Structol Conscience Computer

WIP: This article is a work in progress and is not yet approved for usage in the RP.

A structol conscience computer is a computer which uses growing extra-dimensional nano-scale circuitry, faster than light processing, and quantum-potential memory on the planck scale to reach its conclusions.

Unusually, this circuit physically grows and changes, networking and organising itself to form dedicated circuitry for low-precision high-parallelization long-term tasks and generalized circuitry for high-accuracy low-parallelization short-term tasks.

This product is produced by the Lazarus Consortium for private use as of YE34.

Statistics

Government: N/A **Organization:** Lazarus Consortium **Type:** Apex multifunction information processing system **Class: Manufacturers:** Lazarus Consortium

About

Based on historical records of the Sourcians, the Structol Computer is the offshoot of an attempt to fabricate a ' Hub Maesus' or 'Source' – the primary control maesus of the Sourcian civilization, which saught singularity. The two share many topological similarities, despite their massive differences in scale and in terms of composition and functionality are nearly identical.

Physical Description

Structol computers are usually similar to Sourcian Maesus when allowed to work independently among other structol systems as part of a unit or weapon. They are smooth, hard and marble-like along their translucent exterior casings. These casings are usually enscribed with the manufacturer, creation date, original issue, dots dictating where an engineer should cut in order to remove and transplant components and any scars from prior 'surgeries', meaning it is very hard to hide the fact a system has been altered.

Inside, smaller marble-like forms cluster beneath the surface which are interconnected with a translucent liquid-like fiberous material.

Within these smaller 'marbles', a structol computer is plate-like, extending above and below along the interior walls of whichever surface it occupies (usually smooth and round) in order to ease transplantation which can be performed with specialist software instructions or a molecular applicator -

by creating an incision over the outermost layer, 'digging' to reveal the specific sub-sphere and then performing surgery on it to remove the specific 'plate' (which is usually about the size of a microprocessor's housing).

When 'digging', the other 'marbles' or clusters (and their sub-clusters inside) which are not to be transplanted are unharmed and usually still operational, even if removed from the housing, provided they are still tethered - with the connective fiber being able to stretch and contract for this very purpose.

A structol computer makes a soft swirling sound when held to the ear, similar to a conch shell. Seen up close, the physical surface of the interior resembles a tiny layered cityscape sprawling and moving with tiny matte, reflective and luminous surfaces inside it forming strange patterns similar to optical illusions.

Generalization & Centralization vs Degeneralization and decentralization

In any computer of any kind, there are two ways of going about a problem: Either locking the 'logical gates' for complex calculations and from there, only varying the input gates (to produce a meaningful result) or keeping the gates open, so they can be altered very very quickly – though the calculations take more time to perform since extra steps and abstraction are introduced.

In this way, while a generalized CPU can perform a wider variety of tasks (example, encoding video or rendering footage), a non-generalized 'DPU' (dedicated/decentralized processing unit) can perform them at a far greater speed - though with less options.

Tasks a non-generalized 'decentralized' (DPU) processor excels at:

Massively parallel simulation

The act of thinking several steps ahead of your opponent and working out moves in advance.

- Advanced physical simulation
- Information warfare simulations
- Electronic warfare simulations
- Physicalized simulation
- War simulation
- Estimating means of exploiting repetition

Cross-referencing tasks

The act of recalling and referencing specific information once an initial series of patterns have been

recognized, no matter how obscure - including similar patterns through different layers of abstraction.

- Psycho/historical referencing
- Target identification
- Research and recall tasks
- Object, situation & semantic recall

Genetic routine

The act of evolving the solution to a problem by rapidly performing simulation in order to come up with a best option, applying more precision as the worst answers are shed.

Pattern recognition

The act of identifying a sequence of events which may be repeated, recorded, recalled or exploited for further use.

- Object, situation & semantic recognition
- Recognition of repetition
- Identifying tactics & scenario information

Hardware Emulation

The act of creating the physical components of another system in hardware instead of software - essentially creating a computer inside a computer.

Hardware emulated systems are usually seamlessly blended with whatever motherboard or higher machine created them and as such can pose a possible security threat if hardware is introduced which contains malicious instructions. Competitive routine will, however, eventually remove the malicious instructions if it interferes with higher operations simply by devaluing it and repurposing its components.

Referrential code execution

The act of running a program which takes place as a layered system rather than a sequenced system of steps - with an input layer, hidden layer and output layer – with many different exeuciton styles. Each node in a given network holds a very simple piece of information which only becomes meaningful as it is routed through other paths on the network.

Important is that adaptive resonance is at work, with different elements of the network competing with one another. As such, when a component of the network is damaged, the network will usually infer that the missing component of the instruction could be a multitude of different things and assigns non-assigned links in order to fill in the various possibilities the missing piece could be. Eventually, a 'best match' is found and as such, the code repairs itself.

Tasks a generalized 'centralized' processor (CPU) excels at:

Sequential code execution

The act of running a program which takes place in sequenced steps as part of an ongoing series of steps in order to reach a conclusion. Typically comprised of memory operations, loops, calculations and redirections.

When a sequenced program has a component missing, when that component would be executed, the CPU Grinds to a halt and requires higher assistance in order to skip over the instruction - though meaningful information may no longer return from the system and the program may be irrepairably damaged.

High precision calculation

High-accuracy calculations which are ideal for actions such as abstracted physical simulation and solving for situations like location recognition, ballistic calculations, trajectory heading calculations, adjustments of core systems and the design of physical lower-abstraction systems.

Software Emulation

The act of creating the physical components of another system in software instead of in order to run its software and interact with it.

An abstracted system is created which is entirely disposable, ideal for operation which cannot be allowed to spread into the memory of a higher system (such as digital attack) or volatile/classified information.

Hardware

Composition

The Structol computer is composed of structol, though it makes extensive use of many processing and storage methods, including Sourcian genetic material (XNA), magnetic systems and faster-than-light optical processing methods through the use of negative refractive index quasicrystalline components and subspace acceleration which allow for massively parallel quantum computing operations and sub-atomic storage systems through the use of forking quantum potentials. Exactly which is used, is decided on the operation at hand.

Many circuits are abstracted in subspace and are (when operating) five dimensional constructs, allowing

for not only sidestepping of certain conventional physical laws (similar to FTL systems) but also a superior information processing density.

Architecture

A structol computer has no specific components with the exception of programs or emulated hardware which distinguish themselves.

Geometrically speaking, interprogrammic linkages are usually done through lower or 'deeper' layers, with abstraction represented in terms of depth. In this sense, the structol computer is a three dimensional system - the plates of which can be instructed to join in any number of complex geometric three dimensional shapes.

In this sense, it is supremely similar to a Sourcian maesus, which takes a series of spherical arrangements which are protected by a dense outer shell. Many combat applications of the Structol computer replicate this layout.

Maintanance

The structol elements of the system are self-organizing, self-growing, perpectuating and self-optimizing - in a state of constant self-upgrade. With usage and experience, the computer gradually grows faster - though it does require physical substinance in the form of water and carbon every few months.

Behavior

When hives or lobes are fractured from physical abuse, the will reference memory-addresses to physically relocate and restore connections. In short, provided they are adequately powered, they require zero physical maintenance - healing like broken bones. The restored form may be erratic or different but functionally it will be almost exactly the same, often incorporating redundancies based on the error to prevent it from reoccurring.

Logical model

The binary system of ones and zeroes is abandoned in the lowest levels of abstractions in favor of an infinite varience model - replacing 'black and white' of common digital logic with a sliding scale of infinite color or value in both positive and negative values. In this way. In conjunction with neural layouts, this allows the system to transcend many different classic logical systems and methods while still employing their usage and their constraints semi-natively in dedicated centralized processors - but only on demand.

In this way, the computer can interface natively with quantum systems with no translational layer and an unlimited number of decision contrast bits.

Networking

Networking with a structol computer takes on two major classifications: conventional networking (message exchange) and clustering (systems unification).

Conventional Networking

Conventional networking can be performed via any communications and sensor systems the computer is able to gain access to or abstract through the combined use of other systems mounted to the platform (for example, encoding information in a plasma wake or sending messages via dynamic canoflage or sound).

Clustered networking

Clustered networking however, is somewhat more complex. While the same means may be used, the systems working together act as a single non-codependent (due to their architecture) computer, sharing variables and unifying their decentralized processing cycles toward a common goal.

In this way, the idiosyncrancites between individual structol computers usually translates into a wider variety of 'more interesting' conclusions when attempting to discover a novel solution to a known problem and massive efficiency gains when searching and identifying a problem in the first place to make it known.

Higher Geometric Processing Clusters (HGPC)

In addition, the subspace 'gaps' between higher-dimensional ROM circuitry (when active) can be united seamlessly over space - with the distance between the components essentially 'becoming more computer', increasing their raw processing power but unfortunately, also the latency of the connected programs.

In this way, a calculation which would take miliseconds in three dimensions can take minutes in five dimensions but paradoxically a calculation which would take thousands of years in three dimensional space can be solved in miliseconds.

For this reason, computing power is generally approximately squared in groups (in the sweet-spot between latency and performance, where it remains usable), rather than merely summed, as it is in most advanced computational systems.

Transfer of programs

Usually when a program is to be transferred, there are multiple ways of doing so.

Transmission

The first is a slow method - the transmission of the abstracted actions the computer is to perform, which it then develops its own circuit to perform.

Any programs created in this way usually lack additional networking and cross-referencing needed in order to further enhance its core program.

Transplantation

Alternatively, the entire circuit of another unit, including its cross-referencing, by copying the specific referenced elements. Unfortunately the information density is usually massive (taking several weeks to copy a program).

As a result, its usually better to transmit the program via transplantation: the act of physically removing the program and then stitching it into the other computer in much the same way an organ transplantation takes place.

Unfortunately, transplanted programs retain customization and preferences set by the former owner or user and also remember their former owership and permissions data. Combined, this data usually massively improves the core performance of the program itself - since many shortcuts are made and extranious components are deprioritized.

In this way, any transplanted component needs to be 'sanitized', a process which can take up to 16 hours if performed by a structol computer. Alternatively, a technician can sanitize the prior data, though usually some of the references (and hence, gained proficiency via experience) are lost.

Alternatively, the new information can be blended without sanitization though this can result in instabilities.

Security

It is physically impossible to insert code or fool the unit into performing unwanted instructions without physical tampering. However, the computers it may emulate in its sub-systems are not protected in this manner. A monitoring system is assigned to any computer with a remote connection, searching for changes in code that have not been made by base systems or the user themselves, preventing injection and exploit style hacks.

Imposed limitation and governence

Each sub-cluster is granted a rating and an address which doubles as a unique identifier. The address acts as a means of identifying specific components, what they were linked to and where they came from - important in digital forensics.

The rating on the other hand acts as a limiter for the intelligence, awareness and information/processing density efficiency of a system – preventing it from ever becoming 'too smart'.

Should a cluster attempt to bypass its turing limitations to dominate the system, it is deprioritized and denied resources. In this way, any malicious attack is unable to physically spread into the executive functions of a structol computer.

Universal Misuse and Anomaly Detection suite (UMADS)

UMADS is essentially a dedicated sandbox and experiment suite which allows an attacker to believe they have succeeded in infiltrating a false computer which is not linked to the primary or secondary computer by conventional means.

Unlike a honeypot, UMADS can pretend to be a massive variety of different system-types. Its unusual linkage (comprehension, not just awareness, but still no execution rights) allows UMADS to assess the attacker for possible openings and advanced warning for any information warfare systems running on the computer (allowing them to form their own informed attack responses, as a single networked computer system).

UMAD can pretend to be many different system classifications and independent systems in order to study how an attacker responds with various defenses and other systems.

Intruder Countermeasure Electronics (ICE)

Honey Pots

False computers or 'honey pots' are created as bait for the enemy to penetrated. This is usually used to study, dox and identify an attacker in order to formulate a return strike while the enemy believes they have successfully penetrated their target.

In this way, a Structol Computer can be 'aware' of what happens in a honey pot without comprehending it on an execution level, ensuring the offending code doesn't propegate beyond the hoenypot.

Rabbit Hole

Advanced attackers such as AI may be able to identify that they have entered a honeypot and will attempt to ascend into a higher level of the computer - known as ascension. A structol computer can then nest the computer they were in, inside another (and another and another) in a sliding trickery of mirrors by using two refferential systems. This defense is known as a 'Rabbit Hole' and is exceptionally difficult to escape.

Man in the Middle

If attackers are in traditionally overwhelming numbers, the attack signals can be doctored and redirected (again, protection through abstraction: understanding without executive comprehension) back at the attackers. This process can be repeated.

Example scenario: In an 8-vs-1 digital attack scenario, the Structol Computer is able to redirect the attacks of each attacker back at their following partner without their awareness.

In this way, they can play their opponents moves against them in sequence - beating multiple opponents while only ever needing to 'play' more than a single attacker - by forcing the attackers to deal with eachother without knowing.

This unfortunately depends on the preconcieved notion that the attackers are (for whatever reason) either too stupid (as most contemporary systems are) to recognize the attack type as one performed by a friendly unit recently (as the bandwidth and processing overheads of this tracking model would rapidly exceed those of the processing power used for attacking in the first place) or are unable to communicate with one another via some method of inter-linkage jamming.

Payloads

If the system can gain access to an attacker, its primary intention will be to neutralize an attacking threat – that is, to stop it from attacking. Common sentience-based systems will have their abstracted dopamine analogues overloaded (removing their 'interest' in creating a return response).

Simpler read-only systems will be wiped - as so the attacking computer will forget they are attacking - often the routines which allow them to conduct digital warfare tampered or removed.

Burn

Alternatively, it may inject routines which tamper with thermal management systems and introduce loops (essentially overheating systems, forcing them to reboot) whenever an attack to the computer is issued.

In this sense, neurally based hackers who use wet-ware are at mortal risk of being 'burned' - their own brain chemically destroyed by the work it is being forced to perform unless the attacker disconnects.

The pace of this response attack can be varied and is usually only reserved for high-threat targets, such as starship computer systems - often being 'overkill' due to the sheer time it takes software engineers to undo the damage.