SLHC Calorimeter Trigger Firmware Studies and Performance

T. Gregerson, A. Farmahini-Farahani, J. Liu, B. Buchli,
K. Compton, M. Schulte,
ECE Department , U.Wisconsin

M.Bachtis, S.Dasu,T. Gorski, K.Flood, I.Ross, W. Smith
Physics Department , U.Wisconsin

W. Plishker, G. Zaki, S. Kedilaya, N. Sane, S. Bhattacharyya
ECE Department , U.Maryland

October 28, 2009

Introduction

- Design Platform and Methodology
- Preliminary Designs and Results
  - RocketIO and Buffering
  - Particle Cluster Finder
  - Cluster Weighting and Overlap Filter
  - Cluster Isolation and Jet Reconstruction
  - High-Speed Sorters
- DIF Modeling and Unit Testing
- Next Steps

Initial Design Platform

- **Xilinx Virtex-5 devices contain**
  - **Virtex-5 Slices (4 LUTs and 4 flip-flops)**
  - **DSP48E Slices (multiplier, adder, and accumulator)**
  - **Block RAMs (36 Kbits)**
  - **RocketIO Transceivers**
    - GTP transfers up to 3.75 Gbps/link
    - GTX transfers up to 6.50 Gbps/link
- **Initial designs synthesized for**
  - **Xilinx Virtex-5 LX110T and TX240T FPGAs**

FPGA	Virtex-5 Slices	DSP48E Slices	Block RAM (Kbits)	RocketIO Transceivers
LX110T	17,280	64	5,328	16 GTP
TX240T	37,440	96	11,664	48 GTX

Future Design Platform?

- **Xilinx Virtex-6 devices**
  - **Increase the number of flip-flops per Virtex Slice from four to eight**
  - **Do not modify the DSP48E Slices and Block RAMs**
  - **Will have higher speed RocketIO transceivers**
    - GTX transfers up to 6.50 Gbps/trans (up to 36 trans)
    - HTX transfers up to 11.2 Gbps/trans (up to 64 trans)

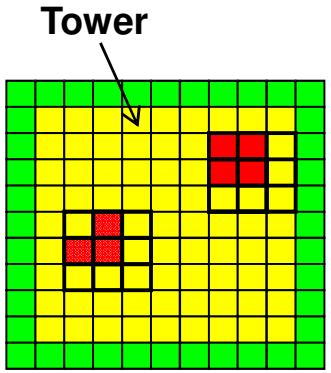
FPGA	Virtex-5/6 Slices	DSP48E Slices	Block RAM (Kbits)	RocketIO Transceivers
5V-TX240T	37,440	96	11,664	48 GTX
6V-SX475T	74,400	2,016	38,304	36 GTX
6V-HX565T	58,560	864	32,832	48 GTX, 24 GTH

Initial Design Methodology

- Designs start with the algorithms
- Physicists and engineers collaborate
  - Evaluate algorithm/implementation tradeoffs
- Designs specified using
  - Verilog and Xilinx Core Generator
- Designs implemented and tested using
  - Xilinx ISE
  - ModelSim Xilinx Edition
- Initial results obtained for
  - RocketIO, buffering, particle cluster finder, cluster overlap filter, cluster weighting, sorting

Rocket IO and Buffering

- **Each pair of RocketIO links provides 17-bit data for 15 towers every 25ns**
  - **An 8 x 8 grid requires 12 RocketIO GTX links**



$$\left\lceil \frac{9 \times 9}{15} \right\rceil \times 2 = 12$$

  - **A 16 x 16 grid requires 40 RocketIO GTX links**

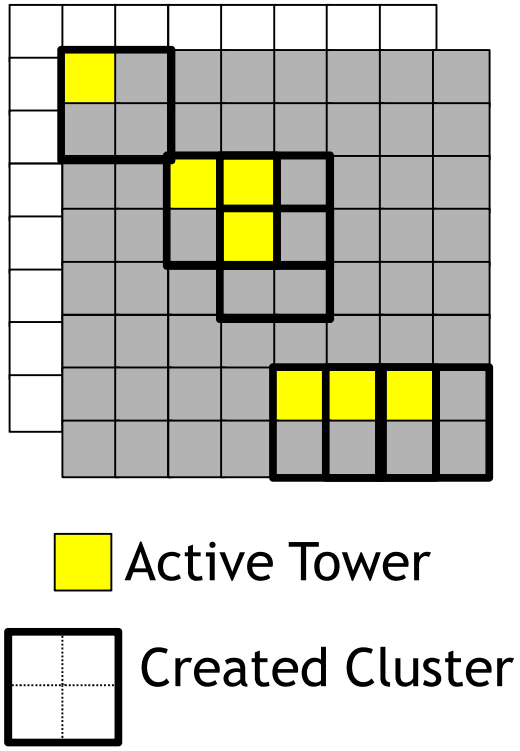
$$\left\lceil \frac{17 \times 17}{15} \right\rceil \times 2 = 40$$

Virtex-5 Resource Utilization for Input and Output RocketIO and Buffering on the TX240T FPGA

Resource	8 x 8 Grid	8 x 16 Grid	16 x 16 Grid
RocketIO Input Links	25%	46%	83%
Virtex-5 Slices	5%	10%	19%

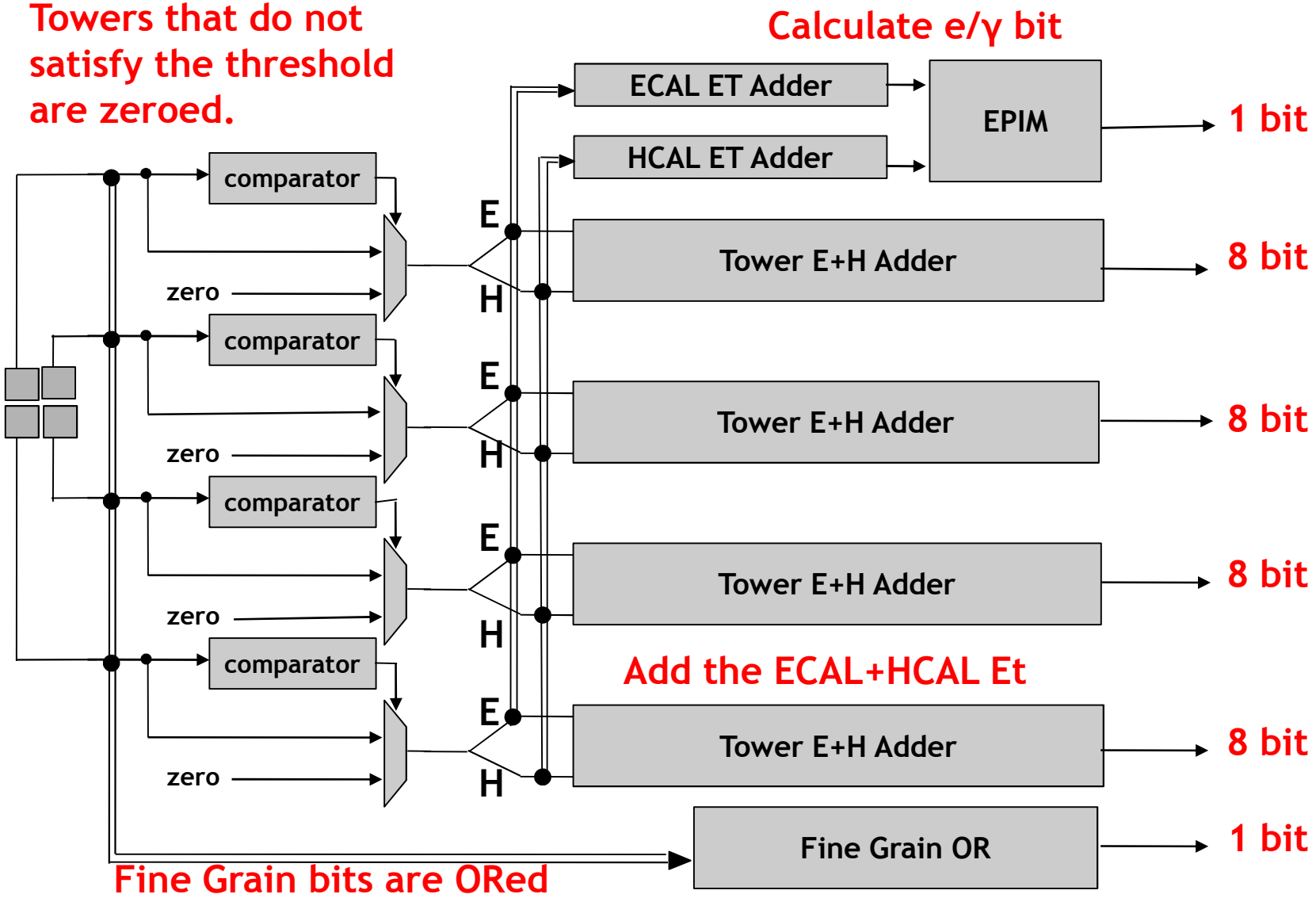
Particle Cluster Finder

- Applies thresholds on the towers
- Creates a 2x2 cluster at each position on the lattice
- Clusters overlap by one tower in eta/phi
- Calculates Electron/Photon ID bit
  - Denotes if the cluster is Photon/Electron like
- Applies OR of the finegrain bits
- Sums the ECAL and HCAL energy for each tower of the cluster



Particle Cluster Finder Logic

Towers that do not satisfy the threshold are zeroed.



Particle Cluster Finder

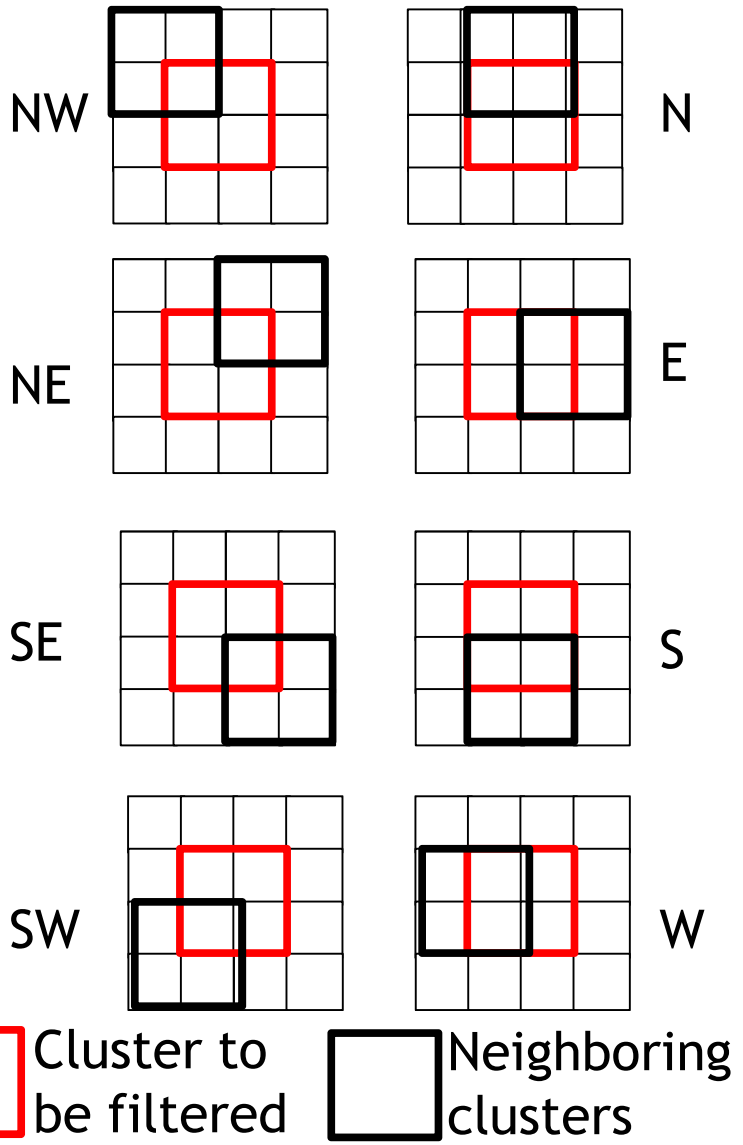
- **Particle Cluster Finder**
  - **Synthesized for a 200 MHz clock (5 ns cycle time)**
  - **Latency of four cycles (20 ns @ 200 MHz)**

Resource utilization for Particle Cluster Finder theTX240T FPGA

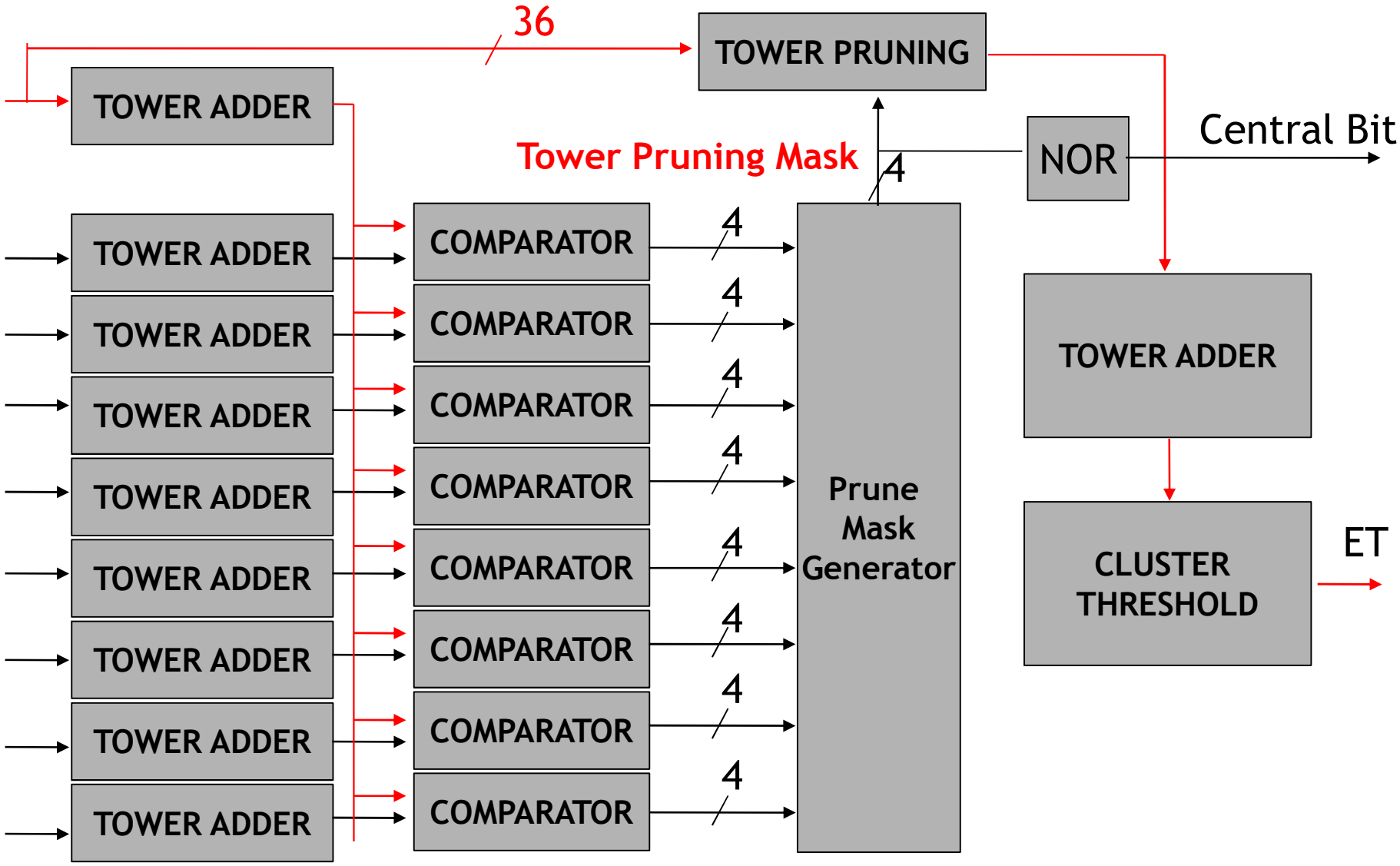
Resource	8 x 8 Grid	8 x 16 Grid	16 x 16 Grid
Virtex-5 Slices	10%	20%	39%
BRAMs	14%	27%	53%

Cluster Overlap Filter

- Compare Cluster ET with neighbor ET
  - If main cluster is less energetic remove the overlapping towers
- After pruning, sum all the towers to obtain cluster ET
- Apply a threshold to the resulting cluster
- Assign a bit to the clusters that were not pruned
  - Local Maxima!



Cluster Overlap Filter Logic



Cluster Overlap Filter Results

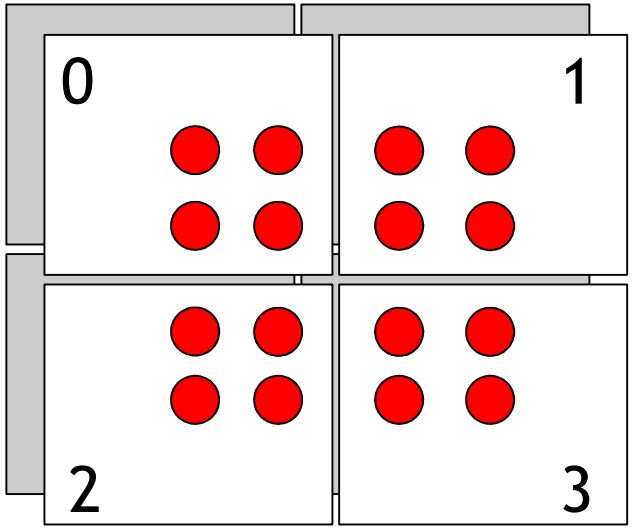
- **Cluster Overlap Filter**
  - **Synthesized for a 200 MHz clock (cycle time of 5 ns)**
  - **Latency of four cycles (20 ns @ 200 MHz)**
    - Operates in parallel with EPIM
  - **No DSP48E or Block RAM resources needed**

Virtex-5 Slice Utilization for Cluster Overlap Filter

FPGA	8 x 8 Grid	8 x 16 Grid	16 x 16 Grid
LX110T	14%	29%	58%
TX240T	7%	14%	27%

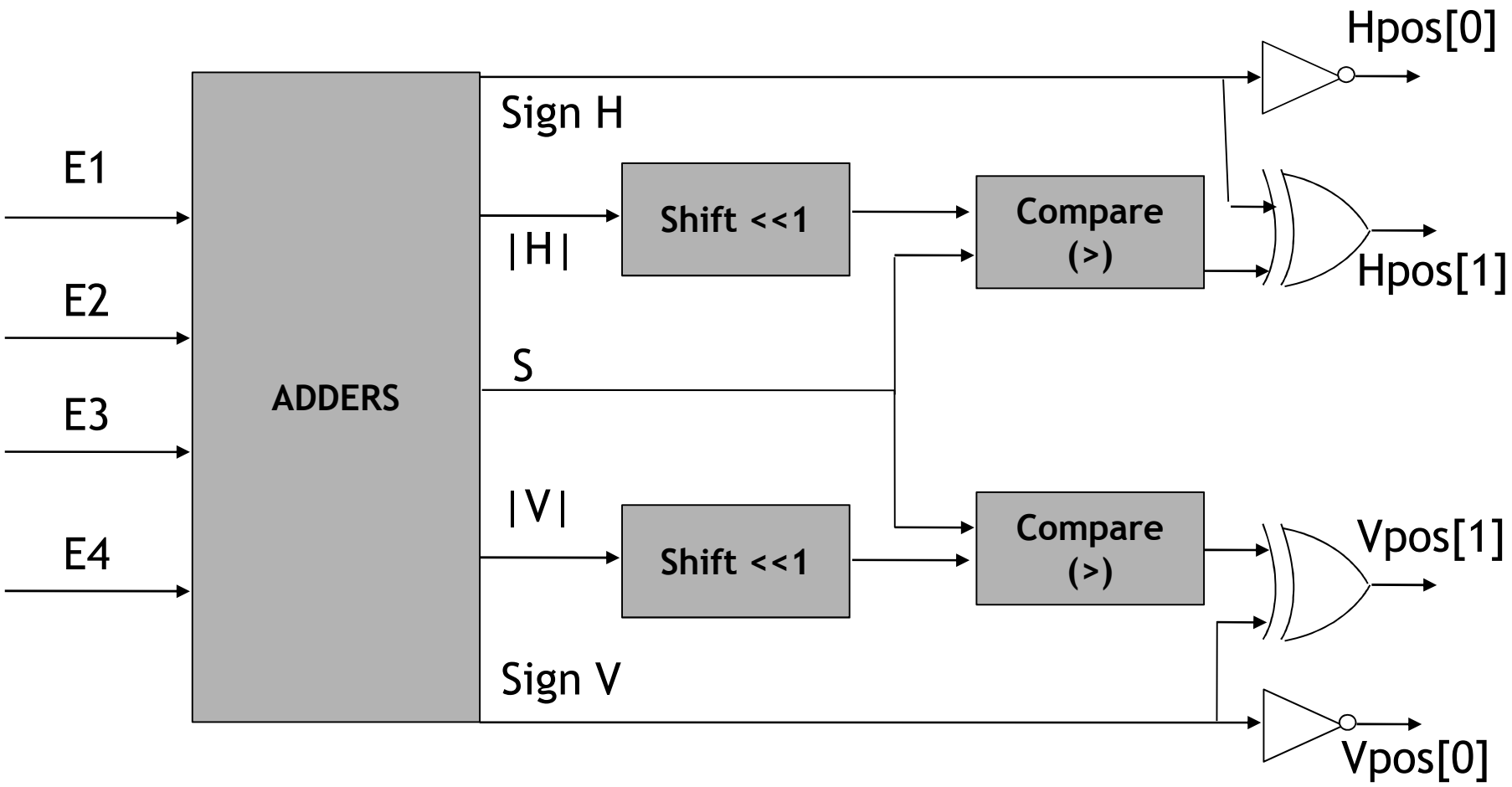
Cluster weighting

- **Weights the cluster to provide position resolution of ½ tower**
  - **Results in one of the depicted 16 points in the cluster**
- **Algorithm**
  - **Calculate horizontal and vertical energy sums**
    - H = E1+E3-E0-E2
    - V = E2+E3-E0-E1
    - S = E1+E2+E3+E4
    - Hpos = H/S, Vpos = V/S



- **No division is needed**
  - **-1<Hpos<-0.5, -0.5<Hpos<0, 0<Hpos<0.5, 0.5<Hpos<1.0**
  - **-1<Vpos<-0.5, -0.5<Vpos<0, 0<Vpos<0.5, 0.5<Vpos<1.0**

Cluster Weighting Logic



- **Signs of H and V plus magnitude comparisons determine Hpos and Vpos (each 2 bits)**

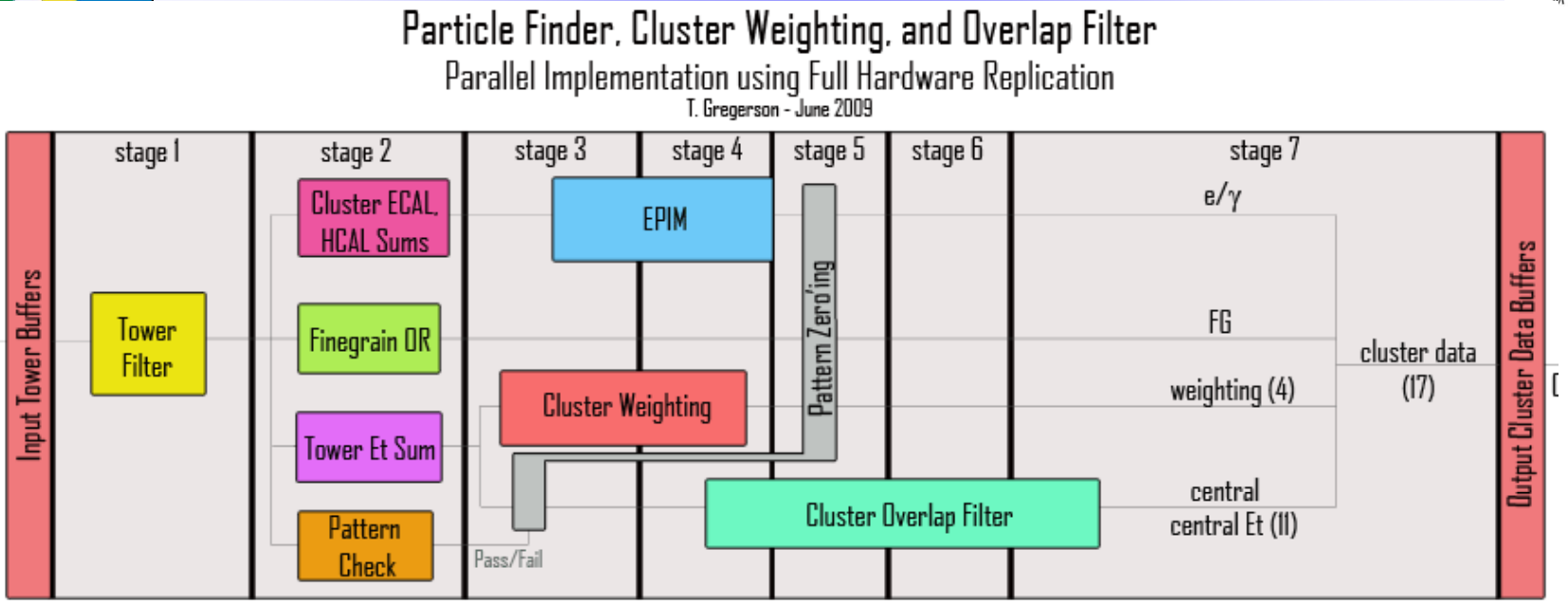
Cluster Weighting Results

- **Cluster Weighting**
  - Synthesized for a 200 MHz clock (cycle time of 5 ns)
  - Latency of two cycles (10 ns @ 200 MHz)
  - Can operate in parallel with the Cluster Overlap Filter with additional hardware cost
  - No DSP48E or Block RAM resources needed

Virtex-5 Slice Utilization for Cluster Overlap Filter

FPGA	8 x 8 Grid	8 x 16 Grid	16 x 16 Grid
LX110T	11%	22%	43%
TX240T	5%	10%	20%

Combined Design



- Cluster finder, overlap filter and weighting logic can be processed in 7 pipeline stages (without buffering I/O)
  - Latency of just 35 ns
  - Lots of hardware is shared
  - Parallel computations decrease delay
- All designs tested with physics patterns from emulator

Combined Resource Estimates

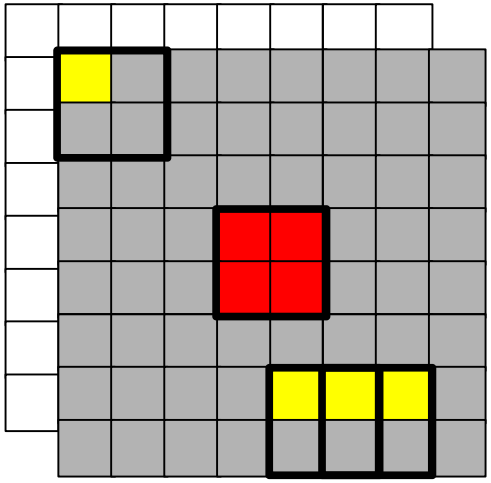
- **Estimated resources are given in the table below**
  - **Includes RocketIO, buffers, particle cluster finder, overlap filter, and cluster weighting**
  - **We plan to examine other grid sizes and FPGA devices**

Overall Resource Utilization on TX240T FPGA

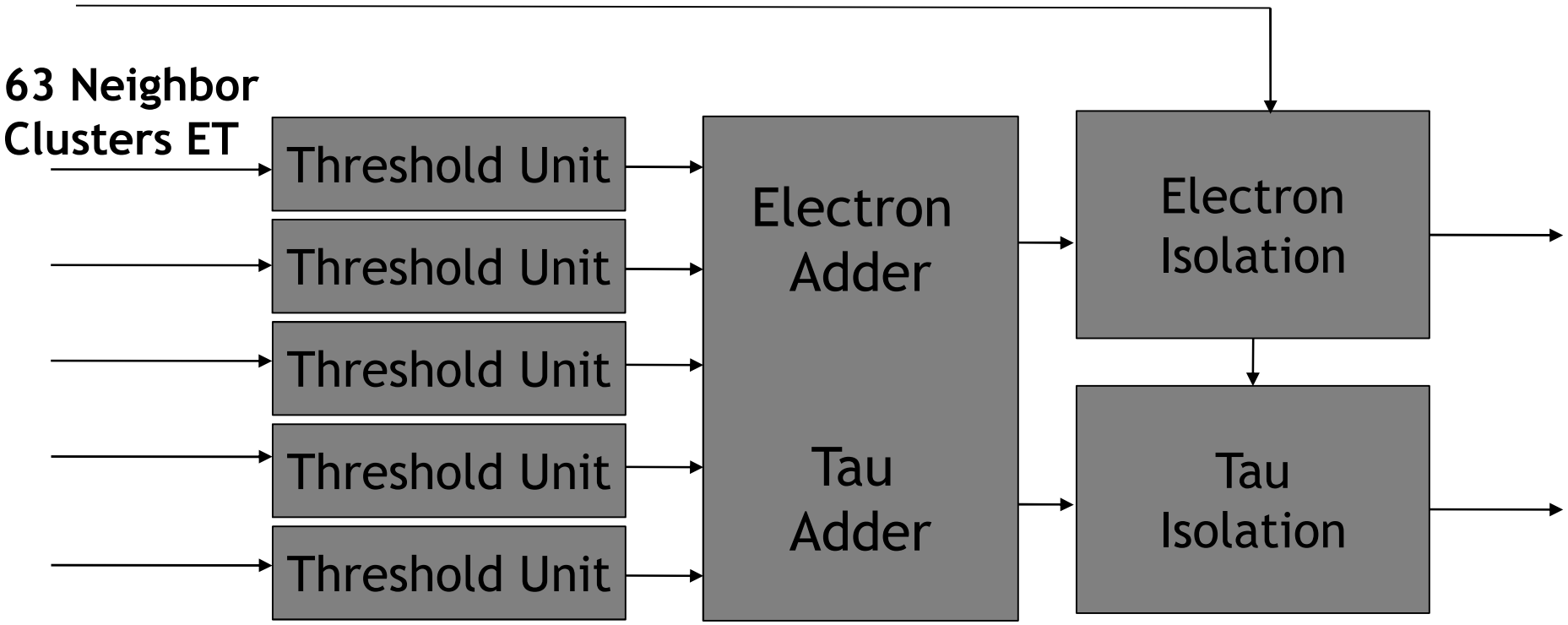
Resource	8 x 8 Grid	8 x 16 Grid	16x 16 Grid
RocketIO Links	25%	46%	83%
Virtex-5 Slices	27%	54%	105%
Block RAMs	14%	27%	53%

Cluster Isolation

- Runs on a 8x8 lattice of filtered clusters
- Counts the number of clusters over a threshold around the central cluster.
  - ET > electron threshold
  - ET > tau threshold
- Adds all the electron/tau threshold bits to obtain electron/tau isolation count
- Uses the central cluster ET and the electron/tau isolation count to obtain the electron/tau isolation bit
- Outputs two isolation bits per cluster



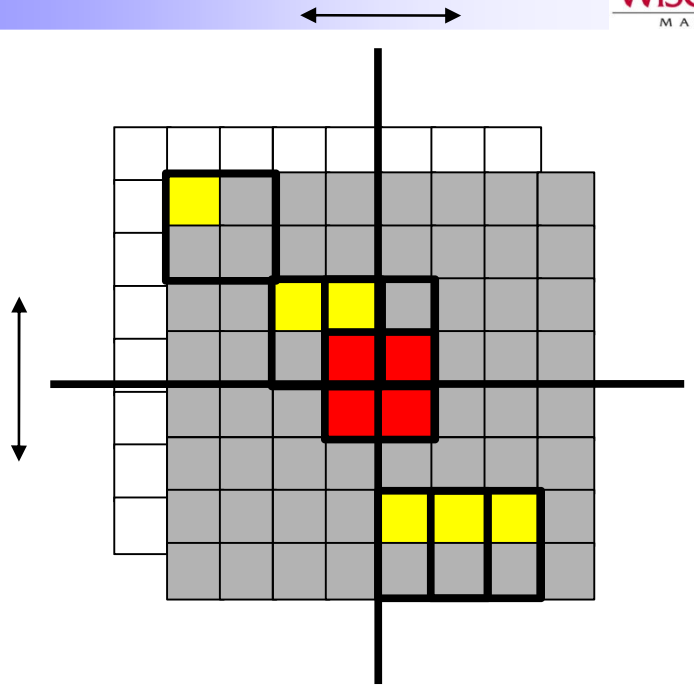
Cluster Isolation Logic



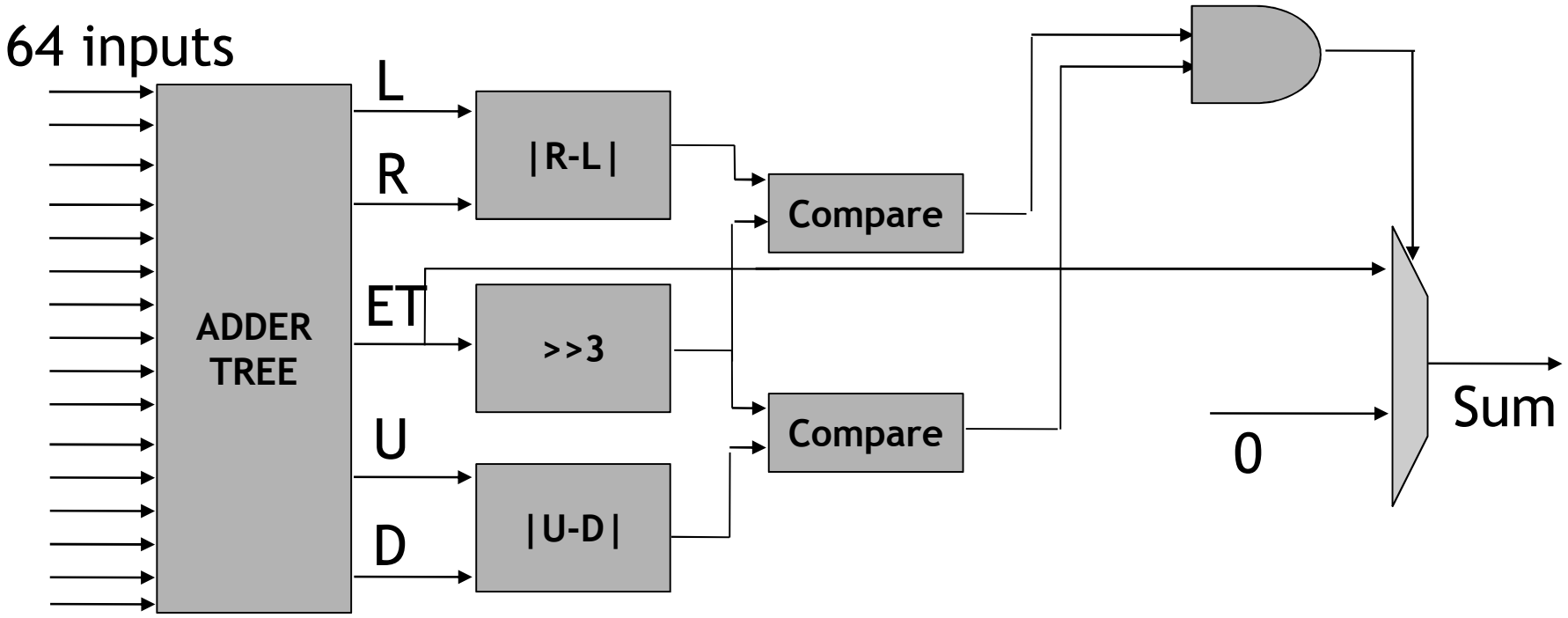
- **Two approaches for Electron/Tau Isolation Modules**
  - Use compressed Central ET plus the count to lookup isolation bits
  - Count < A+ B x CentralET + C x CentralET^2
- **Implementation and evaluation in progress**

Jet Finder

- Runs on 8x8 lattice of filtered clusters
- Starts from a local maximum
  - Central bit set
- Calculate Half Sums
  - UD = | Up – Down |
  - RL = | Right – Left |
  - ET = Total Sum
- Check ratios
  - LR/ET < c AND UD/ET < c
- No need to divide, just multiply-compare or shift-compare
  - e.g. V/ET < 12.5% OR V < (ET >> 3)

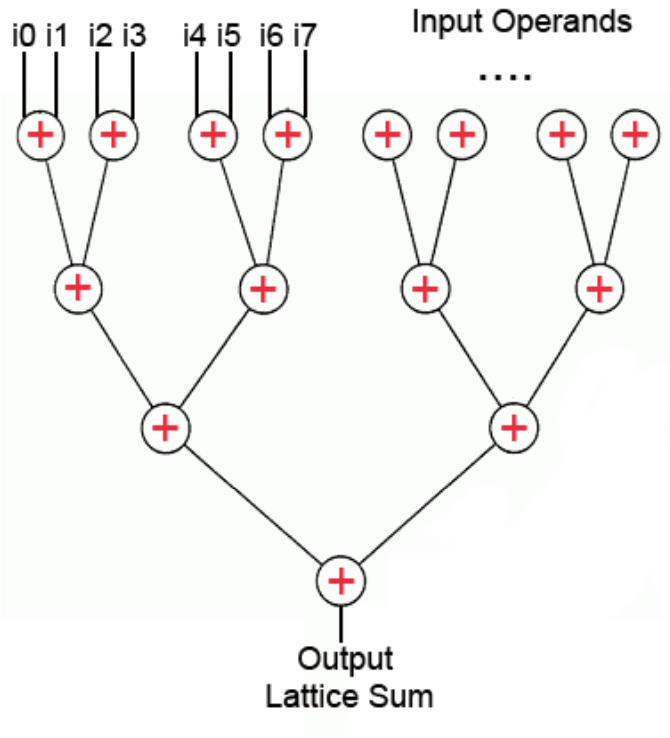


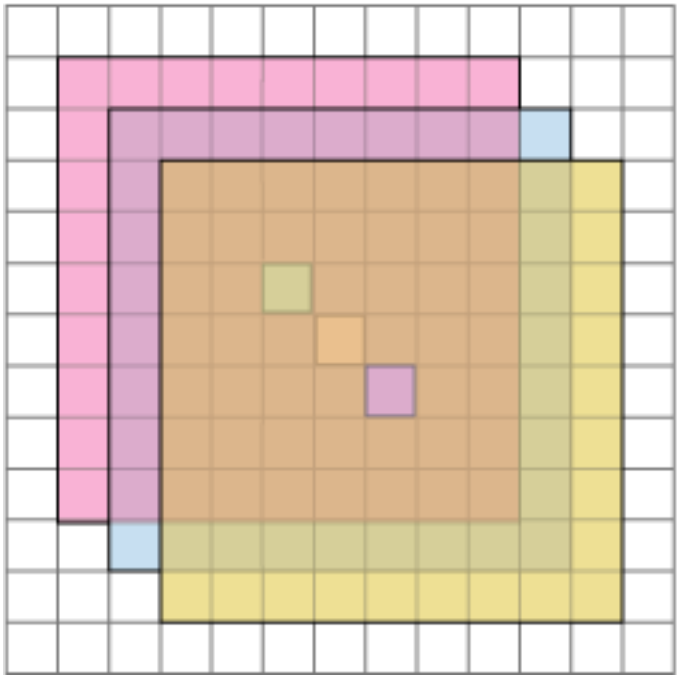
Jet Finder Logic



- 64-input adder tree
  - Computes partial sums L, R, U, and D and total sum ET
  - Overlapping lattices can share adder tree hardware
- Implementation and evaluation in progress

Adder Tree Design



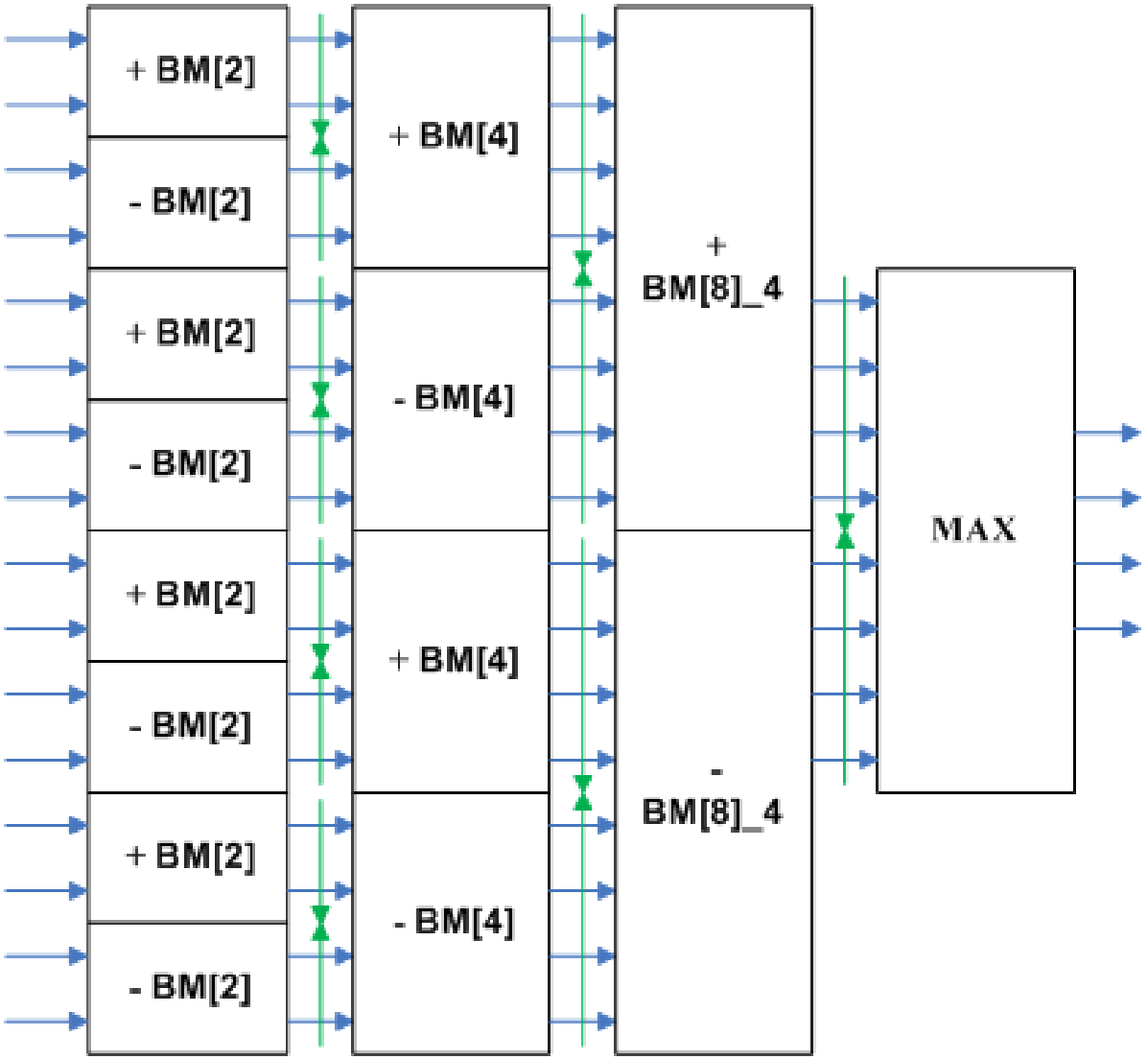


- Have logarithmic delay
- Utilized in cluster isolation, jet finding, and MET / MHT / SumEt calculations
- Lots of opportunity for hardware sharing
- Preliminary designs are being investigated

High-Speed Sorters

- **Electrons, taus, and jets all need to be sorted**
  - Several other CMS subsystems also perform sorting
  - Data to be sorted often has a corresponding position
  - Number of inputs, outputs, and data width can vary
- **Develop modular high-speed sorters**
  - Allow the number of inputs/outputs, data width, and pipeline depth to be varied
  - Utilize bitonic sorting algorithm optimized for few outputs than inputs
  - Develop hierarchical designs that build larger sorters from smaller components
  - Focus initially on *n*-to-4 sorters with *n* a power of two, but techniques work for different numbers of outputs

16-to-4 Sorting Unit



n-to-4-output Bitonic Sorting

Sorting Unit	Structure (level1 + level2 + …+ levelm)	# of Stages (Latency)	# of Comparators
8-to-4	BM[2] + BM[4] + MAX	4	16
16-to-4	BM[2] + BM[4] + 1×BM[8]_4 + MAX	7	44
32-to-4	BM[2] + BM[4] + 2×BM[8]_4 + MAX	10	100
64-to-4	BM[2] + BM[4] + 3×BM[8]_4 + MAX	13	212
128-to-4	BM[2] + BM[4] + 4×BM[8]_4 + MAX	16	436
256-to-4	BM[2] + BM[4] + 5×BM[8]_4 + MAX	19	884

Number of stages: $3 \times \log_2(n) - 5$

Number of comparators: $3.5 \times n - 12$

Synthesis Results

Synthesis uses XST 10.1 SP3 on a V5-TX240T FPGA
Comparator Data Width: 10 bits

Sorting Unit	Pipeline Depth	Estimated Frequency (MHz)	Slice LUTs	Slice Registers	End-to-end Latency (ns)
16-to-4	7	427.6	1200	760	**16.37**
16-to-4	4	232.2	1200	440	17.23
16-to-4	3	160.6	1200	280	18.67
32-to-4	10	427.6	2720	1720	23.38
32-to-4	5	232.2	2720	760	**21.56**
32-to-4	4	160.6	2720	600	24.89
64-to-4	13	427.6	5760	3640	30.39
64-to-4	7	232.2	5760	2040	**30.15**
64-to-4	5	160.6	5760	1240	31.12
128-to-4	16	427.6	11840	7480	37.41
128-to-4	8	232.2	11840	3320	**34.46**
128-to-4	6	160.6	11840	2520	37.34
256-to-4	19	427.6	24000	15160	44.42
256-to-4	10	232.2	24000	8440	**43.07**
256-to-4	7	160.6	24000	5080	43.56

Dataflow Interchange Format Modeling

- **CMS Trigger algorithms should be**
  - **Implemented correctly and quickly, achieve high performance, and be rapidly updateable**
- **Describe modules in The DIF Language (TDL) for**
  - **Formal descriptions of behavior**
  - **A platform independent *golden model* of the application**
  - **A foundation for high level performance analysis**
- **Modules in the calorimeter trigger that have been modeled include:**
  - **Particle Cluster Finder (Cluster Threshold & Cluster Computation)**
  - **Cluster Overlap Filter with Cluster Weighting**
  - **Jet Reconstruction (preliminary version)**
  - **Cluster Isolation (in progress)**

DICE and Unit Testing

- Unit tests are small, local tests that
  - Ensure a particular module is correct
  - Reduce time spent later on system level testing and debugging
- Developed techniques to support unit tests
  - Automatic DIF unit test bench generation
  - Automatic DIF application-level test bench generation
  - SVN repository for firmware, tests, DIF models, and documentation
  - Automatic nightly testing of all modules (DIF, Veriog, C++)
- DSPCAD Integrative Command Line Environment (DICE) enables design frameworks used in trigger to share testbenches
  - C++, Verilog, DIF
  - DICE Website: http://www.ece.umd.edu/DSPCAD/projects/dice/dice.htm
- Cross platform unit tests created for
  - Particle Cluster Finder (Cluster Threshold & Cluster Computation)
  - Cluster Overlap Filter with Cluster Weighting

Next Steps

- **Implement the rest of the Calorimeter Trigger**
  - **Cluster Isolation and Jet Reconstruction**
  - **MET / MHT / Et Sum Calculation**
  - **Particle Sorters**
- **Perform more in-depth testing and analysis of the designs**
  - **FPGA resources, latency, and clock frequency**
  - **Design partitioning across multiple FPGAs**
  - **Enhance testing and analysis frameworks using SVN, DICE and DIF**
- **Prototype the Calorimeter Trigger designs on FPGA hardware**

Questions?

Questions?

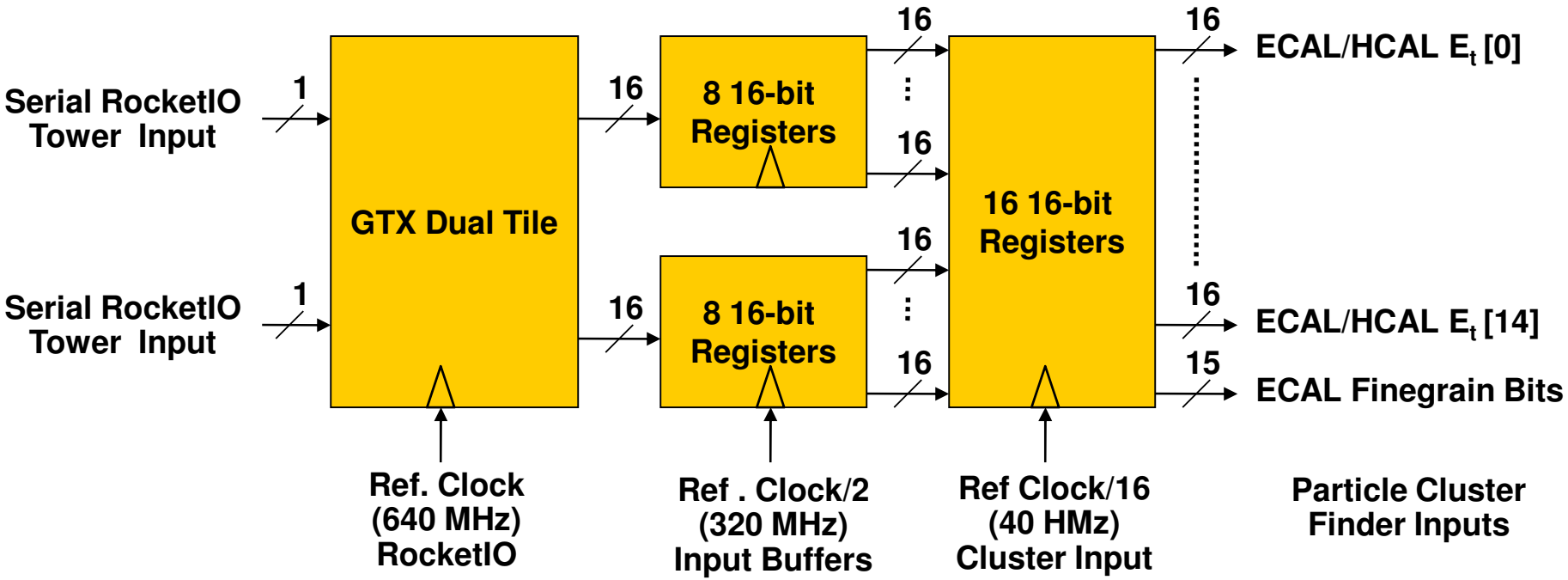
Latency Estimates

- **Estimated latencies are given in the table below**
  - **Clock rate of 200MHz (cycle time of 5 ns)**
  - **Cluster Overlap Filter operates in parallel with part of Particle Cluster Finder**

Estimated Latencies on TX240T FPGAs

Component	Latency (cycles)	Latency (ns)
Input RocketIO and Buffers	15	75
Particle Finder, Overlap Filter, Cluster Weighting	7	35
Output Rocket IO and Buffers	10	50
Total Estimated Latency	**32**	**160**

Rocket IO and Buffering



- Our initial design on TX240T FPGAs uses Xilinx's Aurora protocol for RocketIO inputs
- Each GTX Dual Tile de-serializes 2x8x16 = 256 bits every 25ns.
- 16 16-bit registers store data for 15 towers for 25ns.