



Tachyum™ Prodigy



Dr. Radoslav Danilak
Rodney Mullendore
Igor Shevlyakov
Kenneth Wagner

8/1/2018



Legal Disclaimers

NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. TACHYUM ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF TACHYUM PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, OR MERCHANTABILITY.

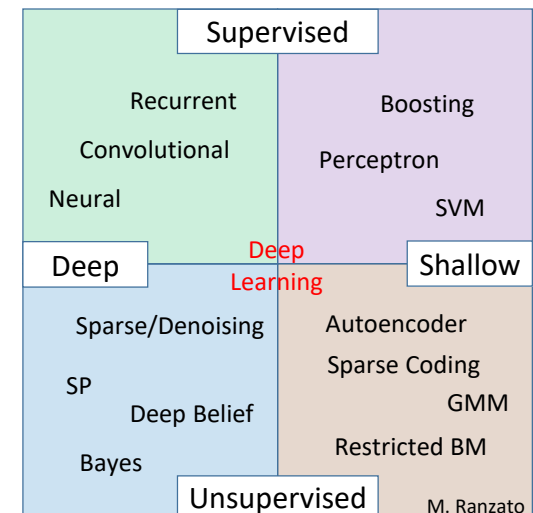
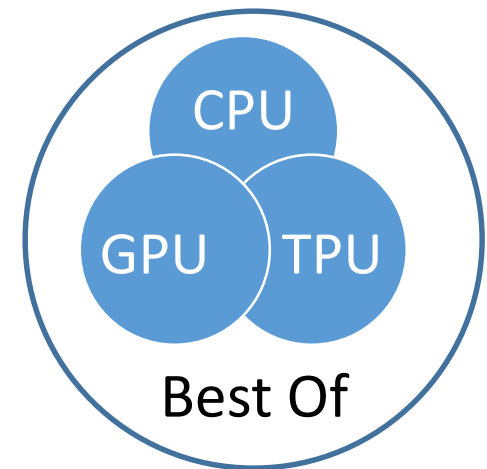
All information provided here is subject to change without notice. Nothing in these materials is an offer to sell any of the components or devices referenced herein.

Tachyum is a trademark of Tachyum Ltd., registered in the United States and other countries, Tachyum Prodigy is a trademarks of Tachyum Ltd. Other products and brand names may be trademarks or registered trademarks of their respective owners.

©2018 Tachyum Ltd. All Rights Reserved.

Prodigy: Universal Processor / AI Chip

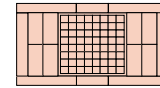
- Prodigy is a Server/AI/Supercomputer Chip
 - For hyperscale datacenters, HPC and AI markets
- **First time humanity can simulate human brain-sized neural networks in real-time**
 - Critical for the Human Brain Project
- Prodigy: a Tachyum Architecture
- Outperforms CPU, GPU and TPU
 - CPU: easy to program, costly & power hungry
 - GPU: much faster but very hard to program
 - TPU: faster but more limited apps than GPU



Tachyum's Prodigy Universal Processor Family

- Faster than Xeon on single/multi-threaded apps

- Prodigy 1 die, 2 packages, multiple SKUs



- T864

- 64 cores and 8 DDR5/4 controllers
- Single and dual-socket Xeon-E5/E7 replacement
- 72 PCIe 5.0, and 2 x 400/200/100G Ethernet



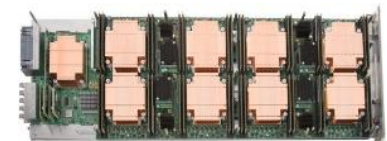
- T432 & T216

- Xeon-D, Xeon-E3 and Xeon-E5 replacement
- 32 cores, 4 DDR4, 32 PCIe 4.0, 2 x 100/50/10GE
- 16 cores, 2 DDR4, 32 PCIe 4.0, 2 x 50/10GE
- Small package, for low cost good 1/2 of die



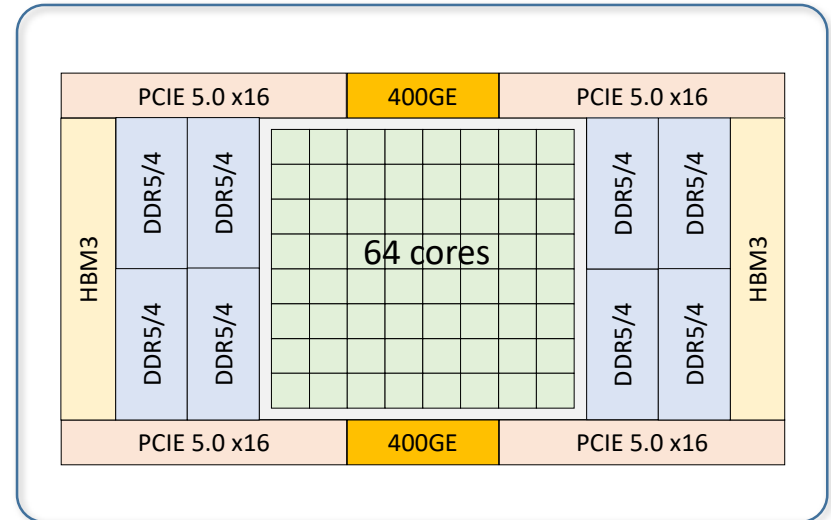
- TH24 - HPC/AI

- 64 cores, 4 DDR5 and/or 32GB HBM3
- HBM3 can be cache or dedicated memory
- Maximum floating-point and AI performance
- Same as T432 package, high density water cooled



Prodigy Chip

- 64 cores, each core faster than Xeon core
 - 8 DDR5/4
 - 72 PCI Express 5.0
 - 2 x 400/100/50/25/10G Ethernet
 - 2 HBM3 (optional)
 - 32MB fully coherent L2/L3 cache
 - 180W, all cores at 4GHz running HPC/AI
 - FCBGA, 66 mm x 66 mm, 1 mm pitch
- Faster than Xeon, smaller than ARM
 - Data travels over very short wires mitigating the “slow wires” problem
 - Out-of-Order execution with Compiler
- 7nm FinFET high-performance process
 - No custom design, standard cells & SRAMs
 - Datapath tiling, place and autoroute
 - 7nm FinFET, 12 metal layers, 0.8V
 - 290sq mm die



Prodigy Instruction Set Architecture

• Instruction Set

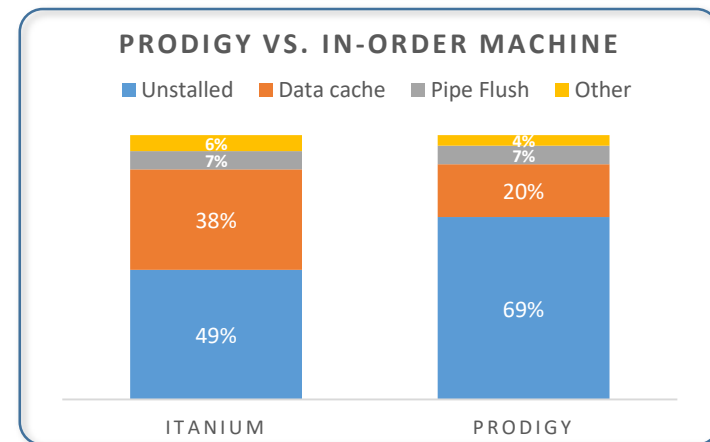
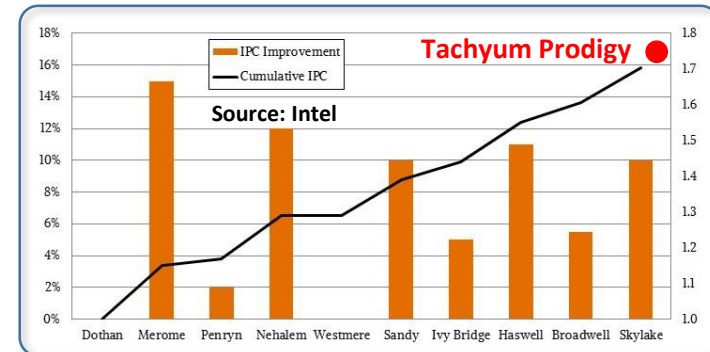
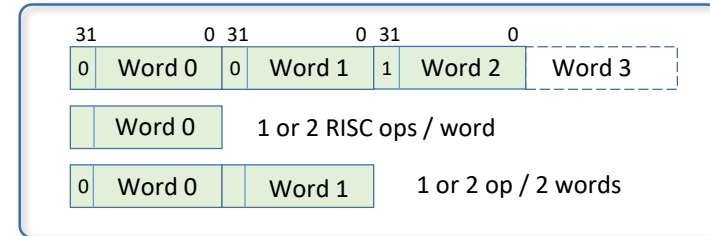
- 32 integer 64 bit registers
- 32 vector registers 256 / 512 bits
- 7 vector mask registers

• Instruction Level Parallelism (ILP)

- Bundle 4, 8, 12, 16 bytes
- Bundle size explicitly marked by compiler
- Sustained up to 8 RISC-style micro-ops/cycle
- 2 LOAD + 2 Multiply-Add + 1 STORE
+ 1 Address Increment + 1 Compare + 1 Branch
- 1.72 Instructions Per Cycle, 2.6 instructions/bundle

• Out-of-order (OOO) execution in software

- OOO performance with In-Order area & power
- Instruction Parallelism extraction using poison bits
- Execute hundred instructions after load miss
- Memory Level Parallelism splits data access and consumption instructions on caches miss



Software

- Tachyum-ported software

- GCC with Tachyum backend, LLVM in 2019
- Porting Linux and Free BSD in 2019
- Device drivers, Boot-loader and Java JIT



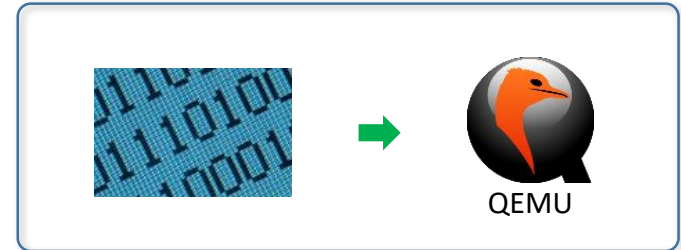
- Existing Applications Recompiled

- Hardware support strong or relaxed memory ordering
- Recompiled application running faster than on Xeon
- Apache, MySQL, Hadoop, Spark, TensorFlow, ...



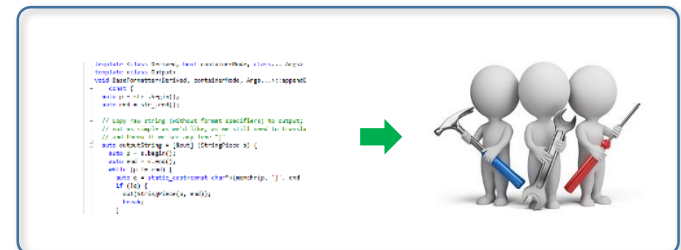
- Existing binaries supported via emulators

- QEMU and emulators transparently launched by Linux
- Deployment of processor before all applications ported
- Port CPU intensive application first, other later



- User applications

- Recompiled by customers
- Porting and support from Tachyum's partners
- Use binary emulators until ported to native execution



Tachyum Prodigy Core

Caches and TLBs

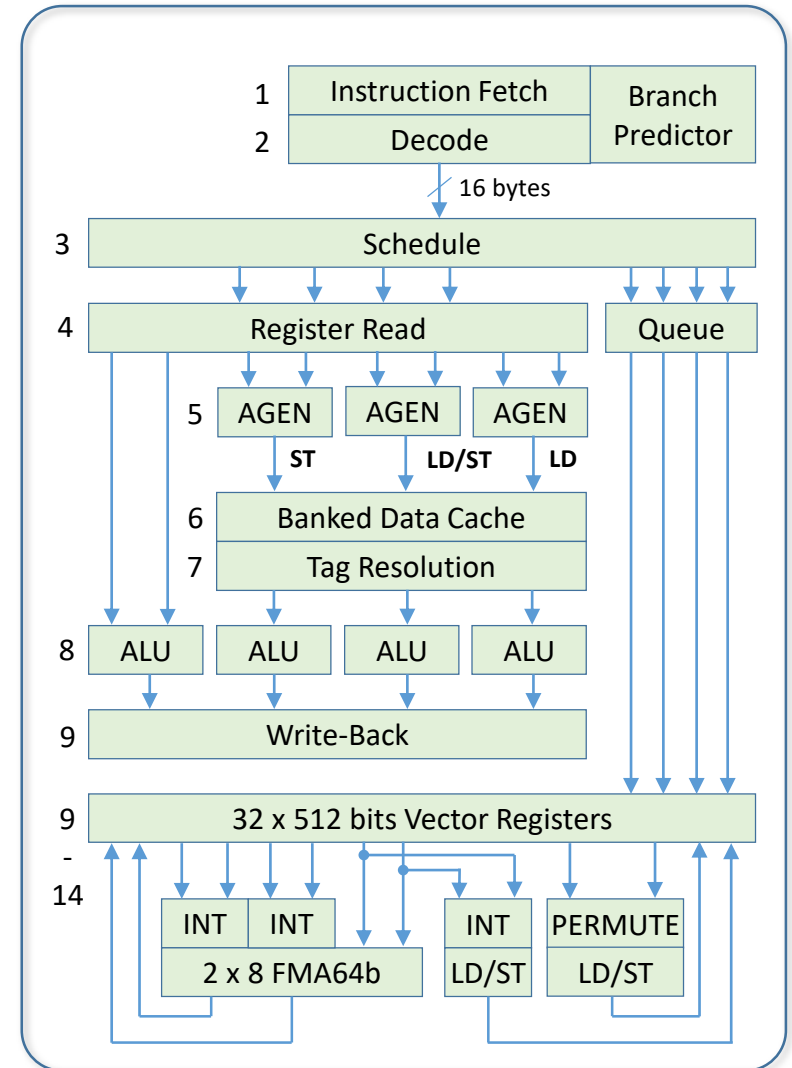
- 16KB 2 way set associative instruction cache
- 16KB 2 way set associative data cache
- 256KB 4 way set associative L2 cache
- 256 entry 2 way set associative TLBs
- 512KB L3 cache slice

Execution units

- 1 load, 1 load/store and 1 store unit
- 3 integer ALU/address generation, shifter, 2 branch
- 2 512-bit vector/matrix integer/FP multiply-add
- 3 512 bit vector ALU + 1 shifter / shuffle unit

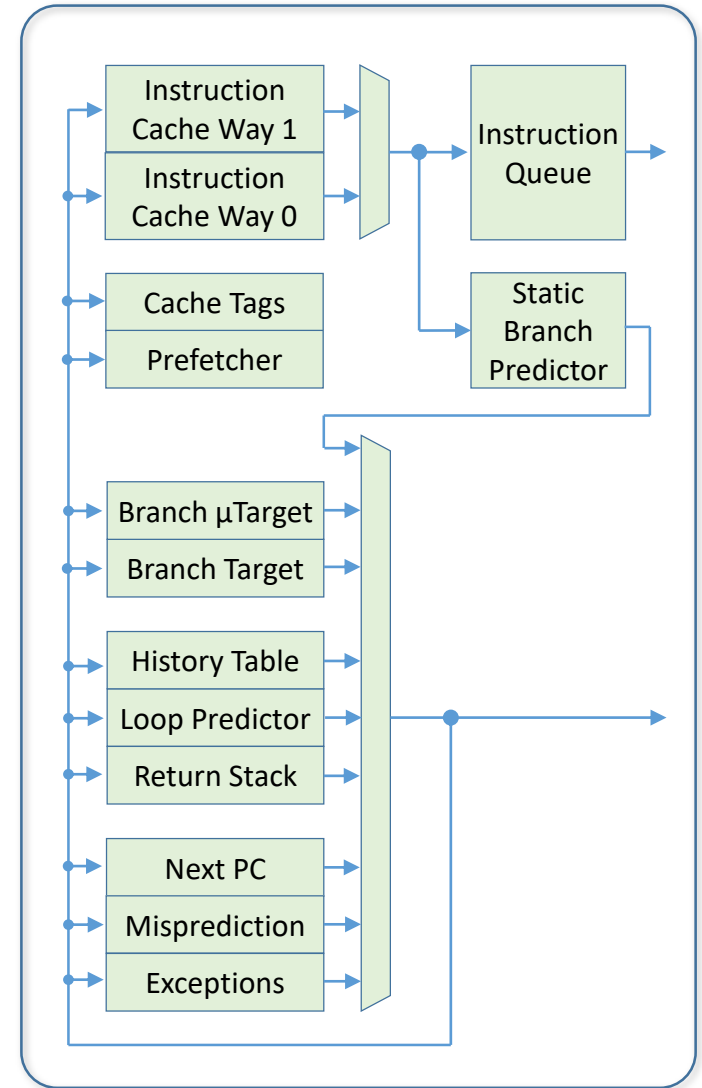
Single Prodigy core

- 9 stage pipeline for integer
- 14 stage pipeline for vector/matrix multiply-add
- 1 load, 1 load/store, 1 store, 256 entry 2 way TLBs
- 3 integer ALU, 1 integer shifter, 2 branch units
- 2 512b multiply-add vector/matrix, shuffle
- 3 512b integer vector units, vector shifter
- Cache directory controller and mesh interface



Instruction Fetch

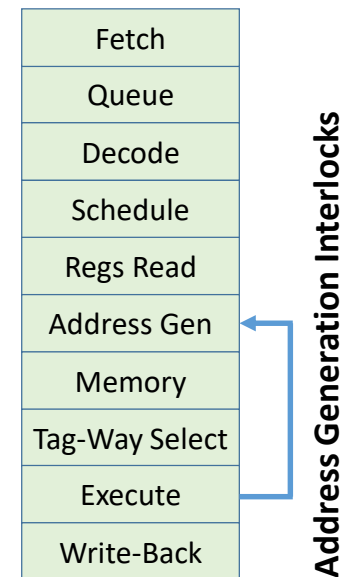
- L1 instruction cache
 - 16KB, 2-way set associative, 128 bytes cache line
 - Hardware prefetch on miss, 64 bytes per clock fill
 - Non-inclusive, SECDED ECC
 - Fetch 4, 8, 12, or 16 bytes per clock
 - 12 entry instruction queue to allow fetch ahead
- Branches
 - Execute up to 2 conditional branches per clock
 - 7 cycles branch misprediction latency
 - Repair branch prediction state on misprediction
- Dynamic and Static Branch prediction
 - Up to 2 branch predictions per clock
 - 0-1 cycle penalty for most predicted branches
 - 2 cycle penalty for statically predicted branches
 - Global history based branch predictor
 - 1024 entry branch target cache
 - 16 entry branch target μ cache
 - Loop, stack and static branch predictor



Instruction Execution

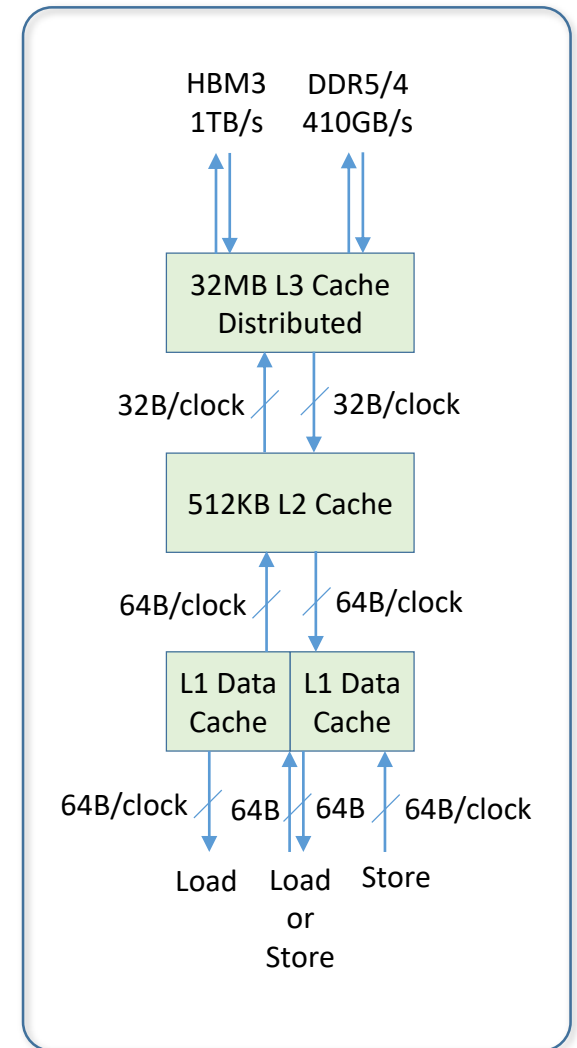
- Integer execution units
 - 3 integer ALU's, 1 shifter, 1 multiplier and 1 divider
 - Early address generation to reduce load to use latency
 - 1 load, 1 load/store, 1 store unit
 - 2 branches per clock
- Control and data speculation
 - Speculative loads defer exception until branch or data use
 - Speculative stores held, uncommitted in buffer, until branch
 - Hardware assists in detecting store mis-speculation
 - Hardware provides exception bit for speculative instructions
- On last level cache miss
 - Continues executing address generation instructions
 - Will execute data consuming instructions in-order later
- Vector instructions
 - Decoupling queue between load and execution stage
 - No stalls on cache L1 and L2 miss data use until queue is full

Reduced Load-to-Use Latency By Early Address Generation



L1/L2/L3 Cache with Directory Coherence

- L1 data cache: 1 load, 1 load/store, 1 store
 - 16KB 2-way set associative, inclusive, SECDED ECC
 - 64 bytes cache line, 64 bytes per clock fill or eviction
 - Sequential and stride hardware prefetch
- Memory management
 - 256 entry TLB with 4KB, 64KB, 2MB, 512MB, 1GB pages
 - Virtualization with nested page tables
- L2 private cache per core
 - 256KB SECDED ECC, hardware sequential prefetcher
 - 128 byte cache line, L1 data cache inclusive
- L3 shared cache
 - 32MB DECTED protected
 - 128 bytes cache line, non-inclusive
- Distributed directory based coherence
 - 128 bytes cache line
 - Non-blocking MESI protocol



Vector and Now Matrix Execution

- Maximum issue rate per clock

- 2 x 512-bit multiply-add
- 3 x 512-bit integer instructions
- 1 load, 1 load/store, 1 store, 1 permutation

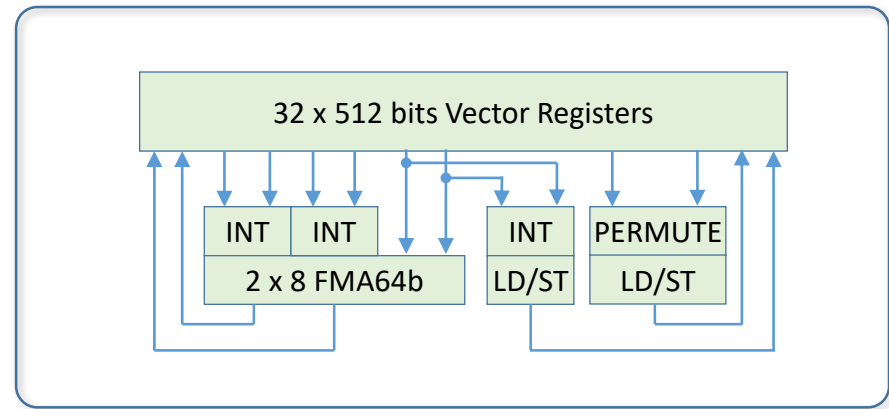
- Floating-Point/Integer execution units

- IEEE double, single and half-precision FPU
- All 8-bit floating-point data type
- 2 x 512-bit multiply-add vector/matrix units
- 3 x 512-bit ALUs 8, 16 & 32 bit integers with no/signed/unsigned saturation

- Vector and Matrix operations

- Matrix operations: 4x less power
- 16b Int/FP 8x8, FP64, FP32 4x4
- 8x8 matrix multiply-add = 1,024 Flops uses 6 source and 2 destination registers
- Can increase performance 2x in the future

64 cores x 4GHz x 512 Flops = 128 TFlops		
2 x 16	Flops	Double-Precision
2 x 64	Flops	Single-Precision
2 x 256	Flops	Half-Precision



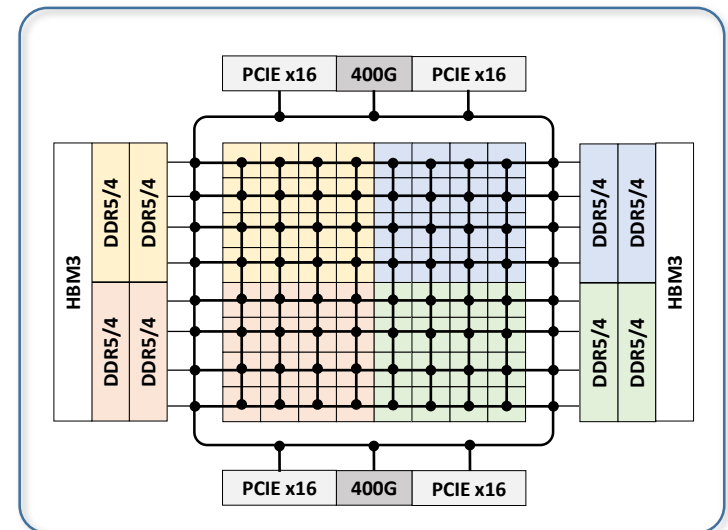
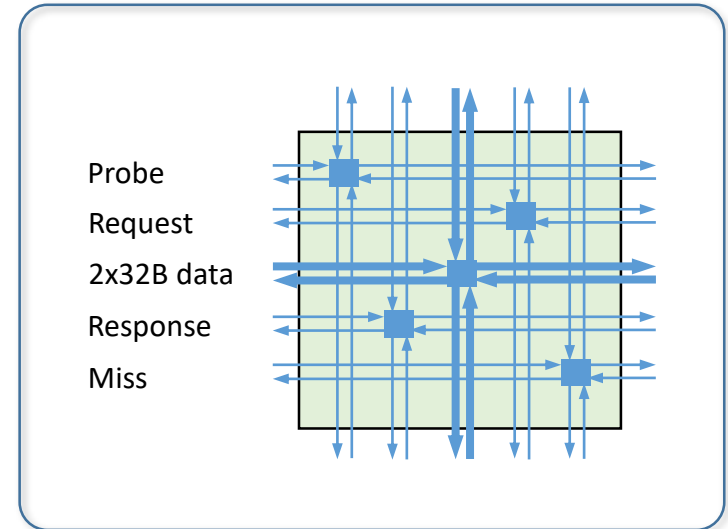
4x4 and 8x8 matrix ops

$$\begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \times \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix} + \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}$$

$$\begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{2,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{3,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{0,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{3,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{0,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{1,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{0,0} & d_{3,1} & d_{3,2} & d_{3,3} \\ d_{1,0} & d_{3,1} & d_{3,2} & d_{3,3} \\ d_{2,0} & d_{3,1} & d_{3,2} & d_{3,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} = \begin{bmatrix} a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \\ a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} & a_{0,0} & a_{1,1} & a_{2,2} & a_{3,3} \end{bmatrix} \times \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \end{bmatrix} + \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \end{bmatrix}$$

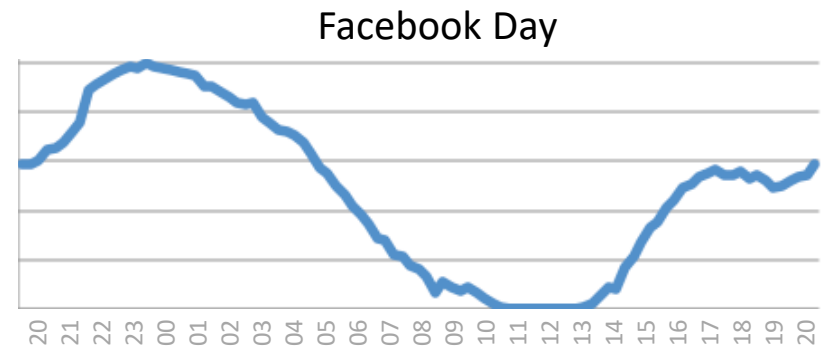
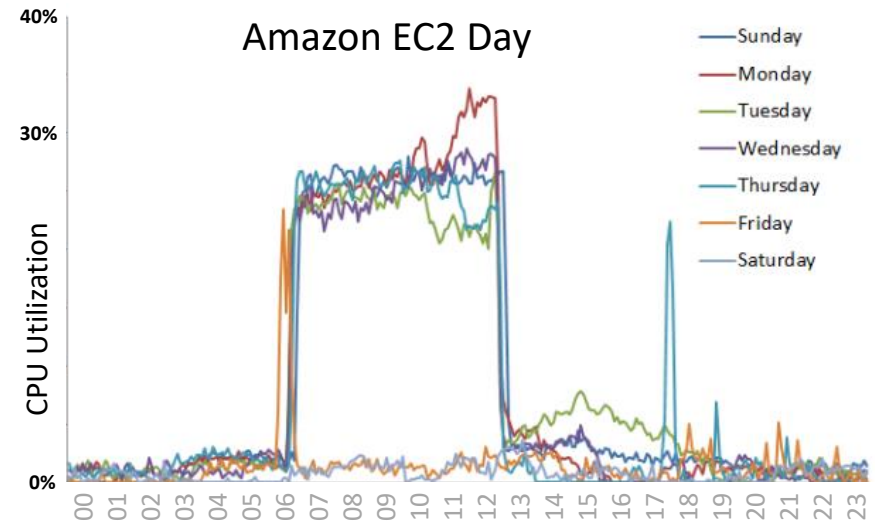
DRAM Controllers, I/O and Mesh

- 32 bytes per clock per direction data mesh
 - Low latency data and request/response meshes
 - Arbitration fairness for quality of service guarantee
 - 1 clock per hop at full clock speed, ECC protected
 - Directories affinity with closest DRAM controllers
- 8 DDR5/4 controllers
 - 2 DIMMs/channel for DDR5/4, 3 DIMMs for DDR4
 - ECC tolerates 2 chip failures/channel with x4 DRAM
- 2 HBM3 controllers supporting 8/16/32GB
 - Unused, or DRAM cache, or memory region
 - Protected by ECC
- I/O ring
 - 32 bytes per clock per direction I/O to DRAM ring
 - 2 x 400Gb / 4 x 100/50/25/10GE PAM4 SERSES
 - 2 x 100Mb / 1Gb Ethernet for management
 - 72 PCI Express 5.0, 36 controllers, x1-16 links



Prodigy Delivers Big AI for Datacenters – CAPEX Free

- Universal Processor / AI chip:
10x more AI using idle servers
- Avg. over 24 hours: 60-80% of servers are idle
<5% of server have AI GPUs
Prodigy enables idle servers to be seamlessly and dynamically reconfigured into HPC/AI systems
- Existing Processors - too slow for AI
therefore, GPU or TPUs are used



Brain Simulation In Hyperscale Datacenter

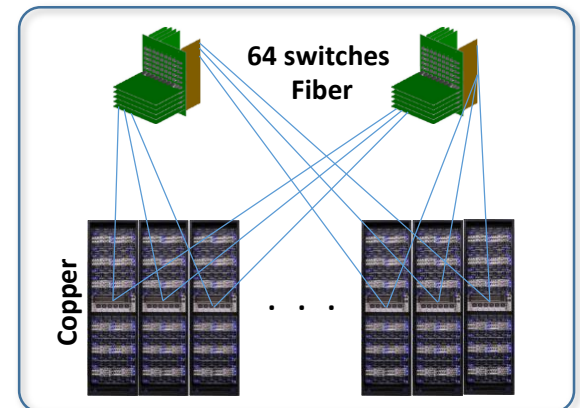
- From Rat Brain to Human Brain real-time simulation

- SpiNNaker system 518,400 processors simulates rat brain
- Human brain simulation requires 1,000x more performance
- The NNSA 20 Pflops Sequoia is 1,542x slower than real-time



- How a system can be built in 2020

- 256K servers, each 4 x 2x100GE with no oversubscription
- Partner's 128 x 2x100GE PAM4 switch chip
- Copper 64 nodes to rack switch, fiber to central switches
- 12U 4K ports x 200GE switch, front-connector-back cards
- Only 1 set of fibers 256 x 2x100 GE vs. 3 to central switches



- 100+ brain-capable datacenters

- Facebook: 100MW datacenter with 442,368 servers
- 40% utilization means 265,420 idle servers
- Use \$100B of underutilized equipment in the world









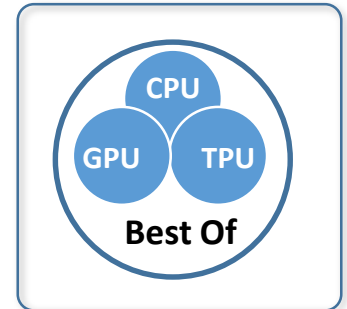
Prodigy Delivers Low Power Cloud

- Datacenters today consume 2% total electricity
 - Consume 40% more power than UK
 - Emit more CO2 than world's airliners
- **10% of planet energy by 2030**
 - 15% growth: is 2x every 5 years
 - 40% of planet energy by 2040
- New Technology is needed
 - 10x lower power to continue growth



Summary and Status

- Tachyum Prodigy is the industry's first Universal Processor
 - Proprietary core architecture & design: optimized for servers, HPC and AI
- Outperforms Xeon on SpecInt & SpecFP 2006 benchmarks
 - One 3.5GHz Xeon E5-2687W v4 core vs one Prodigy core, same GCC 7.2
 - Hand tuned GCC for formal SpecInt2017/SpecFP2017 available next year
- Integer Datapath Proven Post Place & Route at 4GHz at 7nm
- On track for tape-out in 2019
 - Multiple interested and engaged customers
- Visit www.Tachyum.com, follow us on      



Hyperscale



Real-time Human
Brain Simulation



