ASSOCIATIVE MEMORY COMPUTING POWER AND ITS SIMULATION

The Challenge

During LHC high luminosity running, the high rates, multiplicities and energies of particles produced in the p-p collisions pose a unique challenge to the ATLAS Trigger. Online track reconstruction is proved to be an effective solution in separating the most interesting events from the extremely large pile-up background.

The ATLAS FTK Processor

FTK will operate at full level 1 trigger output event rate (100KHz) and will provide high quality global track reconstruction at the beginning of level 2 trigger processing [1].

FTK uses a set of independent engines, each working on a different region of the Inner Detector (ID):
- 64 independent p-towers
- 512 parallel processing units

FTK operates at full level 1 trigger output event rate (100KHz) and will provide high quality global track reconstruction at the beginning of level 2 trigger processing [1].

FTK uses a set of independent engines, each working on a different region of the Inner Detector (ID):
- 64 independent p-towers
- 512 parallel processing units

The Idea

Pattern recognition is carried out by a dedicated device called Associative Memory (AM). Find corre-solution track candidates named roads. The AM comprises in parallel the event hits with all the stored pre-calculated low resolution track patterns and returns the addresses of the matching patterns.

FTK SIMULATION

FTK simulation describes the logic of the main parts inside the processor and verifies how different setups can influence the performance of the system.

The different HW steps are replicated in a detailed simulation. The main blocks are: Road Finding (RF) and Track Fitting (TF).

The memory requirement is extremely high. The simulation tasks are divided in independent jobs. Each job controls only a fraction of the system, otherwise than a single tower, in order to fit the task in standard machine.

Segmentation:
- 64 hardware towers
- 4 jobs per tower
- 256 jobs

The jobs of each tower are executed in sequence, with a partial merge at the end of the last job.

ROAD FINDING EMULATION

Road Finding Diagram

The road finding step reproduces the match between incoming clusters and the pattern bank, as the AM does. The AM emulation module is the most important and consuming part of the whole simulation.

The code needs some internal structures to store the data in the chip, the status of the match during an event and the data to be sent to the next step.

Internal Data Structures

1. Map the patterns’ list in the “In” box for each SuperStrip (SS) in each layer insert the list of the patterns’ id containing the SS.
2. For each input hit:
   - Calculate the correspondent SS, it’s the first hit search for the related patterns in the “Matched SS” table.
   - Append the hit to the list of hits for the SS or create the instance of the container.
3. When a pattern ID has hits on at least all but one layers, put it in the “Selected Patterns” table.
4. Check the “Don’t Care” (DC) bit content and retrieve the final list of hits.
5. Send the output collection to the Track Fitter.

FTK SIMULATION RESULTS

The emulation software is an important test of the hardware because it can evaluate the performance of different types of AM implementations, changing most working points that are relevant for the AM performance [1].

Reaching good time performance for this simulation is important for two main reasons:
1. The emulation will be used to verify the hardware components in order to check for errors and malfunctions of the system.
2. ATLAS needs to emulate the system at the best to parameterize the effect of FTK on quantities that are interesting in physics analysis.

Efficiency of the FTK

Efficiency of the FTK tracks with respect to offline tracks as a function of p_T with 46 and 69 pile-up events.

The ATLAS FTK Processor

AMchips array

AM Computes Power:
Each pattern:
- 4 x 32 bit comparators
Each 10 ns:
- 128 inputs x 4 = 500 K instructions
- 1 MB: sim. ~ 100 INFL/1k hits
- 1 bit of latency
- 1 MB of SRAM
- 1 MB of SRAM

Latency

FTK latency for one event at 69 pile-up.
The timing of the functional blocks is given for the region that takes the most time.

The ATLAS FTK Processor

FTK latency for 2-4 hits events with 69 pile-up.

Efficiency

Efficiency of the FTK tracks matched to truth as a function of η with 46 and 69 pile-up events.

Road Finding Diagram

The road finding step reproduces the match between incoming clusters and the pattern bank, as the AM does. The AM emulation module is the most important and consuming part of the whole simulation.

The code needs some internal structures to store the data in the chip, the status of the match during an event and the data to be sent to the next step.

Internal Data Structures

1. Map the patterns’ list in the “In” box for each SuperStrip (SS) in each layer insert the list of the patterns’ id containing the SS.
2. For each input hit:
   - Calculate the correspondent SS, it’s the first hit search for the related patterns in the “Matched SS” table.
   - Append the hit to the list of hits for the SS or create the instance of the container.
3. When a pattern ID has hits on at least all but one layers, put it in the “Selected Patterns” table.
4. Check the “Don’t Care” (DC) bit content and retrieve the final list of hits.
5. Send the output collection to the Track Fitter.

Road Finding Diagram

The road finding step reproduces the match between incoming clusters and the pattern bank, as the AM does. The AM emulation module is the most important and consuming part of the whole simulation.

The code needs some internal structures to store the data in the chip, the status of the match during an event and the data to be sent to the next step.

Internal Data Structures

1. Map the patterns’ list in the “In” box for each SuperStrip (SS) in each layer insert the list of the patterns’ id containing the SS.
2. For each input hit:
   - Calculate the correspondent SS, it’s the first hit search for the related patterns in the “Matched SS” table.
   - Append the hit to the list of hits for the SS or create the instance of the container.
3. When a pattern ID has hits on at least all but one layers, put it in the “Selected Patterns” table.
4. Check the “Don’t Care” (DC) bit content and retrieve the final list of hits.
5. Send the output collection to the Track Fitter.