



Contribution ID: 12

Type: **not specified**

## The First Prototype for the FastTracker Processing Unit

*Thursday, 3 May 2012 20:00 (1 hour)*

Modern experiments search for extremely rare processes hidden in much larger background levels. As the experiment complexity and the accelerator backgrounds and luminosity increase we need increasingly complex and exclusive selections. We present the first prototype of a new Processing Unit, the core of the FastTracker processor for Atlas, whose computing power is such that a couple of hundreds of them will be able to reconstruct all the tracks with transverse momentum above 1 GeV in the ATLAS events up to Phase II instantaneous luminosities ( $5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ ) with an event input rate of 100 kHz and a latency below hundreds of microseconds. We plan extremely powerful, very compact and low consumption units for the far future, essential to increase efficiency and purity of the Level 2 selected samples through the intensive use of tracking.

This strategy requires massive computing power to minimize the online execution time of complex tracking algorithms. The time consuming pattern recognition problem, generally referred to as the “combinatorial challenge”, is beat by the Associative Memory (AM) technology [2] exploiting parallelism to the maximum level: it compares the event to pre-calculated “expectations” or “patterns” (pattern matching) at once looking for candidate tracks called “roads”. This approach reduces to linear the typical exponential complexity of the CPU based algorithms. The problem is solved by the time data are loaded into the AM devices.

We describe the board prototypes that face the very challenging aspects of the Processing Unit: a huge amount of detector clusters (“hits”) must be distributed at high rate with very large fan-out to all patterns (10 Millions of patterns will be located on 128 chips placed on a single board) and a huge amount of roads must be collected and sent back to the FTK post-pattern-recognition functions. The Processing Unit consists of a 9U VME board, the AMBoard, controlled by an AUX card on the back of the crate. The AMBoard has a modular structure consisting of 4 mezzanines, the Local Associative Memory Banks (LAMB). Each LAMB contains 32 Associative Memory (AM) chips, 16 per side. The proto - AUX card provides hits on 8 buses for a total of 12 Gbits/sec to the AMBoard through 12 high frequency serial links and will sink the found roads through other 16 high frequency serial links (24 Gbits/sec). A special P3 connector allows the communication between the front and rear boards placed on the same VME slot. A custom board profile has been studied and simulated at the CAD to guarantee a perfect board-to-board closure of the P3 connector without a backplane support in that region. A network of high speed serial links characterize the bus distribution on the AMBoard. The hit buses are fed to the four LAMBs and distributed to the 32 AM chips on the LAMB, through fanout chips. The LAMB realization has represented a significant technological challenge, due to the high density of chips allocated on both sides, and to the use of advanced packages and high frequency serial links.

**Primary authors:** LANZA, Agostino (Universita e INFN (IT)); MAGALOTTI, Daniel (Universita e INFN (IT)); PIENDIBENE, Marco (Sezione di Pisa (IT)); CITTERIO, Mauro (Universita degli Studi e INFN Milano (IT))

**Co-authors:** ANNOVI, Alberto (Istituto Nazionale Fisica Nucleare (IT)); STABILE, Alberto (Universita degli studi di Milano); ANDREANI, Alessandro (Universita degli Studi e INFN Milano (IT)); ALBERTI, Fabrizio (INFN Sezione di Milano (INFN)); TANG, Fukun (University of Chicago (US)); TOMPKINS, Lauren Alexandra (Lawrence Berkeley National Lab. (LBNL)); BERETTA, Matteo Mario (Istituto Nazionale Fisica Nucleare (IT)); SHOCHET,

Mel (University of Chicago (US)); BOGDAN, Mircea (The University of Chicago); GIANNETTI, Paola (Sezione di Pisa (IT))

**Presenter:** MAGALOTTI, Daniel (Universita e INFN (IT))

**Session Classification:** Posters