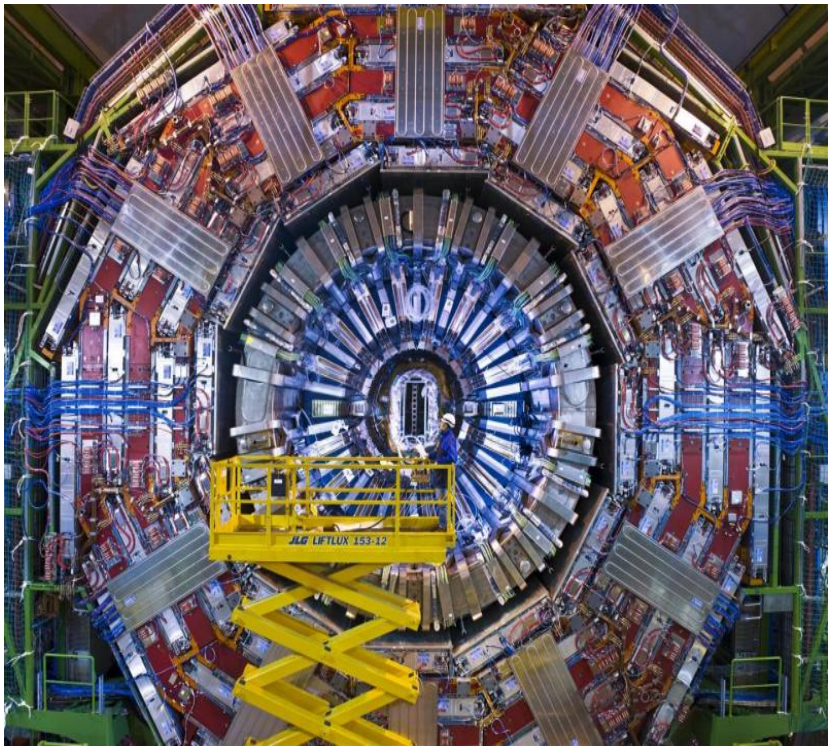
Performance Study of a 2D Prototype of Vertically Integrated Pattern Recognition Associative Memory (VIPRAM)

Sanjay Subramanian
(Mentor: Dr. Ted Liu)

Compact Muon Solenoid (CMS)

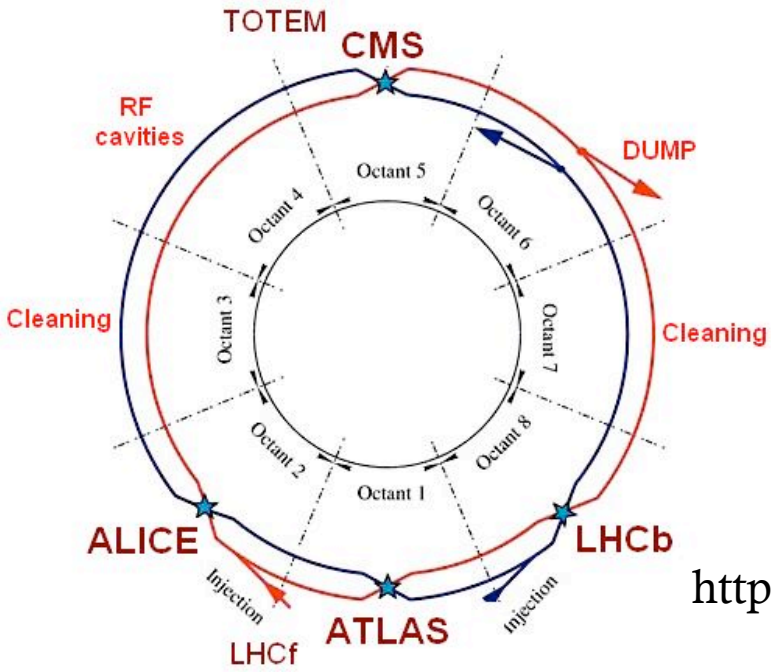
- Detector at CERN
- Looks for Higgs Boson, dark matter particles, extra dimensions
- Huge project – over 4000 physicists, engineers, and others

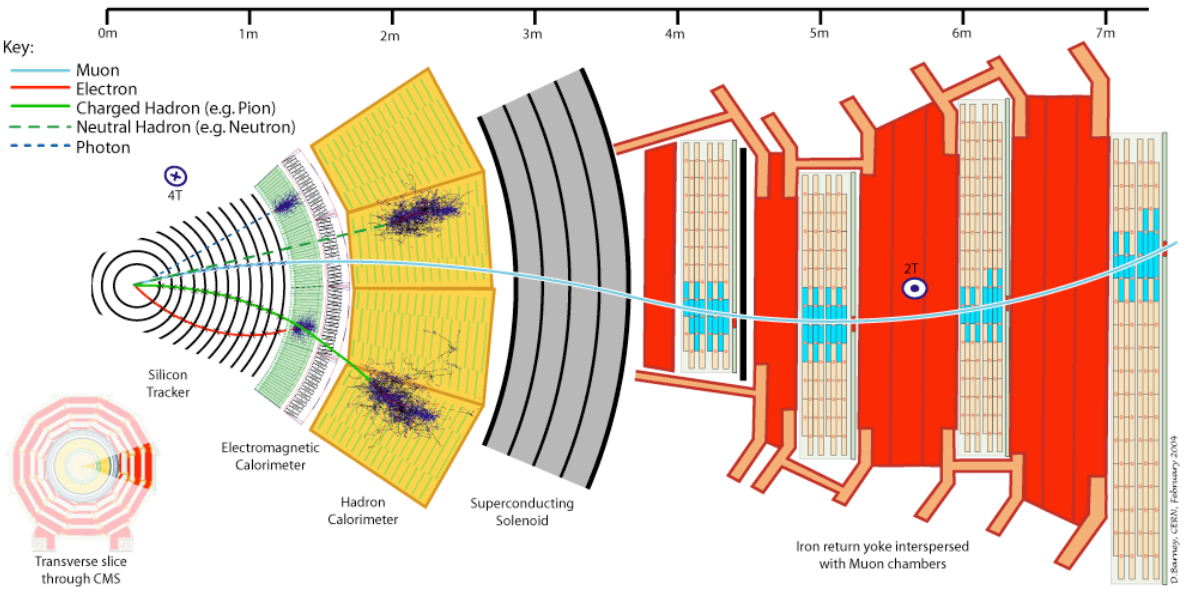


From cern.ch

CMS Challenges

From http://www.lhc-closer.es/1/3/4/0





http://commons.wikimedia.org/wiki/File:CMS_Slice.gif

https://www.youtube.com/watch?v=EVr_7QtQYW8

Luminosity – Number of collisions per unit area per unit time

2011 luminosity: 6 x 10^{33} $cm^{-2}s^{-1}$, planned 2030 luminosity: 5 x 10^{34} $cm^{-2}s^{-1}$

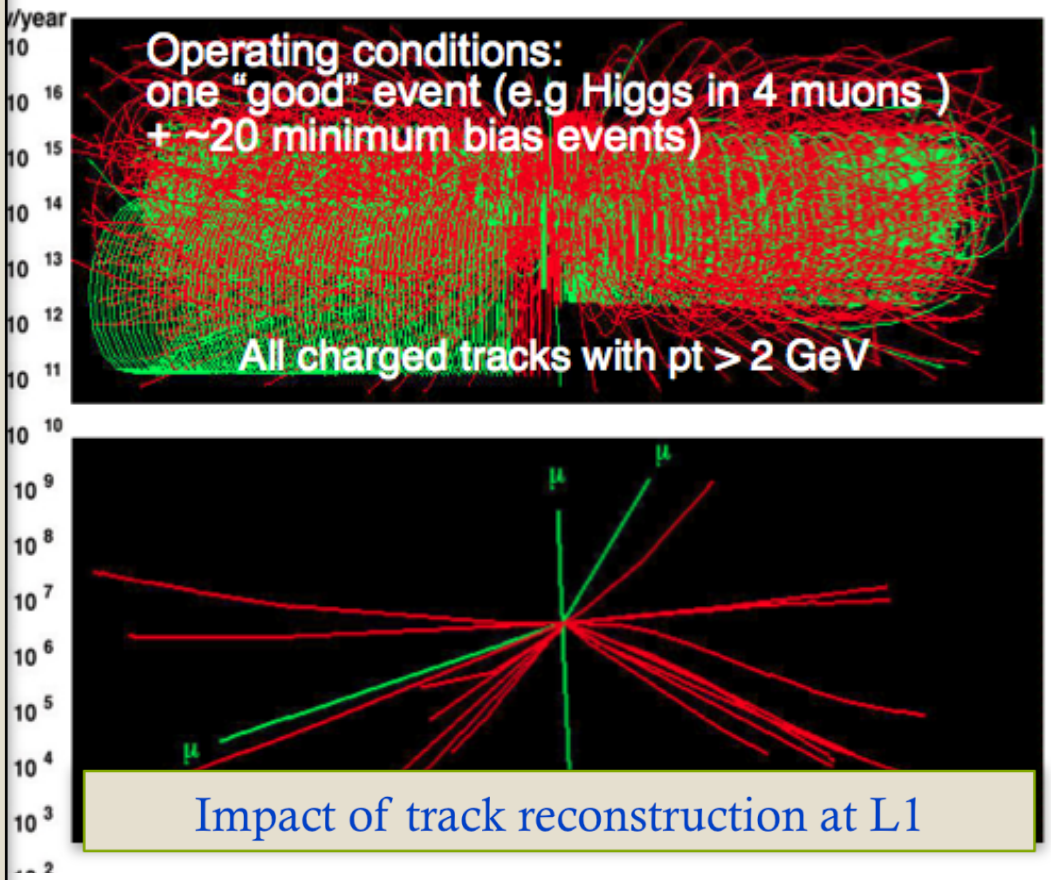
CMS L1 Tracking Trigger:

Will need to reconstruct charged particle trajectories "on-the-fly" for every beam crossing (25 ns, or 40 Million beam crossings per second), from an ocean of input data (bandwidth required to transfer up to ~ 50-100Tb/s)

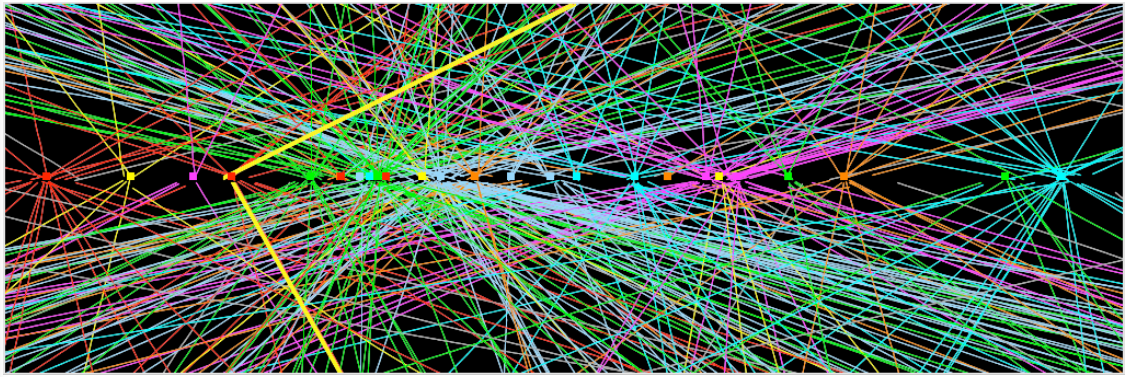
This requires extremely fast high bandwidth data communication as well as massive pattern recognition power,
with lots known patterns to be compared against the multiple input data streams simultaneously with near zero latency (~ few μs)

This is challenging!

Pileup at HL-HC: > ~ 140 (only 20 shown here)

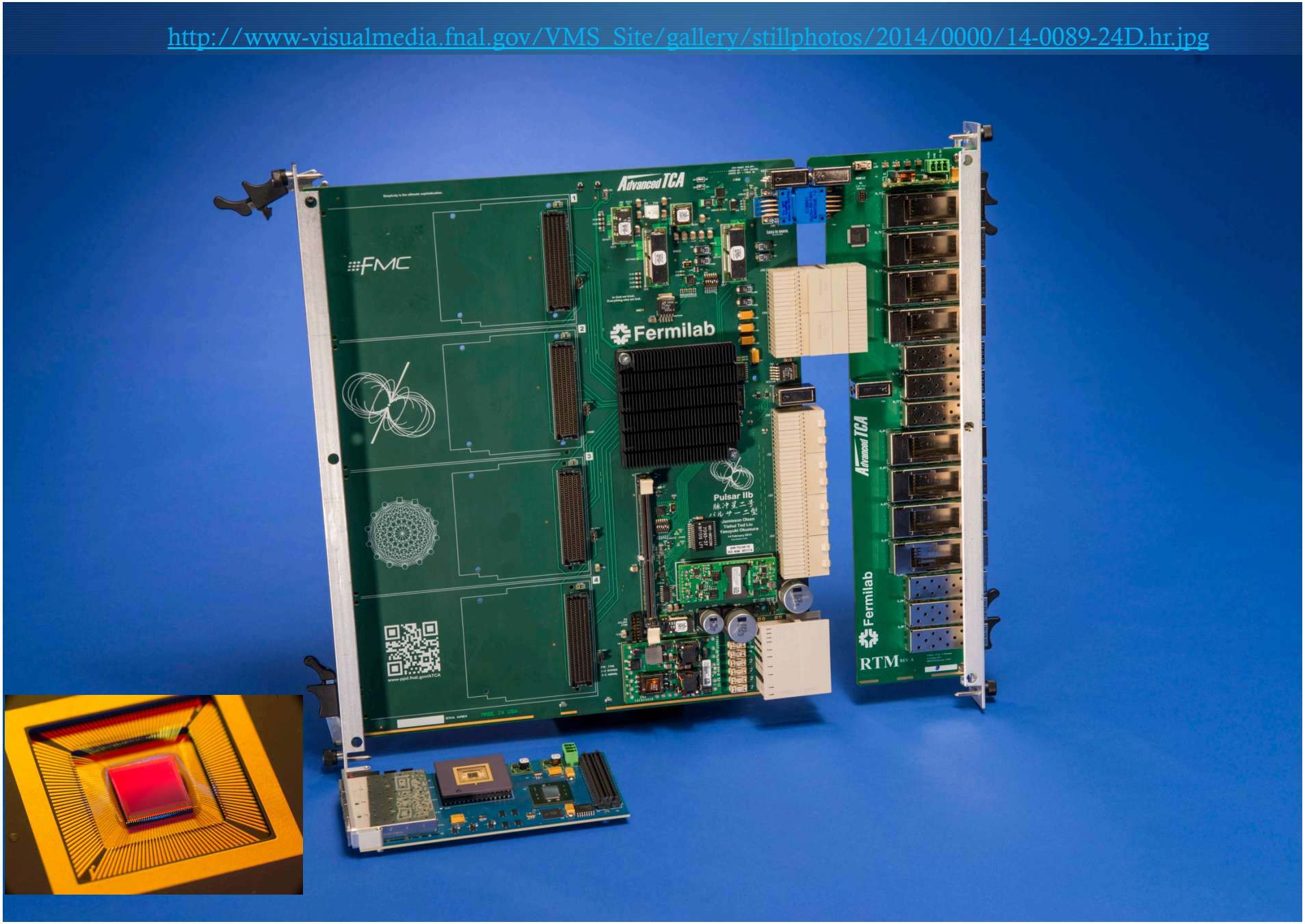


Impact of track reconstruction at L1



Borrowed from Dr. Ted Liu's HL-LHC Tracking Trigger Challenges

http://www-visualmedia.fnal.gov/VMS_Site/gallery/stillphotos/2014/0000/14-0089-24D.hr.jpg

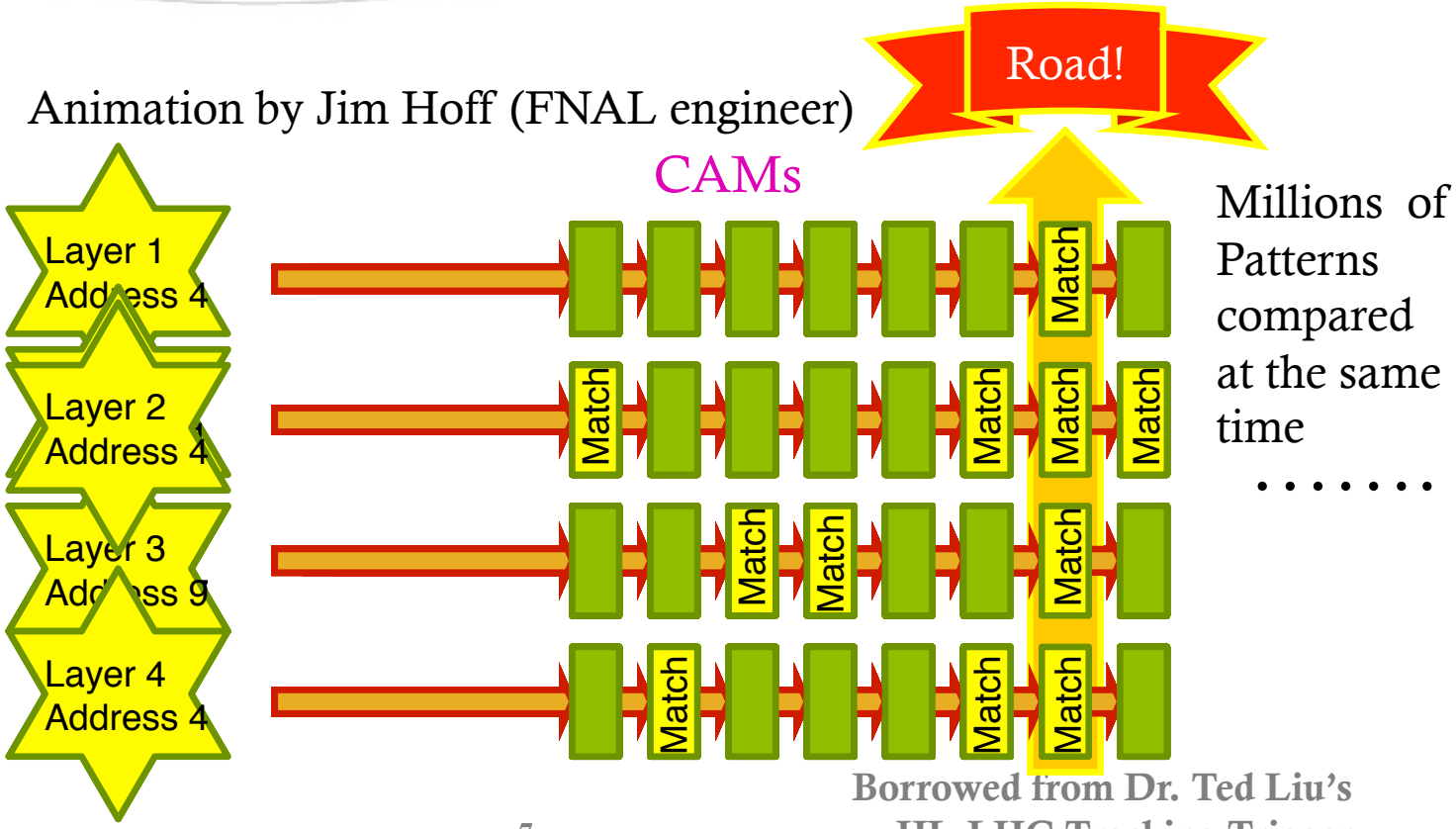


Borrowed from Dr. Ted Liu

What is VIPRAM?

- Uses Content-addressable memory (CAM)
  - Different from RAM
- VIPRAM stores many patterns in CAM cells
- Hardware-based pattern recognition
- The final product will be 3D integrated circuit – higher pattern density and higher speed than in 2D

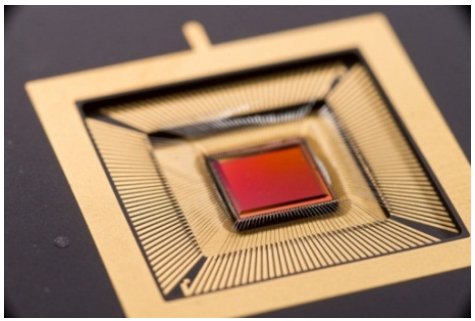
How Associative Memory Works



Borrowed from Dr. Ted Liu's HL-LHC Tracking Trigger Challenges

Deeper into the Design

- Selective-precharge saves power; tradeoff between power and speed (NAND vs. NOR cells)
- 2D prototype: 128 rows, 32 columns (4096 roads total)



From Dr. Ted Liu

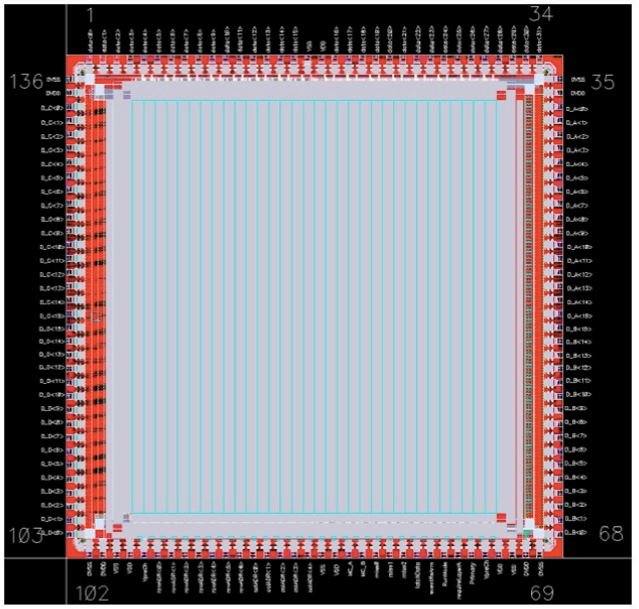
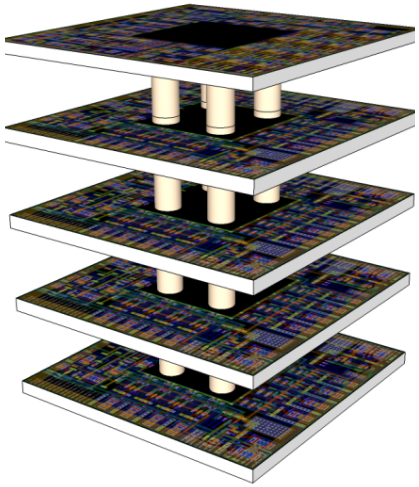


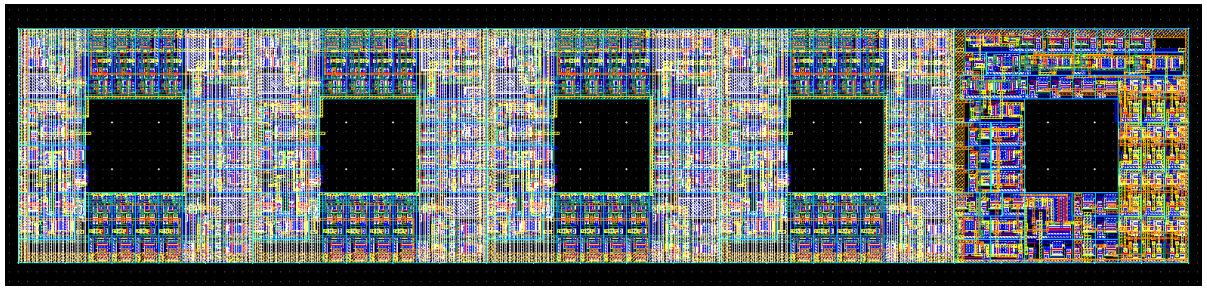
Figure 8 – protoVIPRAM pad arrangement.

2D vs. 3D

- 2D prototype has all 4 layers in same plane
- Majority Logic – an additional layer for each road that indicates if there is a match



From Dr. Ted Liu



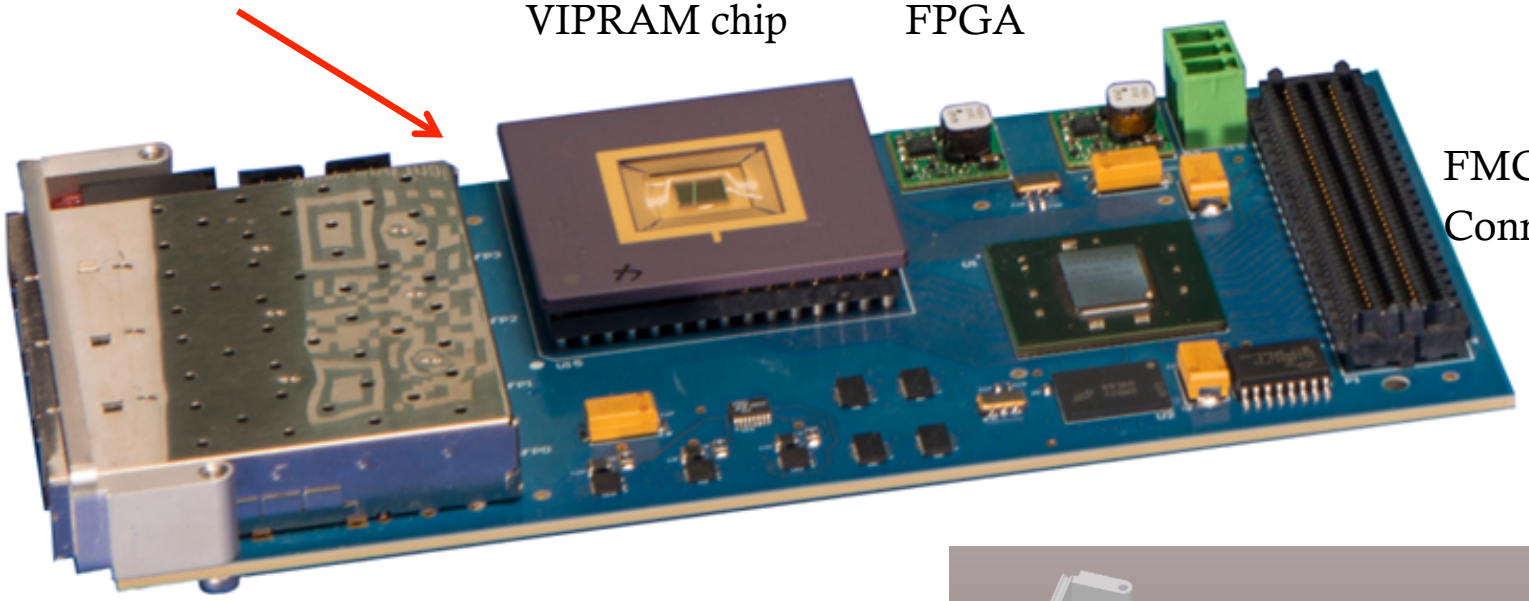
Testing Setup

- 2D VIPRAM prototype mounted on mezzanine card and connected to FPGA
- Load firmware (different for different clock frequencies) using ChipScopePro software
  - Clock frequency dictates rate at which instructions given to chip
- Write test files in Python
- Run from the Terminal (on a Scientific Linux machine)

Prototype Pulsar II Mezzanine card

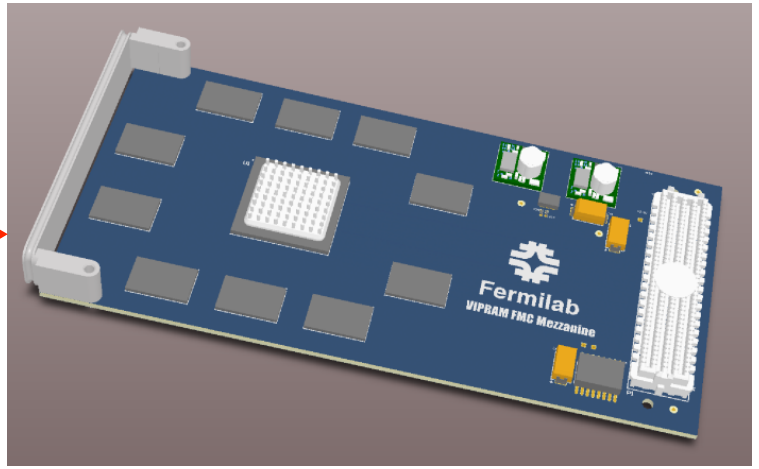
Socket for VIPRAM chip

Kintex XC7K160T FPGA



FMC Connector

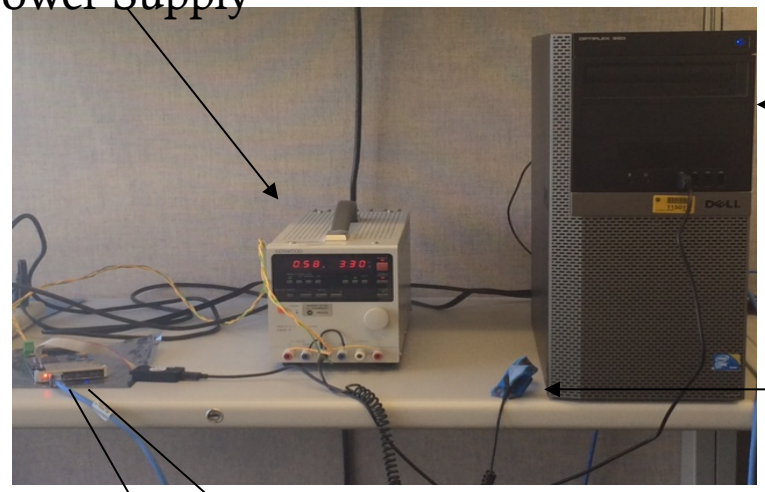
Future versions →



Borrowed from Dr. Ted Liu

Testing Setup (cont.)

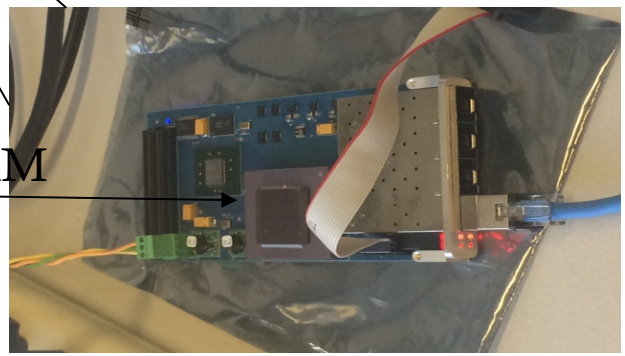
Power Supply



Computer

Grounding Strip

protoVIPRAM



View from Office where I worked

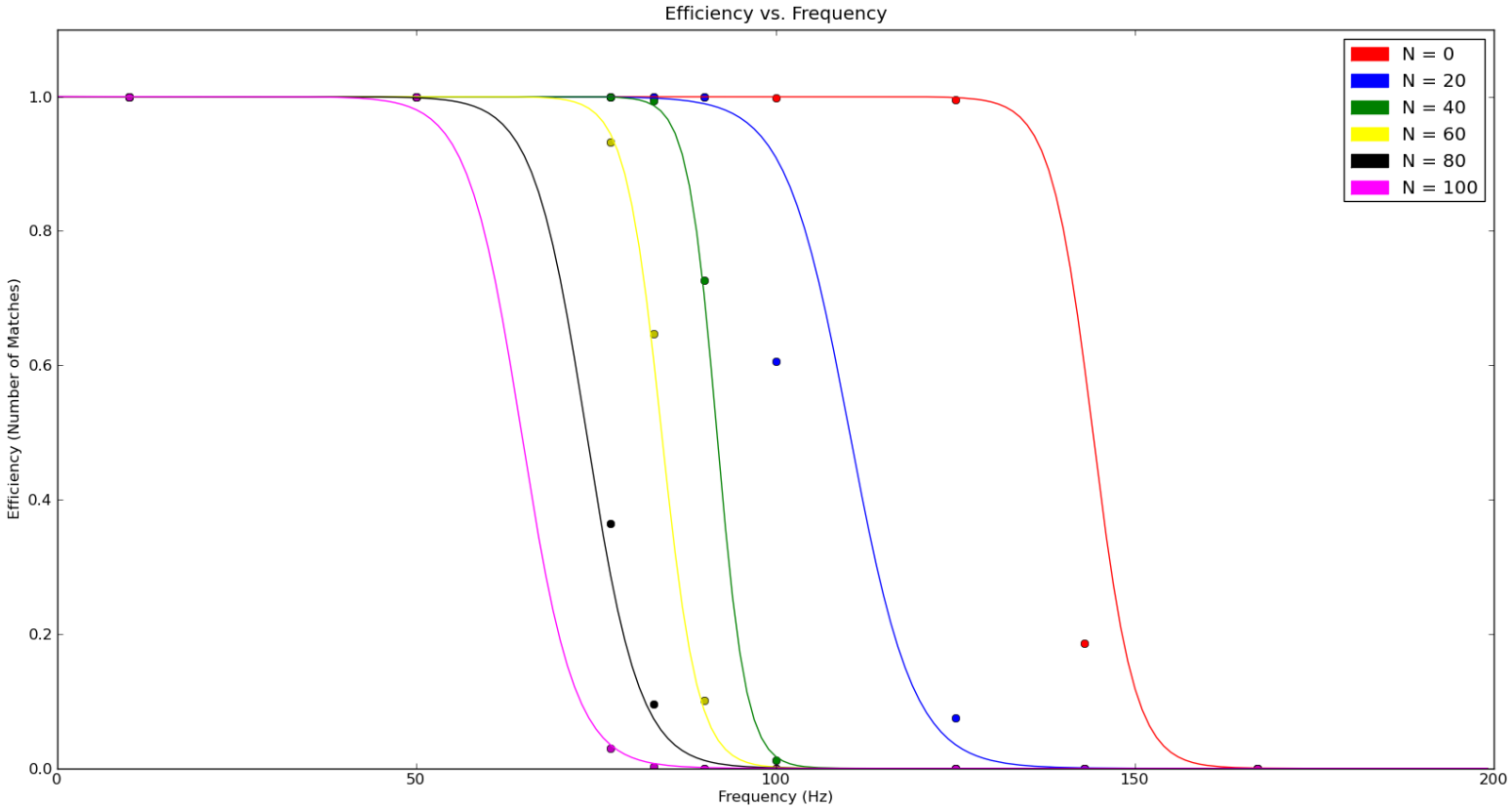
Basic Testing

- Load random numbers into all locations in the chip
- Sequentially check each random number loaded in each location to see if there is a match
- Works perfectly up to 90 MHz
- Testing project outline:
  - Stress (torture) tests – pushing the chip to its limits
  - Realistic tests – using real data to gauge chip performance

Stress Testing (cont.)

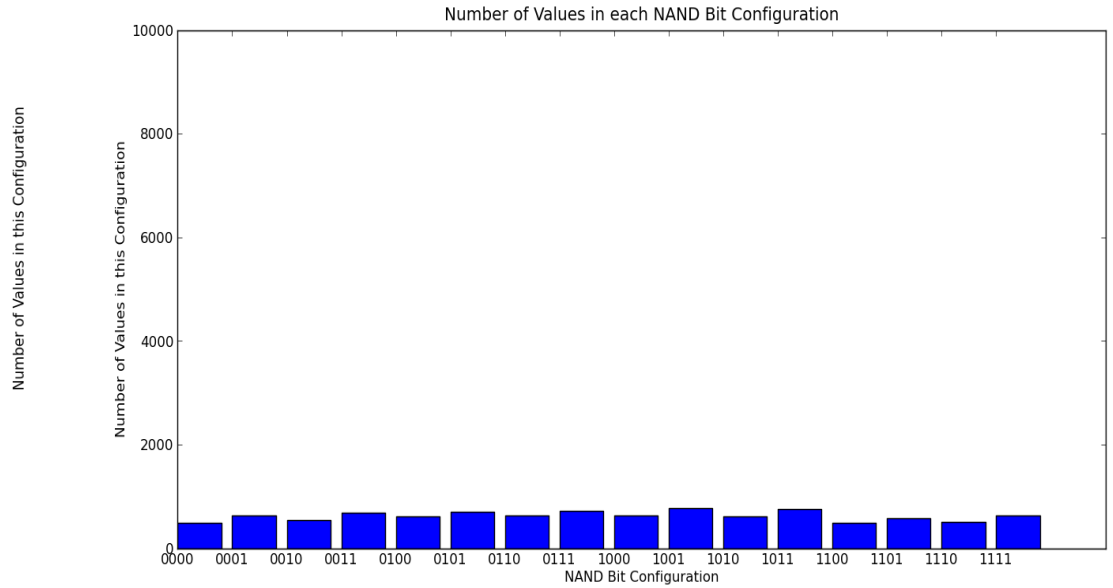
- One possible reason for errors – great deal of power consumption
- Load: 000…00 in about N% of the roads, load 111…11 in about (100-N)% of the roads
- For every row:
  - Load and check: 111…10 every 4 columns
- Load back whatever was in road before Step 2

Stress Test Results



Preliminary Real Data Testing

- Scientists from CERN provided real pattern banks
- 10, 000 patterns in one file
- Loaded 4096 randomly chosen patterns into chip sequentially, checked for these 4096 patterns
- Efficiency – 50 MHz: 100%; At 77MHz: 96.3%; at 90 MHz: 20.2%
- Can we do better?


Number of Values in each NAND Bit Configuration
Number of Values in this Configuration
Number of Values in this Configuration
10000
8000
6000
4000
2000
0
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
NAND Bit Configuration

Shuffling Bits

1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384

000000000001111  111100000000000

What's the best configuration?

- Most ideal: Each outcome equally likely
- 2^4 = 16 outcomes -> 1/16 probability for each outcome
- Optimal configuration minimizes:
  - C = (p1-1/16)^2 + (p2-1/16)^2 + ... + (p16-1/16)^2
- 15 choose 4 = 1365 ways of picking 4 bits
- Brute force scan through possibilities

Optimization Results

Efficiency Chart		
	77 MHz	90 MHz
No Optimization	96.3%	20.2%
Full Optimization	100%	99.8%

Project Status

- Next step is making the 3D prototype
- Rigorous testing contributes to design, provides benchmark
- Characterization of Real Pattern Banks useful for gauging real performance

Future Work

- Varying/Monitoring other variables
  - Power
  - Voltage
- Testing 3D prototype

Personal Impact

- Little hardware background
  - Computer science experience
- Learned about challenges facing modern particle physics
- More programming experience!
  - Bash scripting
  - Over 1000 lines of code in Python, bash, C++

Acknowledgements

- Dr. Ted Liu (Mentor)
- Entire VIPRAM team
  - Dr. Sergo Jindariani
  - Dr. Nhan Tran
  - (Soon to be Dr.) Sid Joshi
  - Dr. Jim Hoff
  - Dr. Jamieson Olsen
- Mr. Dzuricsko, Dr. Stoughton, Ian, fellow students
- Fermilab – Program, facilities, cookies and cheese