ASIC Labor Heidelberg

Joint Beetle/OTIS Review

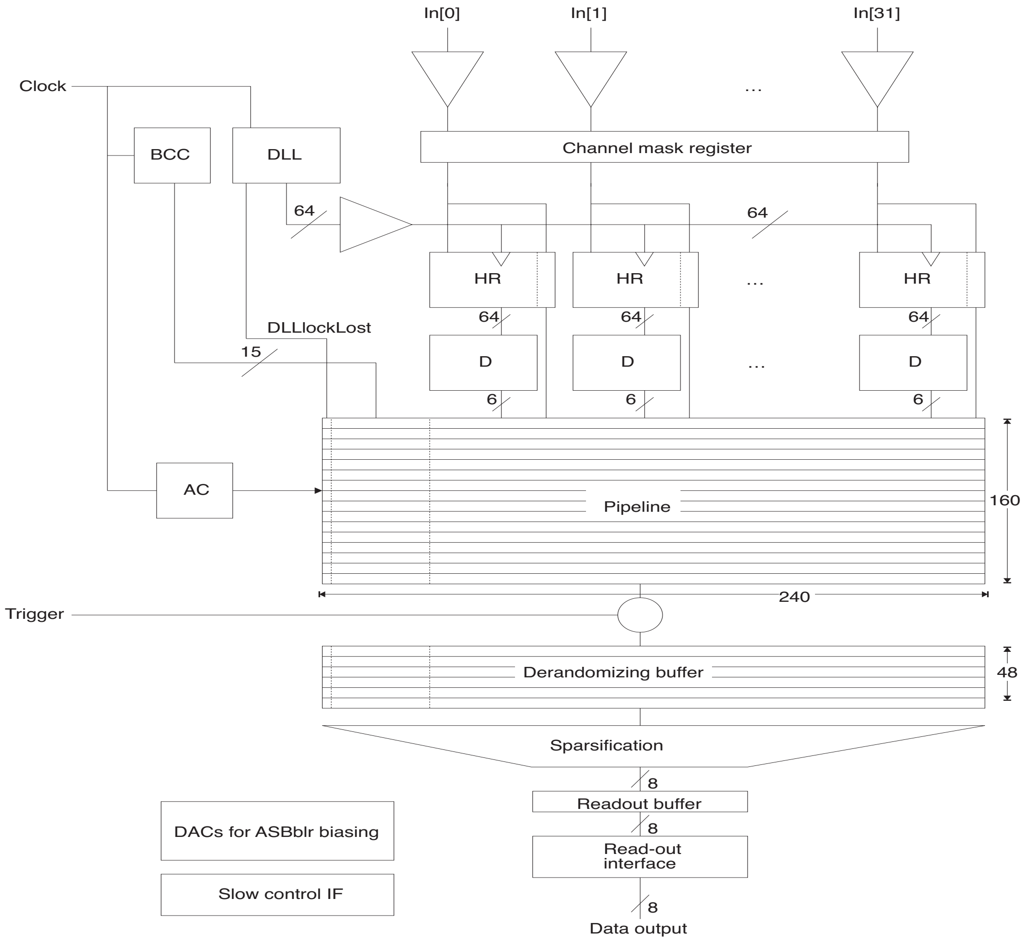
The OTIS Pipeline Control Logic

The pipeline control logic performs the following tasks:

Provide read and write pointer for the pipeline and derandomizing buffer.

Assemble and insert new data words to pipeline.

Copy data word from pipeline to derandomizing buffer upon a trigger.



The OTIS Pipeline Control Logic

Write Pointer:

Points to the pipeline location where new data is written to.

Updates synchronous to LHC clock.

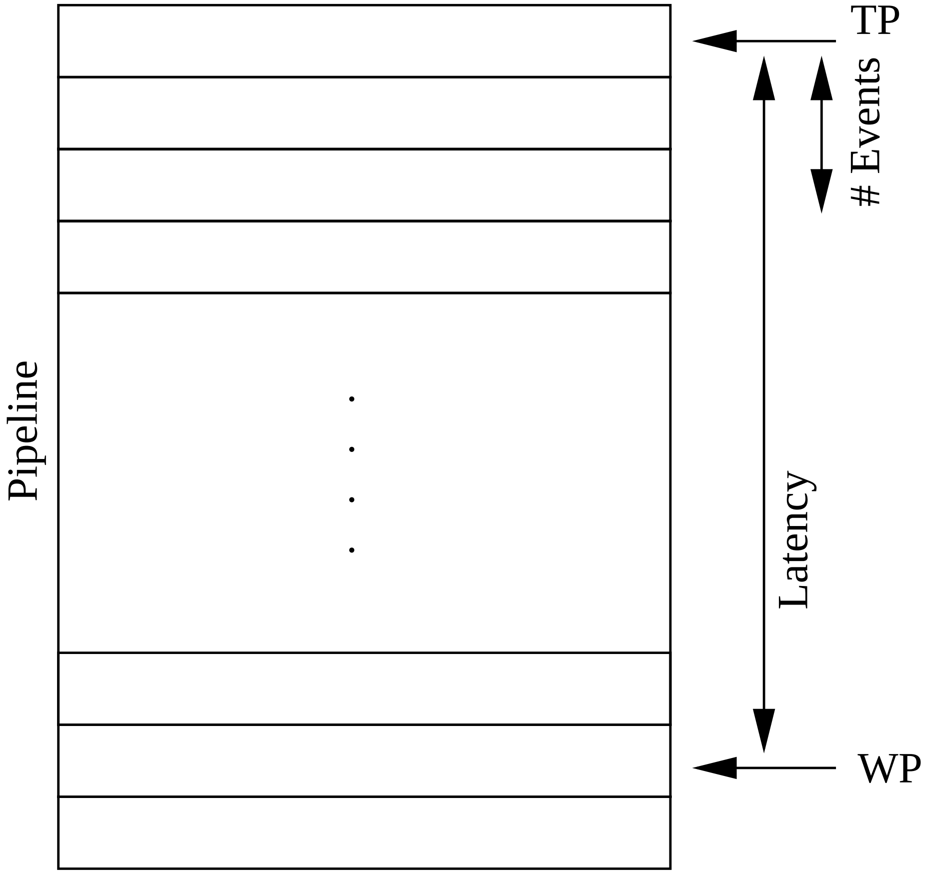
Trigger Pointer:

Points to the pipeline location from where data is copied to DBuf upon a trigger.

Position of TP is a function of WP, latency and pipeline length.

Updates synchronous to LHC clock.

−> guarantees correct distance between TP and WP and correct data (after one complete cycle) if WP had a bit flip.



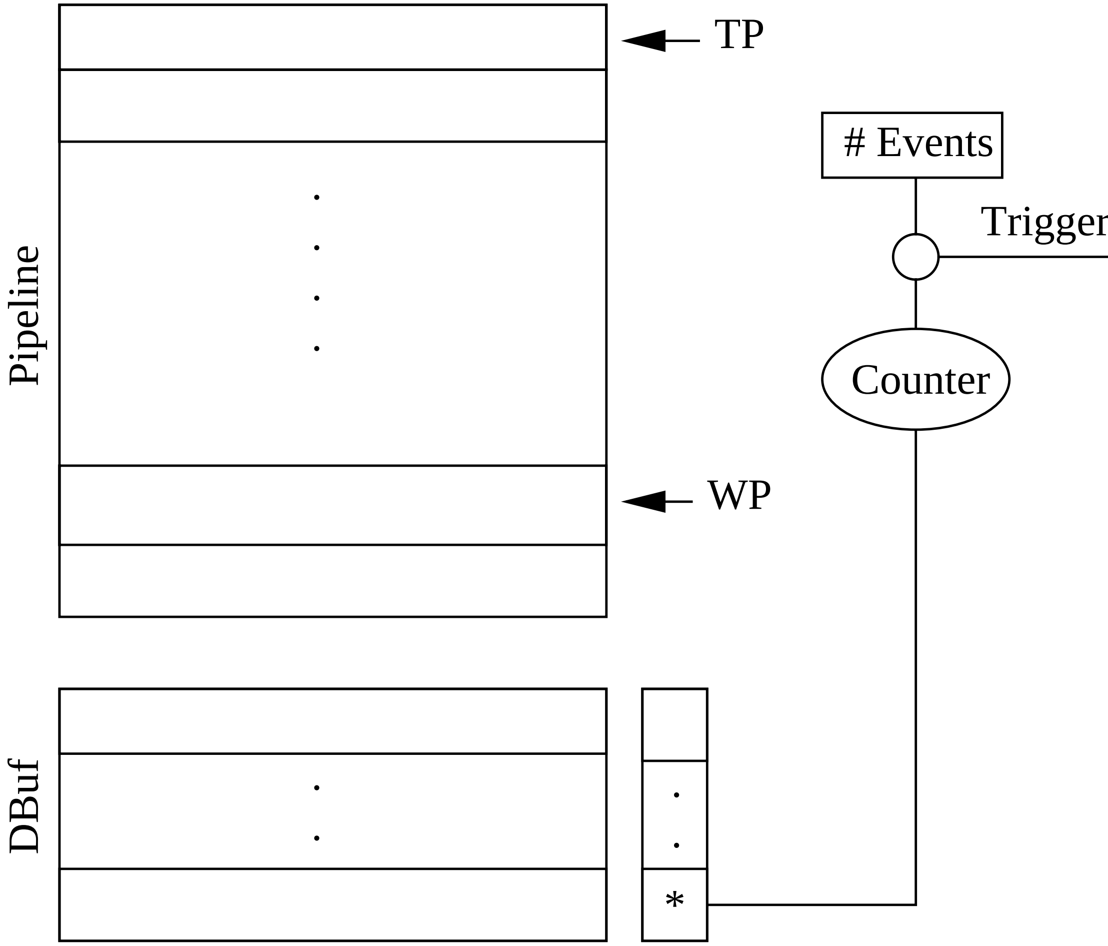
The OTIS Pipeline Control Logic

The control logic behaves equal for each read mode. It has to copy up to three data words per trigger from the pipeline to the derandomizing buffer.

Therefore a preloaded (with the number of events to read out) counter starts to count down to one if a trigger occurs.

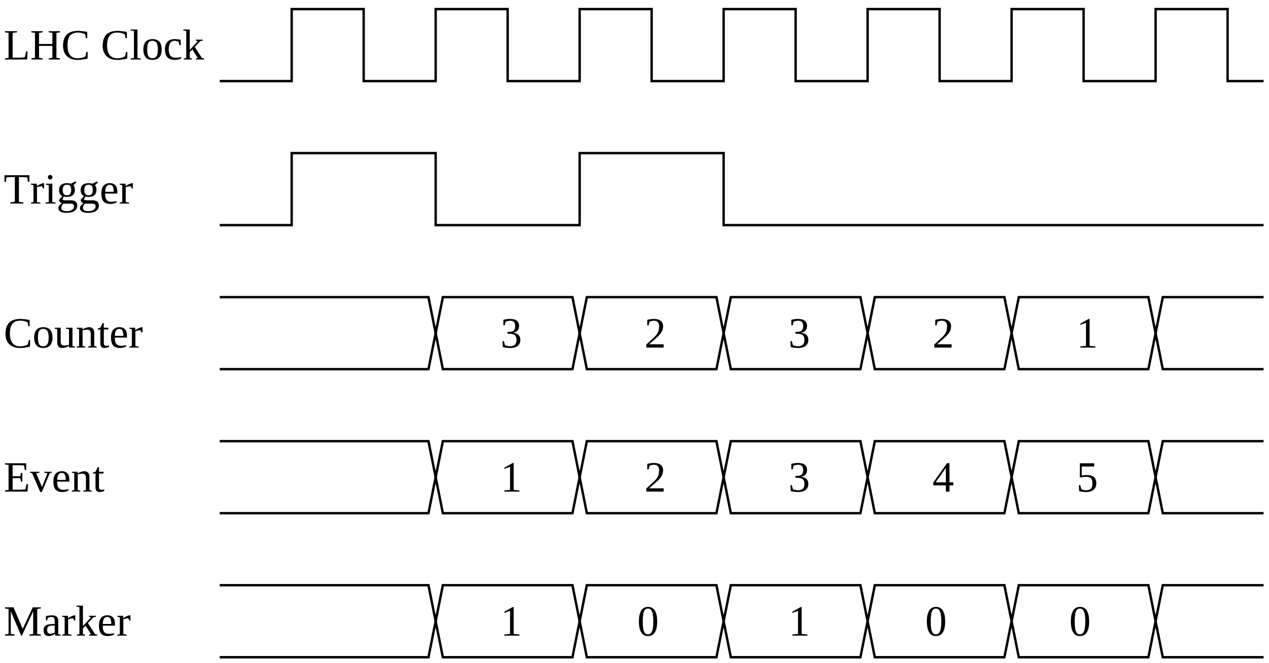
A one−bit register assigned to each data word marks the beginning of a read out sequence.

The following example explains the use of counter and marker bit:



The OTIS Pipeline Control Logic

Timing diagram:



Joint Beetle/OTIS Review
The OTIS Pipeline Control Logic

Testenvironment:

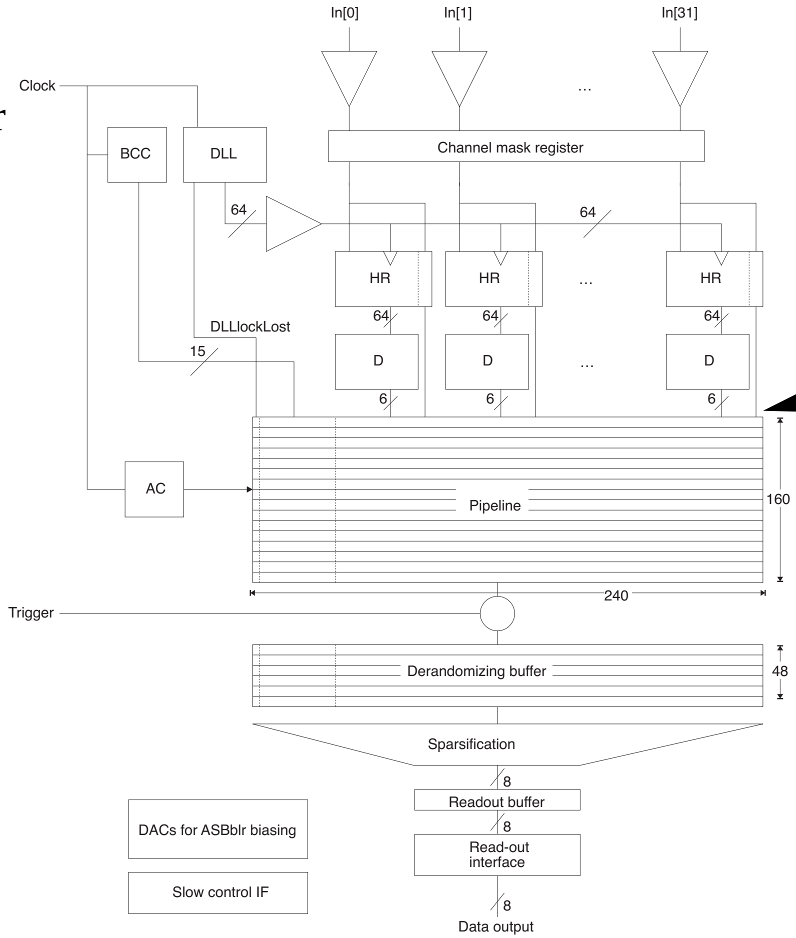
Verilog model of SRAM to take count for setup and hold times.

"Off line" calculation of random hits and drifttimes with parameterised

- ASDblr dead time (20ns),
- number of channels (32),
- maximum drift time (50ns),
- occupancy (10%).

Hits of neighbored channels are not yet correlated.

Simulation data gets inserted direct to the pipeline (clock synchronous).



insertion of simulation data

Joint Beetle/OTIS Review

The OTIS Pipeline Control Logic

Next Steps:

- Finish programming and simulation of the pipeline control logic.
- Synthesis, implementation and test of control logic on an FPGA.
- Start programming of read out logic.