CISC On RISC Engine

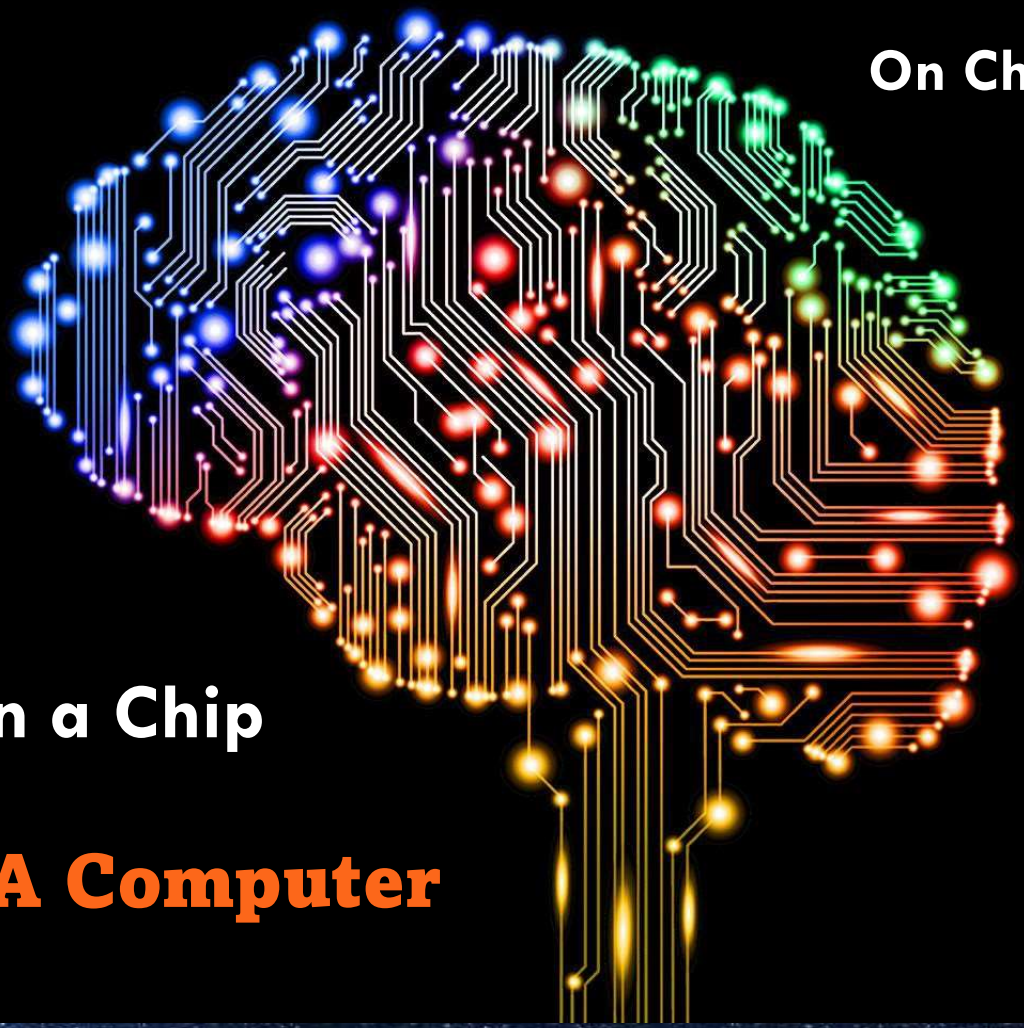
CORE I

32/64 Bit

Internal
FPGA
Networkable

Supercomputer on a Chip

Forth FPGA Computer



On Chip AI Engine(s)

Prolog

Probability
Engine

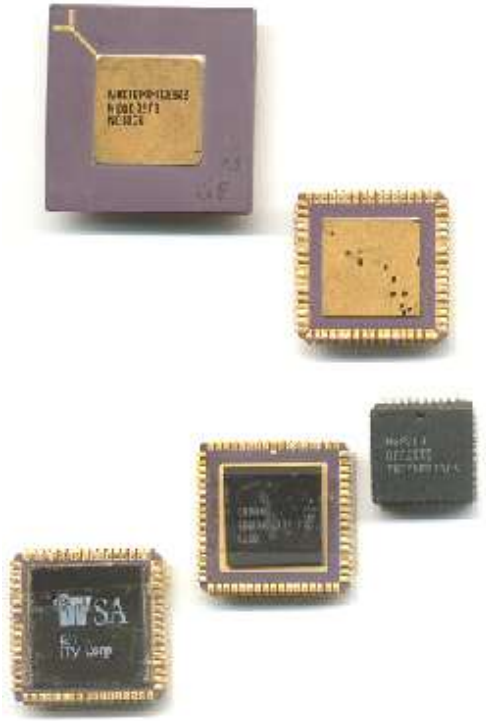
Forth is Not a Language

- Forth is a Virtual Machine, Forth is a CHIP!
- Chuck "Discovered Forth" while writing assembler code
- Forth is THE IDEAL processor architecture
- Forth is easy to implement in HARDWARE
- **Chuck's Chips:** NC4000, Sh-Boom, RTX2000, F21
- https://colorforth.github.io/bio.html
- Green Array's 144 Multi-Computer chip

http://www.greenarraychips.com/

- The World of Forth in Silicon

http://www.ultratechnology.com/chips.htm



Forth CORE I FPGA Computer Goals

- **CORE I** is a CISC Over RISC Engine
- Forth is the RISC engine
- Very high-level System Verilog Code which is very easy to read and understand
- Let the Synthesizer build the circuit
- Most FPGA Forth's are coded using RTL (Register Transfer Logic (eForth – DR. Ting, Ilya, J1, etc.)
- Design a stripped down Forth microprocessor for multiprocessor applications
- Design a complete computer within fabric RAM/ROM, external memory is optional
- Create super fast Forth Opcodes in System Verilog (think assembler)

Forth CORE I Computer Goals

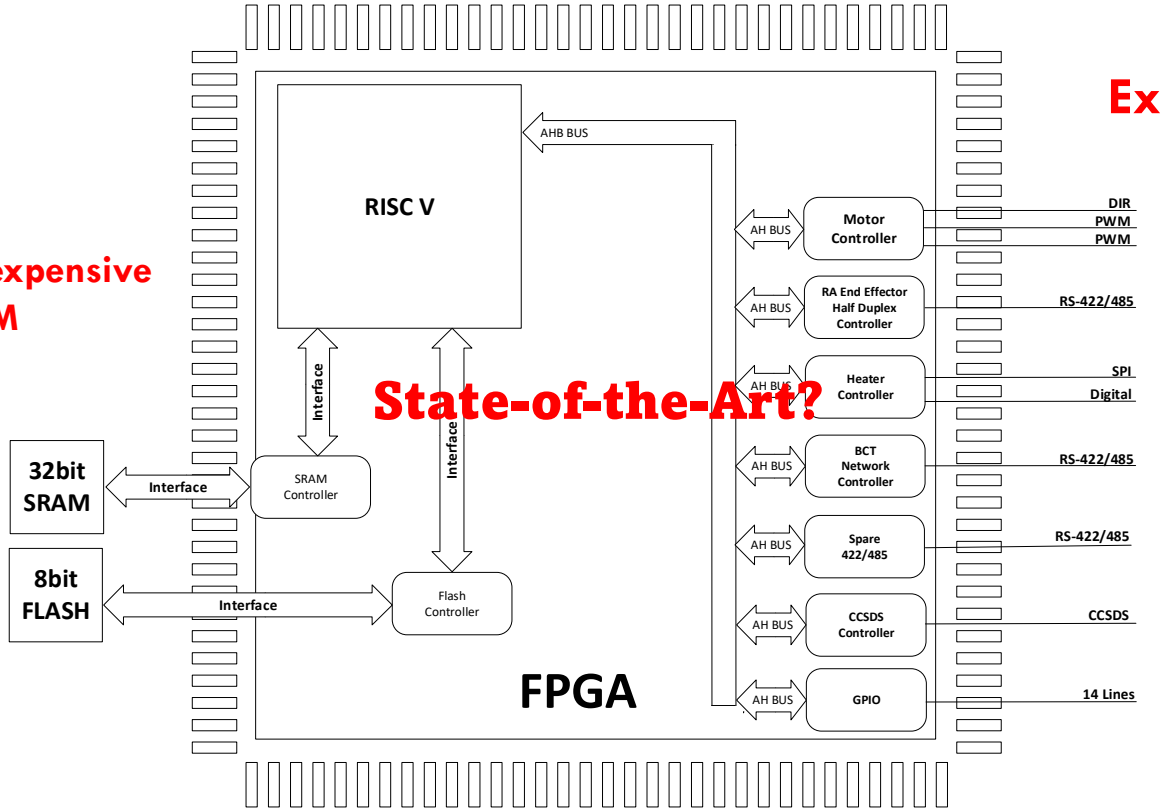
- Microcode Forth programs and Words in BootROM
- CORE extensions (opcodes) can be complex, taking as many cycles to execute as they require.
- Forth "Words" are opcodes
- These "Words" execute at the speed of SILICON
- Steal "Borrow?" high level concepts from other popular languages
- The computer grows as FPGA technology grows: 32 bit, 64 bit, 128 bit, ?
- Network multiple 32 bit Forth Engines together over internal network.
- Store source code in Flash or on disk, compile: on-the-fly as needed

32-bit RISC V FPGA Block Diagram

10,000 LUTs!

Can't run without expensive external Flash/RAM

RISC V opcodes are very primitive!



State-of-the-Art?

Expensive $$$

32 Bit **CORE I** FPGA Forth Basic Architecture

***CORE I** opcodes can be complex!*

1,000 LUTs!

Using $20 Microsemi iGloo2 FPGA (37,000 LUTs)

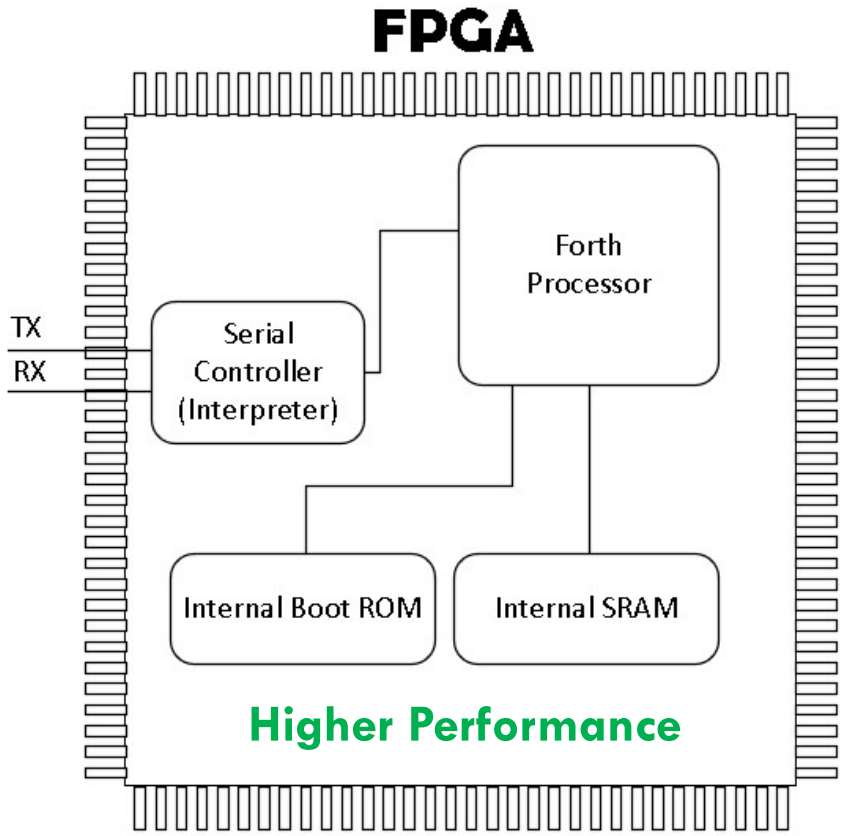
Can run with only internal Flash/RAM

K.I.S.S

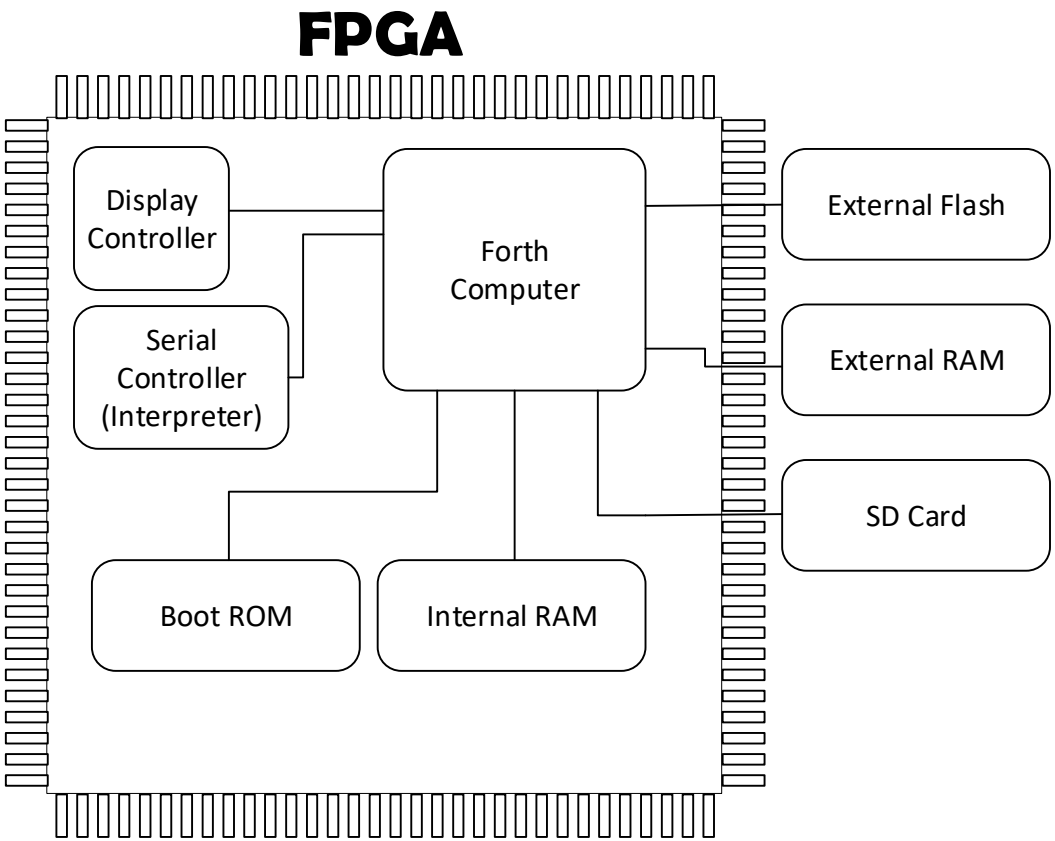
Resource Usage

Type	Used	Total	Percentage
4LUT	928	27696	3.35
DFF	581	27696	2.10
I/O Register	0	414	0.00
User I/O	8	138	5.80
-- Single-ended I/O	8	138	5.80

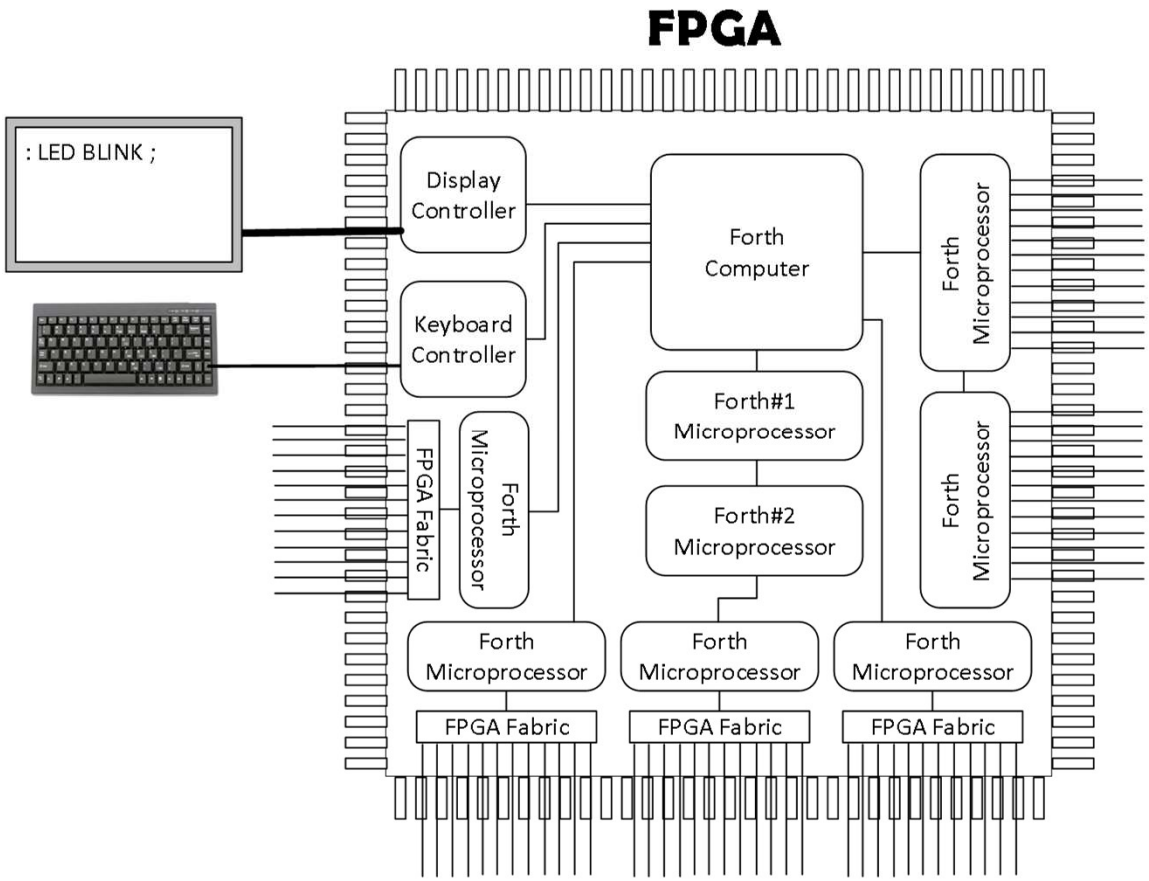
1/10 the Cost $$$ or less



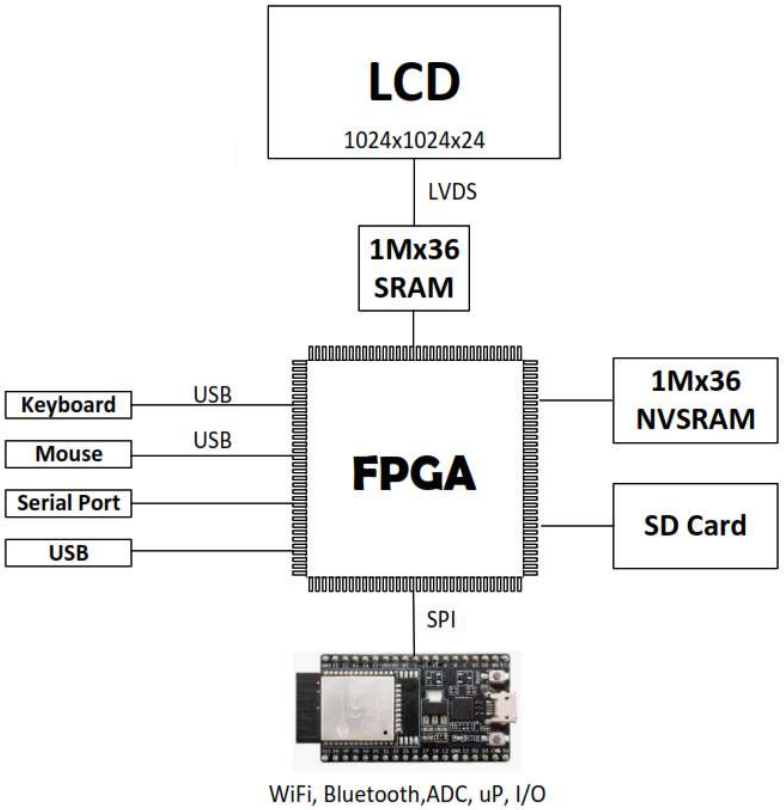
Forth CORE I Supercomputer Basic Architecture



Forth CORE I Parallel Computer



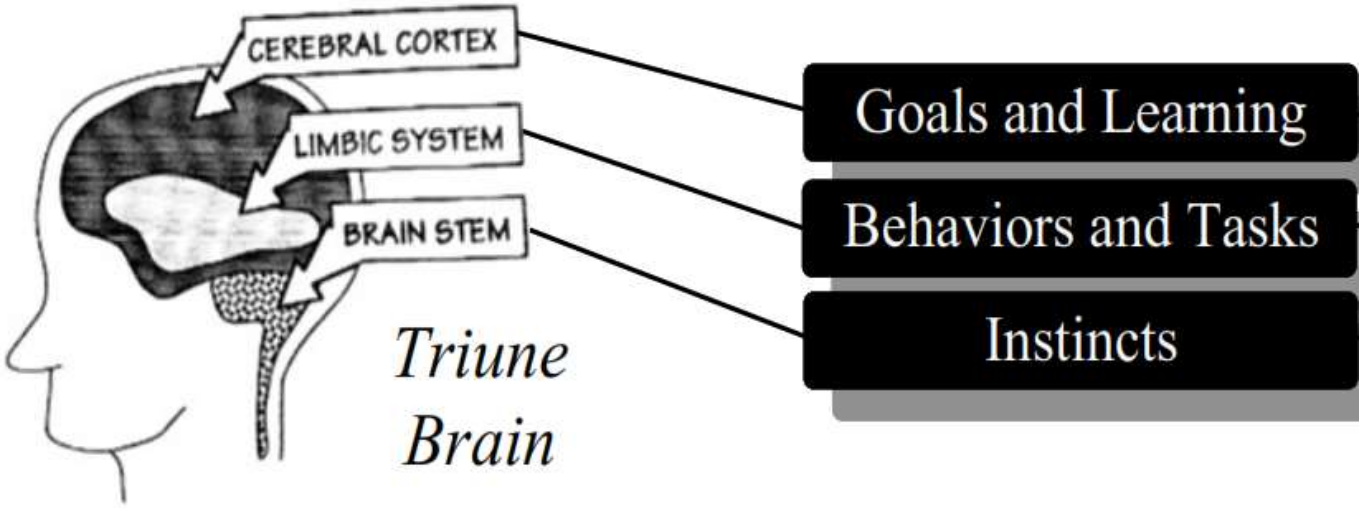
Complete **Forth CORE I** Supercomputer (Tablet?)



Forth CORE I Computer & AI

- Triune OS architecture will be used
- It is easy to implement other languages in Forth
- Dimitri has written Prolog in Forth for the **CORE I** Computer
- Prolog and/or other languages can be implemented in the Boot ROM
- Don Golding wrote Neural Network language in Forth
- Probability engine
- Vision system can be instantiated from other sources
- Voice Recognition can be instantiated from other sources

Triune OS in Silicon



- Emulates Human Triune Brain
- Programming is like “Teaching a Child”

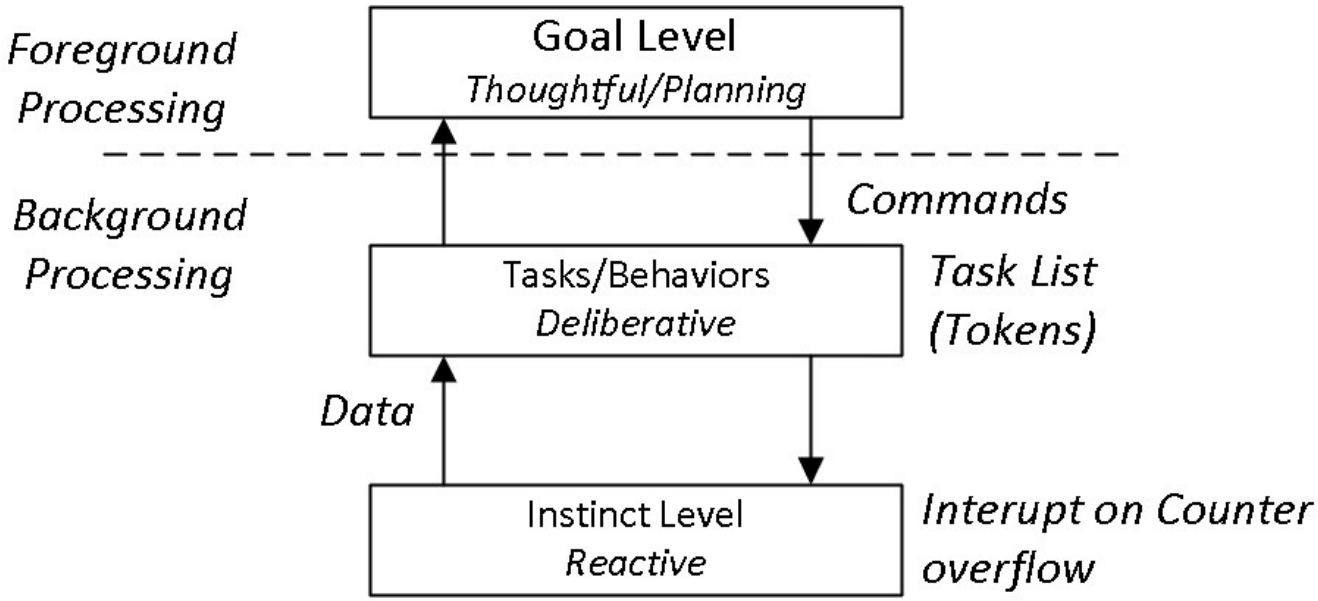
Simplifies Programming and Machine Learning

Triune OS in Silicon

- ***Cerebral Cortex*** - Goal Level
  - The Outer Layer of the Brain
  - Folds under the Skull
  - Decision making, Analysis, and Dreaming
- ***Limbic System*** - Behavior Level
  - The Gray Matter found in the Center of the Brain
  - Hunger, Fear, Feelings
- ***Brain Stem*** - Instinct Level
  - The Base of the Brain Connected to the Spinal Cord and Nervous System
  - Controls Critical Responses and Instinctive Behavior

TOS Emulates Intelligence in Nature

Triune OS Multitasking/AI Model



TOS Emulates Intelligence in Nature

Forth CORE I Development Boards

5,700 LUTs
$4.50

$24 Lattice Board

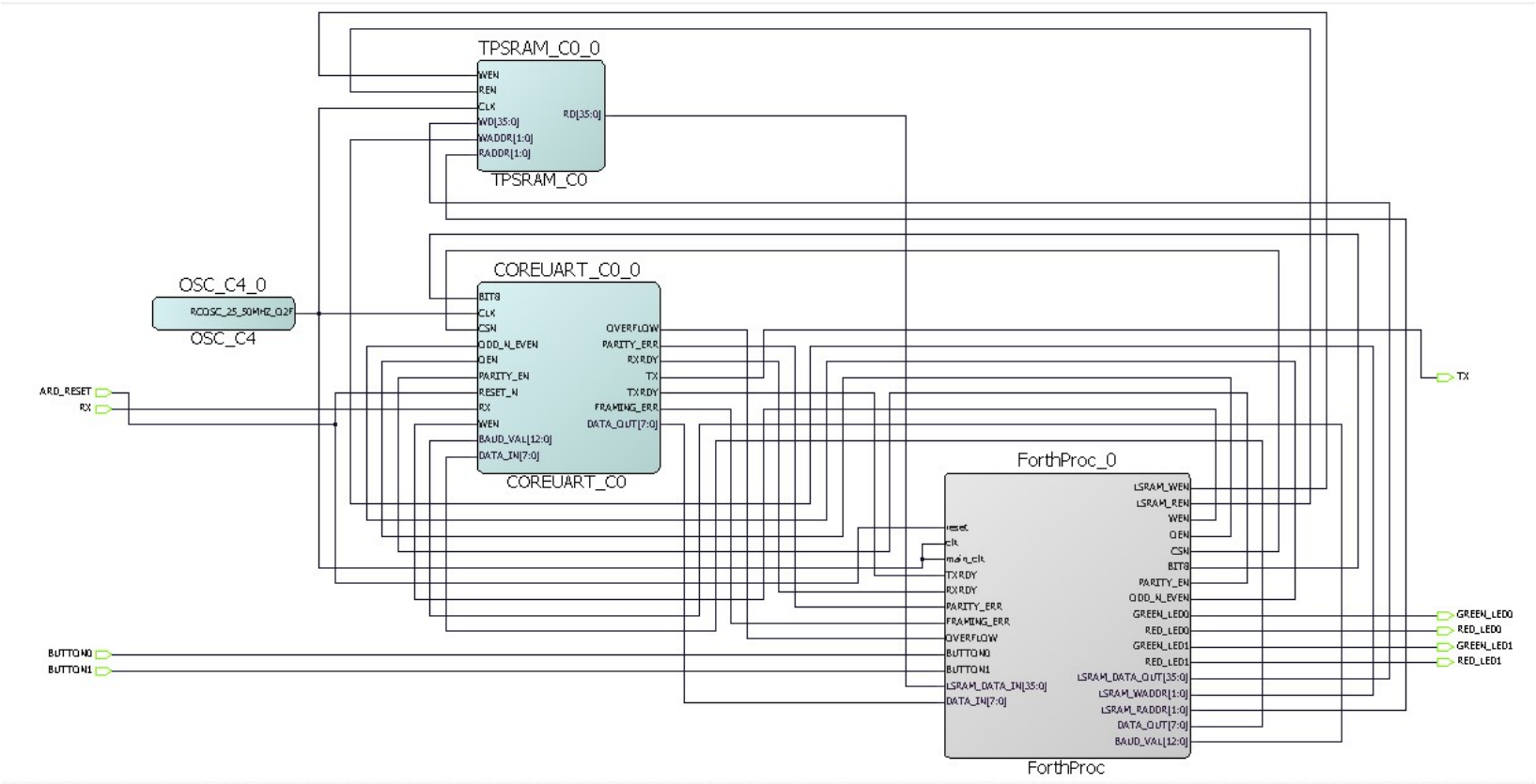
3,700 LUTs
$2.00
Available

$160 Creative RISC V Board

37,000 LUTs
$20

CORE I requires less than 2,000 LUTs, 930 currently

Forth CORE I FPGA Design



Forth & AI, A Long History

- **LISP written in Forth**
  https://forums.parallax.com/discussion/160027/lisp-technically-scheme-written-in-forth
- **Prolog in Forth**
  https://www.researchgate.net/publication/234780918_Compiling_Prolog_to_Forth
- **NASA: Use of a FORTH-based PROLOG for real-time expert systems**
  https://ntrs.nasa.gov/citations/19930073450
- **Simulating Recurrent Neural Networks in Forth**
  http://www.complang.tuwien.ac.at/anton/euroforth/ef16/papers/baranov.pdf
- **Smalltalk**
  http://soton.mpeforth.com/flag/fms/index.html
- **Domain Specific Languages**
  http://mathscitech.org/articles/pol
- **Demitri Prolog - current**

Conclusions/Benefits

- Build High Performance/Low-Cost "Intelligent" Embedded Systems
- Research Novel Forth based Computer Architectures
- Research/Develop AI on Top of Forth
- Add High-Level constructs to Forth the language: ENUM, String, etc.
- Easy Way for beginners to Learn About FPGAs
- Use the Serial Port and Forth to Debug System Verilog Processes
- Basic 32-bit Forth Computer fits in $2 FPGA
- Controller for Intelligent Machines and Robots
- *Bring new people into the Forth World*

The Path Forward

- Forth Micro Engine Development – Completed
- Forth Outer Interpreter/Compiler – in process
- Full project on GITHUB – when Forth Computer is working
- Facebook "AI & Robotics" Private Group – will be open to Forth Community (Jan 2022?)
- Triune OS multitasking engine
- String and Array extension Words
- Prolog? and/or other AI extensions – either microcode in Forth or coded in System Verilog for speed (Assembler)
- Design Printed Circuit Board