

- 10 a: Discussions of milestones deferred to session 5
- b i: Muon needs (especially at the very front ends) are significantly different from pixel. Use the same DSPs, FPGAs operating system, development tools share as much FPGA and DSP code as is practical.
  - b ii: A big question that requires study. Put L1 in collision hall to save on links. Reliability concerns. System must be reliable wherever it is placed. Is there room? We get dibs on shielded region beside magnet.
  - b ii: Possible strategies for increased reliability - CRC embedded into RAM, checksums on DSP code.
  - c i: Time order data. Add necessary time bits. Possible additional ordering (e.g. in phi).
  - c ii: Digital cluster finding at L1
  - c ii: Use verilog/vhdl to do FPGA. Performance is lower, but device independent. Performance requirements limit choices of devices to largest/fastest available (as of to-day). Prove algorithm in simulation. Verify with prototype.

- d ia: Segment processor - 3 candidate algorithms
  - Carnegie algorithms - proponents no longer here
  - Sweep algorithm - requires x and y to be the same size.
  - Queuing simulations needed for assigning figure of merit to leading contenders - BB33 and hybrid of sweep and BB33 algorithm.
  
- d ib: Track processor:
  - assuming a given DSP, trials of different algorithms are much less effort than FPGA tests.
  
- d iia: Rely on FPGA for simulation. Experience is that simulators are conservative. Once architecture is debugged in simulation, validate with a hardware prototype.
  
- d iib: Run code on DSP tools. Simulators are reasonably accurate ("cycle accurate" simulation is an exaggeration). Do a benchmark on an evaluation board to very speed.
  
- d iii: Baseline is TI DSP. Some benchmarks already done. Some further checks done on Intel uP. TI seems reasonably well suited to the task.  
Selection is cost driven. DSP are cheaper / MIP but less familiar. If general purpose processors become "cheap" then their use will be considered.
  
- d iv: use DSP development tools initially. Perhaps a DSP RTOS later. System setup would be private communication from a networked host processor to a number of DSPs. e.g. VxWorks VME processor talking to a crate full of boards.
  
- d v: a - no, b - yes.

