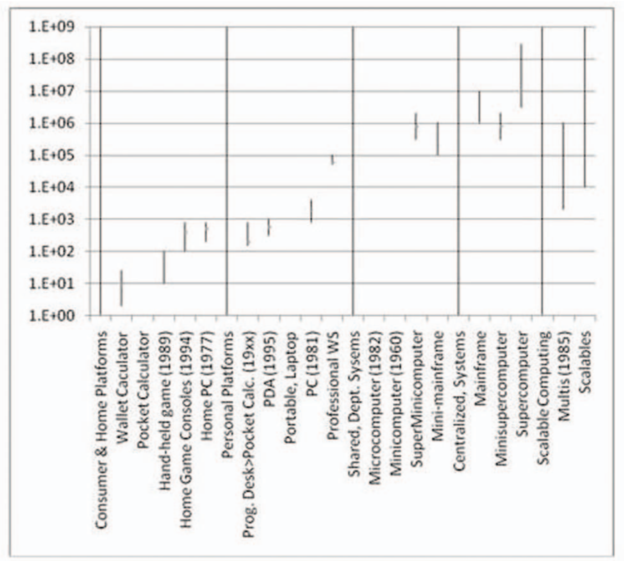

Figure 3. Computer Classes and their Price Range 2005

The taxonomy is divided first into personal and non-personal or invisible and shared, institutional infrastructure systems that would be operated within or for a company, government or institution as a service. This dichotomy of personal versus shared; invisible versus institutional determines characteristics of price and scale, programming environment, user interface, and network. Function though critical, will be neglected.

The named classes and their price range c2010 is given in Figure 3. David Nelson, founder of Apollo. and I (Nelson, Bell 1986) posited that the price of a computer was roughly $200 per pound. Figure 4 gives the introduction price and date of the first or defining computer of a class. Table 1 gives the defining constituent technologies, operating systems, languages, networks, and interfaces of the various classes.

The discussion will use the aspects of Bell's Law described above and follow a timeline of the class formations beginning with the establishment of the first computer classes (mainframe, supercomputer, shared personal professional computers or workstations, and minicomputers) using vacuum tubes, transistors, and bipolar integrated circuits that continue through the mid 1990s. The MOS microprocessor introduced in 1971 ultimately overtook bipolar by 1990 to establish a single line based on CMOS technology.

The section is followed by the three direct and indirect effects of Moore's Law to determine classes:

1 Microprocessor transistor/chip evolution c1971-1985 establish: calculators, home computers, personal computers and workstations, and lower (than minicomputer) priced computers.

2 "Minimal" designs establish new classes c1990 that use a "fraction" of the Moore number. Microsystems evolution using fractional Moore's Law sized SOCs enable small, lower performing, minimal personal computer and communication systems including PDAs, cameras, cell phones, personal audio/video devices.

3 Rapidly evolving microprocessors using CMOS and a simpler RISC architecture appear as the "killer micro" c1985 to have the same performance as supercomputers, mainframes, mini-supercomputers, super-minicomputers, and minicomputers built from slowly evolving, low density, custom ECL and bipolar integrated circuits. ECL survived in supercomputers the longest because of its speed and ability to drive the long transmission lines, inherent in large systems. In the end, CMOS density and faster system clock overtook ECL as shown in Figure 5.

The "killer micro" enabled by fast floating point arithmetic, first subsumed the workstation followed by the minicomputer especially when combined to form the "multi" or multiple microprocessor shared memory computer c1985. "Multis" became the component for scalable clusters when interconnected by high speed, low latency networks. Clusters allow

Table 1. The Computer classes, hardware and software enablers, network, and use.

Class & Lifetime	Hardware Platform (logic, memories)	User Interface & control, Operating System, Languages	Network infrastructure	Use
Mainframes: Commercial & Science. Industry Formation 1951	Inventing 'the computer", vacuum tube totransistor, core, drum & mag tape to disk, card, paper tape. **Drum;**	direct user control evolving to batch operating systems (O/S) 1960: COBOL & FORTRAN	computers operated as independent stand-alone entities	Accounting & Records, Scientific Calculations
Small, shared,personal computers for work (1956-				
Interactive timesharing replaces Batch (1965)	IBM RJE, DEC PDP-10	**glass teletype** & glass keypunch, command language control	POTS using **modem,** and proprietary nets using WAN	Service centers;
Minicomputers 1965-90; Super-minicomputers (1980-1990)	integrated circuit (IC), disk , minicomputer; multiprogramming			Real time for process control & lab; network switch, departmental timeshared "mainframe"
Supercomputers 1965-1995 The largest scientific computer.	**Transistor discrete to ECL small scale ICs**	**FORTRAN**	Batch and remote batch from terminals and card equipment	Science & engineering; design, simulation
Mini-Supercomputers	Cray-ettes. CMOS implementation of Cray vector multiprocesso	FORTRAN		
Scalables: MPPs, Clusters, and Constellations				
Calculators, Home Personal Computers based on 1st microprocessor: 1971-1975	Microprocessor		**Stand-alone use**	
PCs, Workstations, and LANs (1981)	**microprocessor** PCs & workstations, floppy, small disk, dist'd O/S	**WIMP (windows, icons, mouse, pull-down menus)**	**WAN, LAN**	
World-Wide Web. Access via PCs and Workstations (1994)	Evolutionary PCs and workstations, servers everywhere, Web O/S	Browser	**fiber optics backbone, www, http**	
Convergence: Cell Phone, Personal Digital Assistant (PDA), Digital A/V Player (DAP) , Camera …SmartPhone.				
Web Computers: Network-, Tele-, TV- computers (1998)	client software from server using JAVA, Active X, etc.	telephone, simple videophone, television access to the web	**xSDL** for POTS or cable access for hi speed data; 3 separate networks	
SNAP: Scalable Network & Platforms (1998)	PC uni- or multi-processor commodity platform	server provisioning	SAN (System Area Network) for clusters	All computers
Convergence: Ubiquitous IP: phone, TV, data, & videophone (2010)	Video capable devices of all types;	video as a primary data-type	Single high speed network access; Home Net	
Wireless Sensor Nets (2005) Connecting all "things"				
Embedding of speech & vision functions (2020)	$1-10 of chip area for: books, pictures, papers, that identify themselves		**Body Net** , Home Net, other nets	
Body Net: vision, hearing, monitoring, control, comm., location (2015)	artificial retina, cochlea, glasses for display,	implanted sensors and effectors for virtually every part of a body	Body Network, gateway to local IR or radio nets everywhere	

arbitrarily large computers that are limited only by customer budgets. Thus scalability allows every computer structure from a few thousand dollars to several hundred million dollars to be arranged into clusters built from the same components.

In the same fashion that killer micros subsumed all the computer classes by combining, it can be speculated that much higher volume, hundreds of millions, of small form factor devices, may evolve more rapidly to subsume a large fraction of personal computing. Finally tens of billions of dust sized, embeddable wirelessly connected platforms that connect everything are likely to be the largest class of all enabling the state of everything to be sensed, effected, and communicated with.

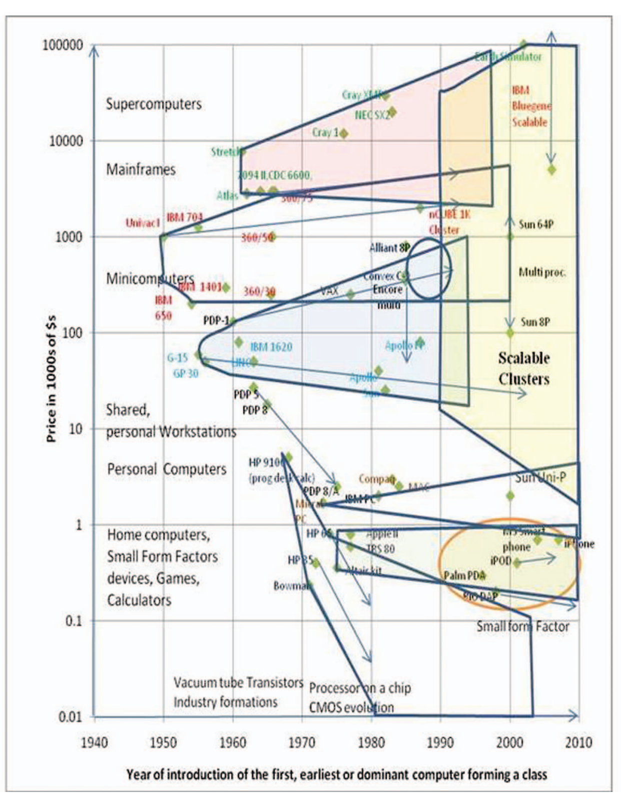

Figure 4. Introduction price versus date of the first or early platforms to establish a computer class or lower priced sub-class orginating from the same company or industry.

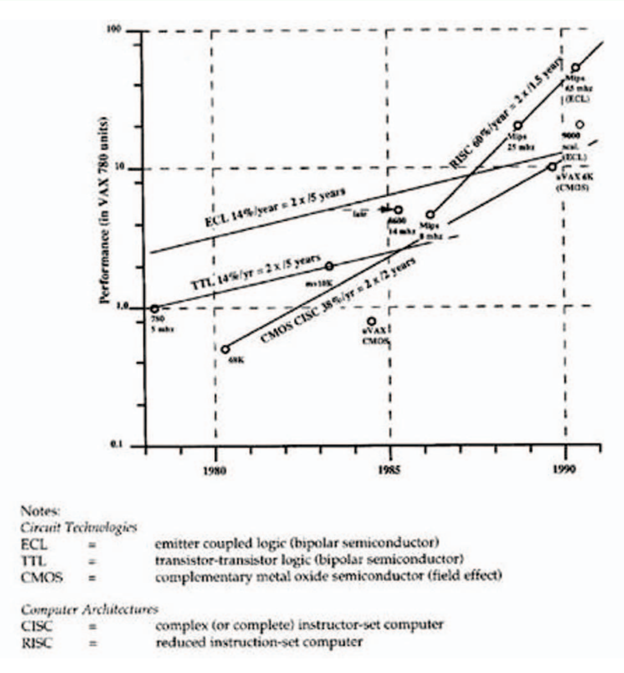

Figure 5. Faster evolving CMOS microprocessors are able to overtake and eliminate slowly evolving TTL and ECL bipolar integrated circuit based computer classes including minicomputers, superminicomputers, mini-supercomputers, mainframes, and supercomputers. A number of companies built one or more Too many ECL computers including CDC, Cray, DEC, Fujitsu, Hitachi, and IBM before switching to ECL.

The Past: How we got here

The Beginning (1951-1990): mainframe, supercomputer, shared personal workstation, and minicomputer classes

By 1970, vacuum tube (50s), transistor (60s), and small scale integrated circuit (late 60s) technologies enabled the establishment of four classes of computers that continued almost without change until the 80s:

1. Mainframes for commercial, record keeping, etc. and mainframes for Scientific and Engineering Computation were the very first computers; a sub-class of smaller computers formed that were used in the same fashion.
2. Minimal design, small, shared computers that were used directly as personal workstations
3. Minimal computers for process and machine control, communication, and embedded apps
4. Supercomputers constructed at the limits of circuit, interconnect, and architectural complexity utilizing clock speed and parallelism

Eckert and Mauchly, operating as the UNIVAC division of Remington Rand delivered the UNIVAC 1 as the earliest commercial computer in 1951, roughly concurrent with the British LEO (Lyons Electronic Office) computer, and followed two years later by the IBM 701 (1953) for scientific applications. These first computers with delay line and electrostatic (Williams Tube) memories of only a few thousand words were priced at $1 million or more ($8.5 million in 2007 dollars) to establish the mainframe class. By 1955, IBM had introduced both scientific (701, 704) and commercial (702, 705) computers that were differentiated by their ability to deal with floating point data of high precision versus the predominately alphanumeric and decimal arithmetic operations typifying data processing. From the graph, the mainframe increased to $4 million and continued to maintain the price range. A set of smaller computers were introduced in the price $0.1- 1 million range e.g. IBM 1401 and 650 for departmental and smaller organization use. These could be classified as subclass of mainframes or super-minicomputers. During the mainframe's formation, eight US and five? European companies competed to establish the class. The US Group was known as Snow White and the Seven Dwarfs or BUNCH (Burroughs, Univac now Unisys, NCR, CDC, Honeywell)+ GE & RCA. With IBM's introduction of System 360 on April 7, 1964, the dominant architecture was established and will doubtlessly remain to run legacy applications "forever" – given the trillions of dollars of software and data that this ecosystem hosts.

Small or minimal computers priced between $60,000 to $120,000 that a person signed up for and used directly for calculation or personal computing at work were introduced beginning in the mid 1950s (Bendix G-15, Librascope LGP-30), as well as the transistorized IBM 1620 that dominated the class. In 1961, the DEC PDP-1 was applied to telegraph line message switching as a prelude to computer networking, peripheral computers for mainframes (like the 1401 or CDC 160), and were used as prototypes for timesharing system.

The PDP-8, introduced in 1965 at a price of $18K is the first "minicomputer." It was minimal, designed as both the smallest computer that could be built and as a component to be used for controlling other devices e.g. process control, lab instruments, terminal concentrators. On occasion it was used as a workstation on a personal basis with an operating system that was a pre-cursor to DOS. The PDP-8 had a dozen implementations following a minimal cost trajectory with single chip versions beginning in 1975 to both define and increase its marketability including its use as a dedicated word processor to the early 1980s. During the minicomputer class formation period, 92 companies formed to establish the minicomputer class with only IBM and HP remaining by 2000 to make computers in this class albeit with substantially changed architectures. Including DEC VAX in this class, the price range increased to cover a range of $10,000 to $1,000,000 servers and covering the entire potential application space of the day. The most expensive VAXen, and VAX clusters competed with IBM smaller System/360 class and sub-mainframes.

The reliable and fast transistor circuitry c1960 enabled a substantially larger number of components to be integrated into a unified system, limited mostly by the maximum feasible selling price, architectural complexity, and interconnection density. Early on, vying for the title of world's fastest computer, were the Manchester Atlas I and the IBM 7030 ("Stretch"), both introduced in 1961. Five years later, the CDC 6600 supercomputer was introduced as the culmination of several years of effort by a small team led by Seymour Cray. It used about 500,000 densely packaged silicon transistors and stunned the world with its performance—easily an order of magnitude faster than any computer shipping at the time or even being contemplated. "Cray sytle" computers based on parallelism functional units, followed by vector processors continued relatively unchallenged for 30 years. In the mid-90s, things had changed somewhat architecturally but bipolar technology still reigned. The fastest machines were shared memory, vector processors using small scale ECL ICs. Successful challengers at Fujitsu and NEC uses the "Cray" formula to build even faster machines with the NEC Earth Simulator holding the title from 2002-2005.

Why Computer Classes Evolve at Constant Price, Increasing Performance

Once a computer class forms, several factors determine the price of the "next" evolutionary model. Building the next model in 3-5 years with chips that have 4 to 6 times more transistors is the natural predicted progression of Moore's Law.

Increases in processing power and memory size are essential for the new data-types such as music, photos, and data-bases. The number of pixels per camera evolve about as rapidly as Moore's Law, requiring more memory and speed to handle the images with constant response. Similarly, disk memories have to evolve rapidly to store the higher resolution photos, higher quality videos, etc.

Nathan's Law, also attributed to Bill Gates, explains software's increasing demand for resources:

1. Software is a gas. It expands to fill the container it is in.
2. Software grows until it becomes limited by Moore's Law.
3. Software growth makes Moore's Law possible through the demand it creates; and
4. Software is only limited by human ambition and expectation.

"Marketing" nominally fueled by user feedback for more functionality, forms the critical link in support of Nathan's Law that minimalist refer to as featuritis, bloat, etc. enabling upgrades to support periodic obsolescence.

We might expect to buy a new computer in three years at 1/4 the price of today's computer using chips that are 1/4 the size of an earlier model perhaps from the same manufacturer. Why not? New microprocessors sell at the same or even a price premium because they have 4x the transistors, faster clock speed and deliver more performance. For example, Intel and AMD are not inclined to build microprocessors with less transistors and lower cost because they don't see such a market – and as such do not participate in establishing the new, lower price classes. Also, a computer is made of other parts e.g. metal and power supplies that may increase in price and act to hold the system price constant with only system manufacturing learning curves left to decrease price.

The "numbers" support a next generation product of constant price and increasing performance, not one of decreasing prices and constant performance. Assume the total cost of ownership is at least 3x the computer's sales price, and for a computer of performance = 4.

performance/total cost = 4/4

Assume a new, constant price, double performance computer performs at 4 x 2 or 8, then

Performance/total cost = 8/4 or 2. Contrast this with a constant performance computer of 4, whose price is just 3/4, giving a total cost of 3.75

Performance/Total cost = 4/3.75 or 1.07

The final and most important incentive to hold price constant and provide more capability is to retain a user's substantial investment in legacy applications and data that have been created together with the implied user and organizational learning. The value of data is most likely to be 10-100 times the hardware cost. A user retains an old computer unless it is unreliable, or there is a substantial increase in functionality – as long as the new model accepts legacy apps and data. The cost to switch to another computer, even with the same capability is so high that the incentive must result in a significant benefit as the above numbers show.

Finally, most goods e.g. cars, construction material, energy, and food not subject to CMOS integration, increase in price with inflation (Table 2). However, computers have defied inflation -- the 1984, 9" monochrome 128 Kbyte, single floppy, integrated $2495, Apple Macintosh costs $1500; in 2007, the same, as a 13" color portable with 1 GB memory and 80 GB disk.

Table 2. Consumer Price Index showing Buying Power since the introduction of computers in 1951 Versus $1 in 2007.

1950	1960	1970	1980	1990	2000	2007
8.5	6.9	5.3	2.5	1.6	1.2	1

Microprocessors 1971: The Technological Force for New Classes in the Second Period

Figure 6 shows the microprocessors derived directly from the growth of transistors/chip beginning in 1971. It shows the trajectory of microprocessors from a 4-bit data path through, 8-, 16-, 32-, and 64-bit data paths and address sizes. The figure shows a second path – the establishment of "minimal" computers that use less than 50 thousand transistors for the processor, leaving the remainder of the chip for memory and other functions e.g. radio, sensors, analog I/O enabling the complete SOC. Increased performance, not shown in the figure, is a third aspect of Moore's Law that allows the "killer micro" formation to subsume all the other, high performance classes that used more slowly evolving bipolar TTL and ECL ICs (Figure 5). The final section will discuss the challenge of having a single chip with billions of computing elements (functional units, processors, computers, wireless links and other I/O).

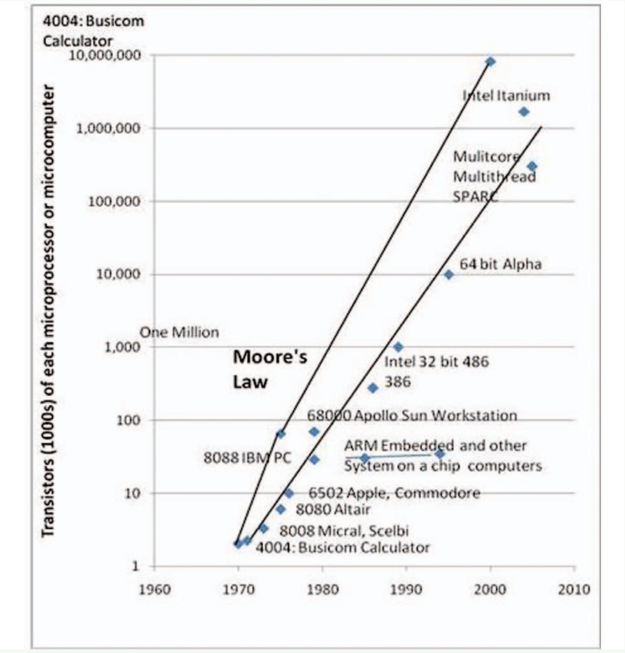

Figure 6. Moore's Law that provides more transistors per chip, has resulted in creating the following computer classes: calcultaors, home computers, personal computers, workstations, "multis" to overtake minicomputers, and clusters using multiple core, multi-threading to ovetake mainframes and supercomputers.

Microprocessor Evolution c1971-1985: Personal Computing (Calculators, Home Computers, Personal Computers, Workstations, and Game Console Platforms)

Calculators, home computers, personal computers, and workstations were established as classes as the processor on a chip evolved to have more transistors with wide data paths and large address spaces as shown in Figure 6.

In 1971, Intel's 4004 with 4 bit data path and ability to address 4KB was developed and programmed to be the Busicom Calculator; instead of developing a special chip as had been customary to implement calculators, a program was written for the 4004 for it to "behave" as or "emulate" a calculator. The 4004 with a 4 bit data path was not suited for storing text and larger numbers other than in a serial fashion, although it was used for numerous applications and to spawn an "embedded computer" market just as the minicomputer had done a decade earlier.

In 1972, Intel introduced the 8008 microprocessor coming from the Datapoint terminal requirement, with 8 bit data path and ability to access 16 KB that allowed R2E's Micral computer (France) and Scelbi to build limited, programmable computers followed by more powerful 8080-based systems that M.I.T.S. used to introduce its "Atltair" personal computer kit in 1975, that incidentally stimulated Gates and Allen to start Microsoft. The more powerful and upward compatible Zilog Z80 was useful in helping to establish a personal computing platform. In 1977, the 16-bit 6502 microprocessor and higher-capacity memory chips enabled personal computers for use in the home or classroom built by Apple, Commodore and Radio Shack—computers that sold in the tens of millions because people bought them to use at home versus corporate buyers. By 1979, the VisiCalc spreadsheet ran on the Apple II establishing it as a "killer app" for personal computers in a work environment. Thus the trajectory went from a 4-bit data path and limited address space to a 16-bit data path with the ability to access 64KB of memory. This also demonstrates the importance of physical address as an architectural limit. In the paper on DEC's VAX (Bell, Strecker 1975), we described the importance of address size on architecture: "There is only one mistake that can be made in a computer design that is difficult to recover from – not providing enough address bits for memory addressing and memory management…" The 8086/8088 of the first IBM PCs had a 20-bit, or 1MB address space, the operating system using the remaining 384KB.

Concurrent with the introduction of the IBM PC, professional workstations were being created that used the Motorola 68000 CPU with its 32-bit data and address paths (4GB of maximum possible memory). Apple Computer used the Motorola "68K" in its Lisa and Macintosh machines. IBM's decision to use the Intel architecture with limited addressing, undoubtedly had the effect of impeding the personal computer by a decade as the industry waited for Intel to evolve architecture to support a larger address and virtual memory space. Hundreds of companies started up to build Personal Computers ("PC-clones") based on the IBM PC reference design c1981. Dozens of companies also started to build workstations based on a 68K CPU running the UNIX operating system. This was the era of "JAWS" (Just Another WorkStation) to describe efforts at Apollo, HP, IBM, SGI, SUN and others based on 32-bit versus 16-bit microprocessors and including specialized systems for Word Processing (Wang, Xerox), Market Analysis (Metaphor), CAD (Intergraph, Daisy, Valid), and high-level programming (Lisp Machines and Symbolics). Virtually all of these "workstations" were eliminated by simple economics as the Personal Computer--based on massive economies of scale and commoditization of both the operating system and all constituent hardware elements) evolved to have sufficient power and pixels.

"Minimal" CMOS Microsystems on a Chip c1990 Establish New Classes Using Smaller, Less Expensive, Chips

In 2007, many systems are composed of microprocessor components or "cores" with less than 50,000 transistors per microprocessor core at a time when the leading edge microprocessors chips have a billion or more transistors cf Figure 6. Such cores using lower cost, less than the state-of-the-art chips and highly-effective, rapid design tools allow new, minimal classes to emerge allow new, minimal classes to form. PDAs, cameras, cell phones, and personal audio & video devices have all been established using this "minimal" computer design style based on small "cores". In 1990, the Advanced RISC Machine (ARM) formed from a collaboration between Acorn and Apple as the basis for embedded systems that are used as computing platforms and achieve two billion units per year in 2006. Other higher volume microsystem platforms using 4-, 8-…64-bit architectures including MIPS exist as core architectures for building such systems as part of the very large "embedded" market.

Rapidly Evolving "Killer CMOS Micros" c1985 Overtake Bipolar ICs to Eliminate Established Classes

In the early 1980s, the phrase "killer micro" was introduced by the technical computing community as they saw how the more rapidly evolving CMOS micro would overtake bipolar based minis, mainframes, and supers if they could be harnessed to operate as a single system and operate on a single program or workload.

In the Innovator's Dilemma, Christensen describes the death aspect basis of Bell's Law by contrasting two kinds of technologies. *Sustaining* technology provides increasing performance enabling improved products at the same price as previous models using slowly evolving technology; disruptive, rapidly evolving technology provides lower priced, products that are non-competitive with higher priced sustaining class to create a unique market space. Over time, the performance of lesser performing, faster evolving products eventually overtake the established, slowly evolving classes served by sustaining technology.

From the mid 1980s till 2000, over 40 companies were established and wiped out attempting to exploit the rapidly evolving CMOS microprocessors by interconnecting them in various ways. Only Cray, HP, IBM, SGI and SUN remain in 2007 to exploit massive parallelism through running a single program on a large number of computing nodes.

Let's look at two potentially disruptive technologies, establishing new classes:

The OLPC (One Laptop Per Child) project of Nicholas Negroponte aimed at a $100 PC (costing about $188 in 2007) is quite likely disruptive as a "minimal" PC platform that relies on the internet for storage of programs and data. Cost reduction is achieved by substituting 500 MB of flash memory for disk, reduced screen size, small main memory, and built in mesh networking to reduce infrastructure cost. An expected selling price of $200 with a $188 cost that is about half the price of the least expensive PCs in 2007, is characteristic of a new sub-class. OLPC will be an interesting development since Microsoft's Vista requires almost an order of magnitude more system resources.

The evolving small form factor devices such as cell phones are likely to have the greatest impact on personal computing, effectively creating a class. For perhaps most of the 4 billion non-PC users, it becomes their personal computer and communicator, wallet... map, etc. since the most common and often only use is of personal computers is for email and web browsing – both stateless applications.

Application of Bell's Law–Planning VAX and the VAX Strategy

In 1975 when VAX was in the planning stage, I used the theory of classes to posit a compatible line of computers that had the same instruction set and programming environment that could be used in a range of uses including personal computers, process control, departmental timesharing, and clusters for large scale apps. The planning was based on the different sized memories resulting in different prices according to the following pricing model:

System Price = 5 x 3 x .04 x memory size/ 1.26 (t-1972) K$

Where 5x: Memory is 20% of cost; 3x: DEC markup; .04x: $ per byte; 26%: price change

Figure 7 shows the prices for systems of various sized memories. The large price declines were in fact one of the root causes of the demise of Digital in the late 90s. In effect, the large memories required to maintain pricing in a price band required larger amounts of processing that were served by clusters of microprocessor based computers connected as clusters. Another cause at DEC was continuing with ECL at a time when CMOS overtook it in speed and especially exorbitant cost when nearly zero cost, microprocessors were outperforming ECL.

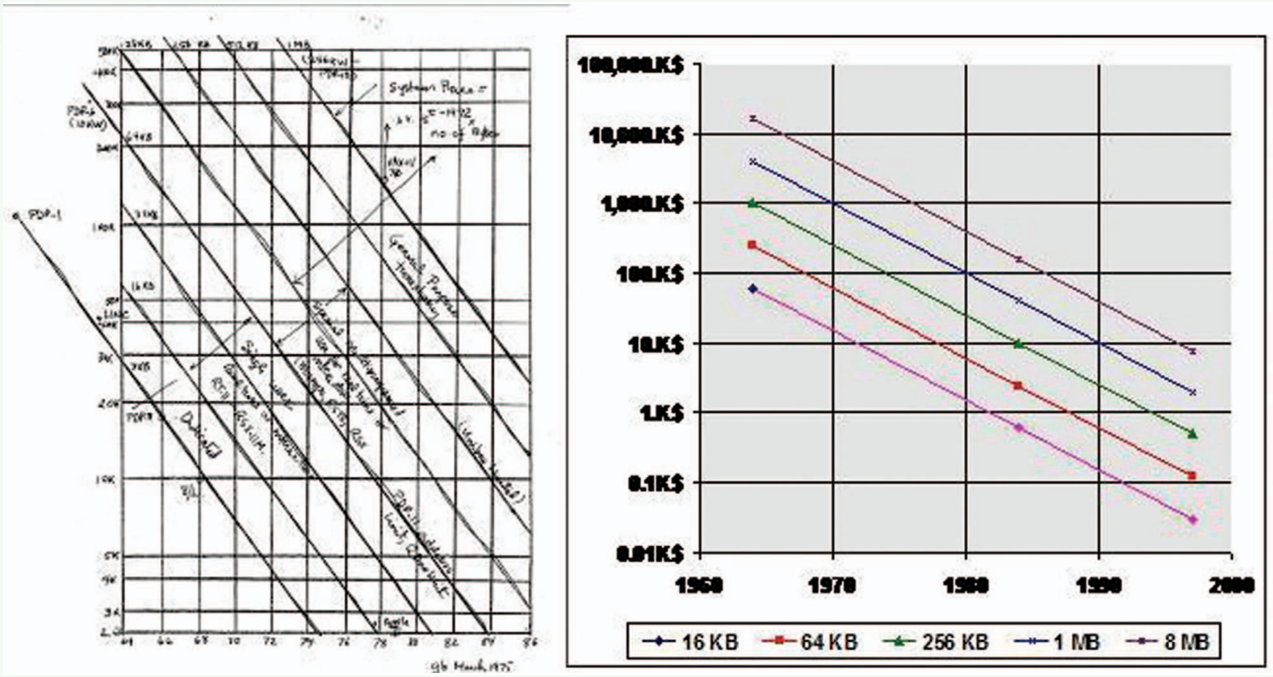

Figure 7. Original VAX Planing model Computer Prices versus time from 1975 showing different memory sizes and resulting prices 1964-1986. In 1998, the model was reviewed retrospecively. The price changes, though accurate, were so rapid to be unbelievable and hardly actionable.

The Challenge of Constant Price, 10-100 Billion Transistors per Chip, Multi-threaded, Multi-processors, for General Purpose Computing

The future is not at all clear how such large, leading edge chips will be used in general purpose computers as used at the desk top. As ever, the resilient and creative supercomputing and large scale service center communities will exploit the largest multi-core, multi-threaded chips. There seems to be no upper bound these systems can utilize!

However, without high volume manufacturing, the virtuous cycle is stopped – in order to get the cost and benefit for clusters, a high volume personal computer market must drive demand to reduce cost. In 2007, the degree of parallelism for personal computing in non-gamer desktop systems such as Linux and Vista is nil either reflecting the impossibility of the task or our lack of creativity.

Several approaches for very large transistor count i.e. 10 billion transistor chips with more than a few (e.g. 2-10) processors could be, in order of difficulty:

1. Small chips with only as many processors that can be gainfully employed e.g. 2-4 processors system with primary memory on a chip for substantially reduced lower priced systems and greater demands that either require or imply proportionally lower cost software
2. Graphics processing, currently handled by specialized chips is perhaps the only well-defined application that is clearly able to exploit or absorb unlimited parallelism in a scalable fashion for the most expensive PCs e.g. gaming, graphical design. In effect, this just cost reduces the system by eliminating graphics chips.
3. Dedicated functional processing for networking, improved user interface including speech processing for text to speech and spoken commands
4. Multi-core and multi-threaded processor evolution for large, high performance scientific systems that are carefully programmed using FORTRAN-MPI, as FORTRAN turns 50.Remodel the desktop architectures at the language level to be able to highly parallelize apps using the vectorization and parallelization that has proven applicability in the multi-vector processor machines, betting on the need
5. Develop image processing enabling "computers to see" and be controlled by motion and emotion using hands and face. The Nintendo Wii seems to have something here.
6. A BKA or "BIG KILLER APP" that exploits these structures, EVERYONE needs, and compatible with our PC environment.
7. Something BIG, based on a dramatic new way to program e.g. Transactional Memories, Functional Programming, block structured dataflow requiring changes in language, tools, training, and new applications. Systems are being introduced such as Microsoft's F# to test this approach, and if successful imply a change akin to the introduction of objects. Software objects, requiring new application architectures may be alternative way of thinking versus the FORTRAN-MPI model.
8. Abandoning general purposeness using FPGAs that are programmed using inherently parallel hardware design languages like parallel C or Verilog that could provide universality that we have never before seen, and

Independent of how the chips are programmed, the biggest question is whether the high volume personal computer market can exploit anything other than the first three paths, and even those require careful programming beyond 2007 operating systems.

Let's apply the Carver Mead 11 year rule – the time from discovery and demonstration till use. Perhaps the introduction of a few transactional memory systems have started the clock using a programming methodology that claims to be more easily understood. A simpler methodology that can yield reliable designs by more programmers is essential in order to utilize these multiprocessor chips.

In a way, the opportunity or rather need for parallelism is reminiscent of the 1982 Japanese Fifth Generation research effort based on parallelization, AI, and PROLOG. (The Denelcor HEP was also installed then.) This time, it's not research. The problem needs a tractable solution. Without it, Moore's Law slows.

Will Small Form Factor Devices Impact Personal Computing?

Users are likely to switch classes when the performance and functionality of a lesser priced class is able to satisfy their needs and still increase functionality. Since the majority of PC use is for communication and web access, evolving a small form factor device as a single communicator for voice, email, and web access is quite natural. Two things will happen to accelerate the development of the class: people who have never used or are without PCs will use the smaller, simpler devices and avoid the PC's complexity; and existing PC users will adopt them for simplicity, mobility, and functionality e.g. wallet for cash, GPS, single device. We clearly see these small personal devices with annual volumes of several hundred million units becoming the single universal device evolving from the phone, PDA, camera, personal audio/video device, web browser, GPS and map, wallet, personal identification, and surrogate memory.

With every TV, becoming a computer display, a coupled SFF becomes the personal computer for the remaining applications requiring large screens. Cable companies will also provide access via this channel as TV is delivered digitally.

Ubiquitous Wireless. WiFi, Cellular Services, and Wireless Sensor Nets

Unwiring the connection around the computer and peripherals, TV set, etc. by high speed radio links is useful but the app is unwiring, and not platform creation. Near Field Communication (NFC) using RF or magnetic coupling offers a new interface that can be used to communicate a person's identity that could form a new class for wallets and identity. However, most likely the communication channel and biometric technology taken together just increase the functionality of small devices.

Wireless Sensor Nets: New Platform, Network, and Applications

Ubiquity: combining the platform, *wireless* network and interface into one to integrate with other systems by sensing and effecting is clearly a new class that has been forming since 2002 with a number of new companies that are offering – "un wiring", and hence reduced cost for existing apps e.g. process, building, home automation and control. Standards surrounding the 802.15.4 link that competes in the existing unlicensed RF bands with 802.11xyz, Bluetooth, and phone are being established.

New applications will be needed for wireless sensor nets to become a true class versus just unwiring the world. If, for example, these chips become part of everything that needs to communicate in the whole IT hierarchy, a class will be established. They carry out three functions when part of a fixed environment or a moving object: sense/effect; recording of the state of a person or object (things such as scales, appliances, switches, thermometers and thermostats) including its location and physical characteristics; and communication to the WiFi or other special infrastructure network for reporting. RFID is part of this potentially very large class of trillions. Just as "billions of clients needed millions of servers" a trillion dust wireless sensing devices will be coupled to a billion other computers.

Summary

Bell's Law explains the history of the computing industry based on the properties of computer classes and their determinants. The paper posits a general theory for the creation, evolution, and death of various priced-based computer classes that have come about through circuit and semiconductor technology evolution from 1951. The exponential transistor density increases forecast by Moore's Law (1965,1975) being the principle basis for the rise, dominance, and death of computer classes after the 1971 microprocessor introduction. Classes evolve along three paths: constant price and increasing performance of an established class; supercomputers – a race to build the largest computer of the day; and novel, lower priced "minimal computers". A class can be subsumed by a more rapidly evolving, powerful, less expensive class given an interface and functionality. In 2010, the powerful microprocessor will be the basis for nearly all classes from personal computers and servers costing a few thousand dollars to scalable servers costing a few hundred million dollars. Coming rapidly are billions of cell phones for personal computing and the tens of billions of wireless sensor nets to unwire and interconnect everything. In 1951, a man could walk inside a computer and by 2010 a computer cluster with millions of processors has expanded to building size. More importantly, computers are beginning to "walk" inside of us[2].

Acknowledgements

Jim Gemmell, Jim Gray , Robert Hasbun, and Chuck Thacker of *Microsoft*; John Gustafson, Craig Mudge, Dag Spicer, *Computer History Museum*, and Doron Swade, *British Science Museum*.

[2] Courtesy of Dag Spicer, Curator, Computer History Museum

Bibliography

[1] Bell, C.G., A. Newell, "Computer Structures: Readings and Examples" McGraw-Hill, 1971.

[2] Bell, C. G., R. Chen and S. Rege, "The Effect of Technology on Near Term Computer Structures," Computer 2 (5) 29-38 (March/April 1972).

[3] Bell, C. G. On the Future of Computers. http://research.microsoft.com/~gbell/Mit288.asx A 1972 talk at M.I.T. describing a model for future computers, including computer classes based on logic technology evolution.

[4] Bell, C. G., "The Mini and Micro Industries", Computer (17) no. 10, pp. 14-30 (October 1984).

[5] Bell, C. G., "Multis: A New Class of Multiprocessor Computers", Science, Vol. 228, pp. 462-467 (April 26, 1985).

[6] Bell, G., W. Strecker, "Computer Structures: What Have We Learned from the PDP-11" IEEE Computer Conference Proceedings, Florida (November 1975).

[7] J. Gray, P. Shenoy, "Rules of Thumb in Data Engineering," Proc ICDE200, San Diego, March 1-4, 2000. IEEE press.

[8] Christensen, C.M. The Innovator's Dilemma, Harvard Business School Press, 1997, pp 225.

[9] Moore, Gordon E. Cramming more components onto integrated circuits, Electronics, Volume 8, No. 39 April 19, 1965. Article predicted transistor doubling annually; revised 1975: 18 months doubling.

[10]Nelson, D. L., C. G. Bell "The Evolution of Workstations", IEEE Circuits and Devices Magazine, July 1986, pp 12-15.

About the Author

Gordon Bell has been a principal researcher at Microsoft Research, since 1995, researching the lifetime capture and storage of everything for an individual. Previous roles include VP of R&D, Digital Equipment Corp. (1960-1983); professor, Carnegie-Mellon University (1966-72); founding assistant director of NSF's Computing and Information Sciences and Engineering (CISE) Directorate (1986-1988); chairman, cross agency committee (FCCSET) for creating the Internet(1987-1988); advisor /investor in 100+ start-up companies; and a founding trustee of the Computer History Museum, Mountain View, CA.

Since 1987 he has sponsored the Association for Computing Machinery's (ACM) Gordon Bell Prizes for parallelism awarded annually at Supercomputing. He has bachelor and master of science degrees from MIT (1956-57), is a University of New South Wales Fulbright Scholar (1957-58), has an honorary doctorate in Engineering from Worchester Polytechnic Institute (WPI) (1993), and is a fellow of the American Academy of Arts and Sciences (AMACAD), Assoc. for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE), and the National Academies of Engineering (NAE) and Science (NAS). Awards include: ACM-IEEE Eckert-Mauchly Award; the IEEE's Computer Pioneer and McDowell Awards; the IEEE's Von Neumann Medal; the Computer History Museum Fellow Awards; the American Electronics Association (AEA) Inventor Award for the economic contribution to New England; the IEEE 2001 Karapetoff Eminent Member's Award of Eta Kappa Nu; and the 1991 National Medal of Technology "for his continuing intellectual and industrial achievements in the field of computer design and for his leading role in establishing…computers that serve as a significant tool for engineering, science, and industry."