**Introduction**

- LHC is to be upgraded to approximately five times design luminosity ($5 \times 10^{34}$ cm$^{-2}$s$^{-1}$) around 2021.
- Requires upgrade of ATLAS Level-1 trigger to continue to trigger efficiently on electroweak scale physics.
- Simulations studies have demonstrated that tracking information at L1 can control rates effectively (Fig. 1).

**The Double Buffer Scheme**

- Tracker readout within tight Level-1 latency is a bandwidth challenge.
- Potential solution: two hardware based levels, Level-0 (L0) + Level-1 (L1).
- L0 uses coarse calorimeter + muon data to reduce rate from 40MHz to 500kHz, and in doing so defines Regions-of-Interest (RoIs).
- Tracker is read-out on L0 accept (L0A): but only the fraction that is within the L0 RoIs.
- Tracks are then reconstructed and available at L1 to further reduce rate.
- Scheme requires double buffer architecture on tracker FE-ASICs (Fig. 2).

**Baseline Stave Readout Architecture**

- Two banks of five daisy-chained FE-ASICs feed a Hybrid Chip Controller (HCC) (Fig. 3).
- FE-ASICs communicate fixed packets of data (60 bits) at 160MB/s along daisy-chain.
- Two types of data: Regional Readout Requests (R3) for modules within L0 RoI, and L1 accept data (L1A).
- Data is transmitted from HCC along stave links to end-of-stave GBT at 160MB/s. Then via a virtual private link from GBT to ROD, and on to the Track Finder/DAQ.

**Latency Studies**

- Have used discrete event simulations to test various options for readout.
- R3 and L1A data transmitted using shared (as opposed to dedicated) links at 160MB/s is found to work well (Fig. 4 and Table 1).
- > 95% data delivered in < 5µs.
- This means that we have 320MB/s into HCC, but only 160MB/s out. Advantageous to have separate R3 and L1A FIFOs on HCC to avoid queuing of R3 data.
- Readout system is shown to be robust to increased occupancy, R3, L0A and L1A rates.

**Conclusions**

- Use of tracking information in the Level-1 decision would greatly benefit the ATLAS upgrade physics programme.
- Tracker readout at Level-1 seems feasible with a split-level hardware system using a double-buffer front-end architecture.
- Next: the challenge of pattern reconstruction and track finding.