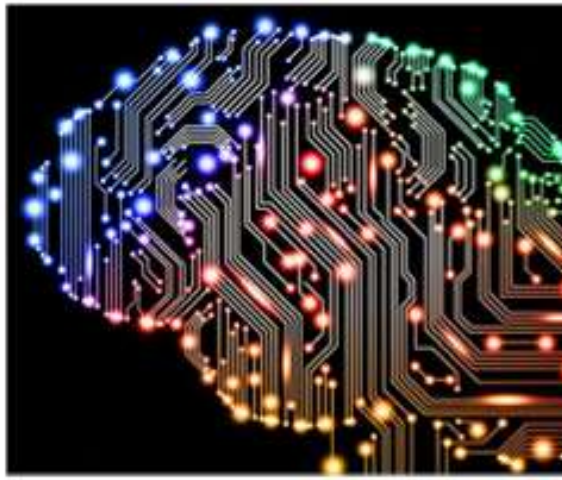
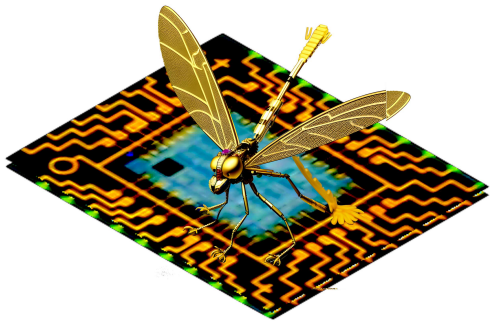
AI, Robotics + IOT Our Future World!





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

Intelligent Machines: AI IOT, Robotics, AI Comp

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

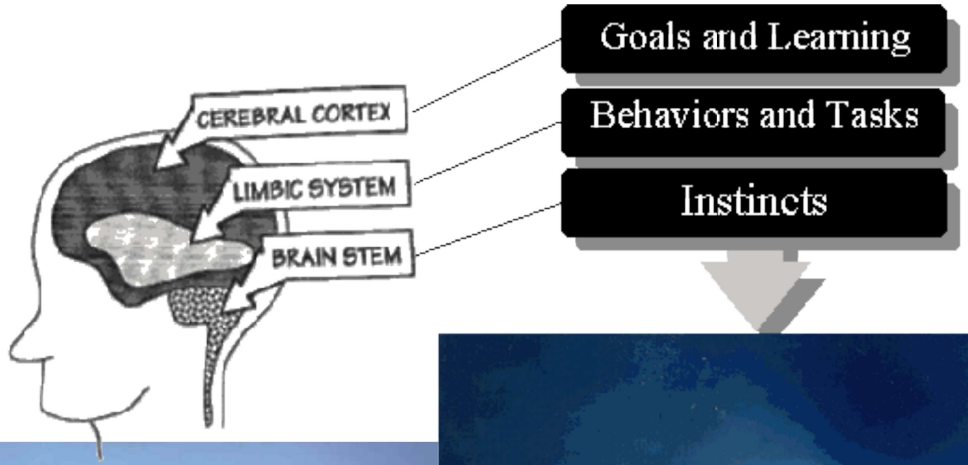
All Teraine 2.5 Ton Advanced Military Robot (2

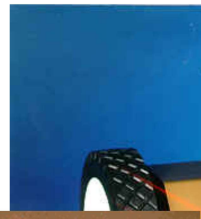
NASA Redwire Archinaut One Mission (OSAM

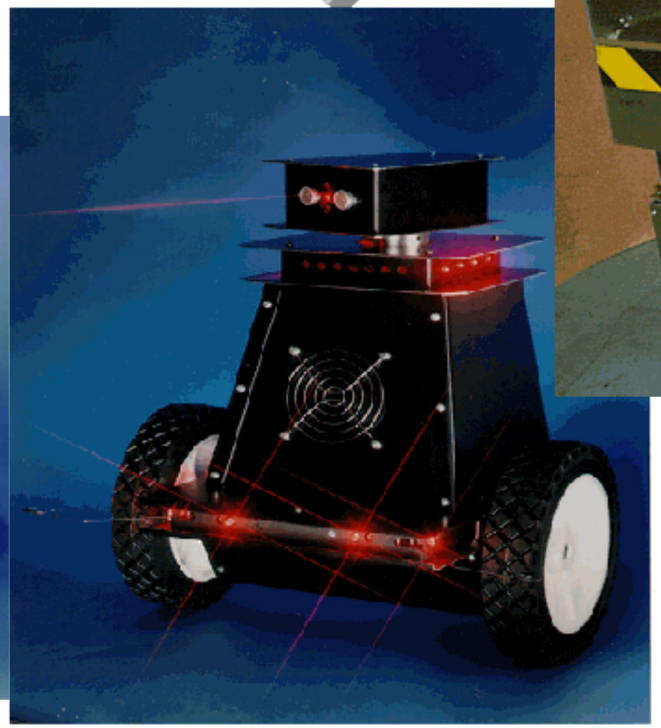
- 10 Meter Lattice Beam (100mmx100mm)
- Fully Autonomous
- Launch in 2024
- Designed the Rad Hard FPGA Command and Control Computer
- Redwire designed a Space Rated 3D Printing Filament
- Simulate the 3D Printing of a Solar Panel in Space
- Solar Film unrolls as Beam is 3D Printed

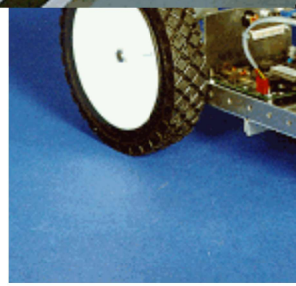
First NASA Mission to Build a Major Structure in

Angelus Research Corp. Don Golding - Founder/









Some of the Intelligent Robots DG Designe

CORE I Applications: IOT Devices

Personal Robots

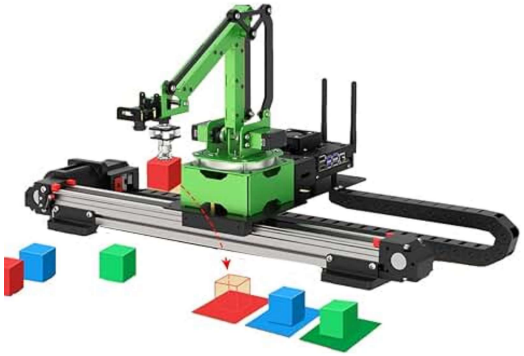
Surveillance Robots

Cell Phones

Drones

Don Xmas

WiFi BBQ Smokers



AI Experimental Robots

Home Security/Automation

Electric Cars

Hobby Bo

Internet Of Things (IOT)

Don Golding Owner
Group created: Jun 2023

Pending posts 0
Requests to join 0

Manage group

Edit group

Recent
- AI & Robotics Group
- SDNx Aerospace Space Rocke...
- Aerospace & Defense Showcase
- AI & ML Engineer, Data Scienti...
- Robotics, Mechatronics and A...

Groups
- AI & Robotics Group
- SDNx Aerospace Space Rocke...
- AI & ML Engineer, Data Scienti...

Show more

Events +
- Aerospace & Defense Showcase

See all

AI & Robotics Group

Public group

Earn an Active Group badge

Start a public post

Media Poll

All Recommended

Pinned by admin Unpin

Don Golding (He/Him) • You
Hardware AI, Space Avionics/Robotics Engineering (SR EE)
Admin • 3d • Edited •

Try Llama yourself...note the upper right hand side drop down list. Choose the model you want to try. Note the code Llama choice, great for writing programs...https://lnkd.in/eX3XESeY

Like Comment Repost Send

936 impressions View analytics

2,997 members
Including Michael Yampol and 1,196 connections

Invite connections

Show all →

Analytics
Last 15 days activity

4,283
Active members

243
New members

62
Posts

15,271
Post views

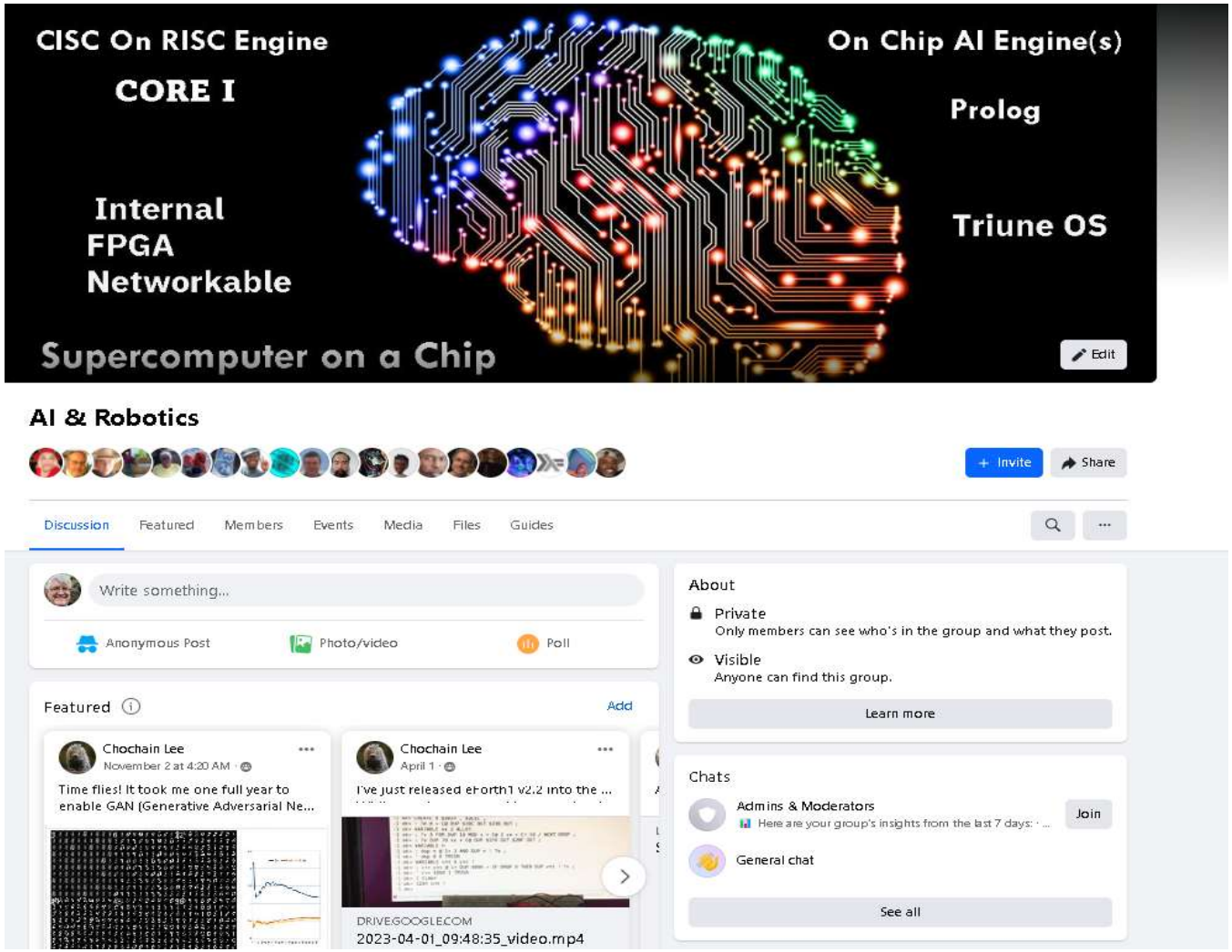
Show all →

Admin

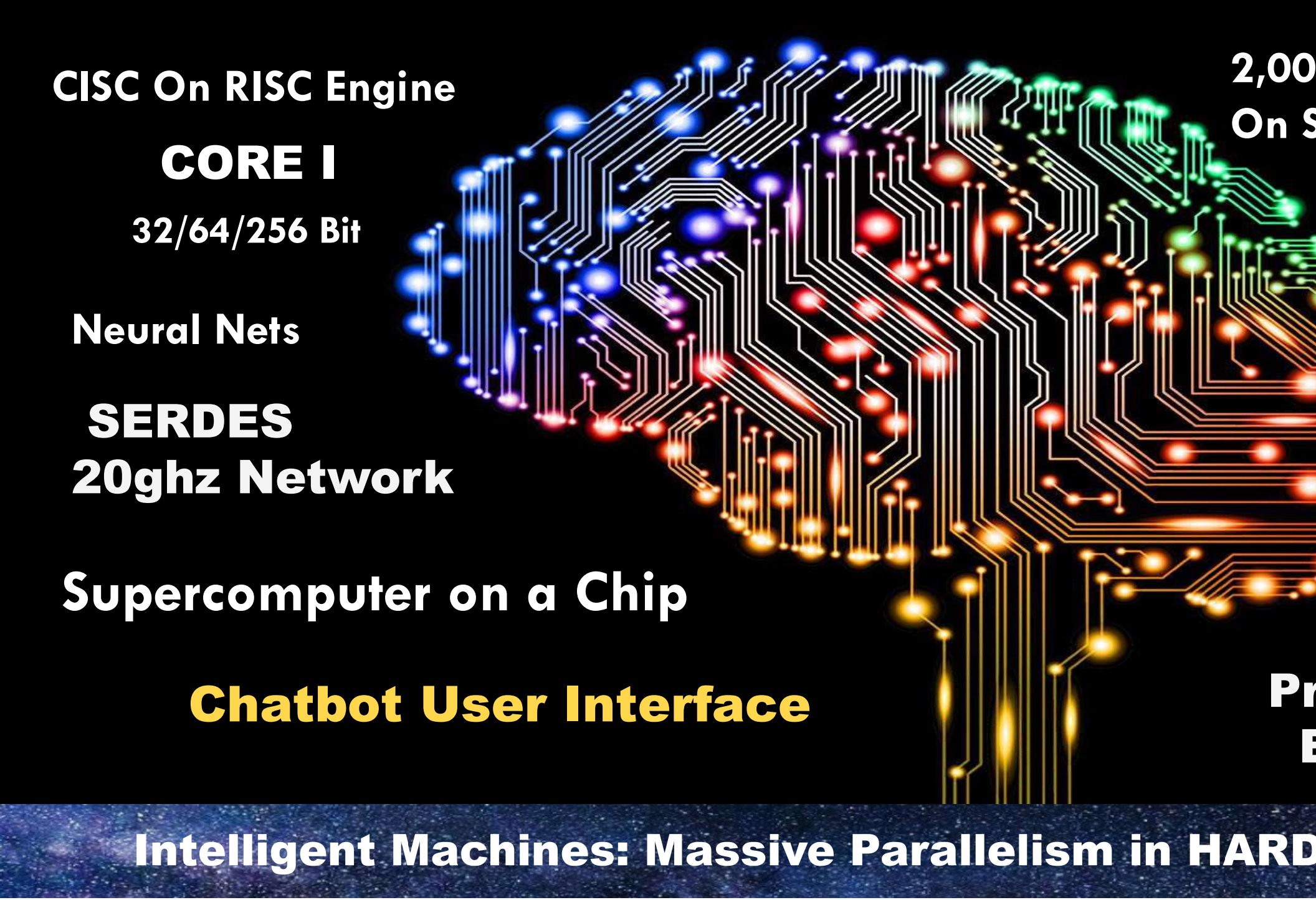
Don Golding
· You Owner
Hardware AI, Space

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

Facebook: AI & Robotics (356 Mem



https://www.facebook.com/groups/1304548976633


CISC On RISC Engine
CORE I
32/64/256 Bit
Neural Nets
SERDES
20ghz Network
Supercomputer on a Chip
Chatbot User Interface
2,00
On S
Pr
E
Intelligent Machines: Massive Parallelism in HARD

Why Forth?

- Forth was Invented in 1963 by Charles Moore
- Forth has used a Chat-bot Interface since the very beginnin
- 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 Langua
- Forth was 60 Times faster than Python Parsing Strings – Dr

Why not use C?

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

CORE I is Efficient Utilize >90 Percent of Fabr

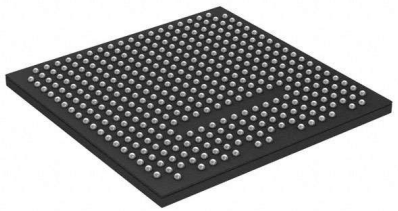
Forth in Hardware History

- Forth is a Virtual Machine, Forth is really a CHIP design!
- 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 ava
CPU opcodes are actually used.

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

CORE I is Efficient Utilize >90 Percent of Fabr

CORE I FPGA Benefits

- Fully Programmable 32 bit(64,128?) Computer on a single C
- 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 Fo

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 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, 12
- Network multiple 32 bit Forth Engines together over internal netwo Internet as a VPN
- Store source code in Flash or on disk, compile: on-the-fly as neede
- 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 Sy
- 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 PyTor state dictionary is a Python dictionary that contains the state of a PyTorch model, including 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 –
- Create Parallel Processing Device Drivers for Peripherals
- Send Source Code over Remote Internet or RF Link, Compi

  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

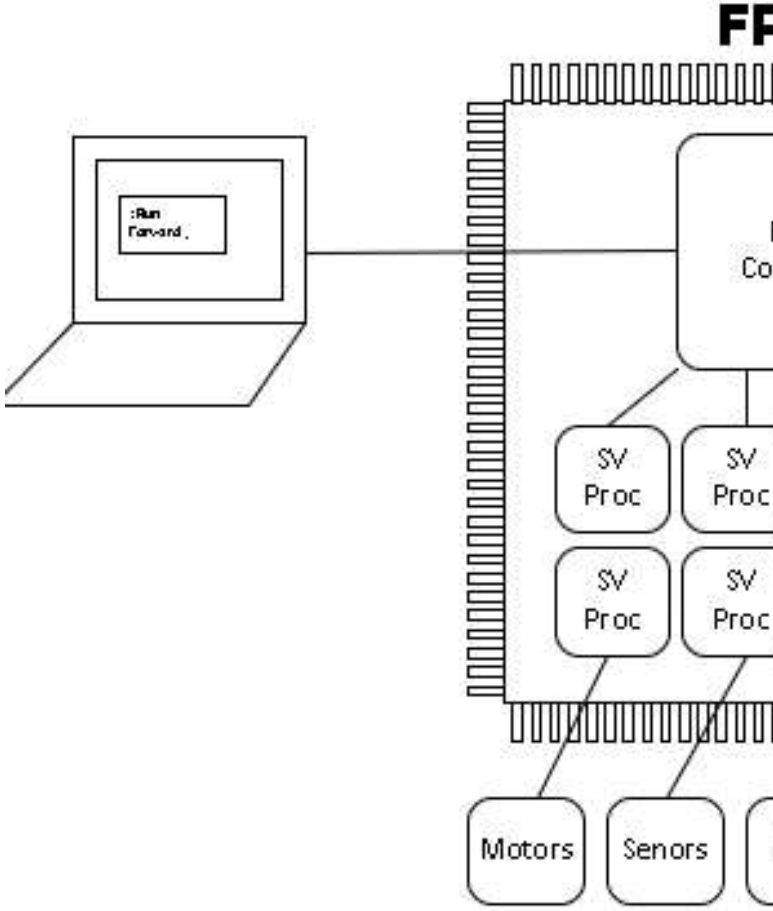
Forth + Syste

Easier to learn

Chip Level Speed, Easy to Program and Deb

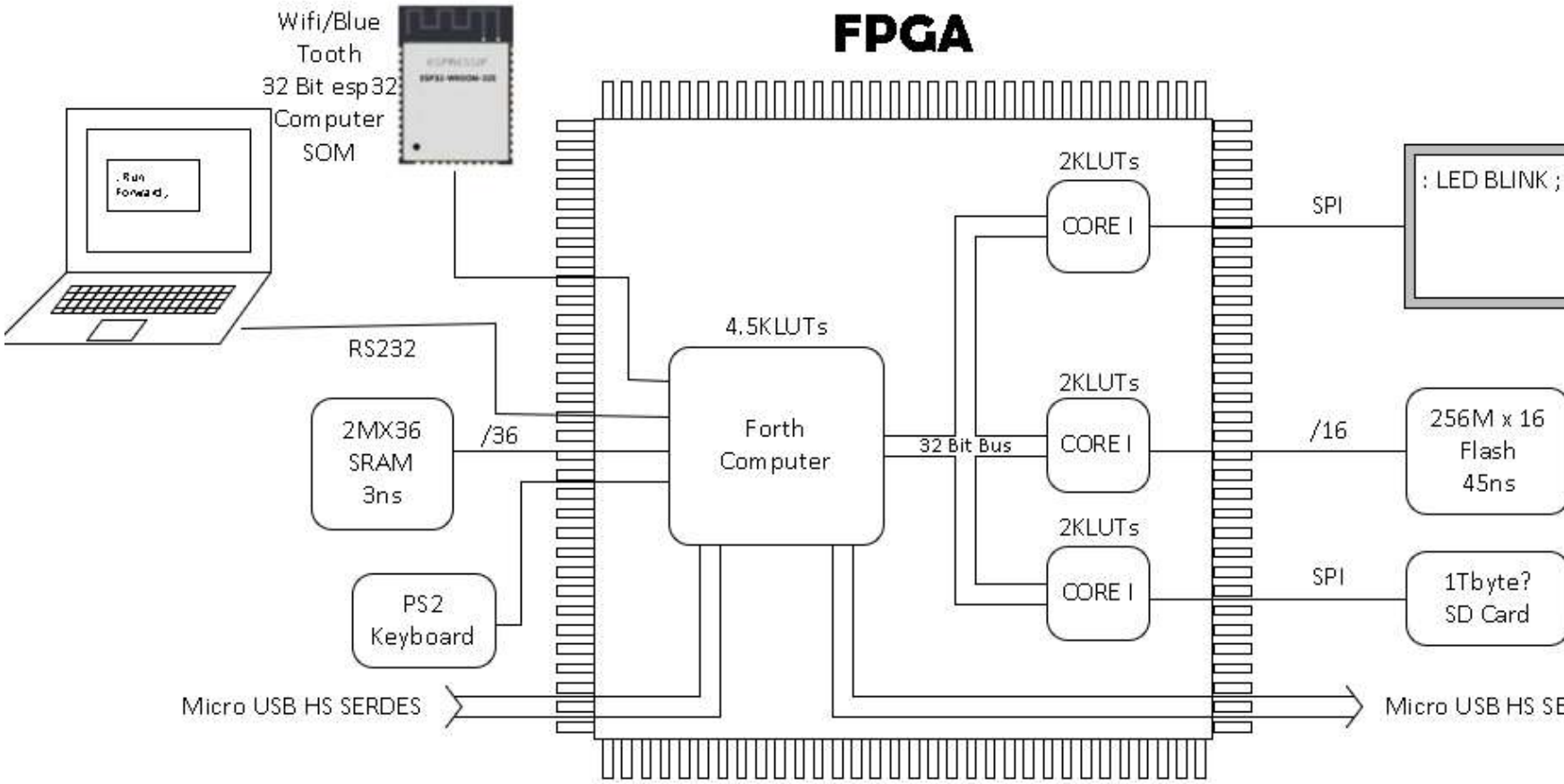
Forth + System Verilog

- 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

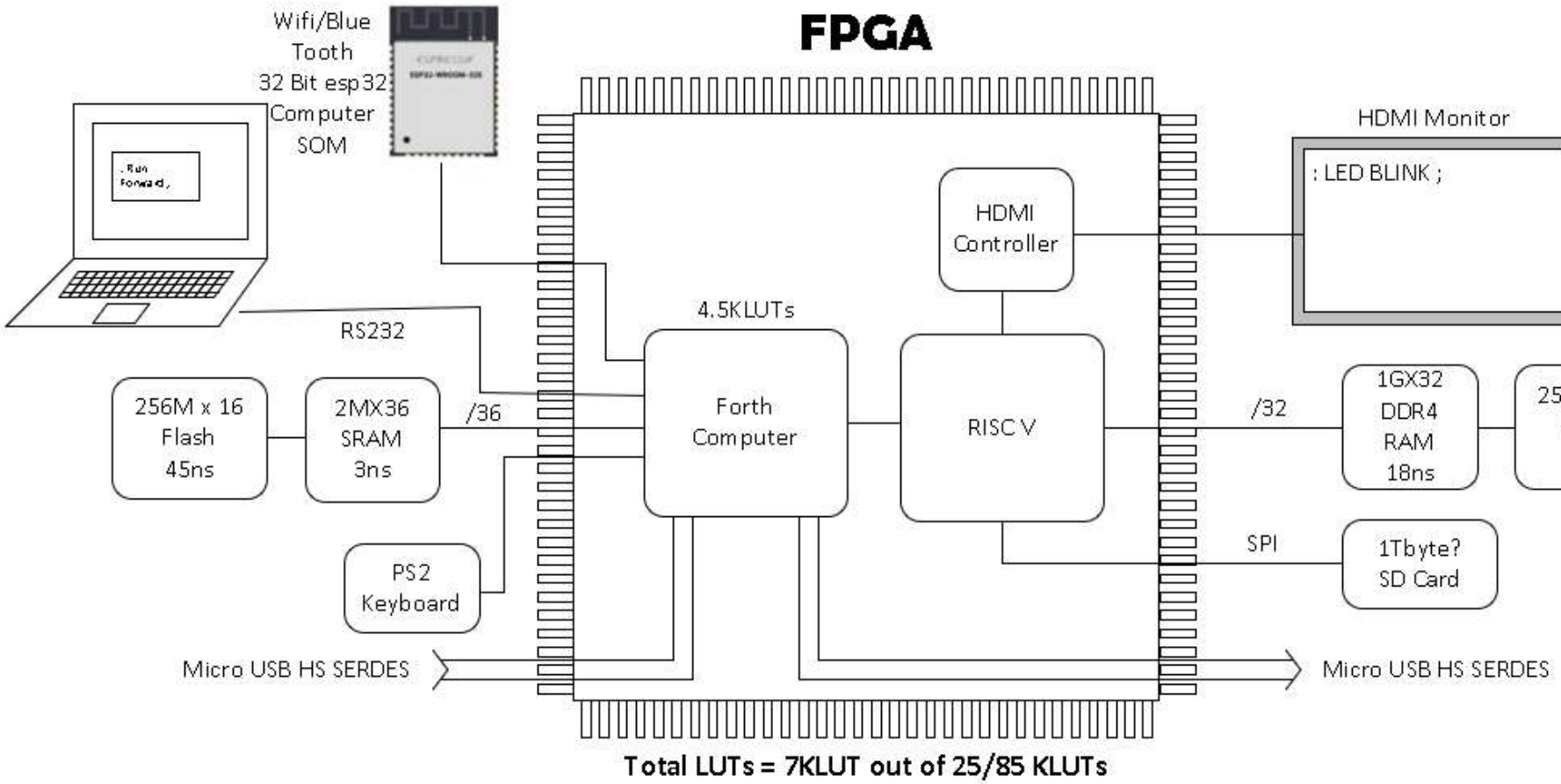
CORE I AI Playground Block Board



Advanced System: Total LUTs = 10.5KLUTs

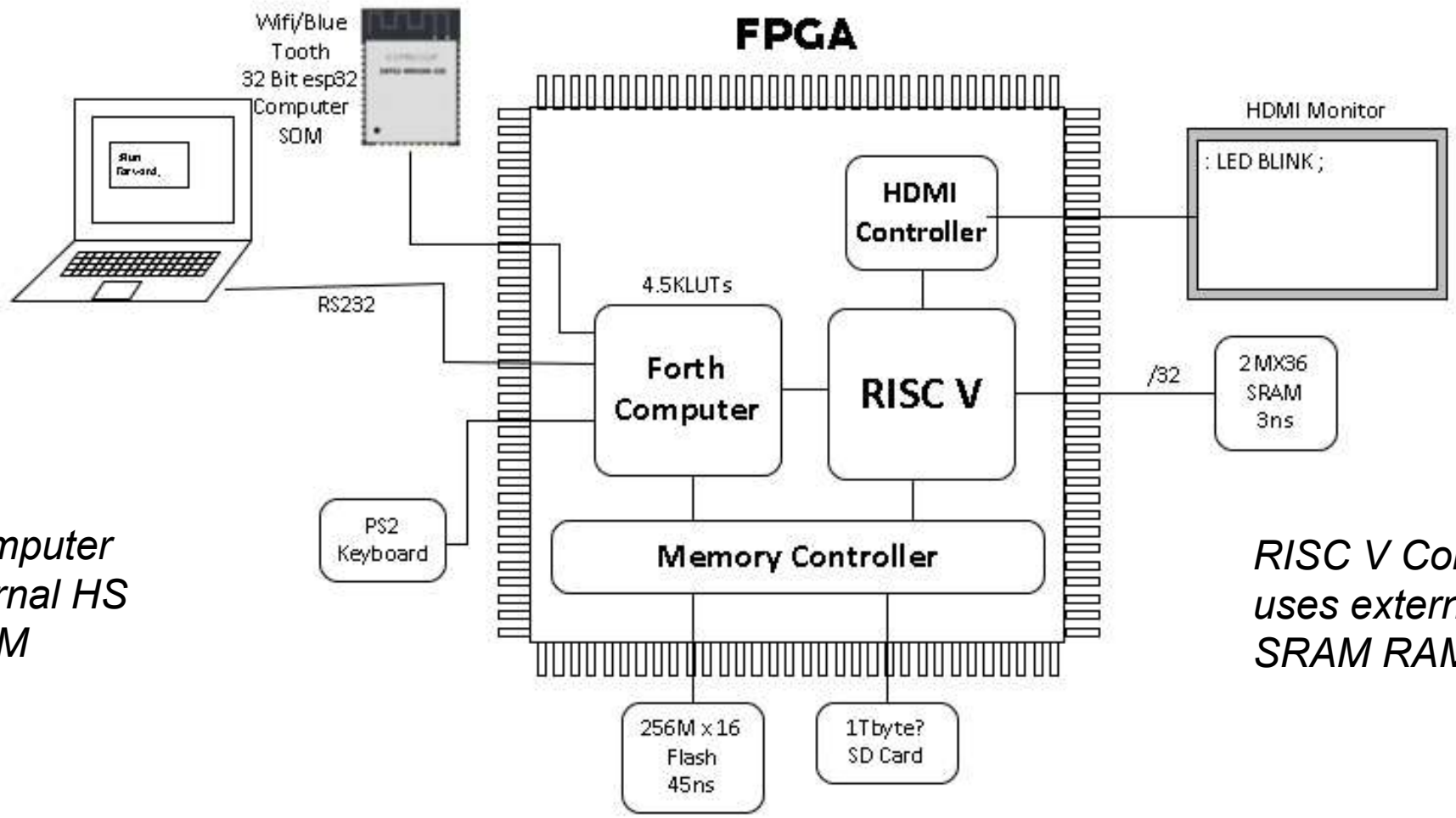
YOU can Experiment and Develop Computer Archit

CORE I + RISC V on a Single Chip



YOU can Experiment and Develop Intelligent Mac

CORE I + RISC V on Current AI Playground B

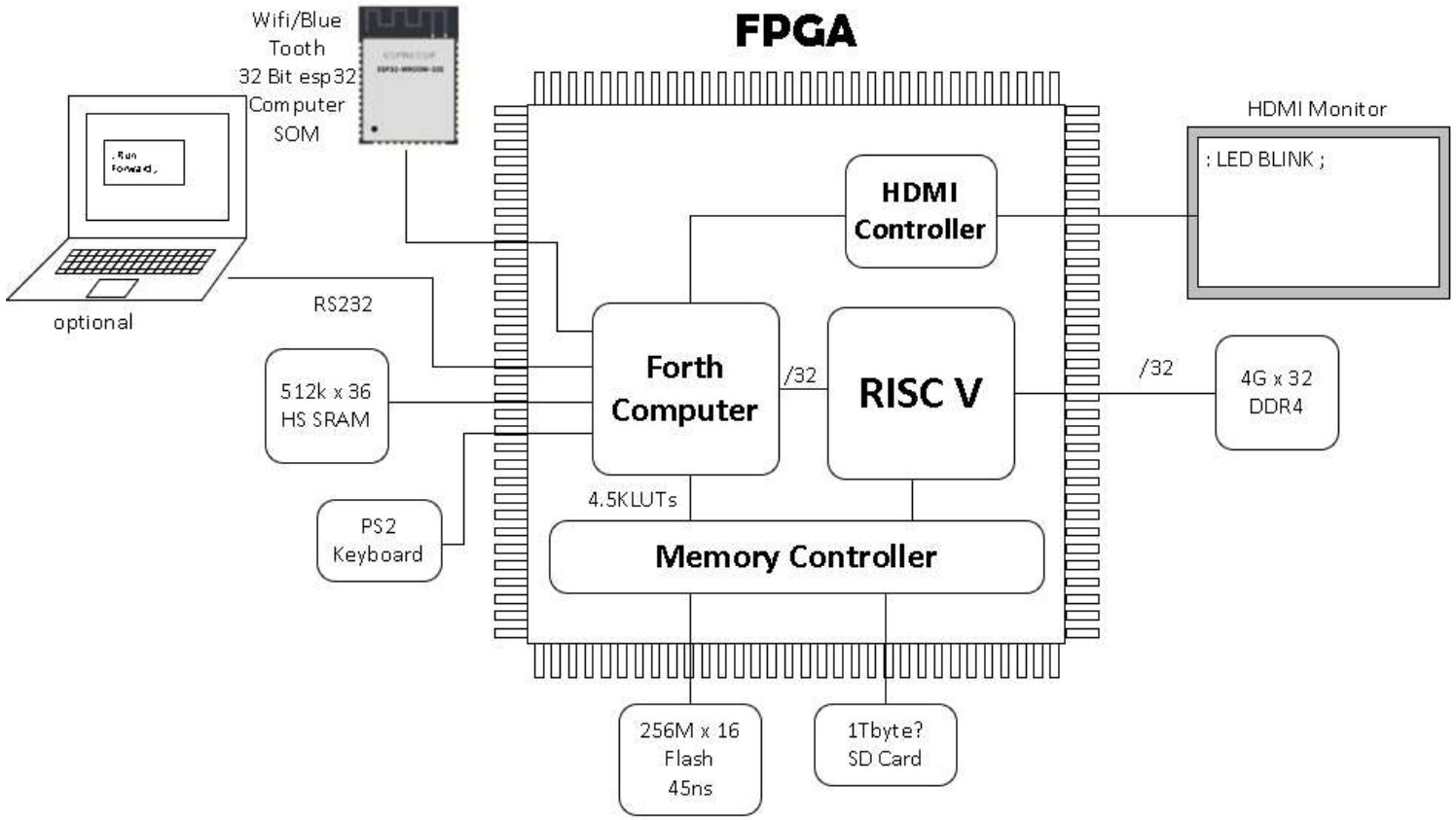


Forth Computer uses Internal HS Block RAM

RISC V Com uses externa SRAM RAM

Forth, CORE I, RISC V on Current Board

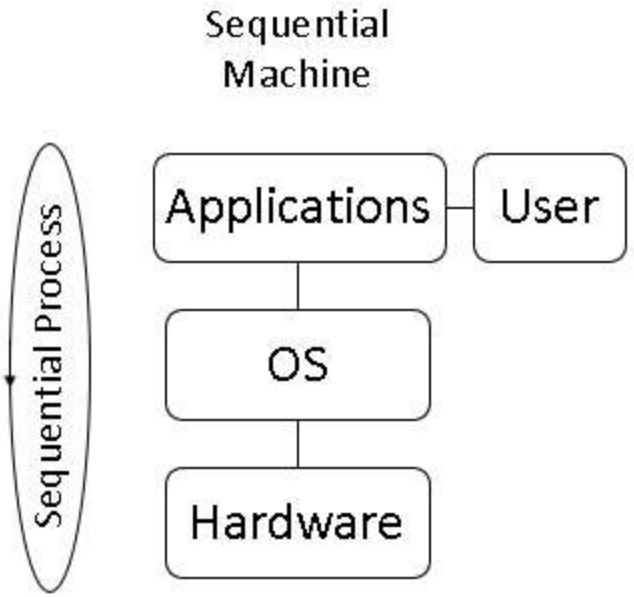
CORE I + RISC V on a Future Development B



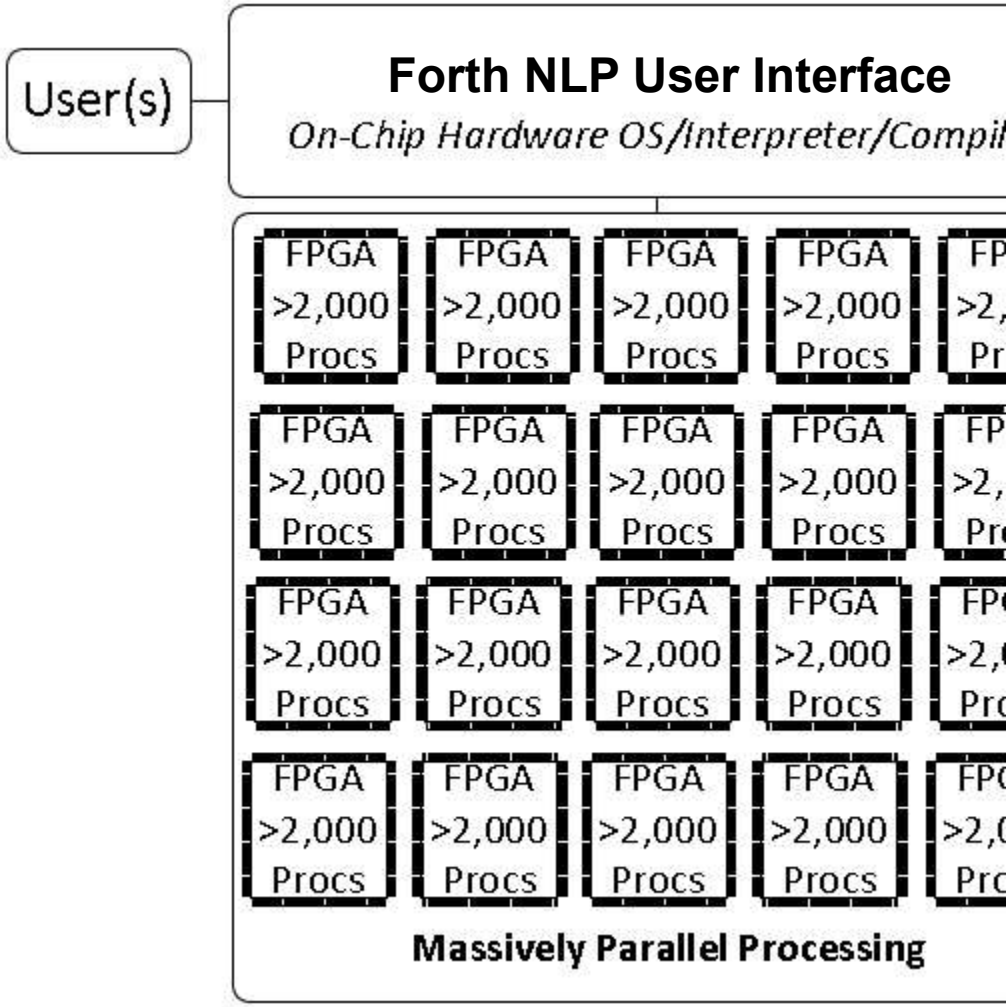
It could use the more Powerful Microchip Polarfire FPGA with 481 KLUTs, 33,792,(
of SRAM, and 584 GPIO pins, more parallel memory can be added to the des

Forth, CORE I & RISC V: Integrate C Based AI C

Sequential vs Parallel Computing



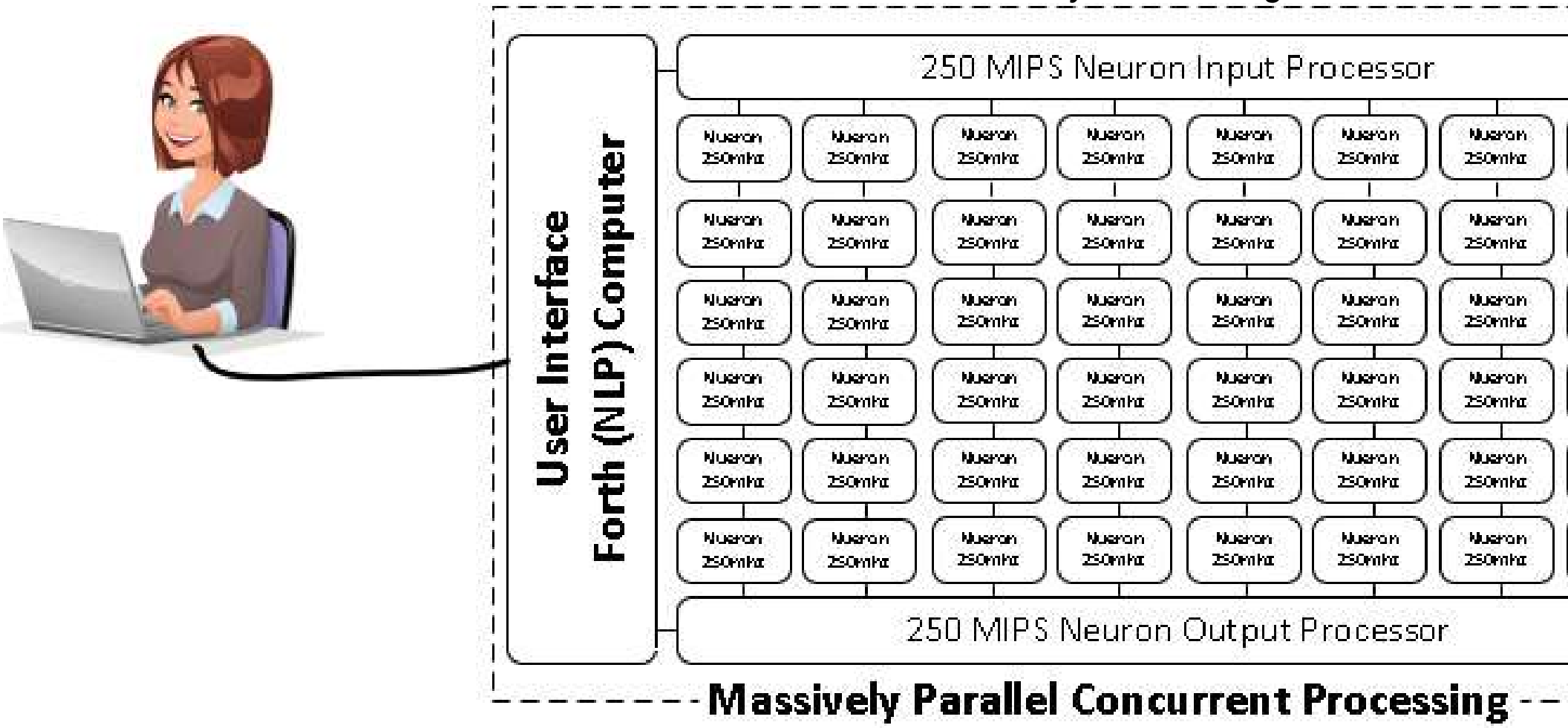
WinTel/Apple/Linux



Massive Parallelism vs Today's Sequencial Com

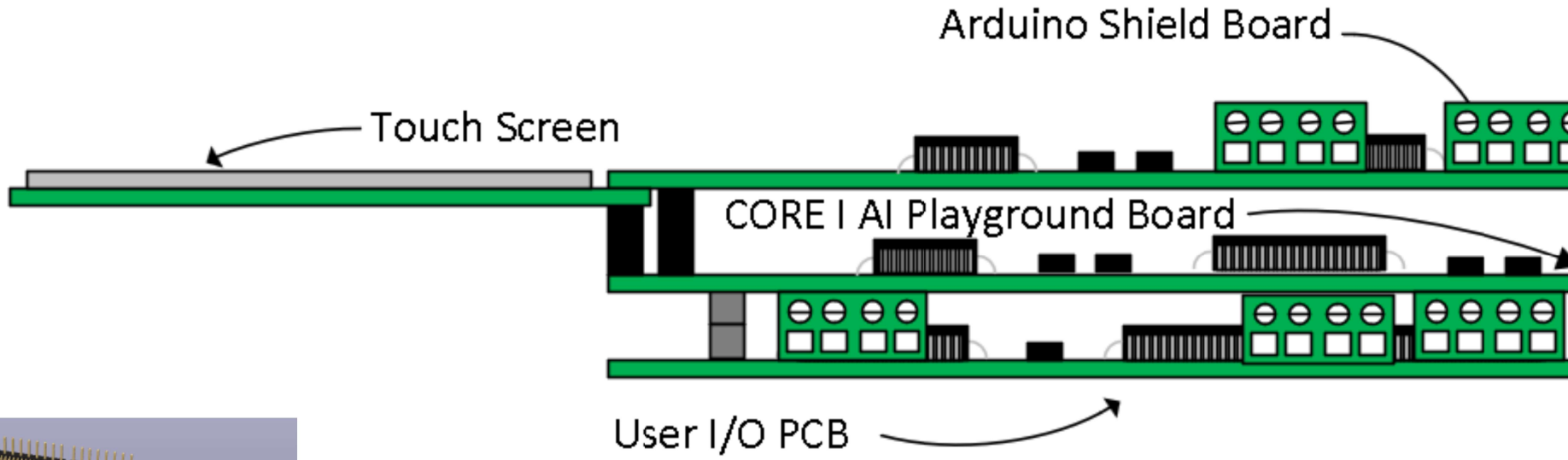
CORE I Massively Parallel Computer

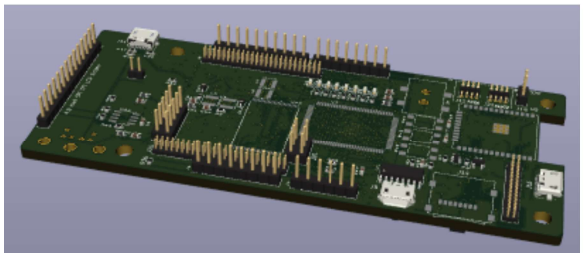
Coded in System Verilog...



Massively Parallel Computer

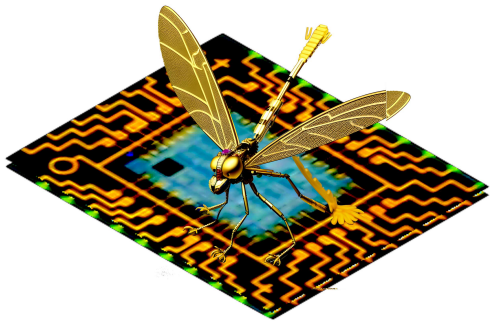
CORE I AI Playground Board



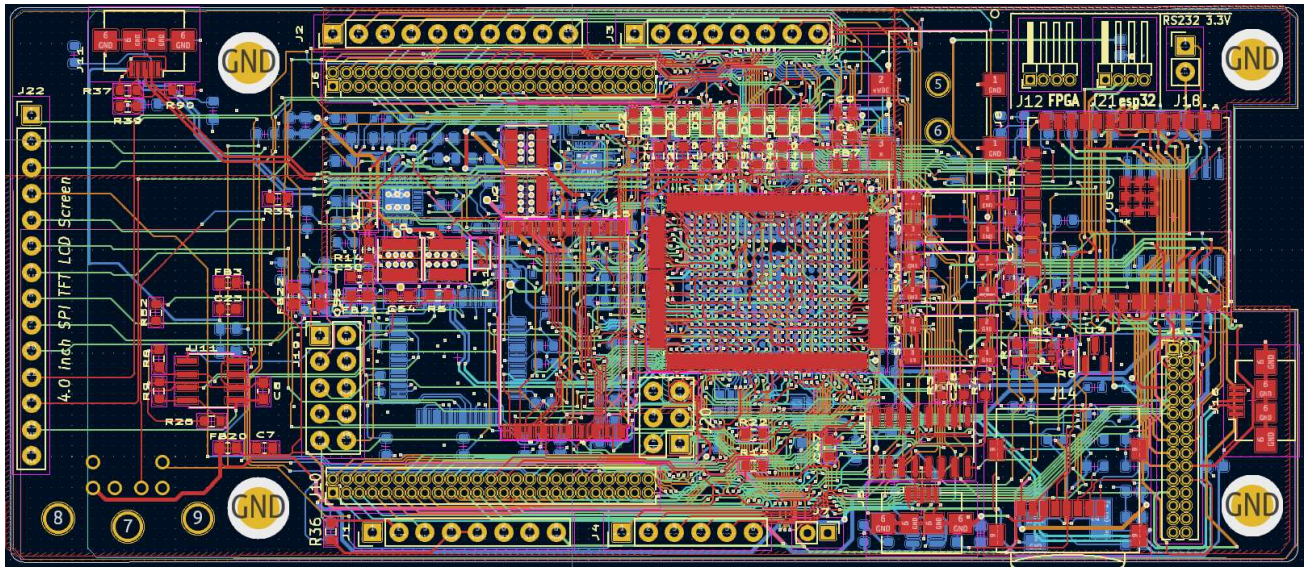


YOU can Experiment and Develop Intelligent Mac

CORE I AI Playground Dev Board - 1st Produ



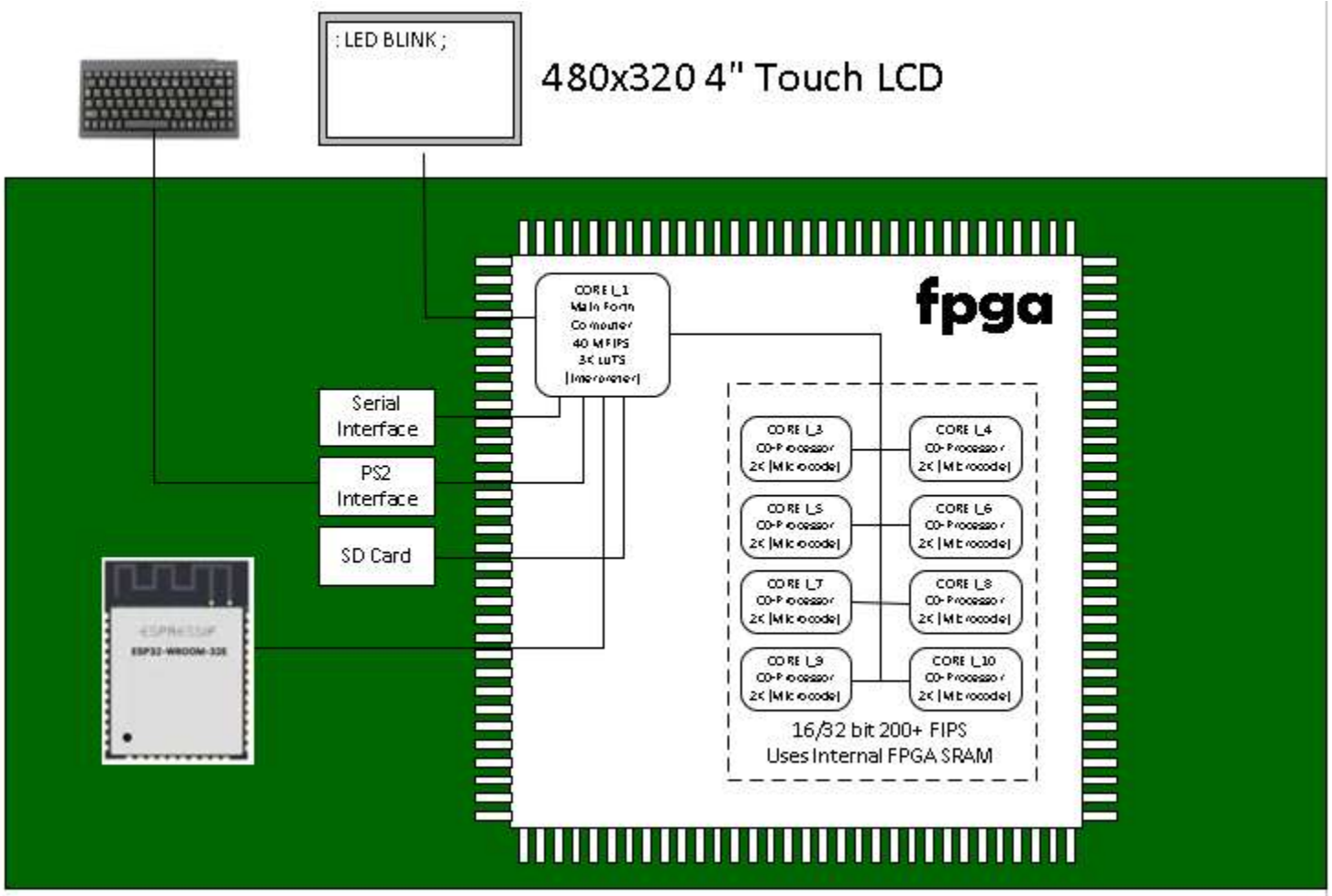
Very Low Power
Runs on Batteries



- Demonstrates COI Technology
- Encourage Early A
- Build Community o Developers
- Benchmark Massiv Computer vs PC

Develop Your Own AI based IOT Products or Just Exp

AI Playground Board Block Diagram



Esp32 8M PSRAM +
400K SRAM
Wifi + Bluetooth

LFE5
F

85KL
Spe

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 LUT
- 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
- Create Input, Output, Semaphore Registers for Forth NLP
- Create Opcodes to update these Registers

Conclusions/Benefits

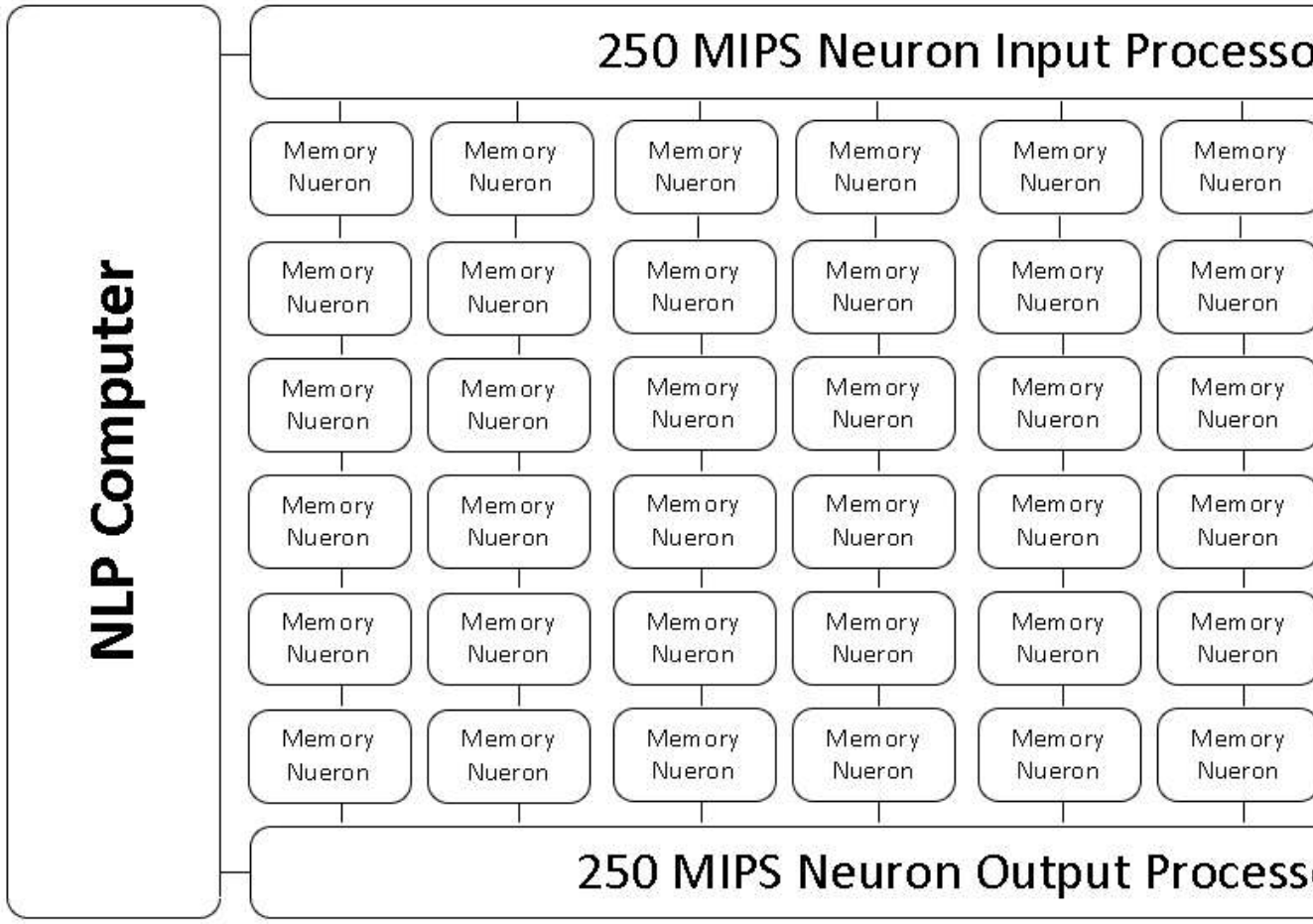
- Build High Performance/Low-Cost “Intelligent” Embedded
- Research Novel Forth based Computer Architectures
- Research/Develop AI on Top of Forth
- Add High-Level constructs to Forth the language: ENUM, St
- Easy Way for beginners to Learn About FPGAs
- Use the Serial Port and Forth to Debug System Verilog Pro
- 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 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, 12
- Network multiple 32 bit Forth Engines together over internal netwo
- Store source code in Flash or on disk, compile: on-the-fly as neede

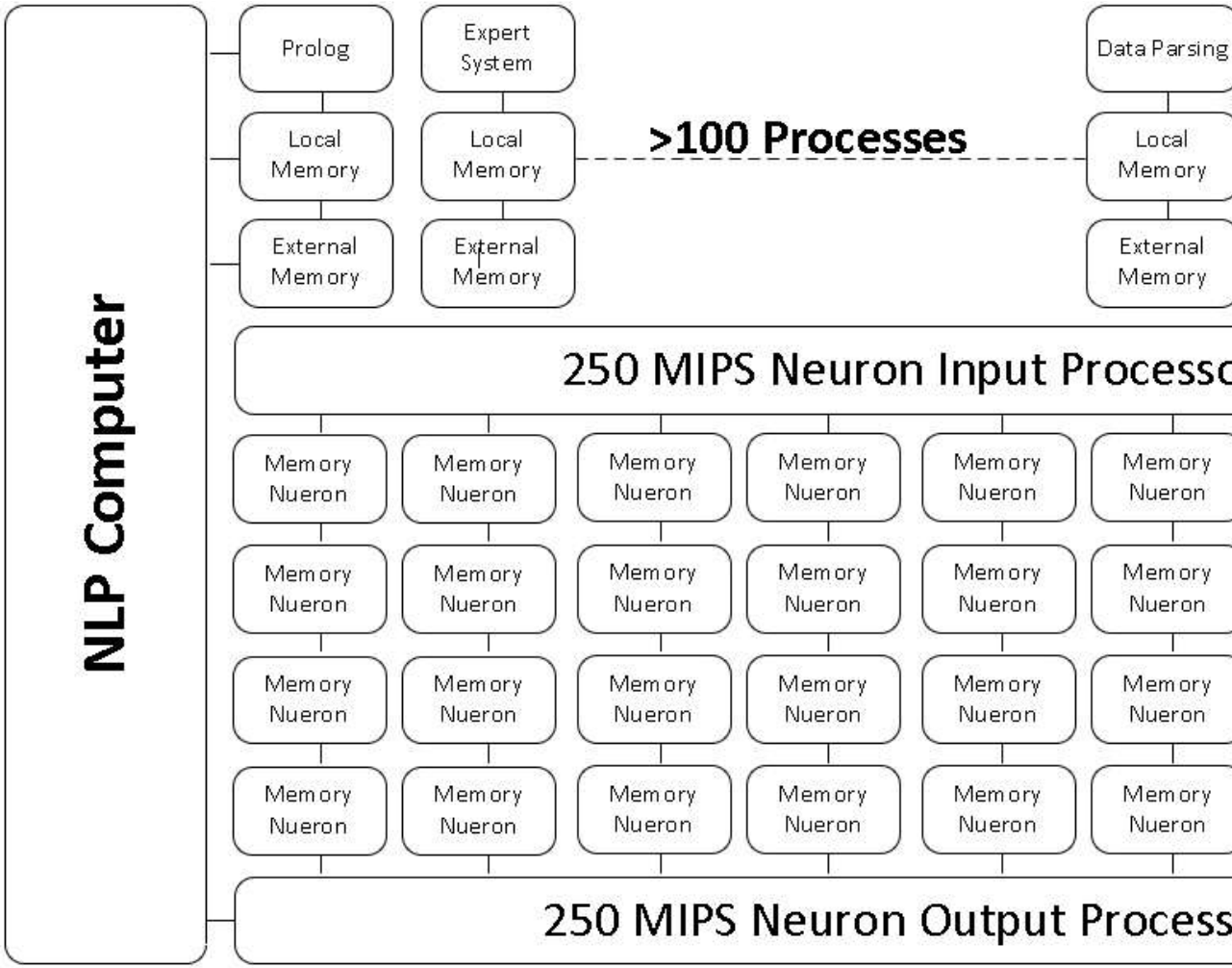
Digital Neural Networks

- 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



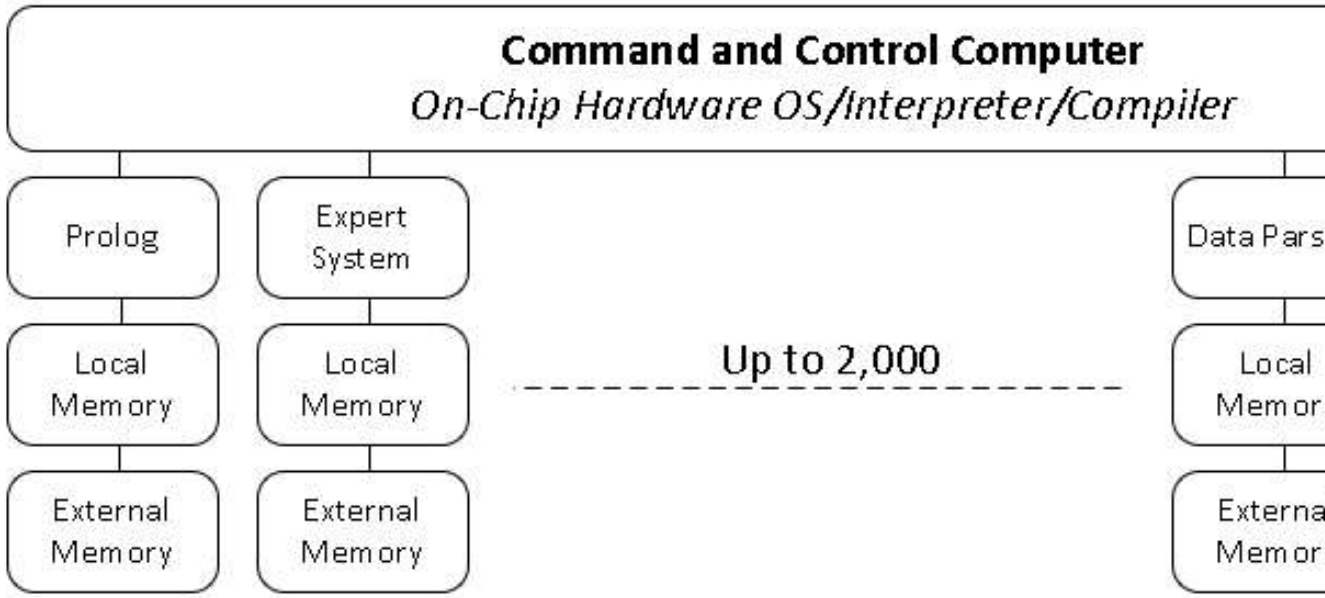
Combine Expert Systems/Neural Networks

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

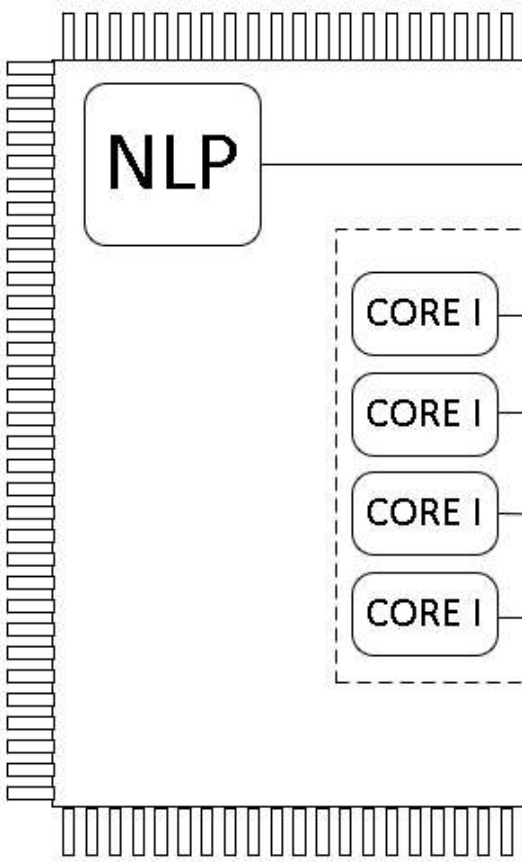


Use Processes as Intelligent Processors

- 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



```
_plus : begin
        busy = true;
       --dp;
•      data_stack[dp] = data_stack[dp] + data_stack[dp+1];
•      busy = false;
end
```



Experiment with novel comp

Easy to Create YOUR Own Op-codes

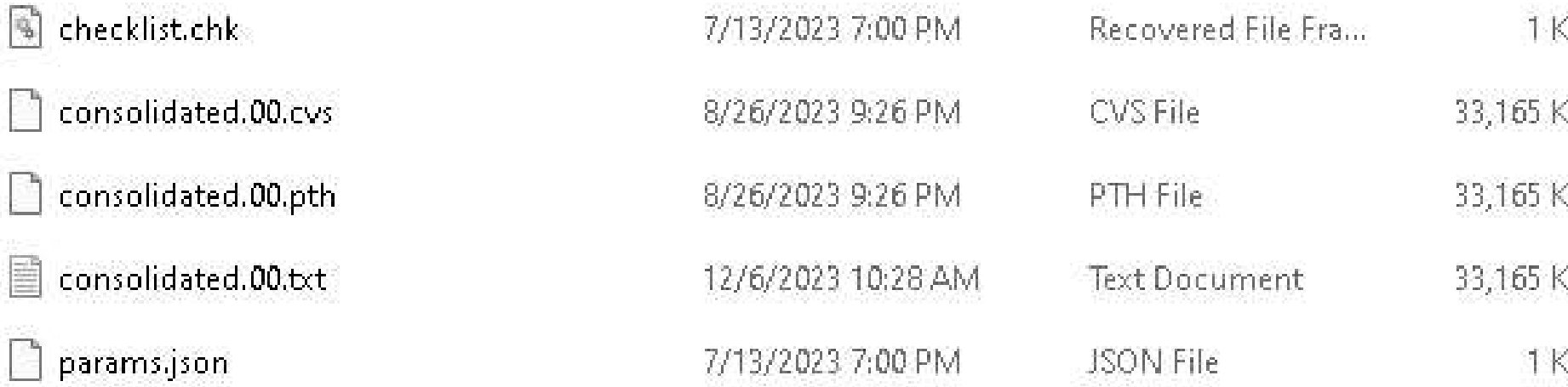
```
Generate
  for (i=0; i < 1000; i++) begin
    function int update_neurons();
      for (int j = 0; j < HIDDEN_LAYERS; j++) begin
        for (int z = 0; z < NUM_NEURONS; z++) begin
          hidden_neuron_regs[i].synapse = input_neuron_regs[i].synapse;
          hidden_neuron_regs[i].weight = hidden_neuron_regs[i].weight;
          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
```

CORE I Integrate Lla

- Store Llama 7B Dataset in Flash Memory
- Create the Inference Engine to Access/Parse Dataset in S
- Using Llama 7B as the base, “Fine Tune” for Personal Assis
- Optionally add RISC V Co-Processor – Use PyTorch Codi
- Build Internet Search Engine on esp32 Co-Processor

CORE I Hardware Integrate

- 7B Llama dataset fits into a single 100 Mbyte Flash Memory Ch
- This Memory Chip has a Dedicated I/O Pinsbuilds your
- Inference Engine written in System Verilog
- This builds your AI Chatbot Machine
- Use “Fine Tuning” to create your Application Specific AI Comput



checklist.chk	7/13/2023 7:00 PM	Recovered File Fra...	1 K
consolidated.00.cvs	8/26/2023 9:26 PM	CVS File	33,165 K
consolidated.00.pth	8/26/2023 9:26 PM	PTH File	33,165 K
consolidated.00.txt	12/6/2023 10:28 AM	Text Document	33,165 K
params.json	7/13/2023 7:00 PM	JSON File	1 K

Massive Parallel Processing will change Computer S

Natural language Processing

Fo

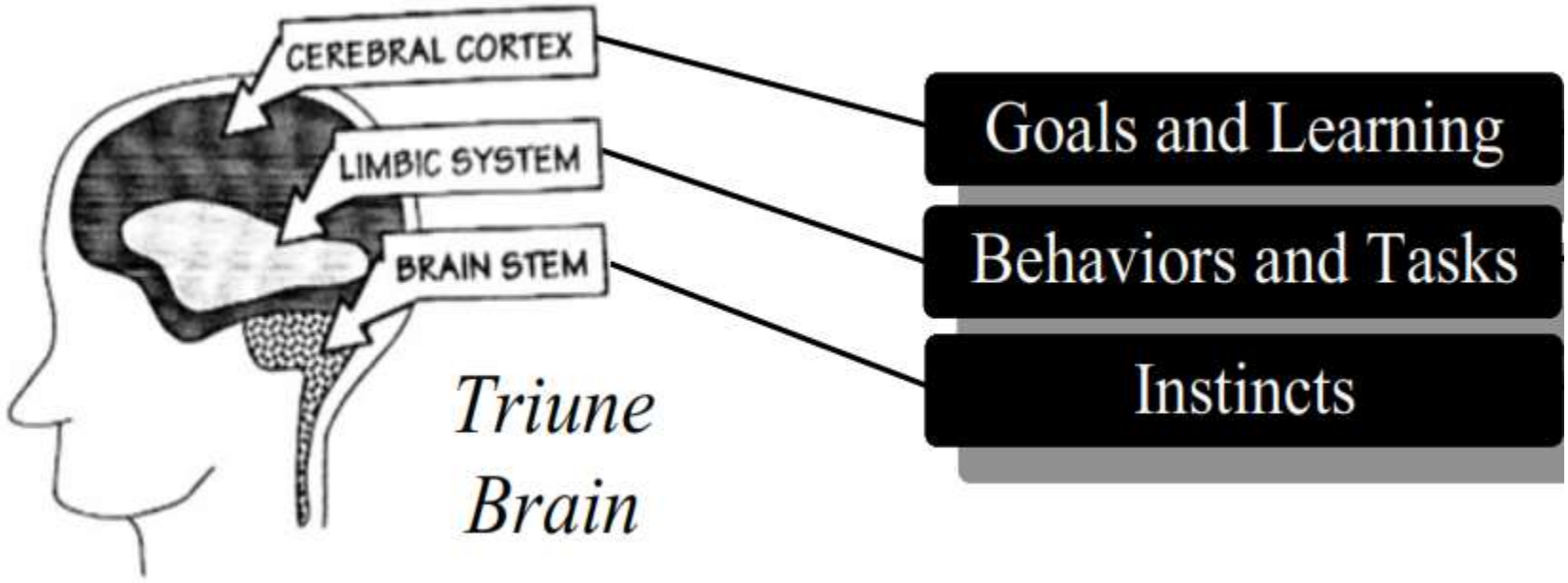
- Forth 2.0 Conversational Forth
- Talk to your Computer in English
- Forth is already a Programmable Chatbot
- Interpreter Front end (Chatbot Interface)
- Incremental Compiler (On the fly!)
- New: Named Local Variables
- No Stack Manipulation Words Required
- Higher Level Language Built on top of Forth
- Built with Voice Input in mind
- Hardware Optimized on FPGA

Courte

PS: B

Anyone can Program their Computer!

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

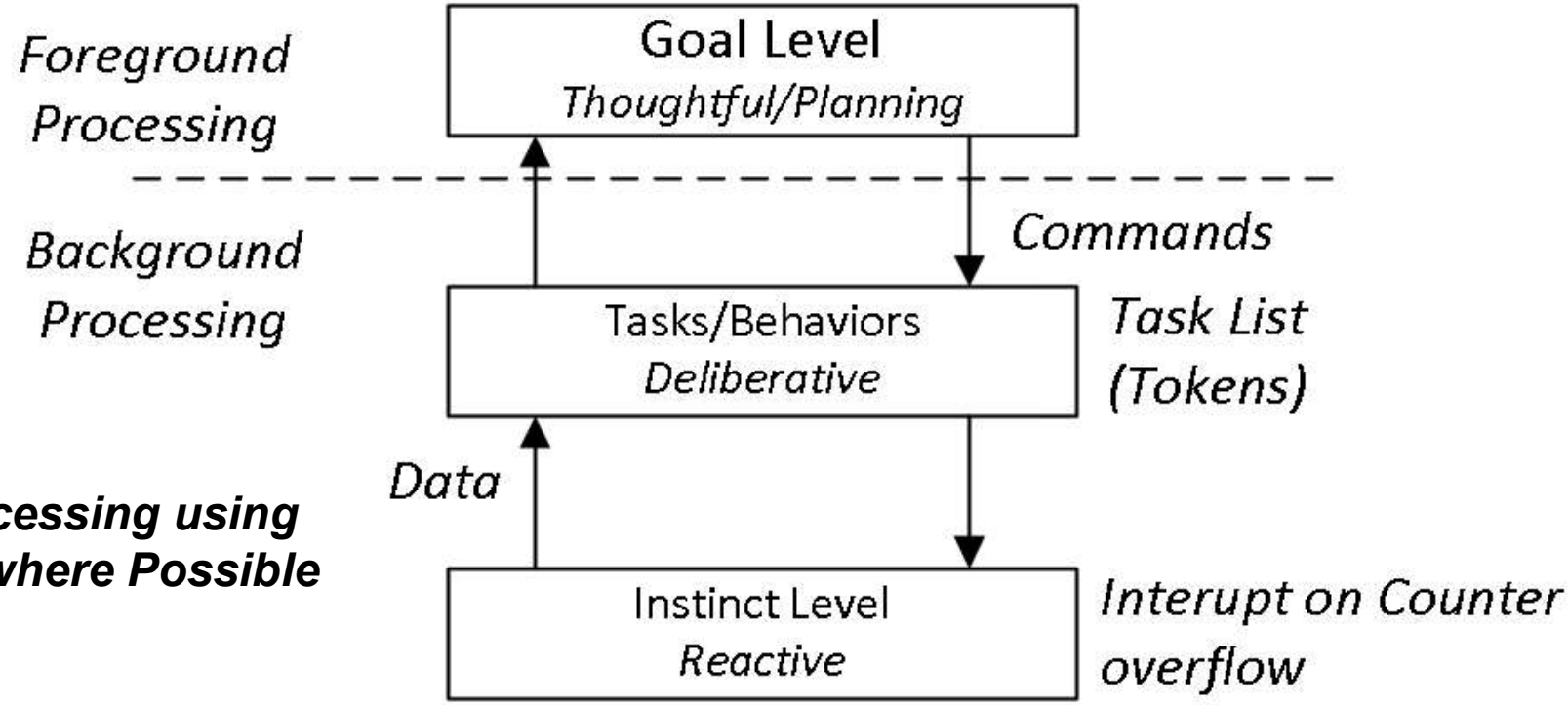
Triune OS in Silicon

- ***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 B

Triune OS Multitasking/AI Model



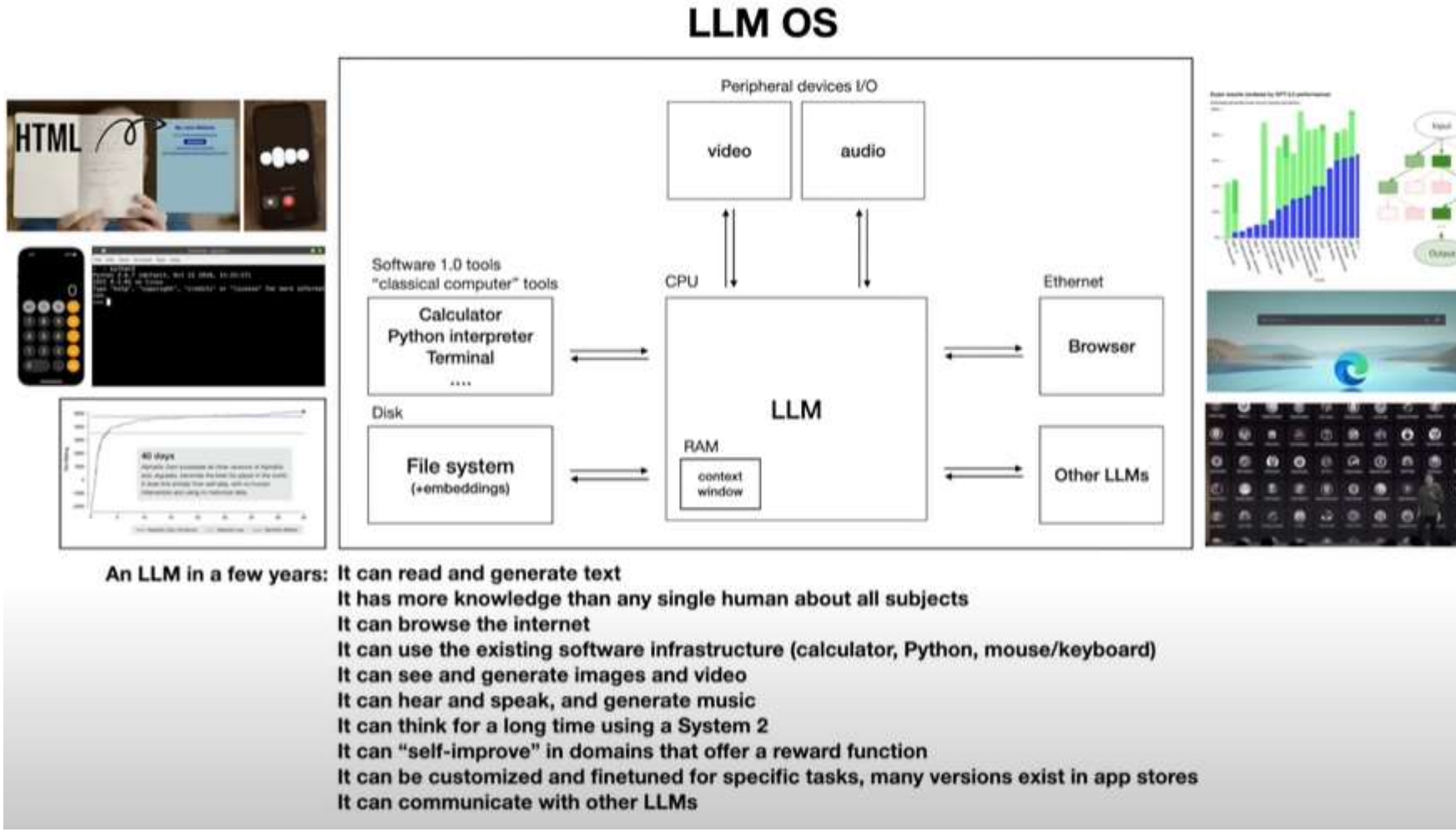
Use Parallel Processing using System Verilog where Possible

TOS Emulates Intelligence in Nature

CORE I AI Playground Board Mimics Human Intelli

LLM OS AI Hardware Architecture

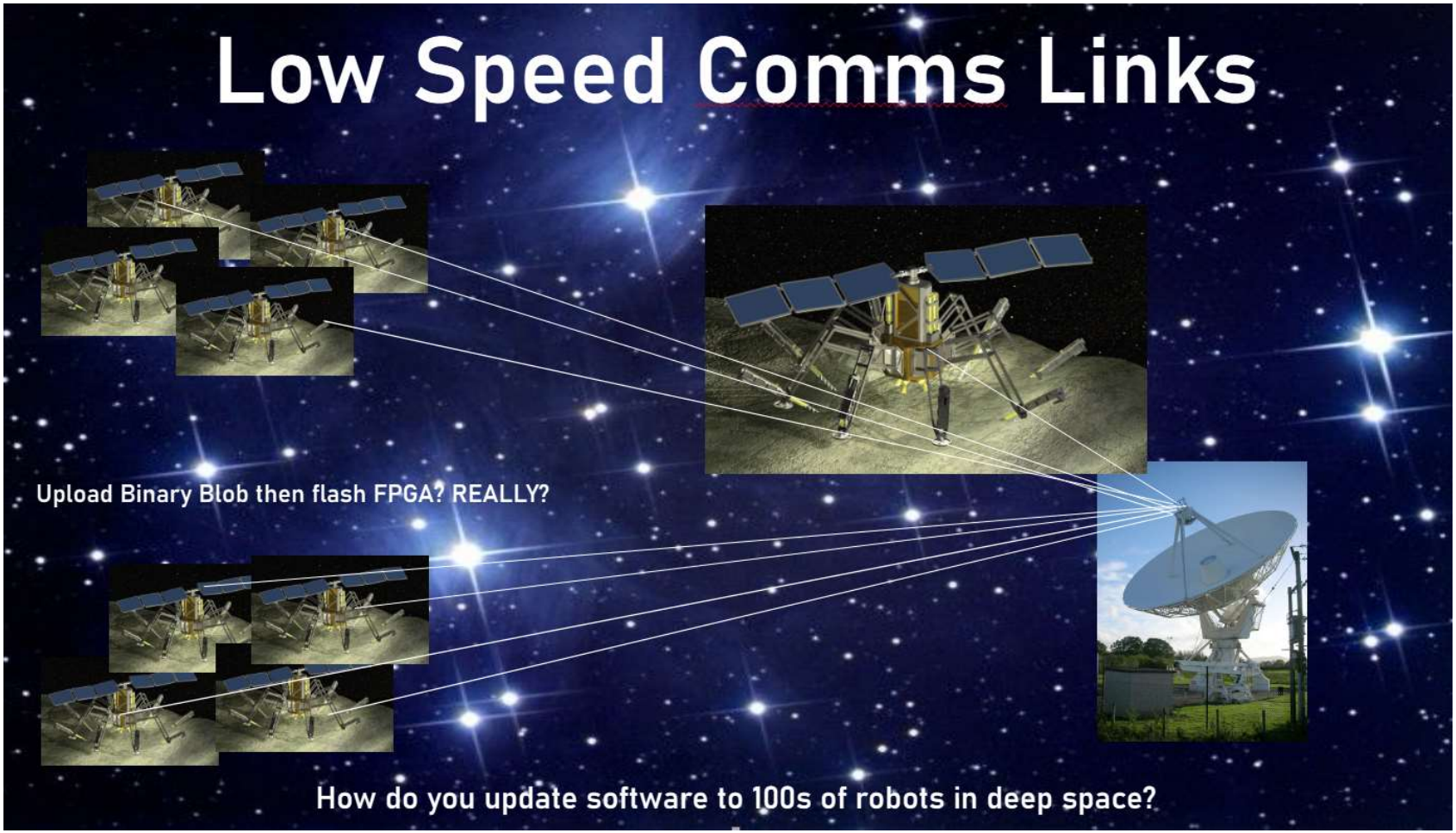
Andrej Karpathy



https://www.youtube.com/watch?v=zjkBMFhNj_g

CORE I AI Playground Board Mimics Human Intelli

CORE I Processor REQUIRED for Space Robo



- Increme
  Software
- Comput
  Stop Ex
  While N
  Function
  Updated
- Updates
  in One C

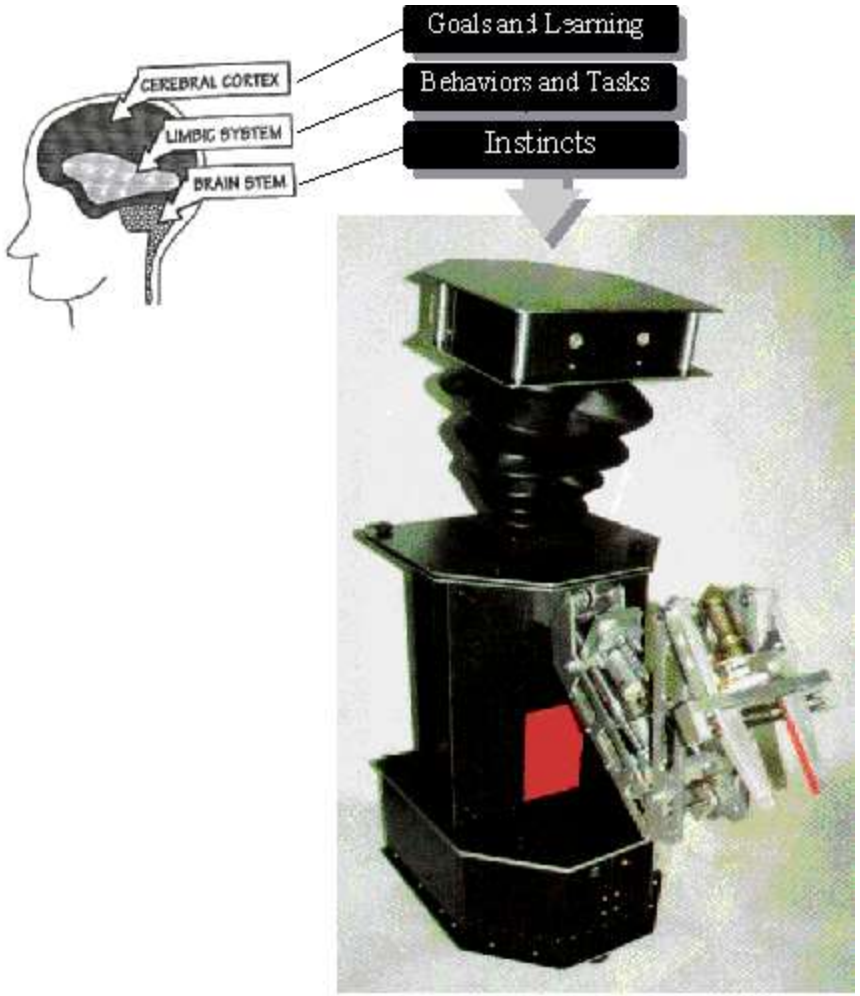
Space Applications

Intelligent Drones

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

Drone Applications

Home Assistant Robots



- Uses Wifi to Communicate with Online Chatbots, Medical, Polic
- 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

Make the Personal Computer a Personal Assi

- The Computer of the Future doesn Crash!
- Anyone can program it using Voice
- It uses very little Power
- It can be As Small as a Phone or Tablet
- Communicate using Voice
- Program using English or Voice
- Even Children can Program it

Computers become Personal Assistants (PAs

The Rabbit R1 is an AI-powered gadget

- Voice Controller interface
- Rabbit OS, and the AI tech undern
- similar to Alexa or Google Assistar
- Rabbit OS can control your music, a car, buy your groceries, send you messages
- The large action model, or LAM, w by humans interacting with apps lik and Uber

https://www.theverge.com/2024/1/9/24030667/rabbit-r1-ai-action-model-price-release-date?fbclid=eyqKxaQwJbl9sspwuYLPzj4nDtWR0wfqFuYC6nVF1iMb_dw

The Rabbit R1 is an AI-powered gadget

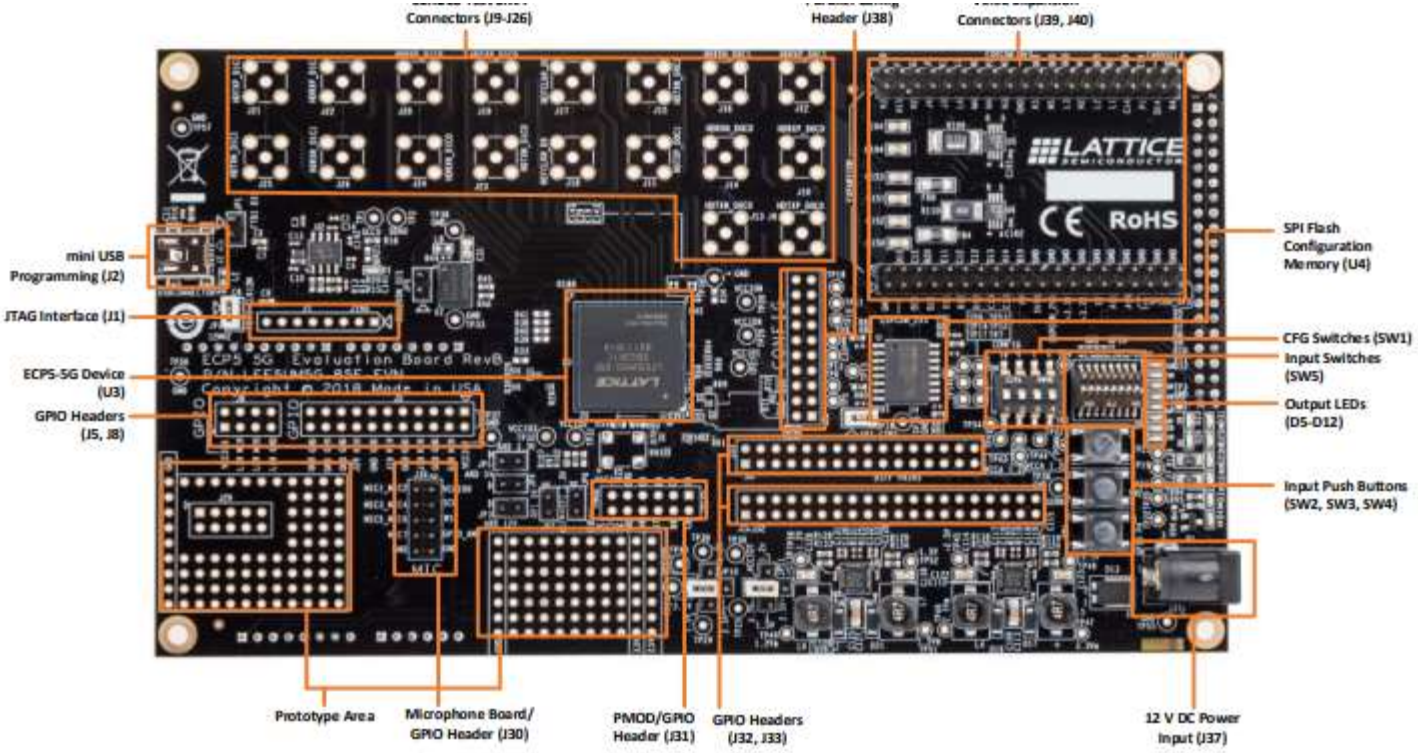


Figure 1.1. Top View of ECP5 Evaluation Board

- Same FPGA as C Playground
- 85KLUTs FPGA
- Use Lattice Diam coding
- $99

https://www.mouser.com/ProductDetail/Lattice/LFE5UM5G-85F-EVNG?qs=Li%252BoUPsLEnuSZ%3D%3D

Computers become Personal Assistants (PAs