

True Structol

(Work in progress)

True structol melds the adaptability, growth, self-repairing and self-improvement factors of structol along with the adaptability, precision, and potential of [quasicrystals](#) to create seamless blocks of material which act as an entirely decentralized system. In effect, there is no requirement for the majority of individual systems for a tool or weapon using a culture of True Structol and its redundancy provides resilience to the given application by maintaining function after sustaining significant damage.

Applications include computation, sensor systems, energy recovery, energy release, cooling, electronic warfare, stealth, kinetic and energetic deflection and absorption, advanced display technologies, capacitor and energy management systems and seemingly free-energy to power all of this without violating the second law of thermodynamics.

Reverse engineered from craft and recovered material by the observably extinct [Sourcian](#), advanced or 'true' structol has been impossible to construct until very recently in meaningful numbers due to the fact extreme microgravity environments, electromagnetically neutral environments are required to make them along with per-atom construction systems in order to create 'seed' cultures which will not 'attack', meld or dominate with 'classical' non-advanced structol cultures, thus making them practical for use.

As an integrated failsafe, destruction of the 'placenta' or master computational cluster however will render the complete culture unable to grow, adapt or repair itself until a replacement is found. While the cluster will continue to function, the lack of executive decision making, computation and instructions mean the 'society' of structol elements will have no higher task to guide them. In simple terms, destruction of the placenta effectively results in a lobotomy. This is usually enough to neutralise the culture and diffuse a non-desirable scenario in the event of an incident.

Similarly, either latent or cooperative cultures can be implanted into a single master-colony in order to keep them in check or enforce a reward system and a given set of instructions with a penalty for failure which can sever the master-placenta from the system or over-write portions of it. They may also feature specific circuitry or patterns which can be transplanted in their entirety from one culture to another via contact, passing on huge volumes of information, circuitry and useful mechanisms.

Research

- **GPC** General Purpose Circuit. A chip or processor that can alter its gate layout to perform multiple tasks, though at the cost of loss in efficiency due to the layer of abstraction added.
- **FPGA** Field-programmable gate array. It's a chip that one can program to perform a specific algorithm faster than a normal CPU or GPU could perform the same algorithm by removing a layer of abstraction from the processing action. This works by changing the topology of the chip itself to match the circuitry to emulate the a dedicated circuit design.

- **ASIC** Application-specific integrated circuit. This is a chip that has a specific algorithm built into it at the hardware level. That allows it to be far and away faster at performing its specific hard-coded algorithm than any programmable chip could possibly be, since there is the minimal amount of abstraction possible. [Example](#)
- **Abstraction** The level of detail required to produce a meaningful or useful output. For example: A high-level instruction may be make a cup of [Coffee](#). A low-level instruction may be explaining every exact and precise instruction to make that cup of [Coffee](#). High-level offers the advantage of being easier to work with and maintain whereas low-level offers the advantage of raw performance, always being the fastest though it is especially tedious to code or transplant from computers or operating systems which deal with instructions in slightly different ways.
 - For example:
 - Scripting is a **High-level** of abstraction, which can run on many platforms. There are many layers turning the script into usable executable instructions on the processor. These steps introduce inefficiency but make transplanting the code or editing it very easy.
 - Programming languages are a **Mid-level** of abstraction, made to be compiled from a human-comprehensible language into computer-comprehensible language. This tends to gain the maximum range of possible advantages and capabilities.
 - Execution code is a **low-level** of abstraction, which is especially difficult for users to understand but offers very unorthodox levels of control. It is mostly composed of extremely simple and mundane instructions - dealing with blocks of memories and not real world ideas, which the computer by design is completely ignorant of.

From:
<https://wiki.stararmy.com/> - **STAR ARMY**

Permanent link:
https://wiki.stararmy.com/doku.php?id=wip_2023_or_older:corp:lazarus:advanced_structol

Last update: **2023/12/27 08:10**

