Moore's Law is dead, long live the CapCasc!

CapCasc.org

In the last 10 years the IT Industry struggled to keep pace with Moore's - first - Law predictions. And the gap is increasing exponentially. Unfortunately, instead, investments required to develop new ICs are still growing exactly in accordance with Moore's second Law! During the years such a scenario diverted most of the overall IT Industry Resources to IC development. As a result funds for R&D on Computer Systems dropped. All that, made the innovation in the IC Sector the only driving force in the IT market. Now the scene has changed. We need a new propelling technology to foster innovation. And we must start from the area we neglected so far. For instance, from the Computer System perspective. In fact, we have dozens of new - or old but not yet properly implemented - Methodologies and Architectures, we could exploit to overcome today's main pitfalls:

- Limited and Inefficient support to real Parallel Computing - for all kind of Users, disrupting any chance of effective and efficient Scalability and slowing down the development and adoption of really innovative - but resource demanding - softwares
- Incomplete Encapsulation for Software Objects, voiding any effort toward a wider practice of real Software Reuse and Interoperability, and most of all rendering Parallelization of existing Algorithms an impossible task

The CapCasc Technology, - implementing and extending UK patent application GB20060012776 - is the right answer for both issues. Let's see its unprecedented wealth of advantages for End Users and the Industry, and how it works!

An Unprecedented Wealth of Advantages

For End Users

- Processing Power at will
- Plug-n-Play Applications
- No more HW and SW incompatibilities
- Instantaneous Upgrade
- Lowest costs due to Open Source
- Superior Browsing and Data Security
- Unprecedented AntiVirus Capabilities
- Multiple OSs at the same time

For Businesses

- Processing Power at will
- Instantaneous Deployment
- FailSafe SwitchOver
- 100% Replicable Environments
- HW Data and Comms Containment
- Open Standards simplify Procurement
- Superior AntiIntrusion Capabilities
- Granular Power Management

For Developers

- Truly Encapsulated Objects
- Control of System Level Resources
- Freedom of choice between OSs
- Open Source
- Solid Distribuited Computing Support
- Function Specific OS Tailoring
- Customer Platform Independence

For the Software Industry

- A completely new Market
- No royalties to pay
- 100% OS indipendence
- Completely new Sales Channels
- Real World flawless Interoperability
- Hardware Licensing Control

For the Hardware Industry

- A completely new Market
- A vehicle to access Proprietary Platforms
- Open Hardware
- Completely new Sales Channels
- Real World flawless Interoperability

For Silicon Vendors

- Volumes multiplied by 10 and 100+
- Real Interoperability
- Direct reach to the end user
- Mass market for Dedicated Processors
- Mass market for Reconfigurable Logic

discover how at capcasc.org/advantages

CapCasc Architecture

CapCasc - but also CupCask or BallCask - are the names of a new kind of computer. They are composed as follows:

- **CAP** is the name given to a "ball", and it stands for Completely Autonomous Processor. They can also be called CUPs (Contained Universal Processor). Or just simply Balls
- **CASC** is the name of the remaining parts, and it stands for Communications and Support Carrier. But it can also be called CASK (Communications and Support Keg)

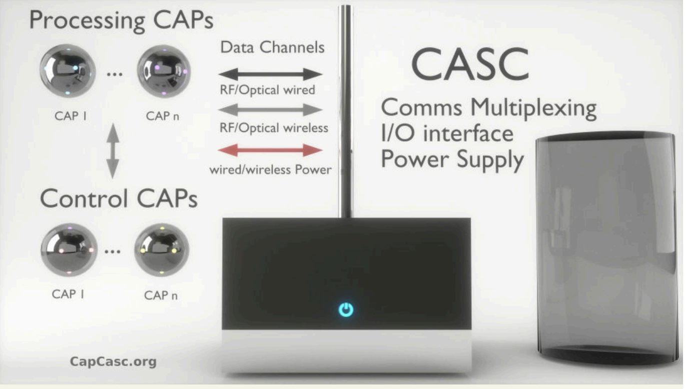
In a summary the CapCasc Recipe for a truly Parallel Processing System follows a few basic criteria:

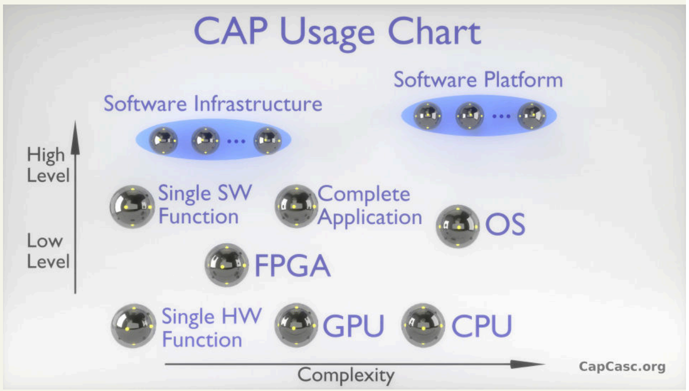
- **True Encapsulation** - any given Software Function or set of Functions - implemented in a single CAP - must encapsulate their own Hardware and Base Software (Firmware, OS, etc.) implementation - as an extension (or completion) of the Encapsulation concept used in Object Oriented Programming Languages. Orthogonally, any given set of Hardware Functions can be also implemented within a CAP, but it also has to bring along all its supporting Software
- **Universal Plug-n-Play** - the aforementioned True Encapsulation permits the extension of the Plug-n-Play model to all HW and SW objects. In other words, any IT product, either Hardware or Software can be now completely implemented within a CAP - or series of CAPs - and subsequently installed instantaneously and without any adaptation work inside any CapCasc System
- **Parallelization by Replication** - True Encapsulation is also the tool of choice, in a CapCasc System, to facilitate Parallelization of existing Algorithms. In fact it makes it extremely easy to replicate any set of Tasks or Functions within any number of CAPs and then distribute the load among them with the usual balancing methods
- **Efficiency by Specialization** - finally, True Encapsulation - enabling a straightforward integration of Specialized Hardware and Software within existing Platforms - supports extremely efficient System Designs. For Example, an Application can run processing intensive parts of the program, in a Specialized or Reconfigurable Processor, or, a networking SW can work within a Real Time and Security Hardened OS, and both can still operate along with the rest of the system which in turn can use a mainstream OS version and General Purpose CPUs
- **Unlimited Processing Power Scalability** - The amount of CAPs allowed in a System is only limited by the available space and Power resources
- **Unlimited Inter-Processor and I/O Bandwidth Scalability** - The amount of inter-Processor and I/O Data Channels allowed in a System is only limited by the information density allowed within the RF and Optical Spectra by the technology available at any given time. Forward and backward compatibility is enforced
- **Virtually Unlimited Physical Scalability** - The system is able to scale physically to accommodate increasing numbers of Processors and Data Channels

other ancillary Architectural and Implementation concepts are:

- CAPs can be used alternatively for Generic Processing purposes - and they are named Processing CAPs, or to control the system itself - and they are named Control CAPs
- Control CAPs, implement and control all system level tasks, and arbitrate the access of Processing CAPs to all I/O and Data Communication resources
- All CAPs have one or more wireless data communication channels
- Some CAPs can also have one or more wired data communication channels
- Both Wireless and Wired data channels use RF and Optical signals, which are routed and multiplexed by the CASC Multiplexing Infrastructure
- The CASC Multiplexing Infrastructure is also used to route all Available Analog and Digital Video Signals of both Processing and control CAPs, so that they can be shared and merged using Overlay and Compositing Techniques.
- All CAPs can have an either wired or wireless power supply - provided by the CASC
- Control CAPs can selectively switch on and off Processing CAPs
- All Processing CAPs can both bootstrap and have their firmware updated remotely - from Data Sources managed by the Control CAPs
- Some Processing CAPs - namely those implementing Proprietary Functions - can implement only a subset of the required features, but still be able to operate within a CapCasc

CapCasc.org





Open Software and Hardware Policies

The CapCasc Architecture is at present under further development and all details are being kept proprietary to allow for a closer control of all the final features. But when the first production unit is ready, all aspects of the Design - Hardware and Software - will be made available under the CC BY-SA or GNU GPL license

more insights at capcasc.org/howitworks

Current and upcoming CapCasc Specifications

The current CapCasc implementation include the following features:

- Processing and Control CAPs, with either **ARM Cortex A8 or A9 Processors** and **1, 2 and 4 cores** - based on off-the-shelf SOMs
- **0.5Gb/s** Wireless Optical Data Communication Channels for **High-Speed USB** and **Fast-Ethernet** Payloads - implemented with original circuits
- An original **Modular RF and Optical Data Communication Multiplexing Infrastructure**, supporting 4 CAPs per module - all implemented with custom circuitry, except for the Ethernet and USB switching Hubs which are based on common off-the-shelf modules
- All Wired Connections - optionally used by a CAP - are consolidated within a single non standard micro-HDMI cable. A reconfigurable number of USB, Ethernet, Serial, Power and Control connections are supported
- **CapCasc OS** - a tailored version of Linux, derived from the Debian Distribution
- **Single Server Image** and **Simulated SMP** capabilities - implemented integrating and extending openSSI and Kerrighed
- Compatibility with all the File System Clustering Platforms available in Linux
- Video Sharing and Compositing through XServer and FreeRDP

upcoming versions will comprise:

- CAPs based on off-the-shelf Intel Atom and/or AMD G-Series SOMs
- Integration with Apache Hadoop
- Open-Hardware Processor Modules based on both Atom and ARM CPUs
- Hardware Multiplexing of Analog (VGA) and HDMI video signals
- CAPs running Windows
- Optical Wireless Gigabit Ethernet