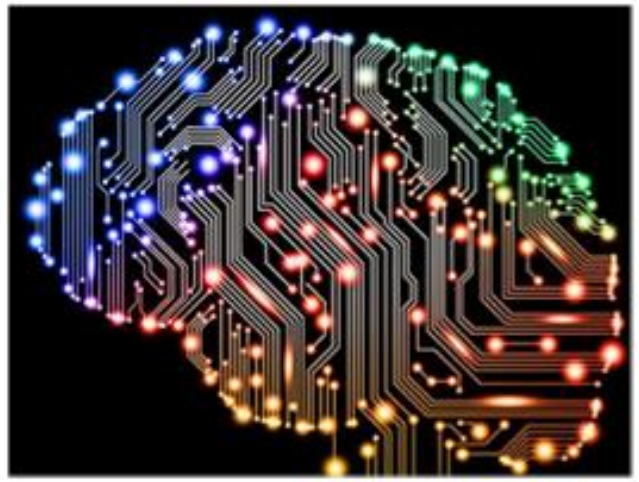
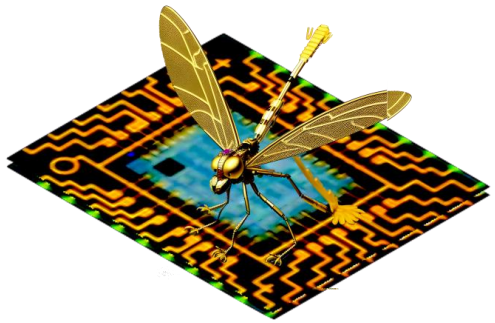
AI, Robotics + IOT Our Future World!





SUMMIT OF
THINGS
Virtual Event & Expo
Oct. 25-27

About Don Golding

- SR Staff Engineer
  - Designed NASA's Archinaut One Computer
  - Currently Working on a new Space Computer for NASA Goddard/JPL
- Founder: Space-Tek, Inc.
  - Medical Imaging Computers & Systems
- Founder: Angelus Research Corp.
  - Intelligent Robotics: STEM, Military, AGV
  - Triune OS & Language for Intelligent Robots

Lockheed Martin M.U.L.E Robot

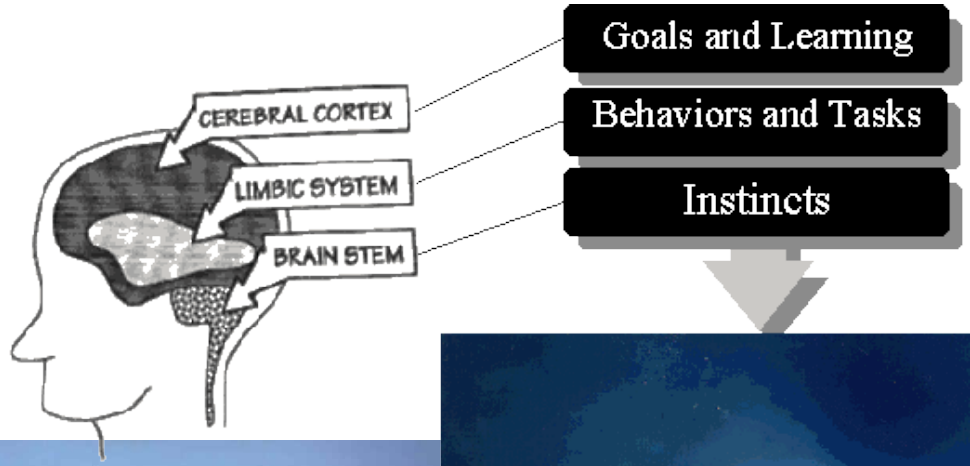
- 6 Articulated Arms with Powered Wheels
- Fully Autonomous
- Canceled 2011
- Six Cyclone FPGAs with two 32 bit Processors
- One FPGA Computer per Wheel/ARM
- 100 MBIT Network
- Each Arm Controller 6KWatt Brushless Motor

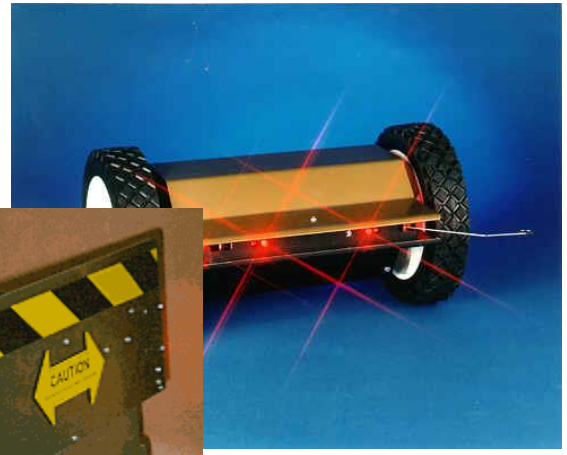
NASA Redwire Archinaut One Mission (OSAM2)

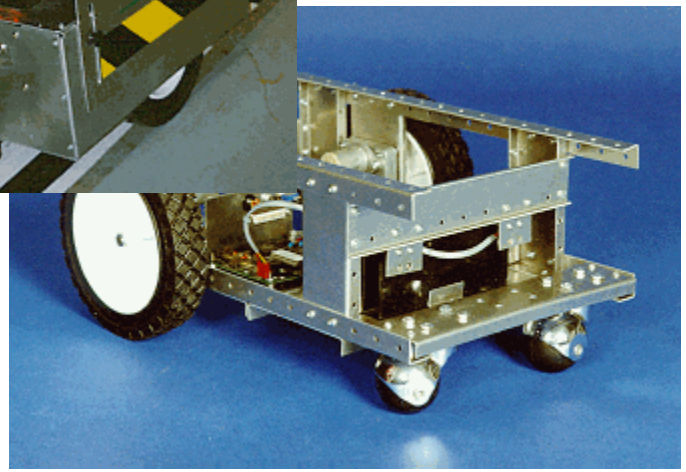
- 10 Meter Lattice Beam (100mmx100mm)
- Fully Autonomous
- Launch in 2024
- Designed the Rad Hard FPGA Command and Control Computer
- Redwire designed a Space Rated 3D Printing Filament
-
- Simulate the 3D Printing of a Solar Panel in Space
-
- Solar Film unrolls as Beam is 3D Printed

First NASA Mission to Build a Major Structure in Space

Angelus Research Corp. Don Golding - Founder/Eng







Some of the Intelligent Robots DG Designed

CORE I Applications: IOT Devices

Personal Robots

Surveillance Robots

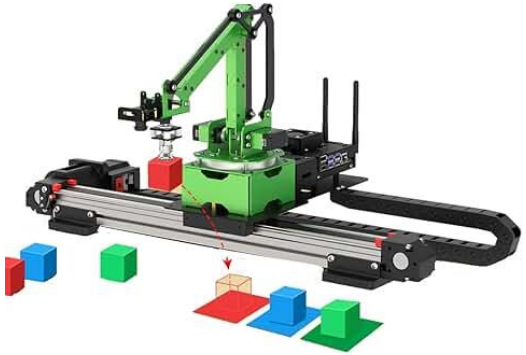
Cell Phones

Drones

Smart Watches

WiFi BBQ Smokers

Robot Vacuums



AI Experimental Robots

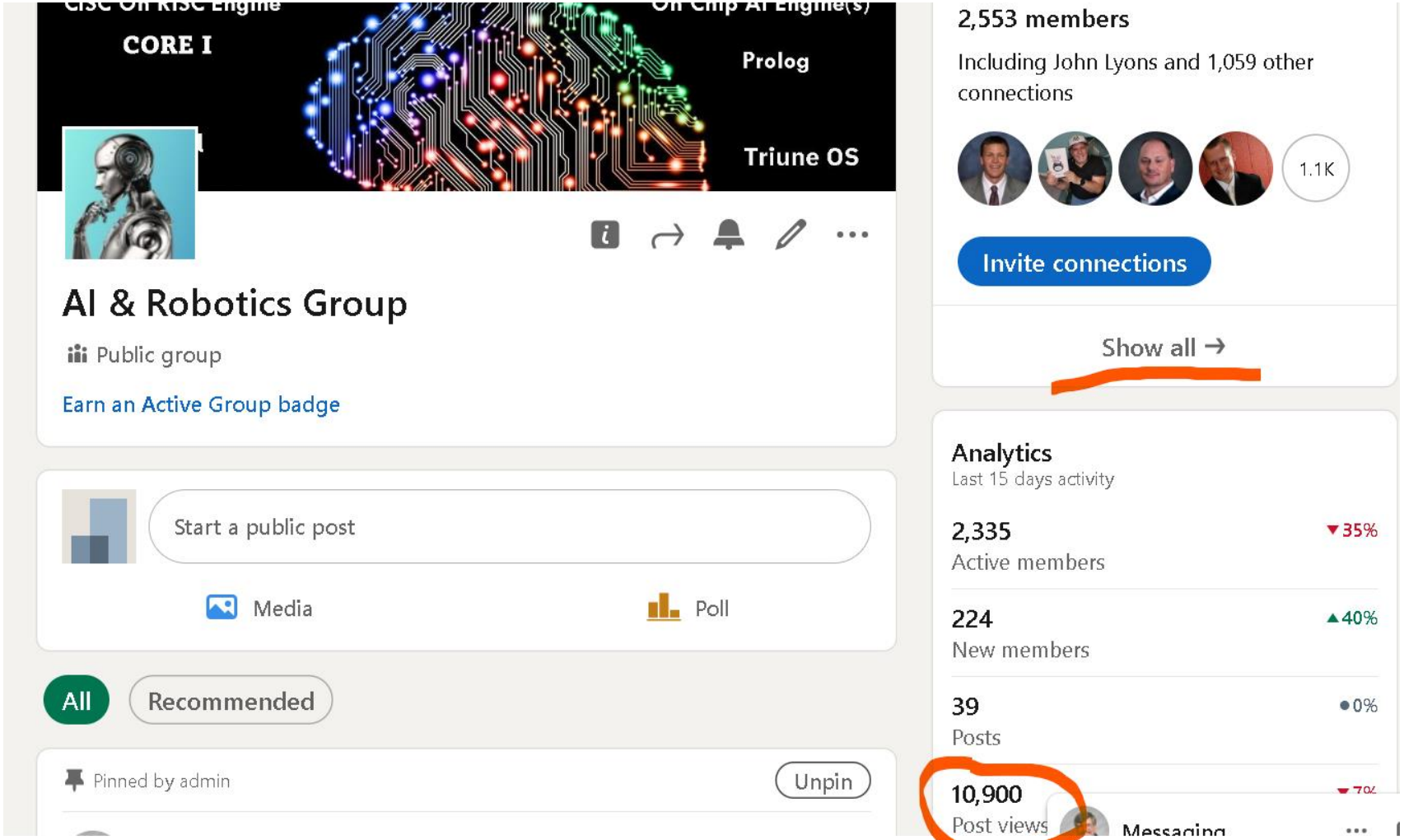
Home Security/Automation

Electric Cars

Hobby Bots

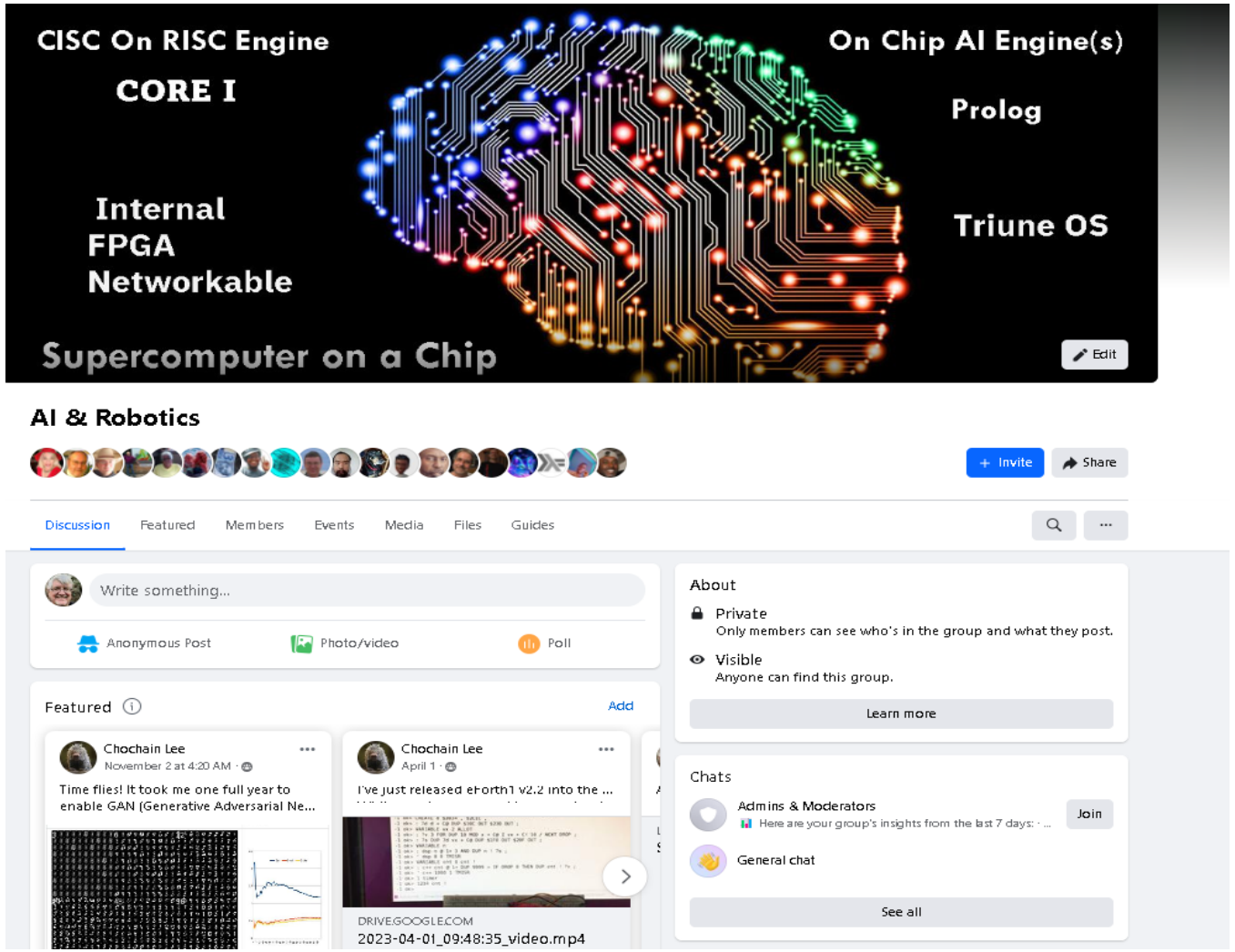
Internet Of Things (IOT)

LinkedIn AI & Robotics Group



https://www.linkedin.com/groups/12858138/

Facebook: AI & Robotics (321 Members)



https://www.facebook.com/groups/1304548976637542/

CISC On RISC Engine

CORE I

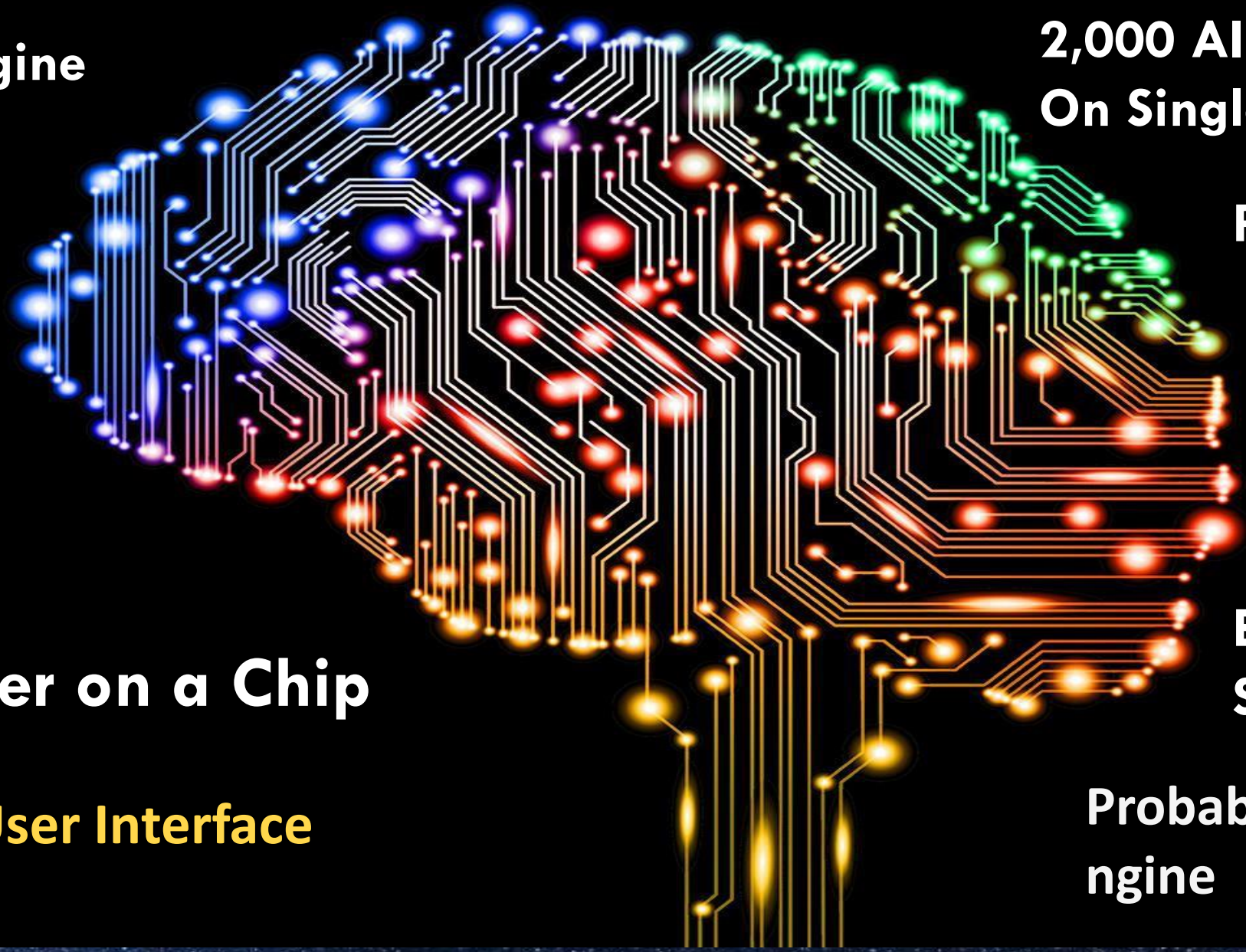
32/64/256 Bit

Neural Nets

SERDES
20ghz Network

Supercomputer on a Chip

Chatbot User Interface



2,000 AI Engines
On Single Chip

Prolog

Expert
Systems

Probability E
ngine

Intelligent Machines: Massive Parallelism in HARDWARE

Why Forth?

- Forth was Invented in 1963 by Charles Moore
- Forth has used a Chat-bot Interface since the very beginning
- Outer Interpreter, promotes simple Testing and Debugging
- Simple incremental Compiler
- Forth is a Virtual Machine easily Implemented on a Chip
- Forth is a Meta Language, easy to implement other Languages
- Forth was 60 Times faster than Python Parsing Strings – Dr. Ting

Why not use C?

- Not interactive, unless you use a derivative like Java, Python
- Very complex Compiler – Requires Powerful Computer
- You must Compile entire Application if you change one line of Code
- JTAG binary file into CPU
- Need more Memory and Processing Power
- Reprogramming on Same Machine requires and OS, IE: Linux

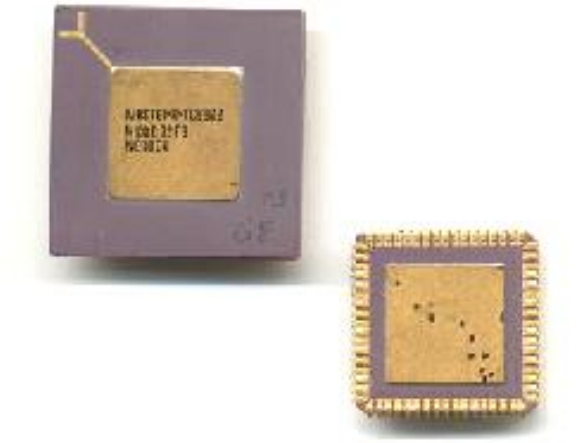
Forth in Hardware History

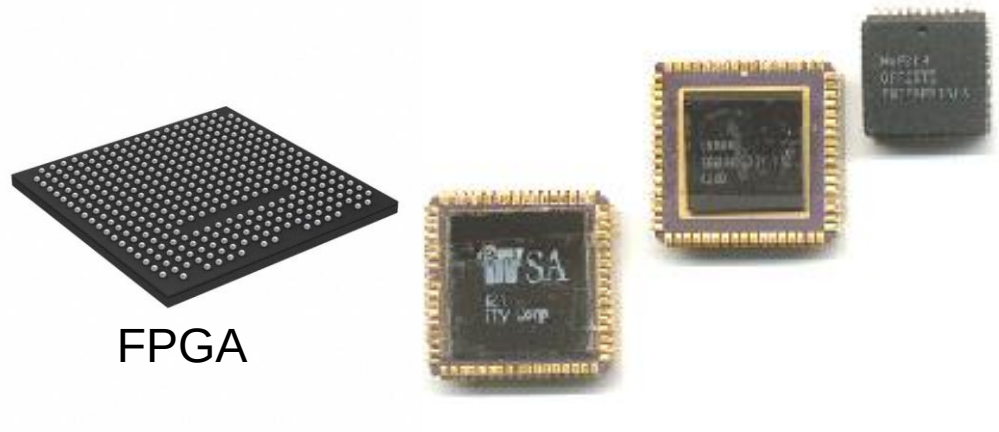
- Forth is a Virtual Machine, Forth is a CHIP!
- Chuck "Discovered Forth" while writing assembler code
- Forth is THE IDEAL processor architecture
- Forth is easy to implement in HARDWARE
- **Chuck's Chips:** NC4000, Sh-Boom, RTX2000, F21
- https://colorforth.github.io/bio.html
- Green Array's 144 Multi-Computer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

http://www.greenarraychips.com/

- The World of Forth in Silicon

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





FPGA

General Purpose Computers vs CORE I

On a typical software program, only a small fraction of the available CPU opcodes are actually used.

- Some estimates suggest that around 5-10% of a CPU's instruction set is utilized on average for most programs.
- Today’s Computers require an General Purpose Operating System slowing Program Execution.
- High Power Consumption, Low Efficiency
- CORE I – Implement only the Opcodes you need – 95% Utilization

CORE I is Efficient Utilize >90 Percent of Fabric

CORE I FPGA Benefits

- Fully Programmable 32 bit(64,128?) Computer on a single Chip.
- Outer Interpreter - On Chip (UI), like Chatbot
- Incremental Compiler – On Chip
- Built on Forth Virtual Machine
- Send Source Code through Terminal Interface
- Move Application Software into Hardware (FPGA)
- No External Development Tools Required for Forth Code
- Future: Build Very High Level Natural Language on top of Forth

Build most Converational Computer on a Chip

Forth CORE I Computer Features

- 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
- Take SRAM Snapshots and store in Flash or SD Card

CORE I FPGA Development Goals

- Develop Vast Open Source User Community in AI
- Extend the System for an Intelligent Chat-bot User Interface
- Users can quickly and easily add their own Words to the System
- Load LLAMA 7B Dataset into Flash Memory (33 Megabytes)
- Build Inference Engine to Interrogate LLAMA LLM Dataset
- *Need to Figure out how to build a Pytorch Model*

A file with a . pth extension typically contains a serialized PyTorch state dictionary. A PyTorch state dictionary is a Python dictionary that contains the state of a PyTorch model, including the model's weights, biases, and other parameters.

CORE I FPGA Benefits, Continued

- Create your own Complex Opcodes in System Verilog
- Share Opcodes on Github
- Opcodes run at the Speed of Silicon 200mhz+
- Using System Verilog, create 1000's of Parallel Processes – On Chip
- Create Parallel Processing Device Drivers for Peripherals
- Send Source Code over Remote Link, Compiles directly On Chip

Forth + System Verilog

- The User Interface Computer is On-Chip in Forth

    1 Easy Programming
    2 NLP/Interpreter
    3 Incremental Compiler
    4 Extensible High Level Language

- Chip Processing in System Verilog

    1 Code becomes a Chip Circuit (Fabric)
    2 All Processes Run in Parallel
    3 Massively Parallel Processing Capable

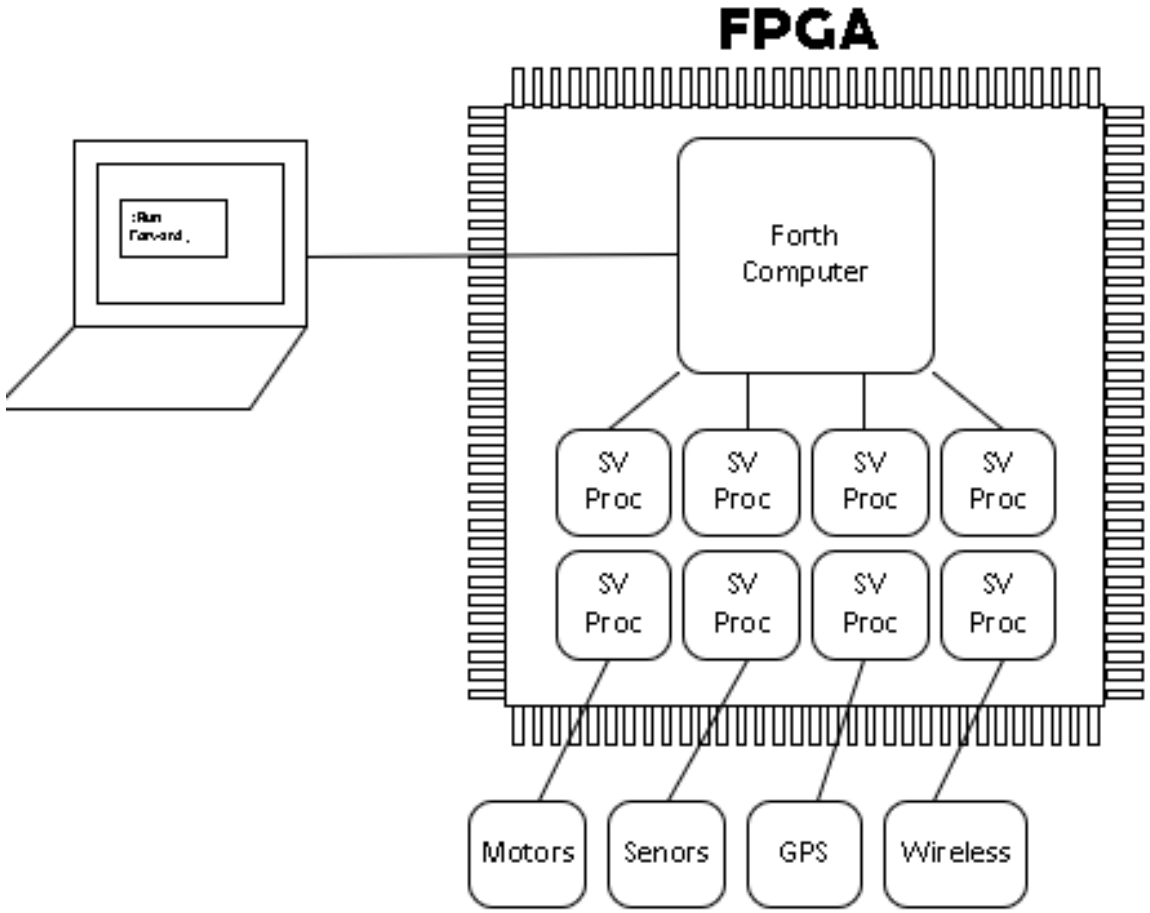
Forth + System Verilog

Easier to learn Chip Design

Chip Level Speed, Easy to Program and Debug

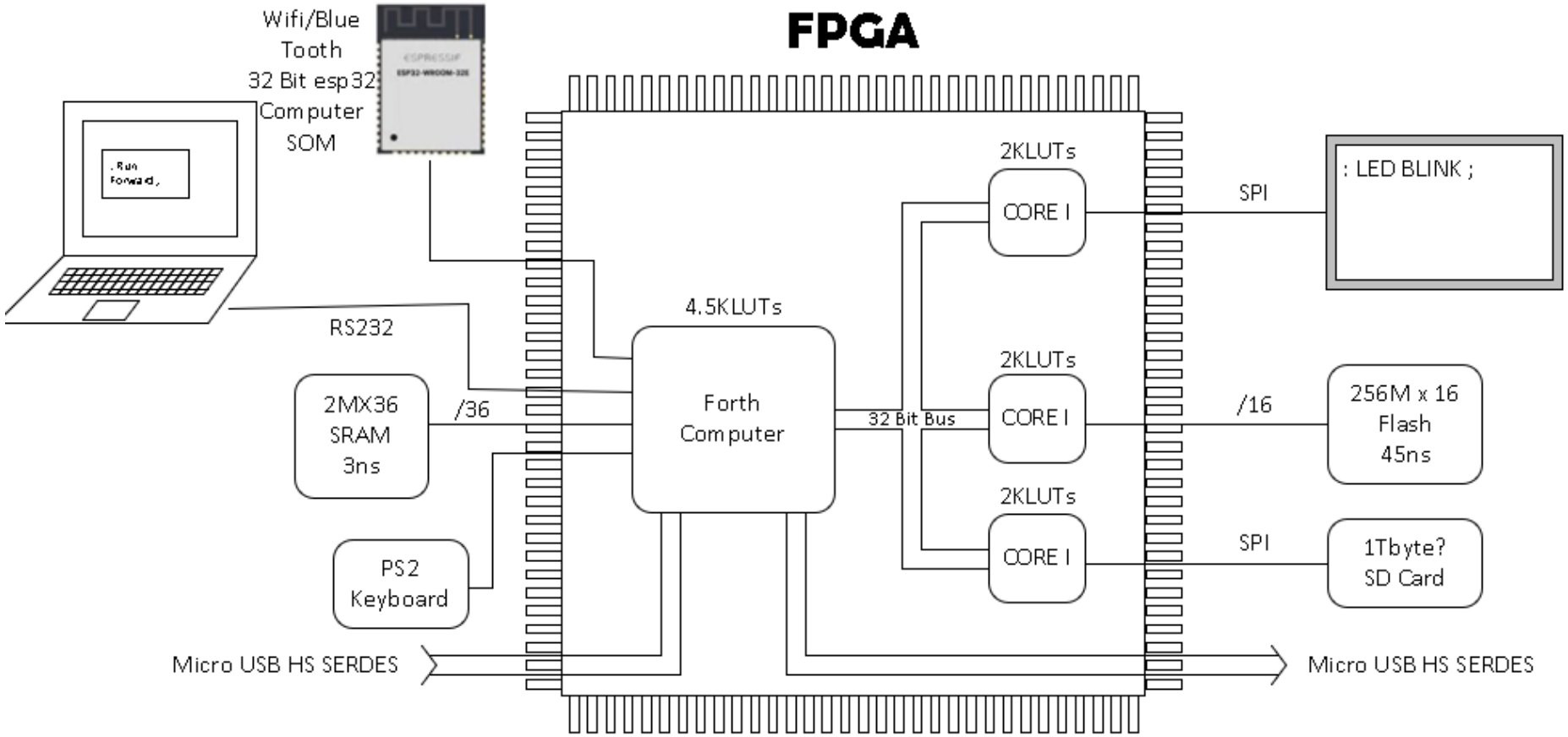
Forth + System Verilog

- Forth is the User Interface Chat-bot
- System Verilog (SV) is the Assembler
- Program All Device Drivers in SV
- All SV processes run at 200mhz+
- All SV processes run in Parallel
- FPGAs can run Thousands of Parallel Processes, Concurrently
- Multi-Tasking Supported in Forth



Simple UI, Massive Parallel Processing

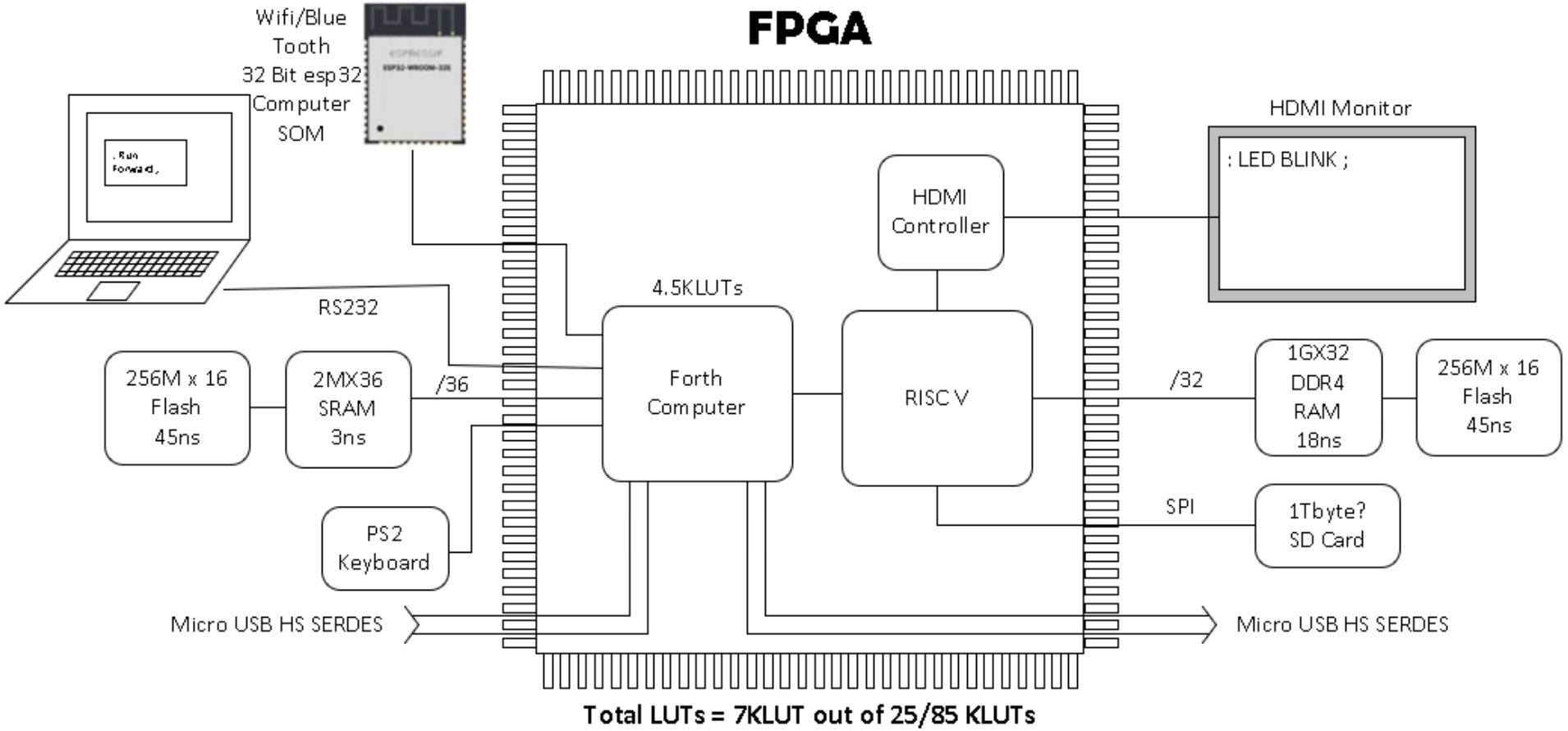
CORE I AI Playground Block Board



Advanced System: Total LUTs = 10.5KLUTs

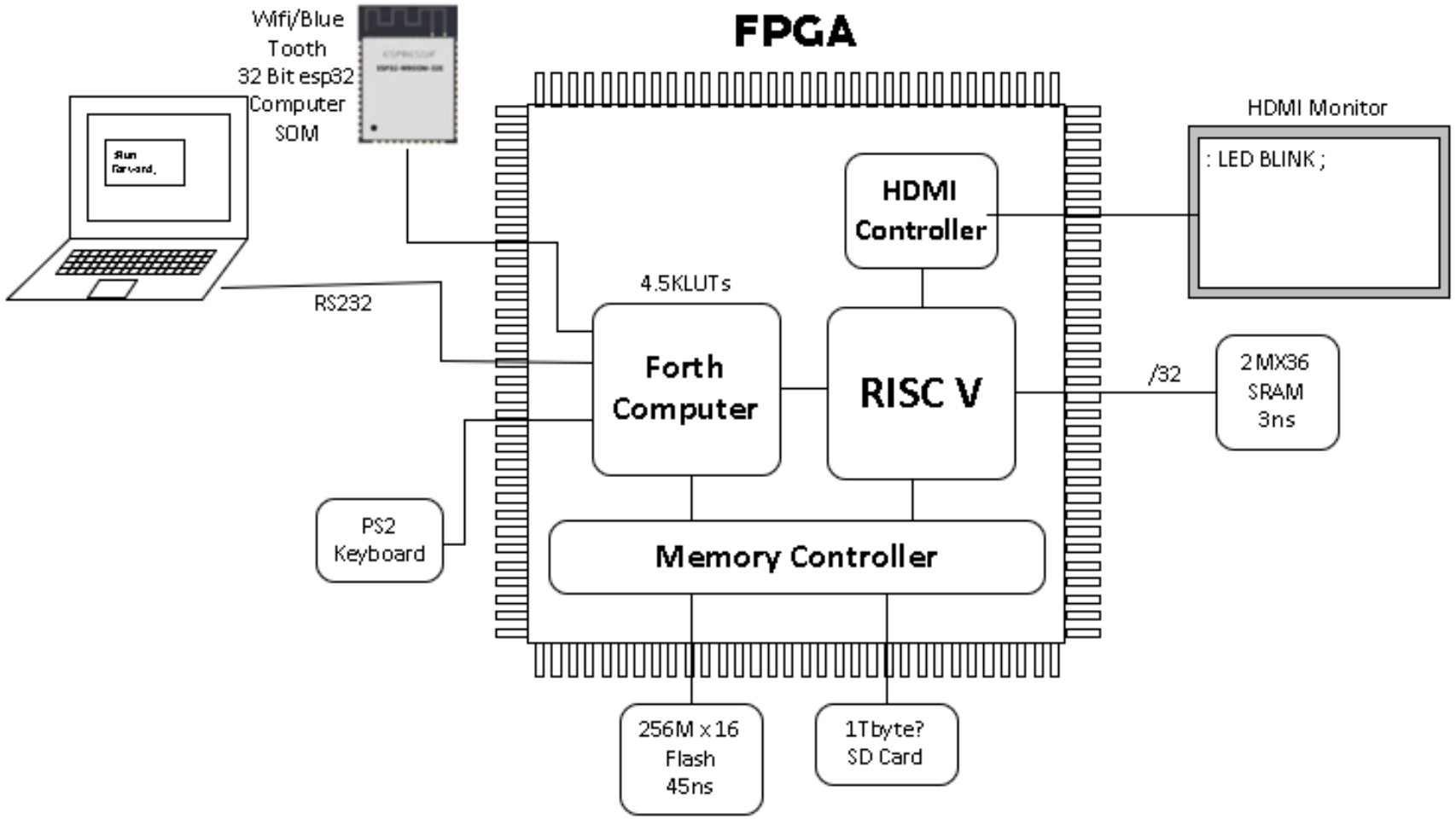
YOU can Experiment and Develop Computer Architectures

CORE I + RISC V on a Single Chip



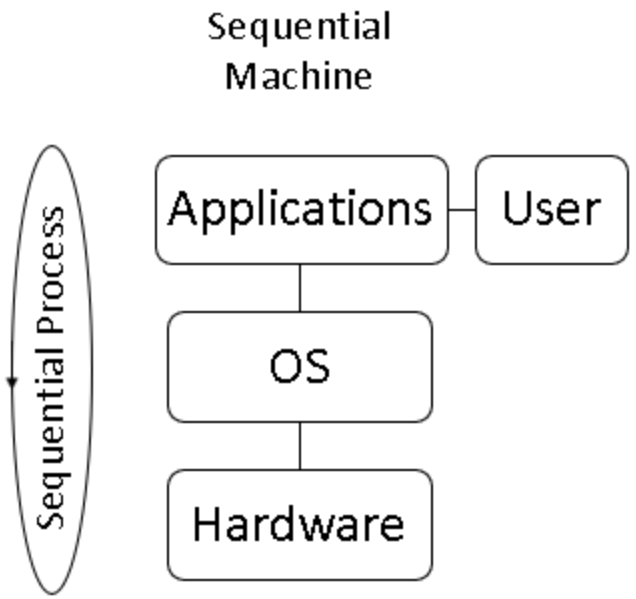
YOU can Experiment and Develop Intelligent Machines

CORE I + RISC V on a Single Chip

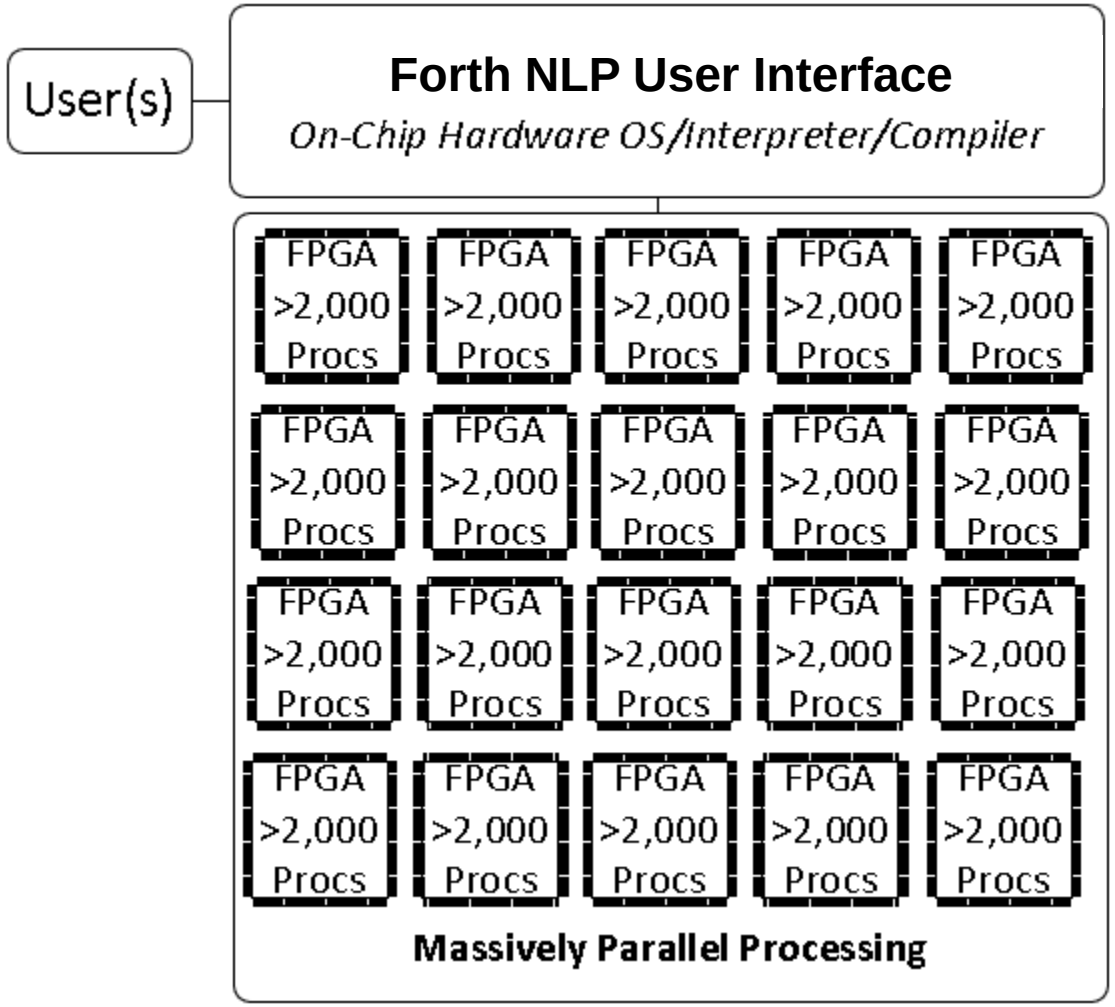


Forth, CORE I, RISC V on Current Board

Sequential vs Parallel Computing



WinTel/Apple/Linux



Massive Parallelism vs Today's Sequencial Computers

CORE I Massively Parallel Computer

Coded in System Verilog...

User Interface

Forth (NLP) Computer

250 MIPS Neuron Input Processor

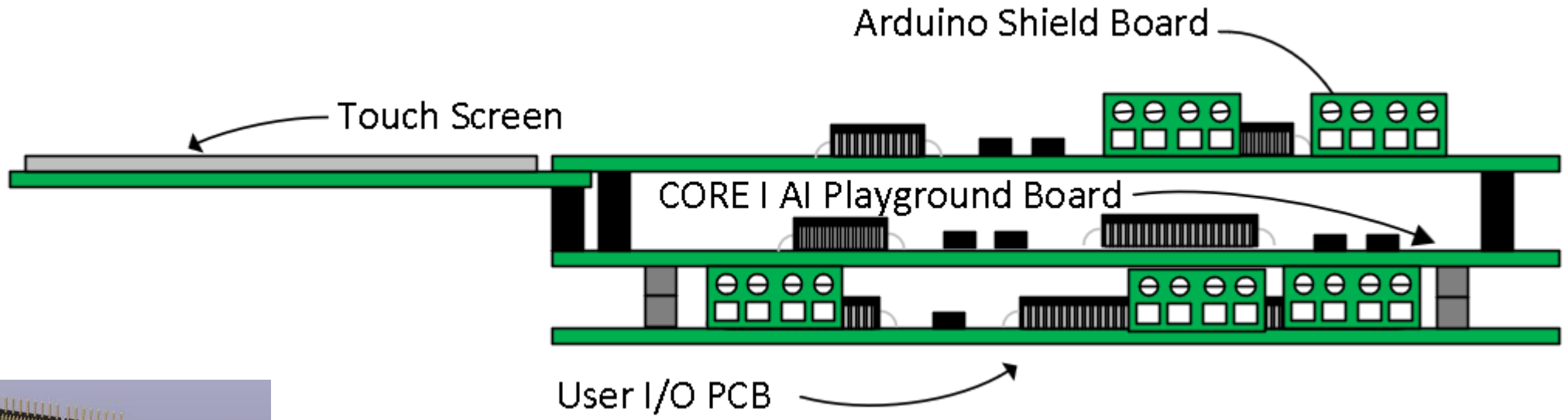
Nueron 250mhz

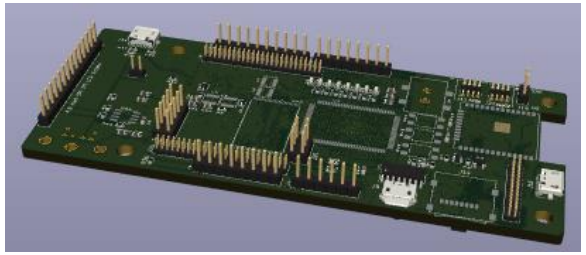
250 MIPS Neuron Output Processor

Massively Parallel Concurrent Processing

Massively Parallel Computer

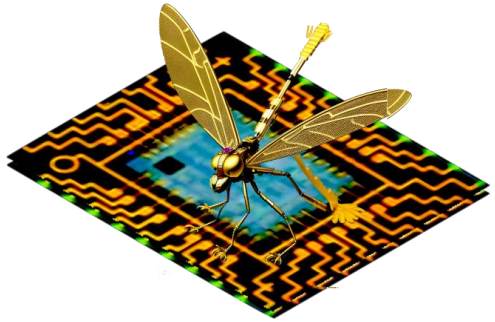
CORE I AI Playground Board



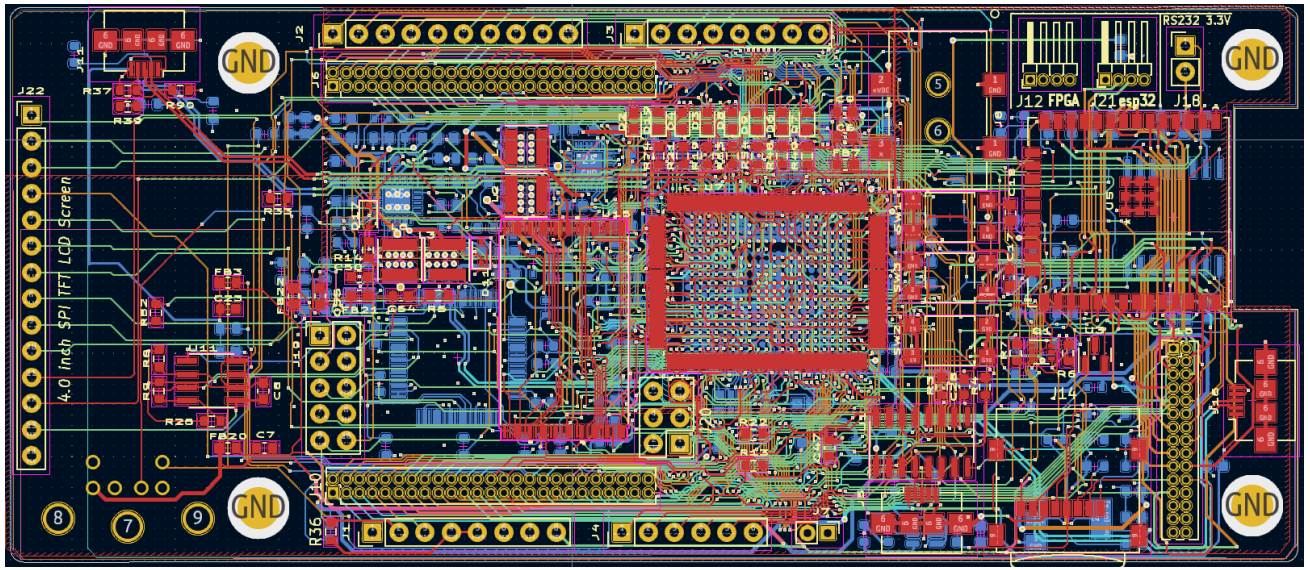


YOU can Experiment and Develop Intelligent Machines

CORE I AI Playground Dev Board - 1st Product



Very Low Power
Runs on Batteries

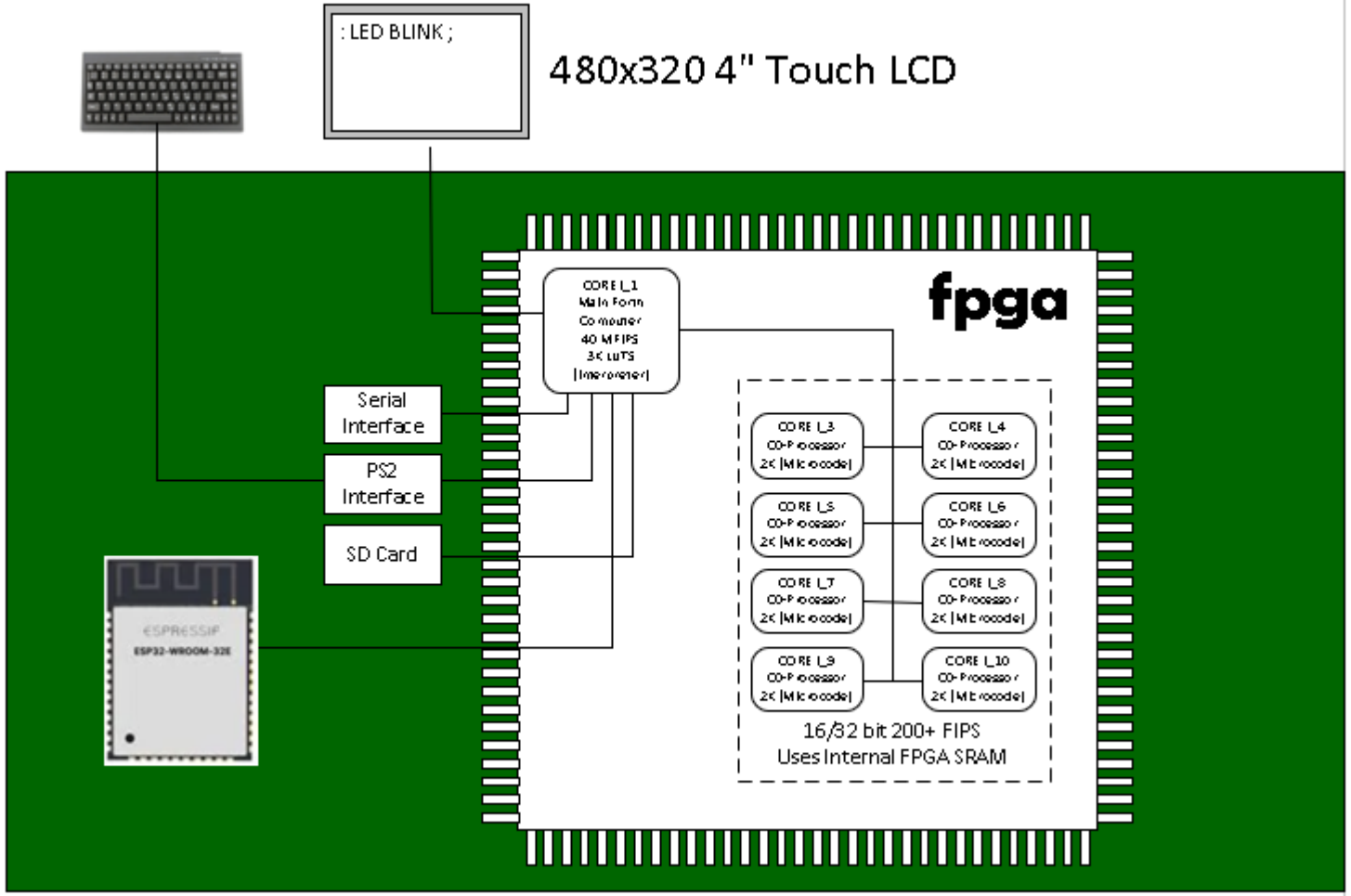


- Demonstrates CORE I AI Technology
- Encourage Early Adopters
- Build Community of Developers
- Benchmark Massive Parallel Computer vs PC

Develop Your Own AI based IOT Products or Just Experiment

AI Playground Board Block Diagram

Esp32 8M PSRAM +
400K SRAM
Wifi + Bluetooth



Lattice
LFE5UM-25F-6BG381C
FPGA

85KLUT with 512K High Speed SRAM Version Capable

A Full IOT Computer

How to Build AI/ML on CORE I Technology

- Define YOUR Functions as OPCODES
- Use Opcodes Interactivity (They are now Words)
- Extend Opcodes to Fit your Application
- Over 2,000 Parallel Processing Engines per Chip (depends on LUTs)
- AI/Machine Learning (ML)
- Neural Networks
- Expert Systems
- Implement other Languages: Lisp, Smalltalk, Prolog, Python?
- Use CORE I Processors for Top Level Processors: Display, Flash, SD Card, Net
- Create Input, Output, Semaphore Registers for Forth NLP
- Create Opcodes to update these Registers

Conclusions/Benefits

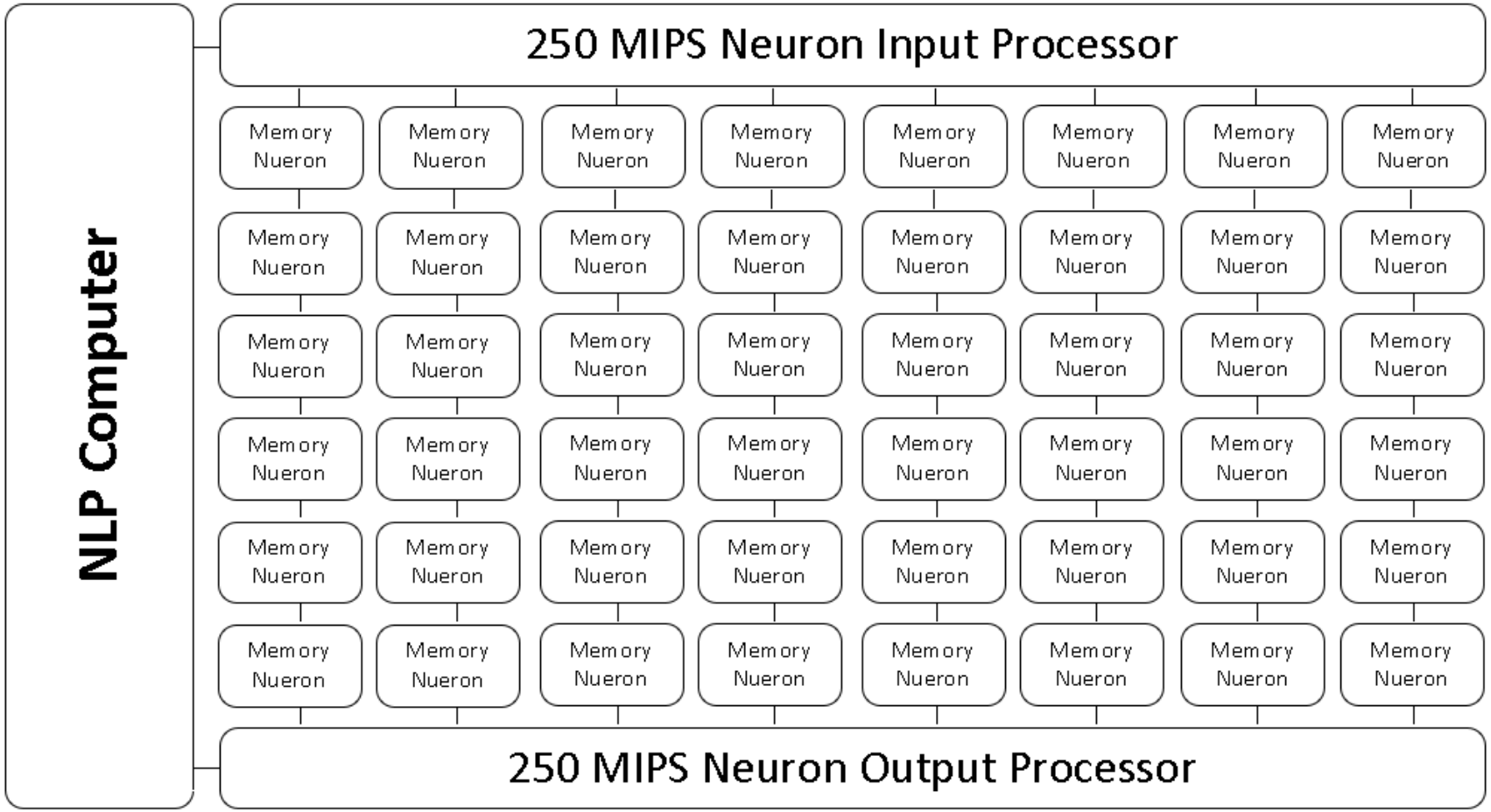
- Build High Performance/Low-Cost “Intelligent” Embedded Systems
- Research Novel Forth based Computer Architectures
- Research/Develop AI on Top of Forth
- Add High-Level constructs to Forth the language: ENUM, String, etc.
- Easy Way for beginners to Learn About FPGAs
- Use the Serial Port and Forth to Debug System Verilog Processes
- Basic 32-bit Forth Computer fits in $5 FPGA
- Controller for Intelligent Machines and Robots

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

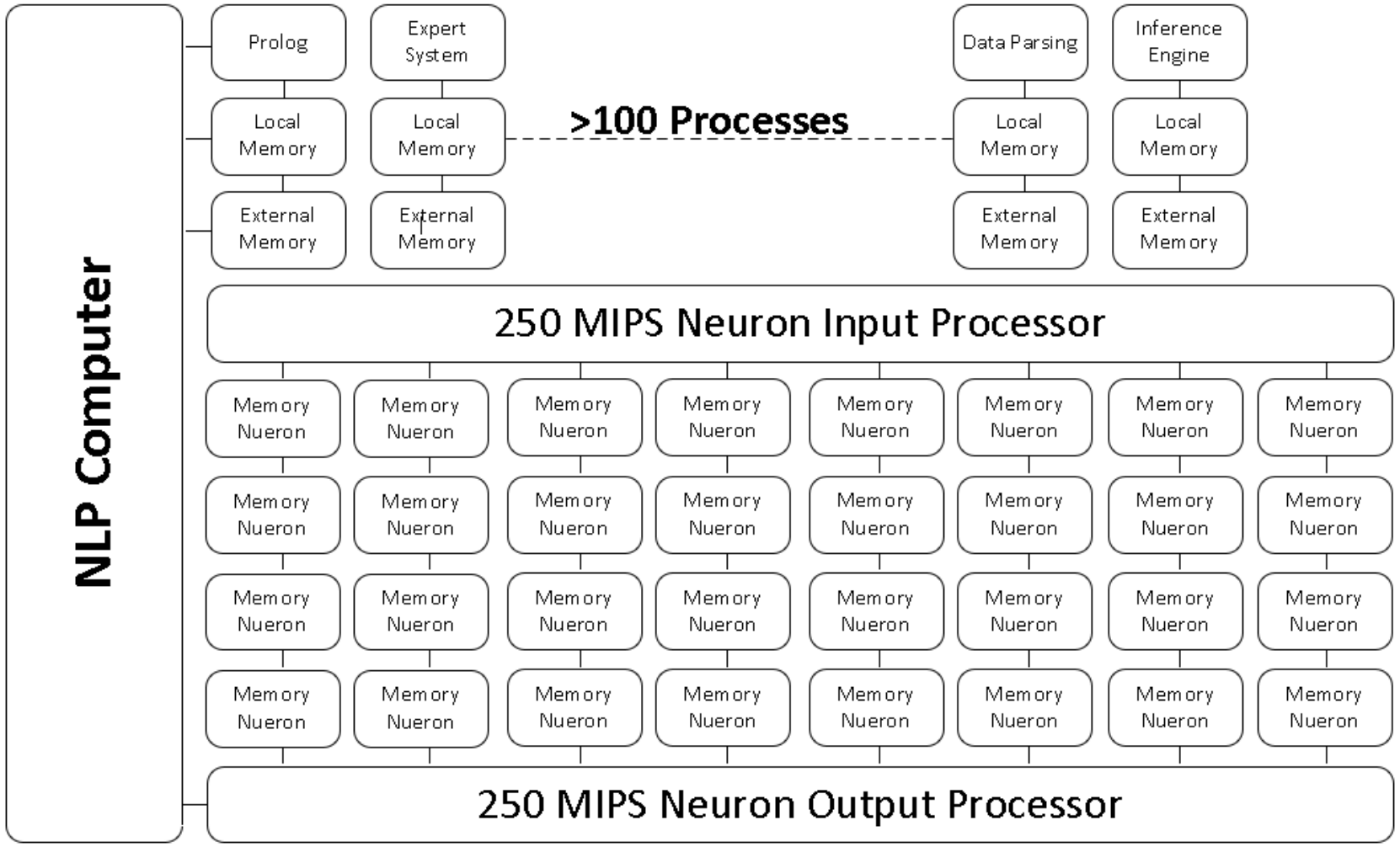
Digital Neural Networks

- Each Neuron Processor Updates Thousands of Neurons Depending on Application
- Thousands Neurons per FPGA Chip
- Size only Limitied to the Number of LUTs on the FPGA
- Each Neuron Processor Executes at 250 MIPS



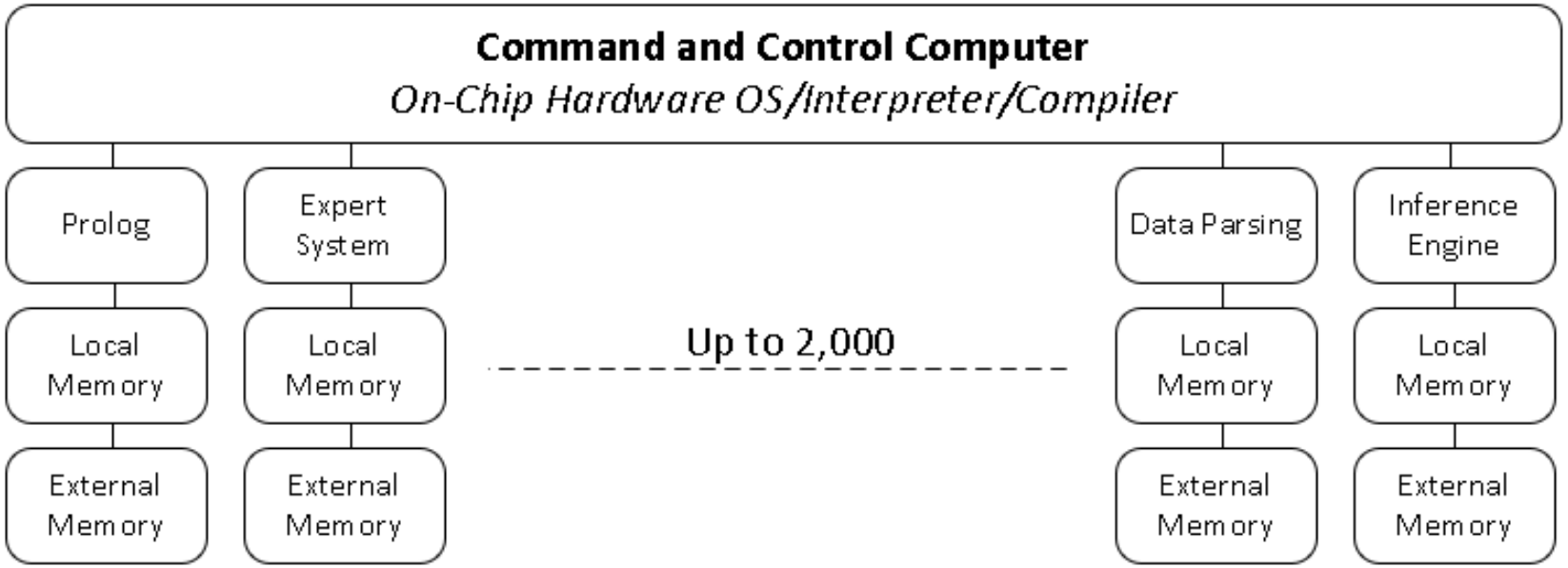
Combine Expert Systems/Neural Networks

- Massively Parallel Processor
- Neuron Processor/Expert System executes in Parallel
- Thousands of Processes per FPGA Chip
- Each Neuron Processor Executes at 250 MIPS



Use Processes as Intelligent Processors

- Processes can be used for Very High Level AI Engines
- More than 2,000 Processors per FPGA Chip (depends on LUTs)
- Typical Processes for High Level Functions might be in the Hundreds of AI Engines



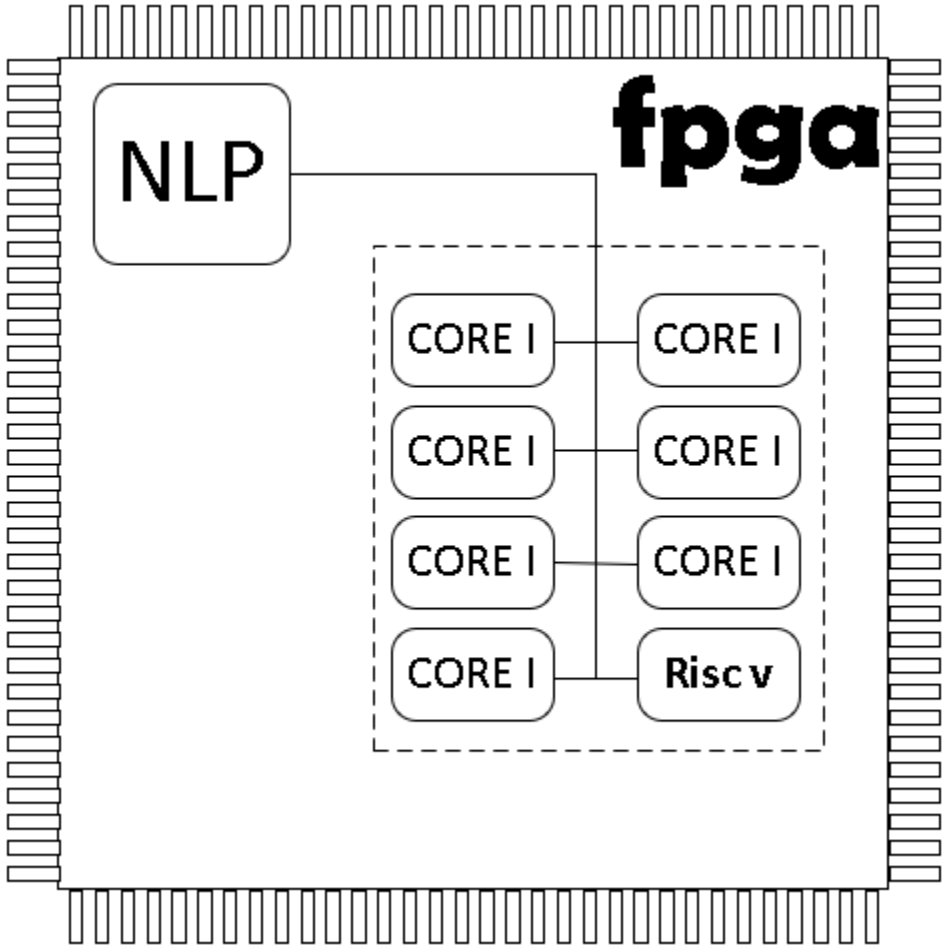
CORE I

System Verilog: ADD

```
 _plus : begin
busy = true;
--dp;
```

- `data_stack[dp] = data_stack[dp] + data_stack[dp+1];`
- `busy = false;`

```
end
```



Easy to Create YOUR Own Op-codes

CORE I

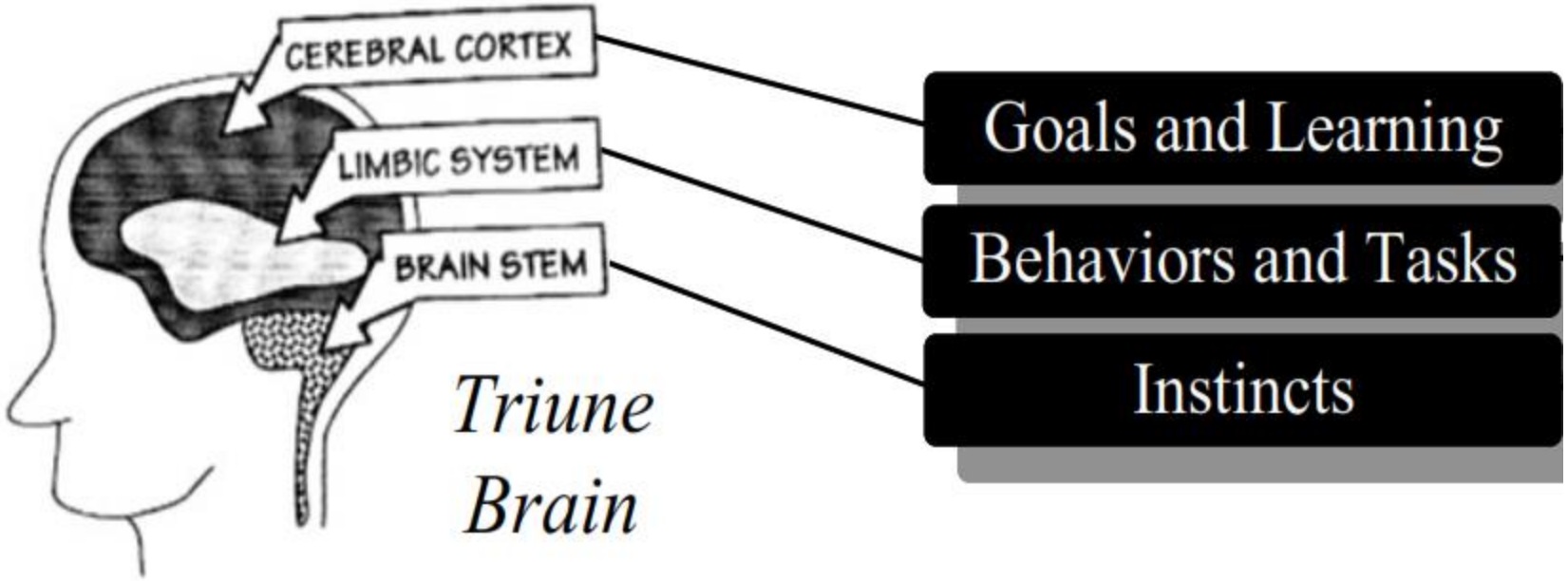
Massive Parallelism

```
Generate
  for (i=0; i < 1000; i++) begin
     function int update_neurons();
        for (int j = 0; j < HIDDEN_LAYERS; j++) begin
          for (int z = 0; z < NUM_NEURONS; z++) begin
                hidden_neuron_regs[i].synapse = input_neuron_regs[i].synapse;
                hidden_neuron_regs[i].weight = hidden_neuron_regs[i].weight;
                hidden_neuron_regs[i].bias = 8'd50;
                hidden_neuron_regs[i][j].fire = 1'b1;
          end
        end
  endgenerate

     _update_neuron : begin
        busy = true;
        update_neurons();
        busy = false;
  end
```

Massive Parrallel Processing (MPU) will change Computers Science

Triune OS in Silicon



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

Simplifies Programming and Machine Learning

The CORE I AI Board Mimics Human Brain

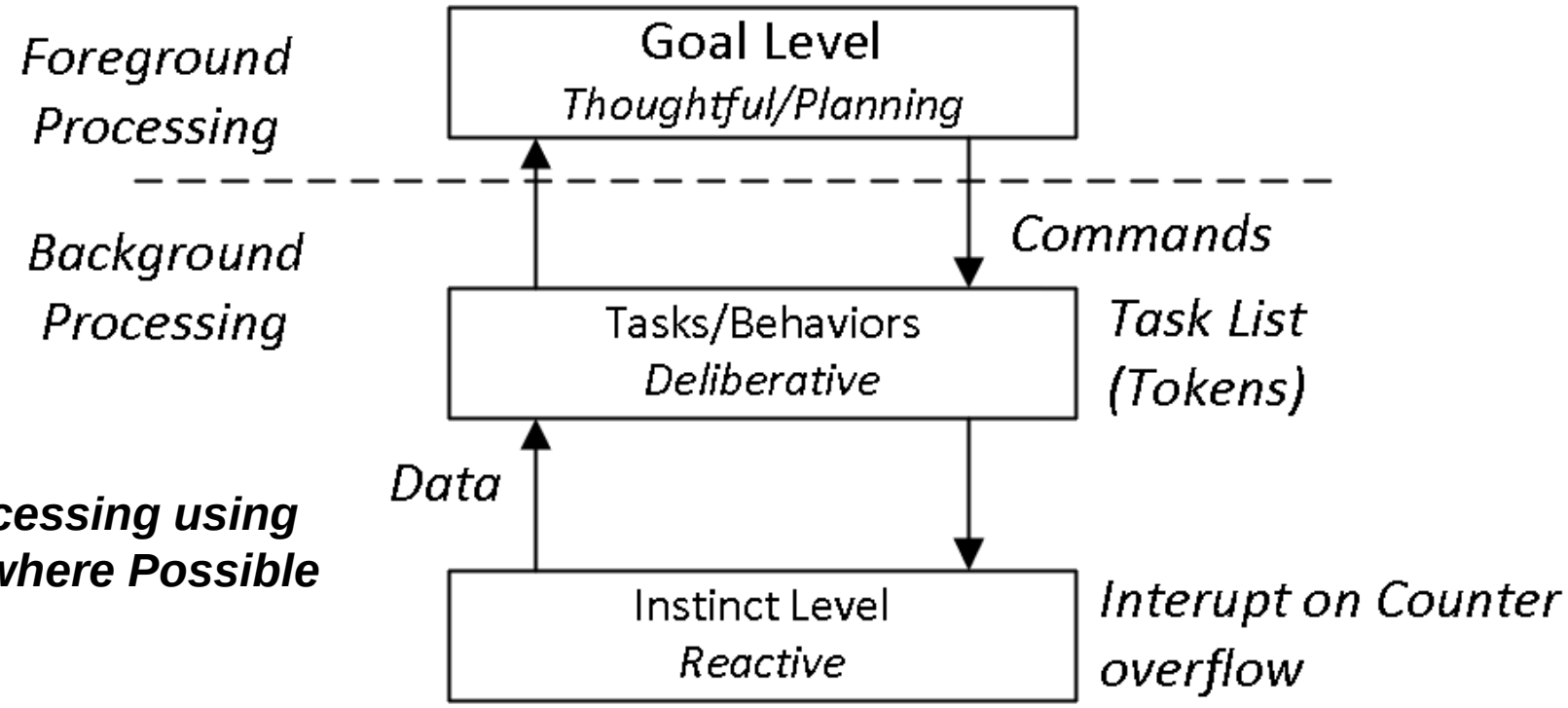
Triune OS in Silicon

- ***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

CORE I AI Playground Board Mimics Human Brain

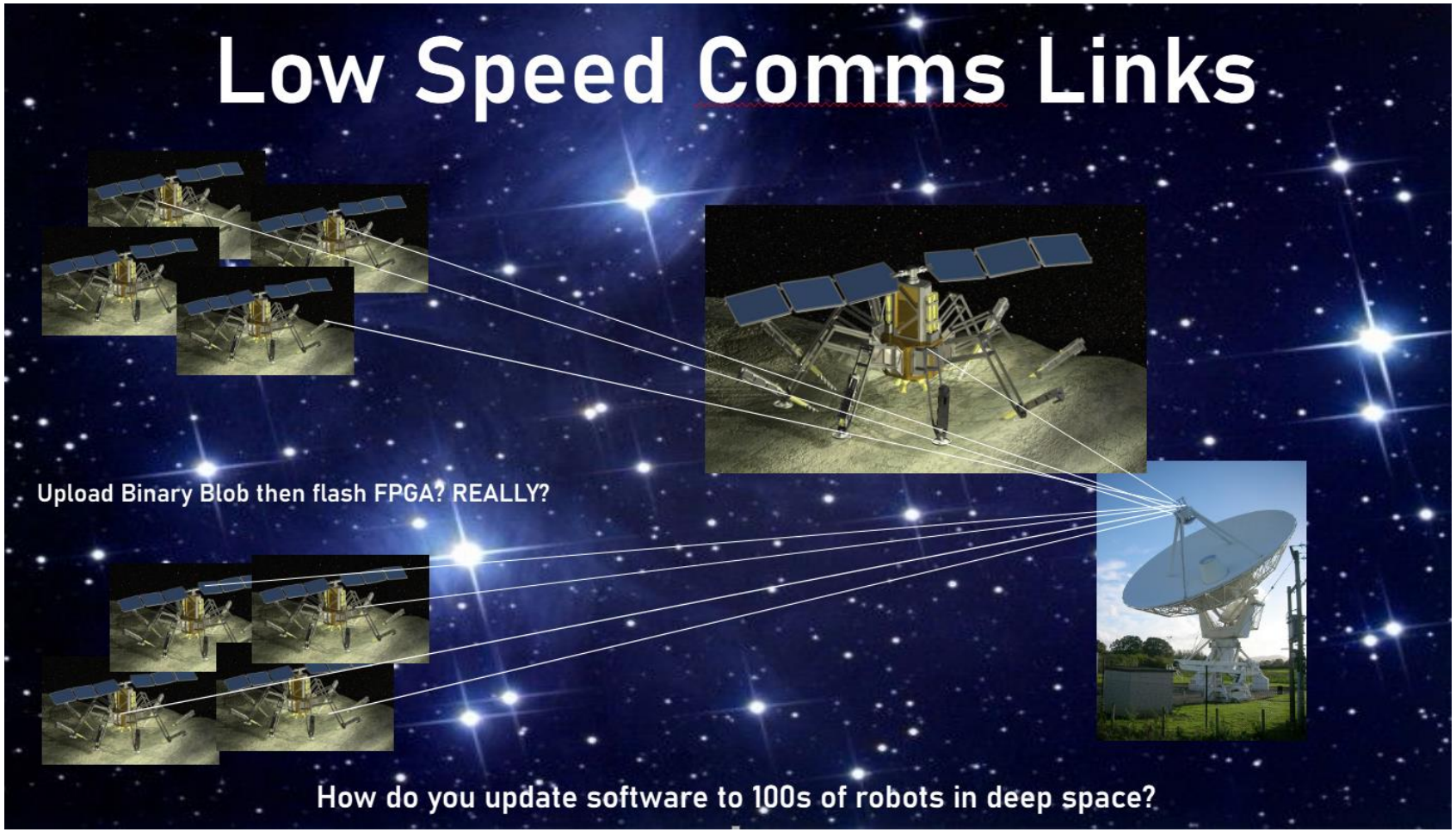
Triune OS Multitasking/AI Model



Use Parallel Processing using System Verilog where Possible

TOS Emulates Intelligence in Nature

CORE I Processor REQUIRED for Space Robotics

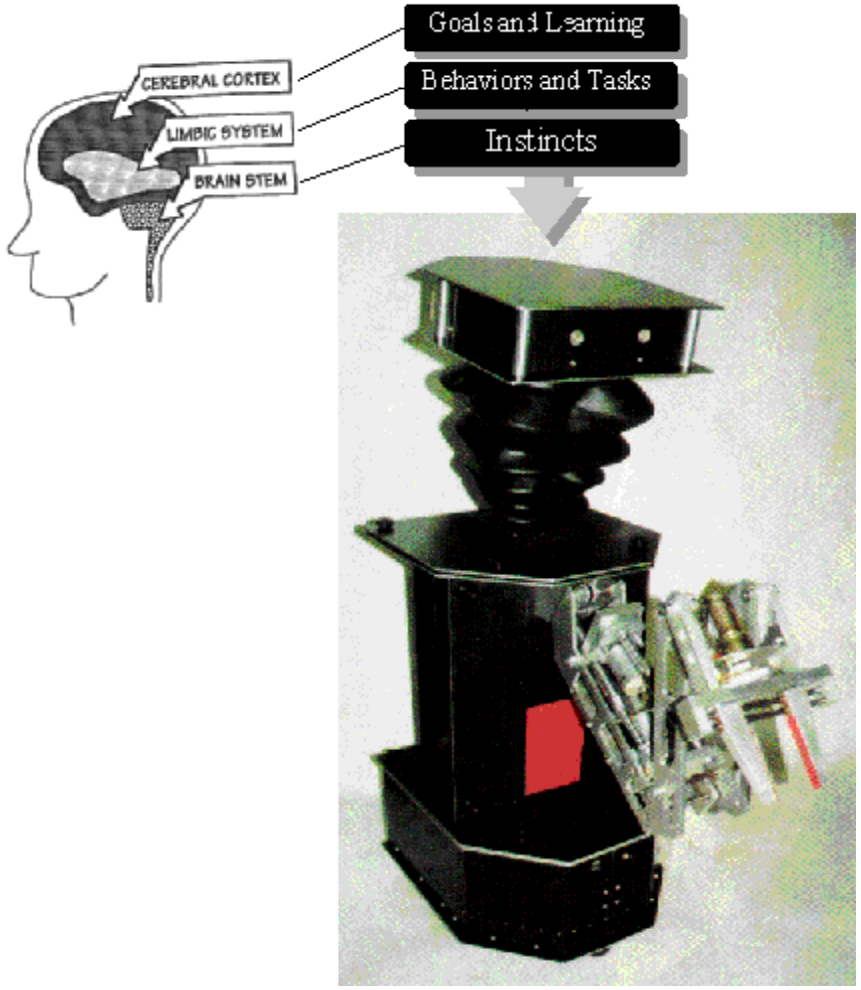


- Incremental Software Upgrades
- Computer does not Stop Execution While New Functions are Updated
- Updates can Occur in One Clock Cycle

Intelligent Drones

- Real-time AI
- Behavior Based Intelligence
- Motion Sensor Fusion
- Software Upgrades in Real Time

Home Assistant Robots

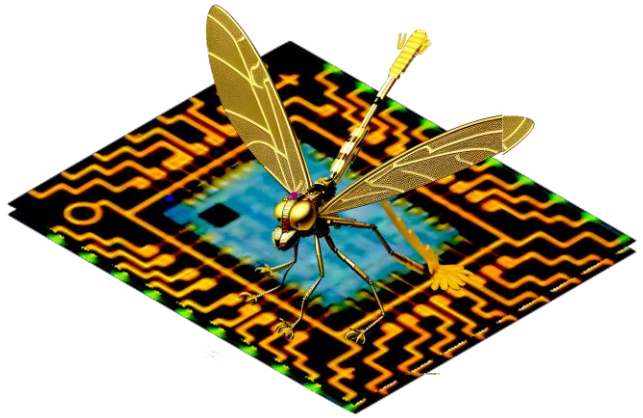


- Uses Wifi to Communicate with Online Chatbots, Medical, Police
- Boomers need Help
- Connected to Alarm System
- Call for Help
- Help get up from a fall

Help 10's of Millions of Seniors Live at Home

DragonFlys.ai

Contact Us



- Website: Dragonflys.ai

- AI & Robotics Group on LinkedIn
- https://www.linkedin.com/groups/12858138/

- AI & Robotics Group
- https://www.facebook.com/groups/1304548976637542/

- donaldrgolding@dragonflys.ai

- https://www.youtube.com/watch?v=QxDRVTXuJH8

If you are Interested in Intelligent Machines and AI IOT