

Potential Enhancements to the XS Trigger Firmware

- Current Implementation
- Potential Enhancements
 - Functional overview
 - Resource Requirements
- Conclusions

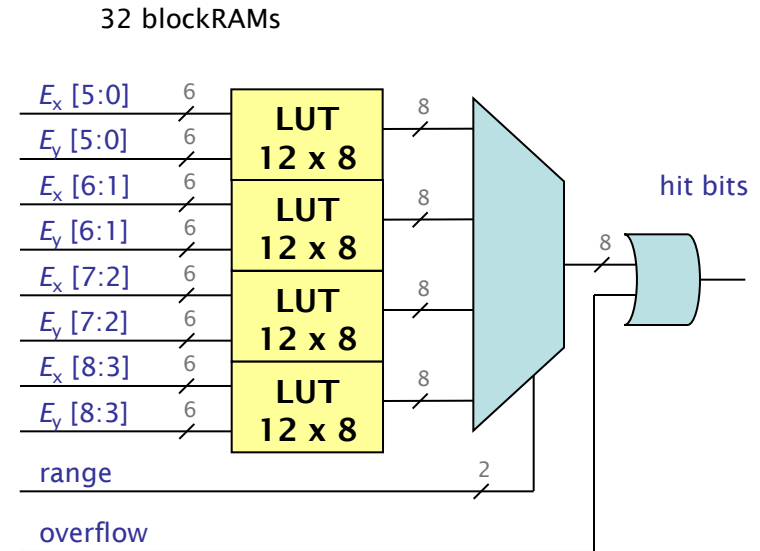
(Missing Energy = E_T^{miss} , MET, XE
Missing Energy Significance = XS)

Implementation

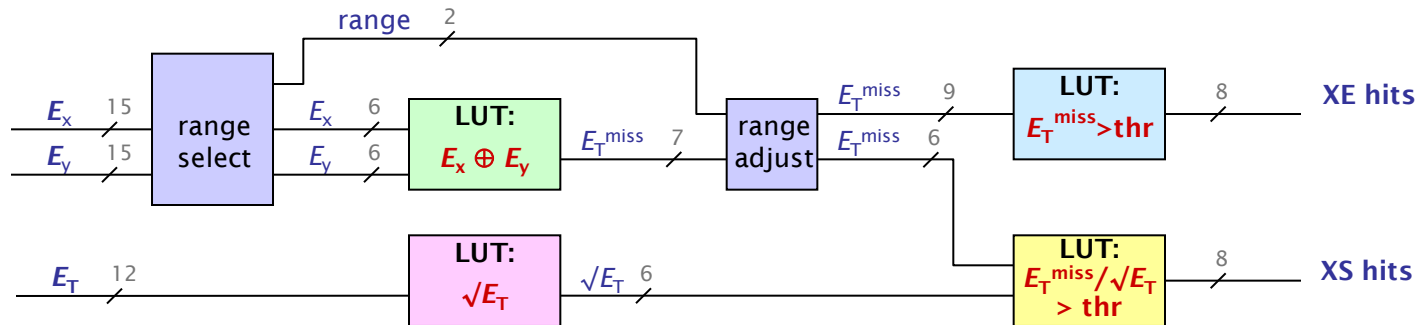
- XS is computed in the System FPGA of the CMM-E
 - Receives 15-bit data for E_x and E_y
- When considering enhancements to this trigger, the first limits we reach are due to the internal resources of this FPGA
- XS algorithm features square roots, division (or multiplication by dynamic numbers)
- System FPGA = XCV1000E
 - The only way to implement such operations in a Virtex-E device is via LUTs
 - LUTs of the required size must be implemented in blockRAM
- BlockRAM is 4K memory block with flexible configuration
 - Used for LUTs, RAM, FIFO...
- Current blockRAM usage in CMM-E System FPGA:
 - LUTs for XE and XS: 22 blocks
 - DAQ & ROI buffering 32 blocks
 - spare: 42 blocks

The Original XE Implementation

- 4 x 6-bit ranges of E_x and E_y extracted from incoming 15-bit values
- For each range, a LUT receives E_x and E_y
 - calculates E_T^{miss} (vector sum)
 - sets hit bits
 - single operation
- Results from most appropriate range then selected, determined by most significant populated bit in original 14-bit E_x and E_y values.
- 'Out of band' overflow signal: hit results saturated at end of RT path.
- Parallel architecture reduces latency at expense of resources.



The Current XE + XS Implementation

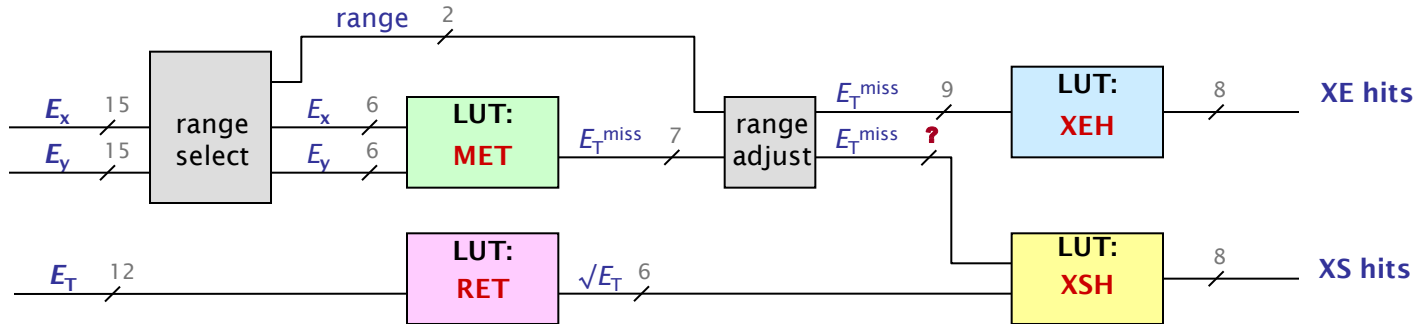


- Move to efficient use of resources at expense of latency
- Split calculation of E_T^{miss} & thresholding into separate LUTs
 - Fewer LUTs required — range adjustment handled using combinatorial logic
 - E_T^{miss} results used in XE and XS triggers
 - XS receives 6-bit E_T^{miss} saturating at 63 GeV — leave greater values to XS trigger
- MET sig trigger implemented using 2 further blockRAMs: (\sqrt{i}) and ($j/k > threshold$)
- No propagation of overflow signals; propagate saturated data values

Potential Improvements to XS

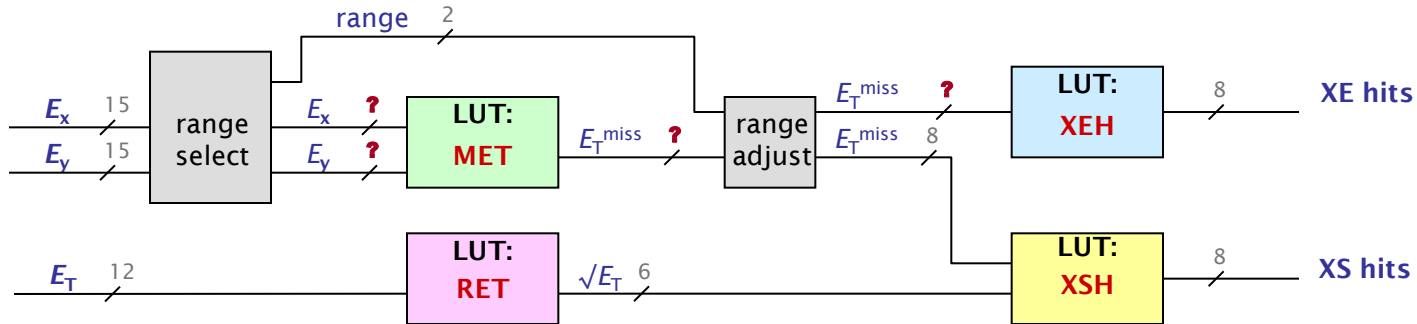
- Concern expressed at 63 GeV ceiling of E_T^{miss}
- Currently XS receives 6 bits of E_T^{miss} :
 - derived from 7 bits of data + 2 bits of range used for XE
⇒ 6 LSBs, saturate if bit 7 or range asserted
- Potential upgrade 1:
 - Keep $E_T^{\text{miss}} \rightarrow$ XS at 6 bits, double range by halving precision:
 - drop LSB, use bits (6:1), saturate if range asserted
 - trivial *firmware* change
 - simplifies logic
 - very unlikely to have repercussions for latency
- Potential upgrade 2:
 - Increase no. bits used for $E_T^{\text{miss}} \rightarrow$ XS
 - Necessitates increase in width of 2nd XS LUT
 - Possible increase in latency (< 1 tick as multiphase clocks are used)
 - Note: size of LUT rises linearly with output width but exponentially with input width (input comprises address to underlying RAM)

The Current XE + XS Implementation



$E_T^{miss} \rightarrow XS$	XSH LUT		MET LUT		XEH LUT		RET LUT		Total	Spare
bits	I x O	RAM blocks	IXO	RAM blocks	IXO	RAM blocks	IXO	RAM blocks	RAM blocks	RAM blocks
6	12x8	8	12x7	7	9x8	1	12x6	6	22	42
7	13x8	16	12x7	7	9x8	1	12x6	6	30	34
8	14x8	32	12x7	7	9x8	1	12x6	6	46	18
9	15x8	64	12x7	7	9x8	1	12x6	6	78	-14

The Current XE + XS Implementation



$E_T^{miss} \rightarrow XS$	XSH LUT		MET LUT		XEH LUT		RET LUT		Total	Spare
bits	I X O	RAM blocks	IXO	RAM blocks	IXO	RAM blocks	IXO	RAM blocks	RAM blocks	RAM blocks
8	14x8	32	12x7	7	9x8	1	12x6	6	46	18
8	14x8	32	14x8	32	10x8	2	12x6	6	72	-8

- Potential to expand precision of XE trigger to 8 bits + range? **No.**
- Similarly, can't expand range of $\sqrt{E_T} \rightarrow XS$ (without sacrifices elsewhere)

Implications

- Compared to initial addition of XS trigger, all of these upgrades are minor f/w modifications
 - but still involve new f/w, new LUT contents, new on-line & off-line s/w
- How much spare block RAM do we wish to retain?
 - Currently: 22 blocks used for XE + XS triggers
32 blocks used on RoI + DAQ paths
42 blocks spare
96 blocks total
 - Recent implementation of XS required 8 new RAMs on RoI + DAQ paths for 12 new bits of data
- 18 spare RAM blocks used left by 8-bit $E_T^{\text{miss}} \rightarrow \text{XS}$ could be consumed very quickly
- 7-bit $E_T^{\text{miss}} \rightarrow \text{XS}$ is the safer option
 - (Nobody has actually proposed more than this yet)

Slight Aside About Block RAMs

- VME access to some of the new LUTs features unwritable bits
 - Always read as zero
 - eg, new MET LUT (12x7) = RAM, 4K deep x 7 bits wide
 - VME access 16-bit via word:



- Unfortunately software can't handle unwritable RAM bits
- However, these bits are unwritable because there is no RAM behind them
- To eliminate this feature would require addition of 3 RAM blocks
 - containing redundant and potentially confusing information

Summary and Conclusion

- Feasible to double range of XS trigger to E_T^{miss} of 127 GeV
 - Trivial f/w modification if done by halving precision
- Potential to quadruple range to 255 GeV
 - Leaves few spare RAM blocks given how much CMM-E f/w has changed this far into active life
- A slight increase in latency may be necessary
- Firmware changes for these modifications are minor
 - Few days effort at worst — ie, if design needs to be retimed
- Firmware design effort is only part of the picture. Any change also requires
 - Modification of specification
 - Modification of on-line software
 - Modification of off-line software
 - Simulation of firmware
 - Testing of software and firmware in situ
- Last modification to implement XS functionality took considerable work by number of people
- We're doing the right thing by talking about this before rushing into it