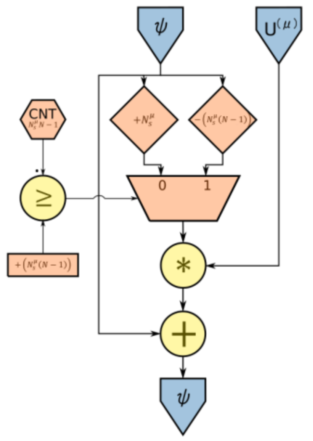
Data Pre-Processing with High-Level-Synthesis and Dataflow Programming using HLS C++ Dataflow Template Library

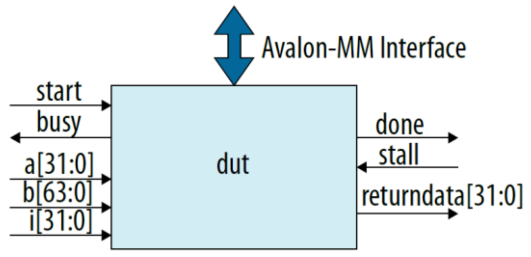
Thomas Janson and Udo Kebschull

IRI GOETHE UNIVERSITÄT FRANKFURT AM MAIN

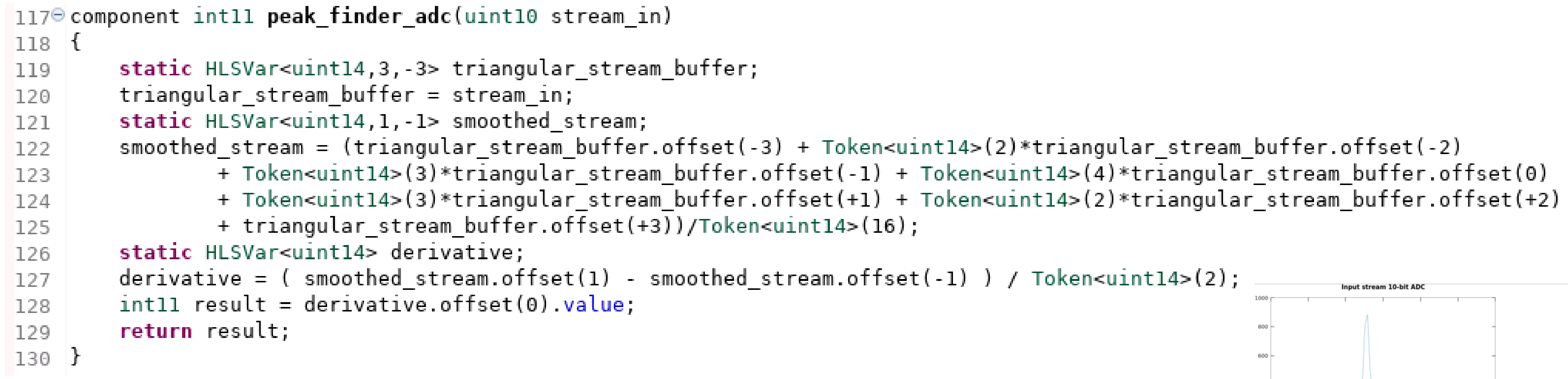
- Variables are static stream buffers (FIFOs) – HLSVar
  - Arcs of the graph
- Offset Operator picks up data items from the stream buffer at indexed position
- Each function call moves a data one step through the pipeline (static variables)
- Component is pipelined by default and you get always II=1, so you can invoke the function for each clock cycle before the previous call returns.
- The algorithm is described as a deep pipelined dataflow graph.



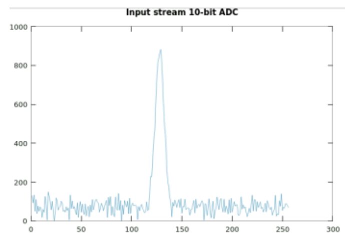
Component Function becomes an IP which can be instantiated into an VHDL design:

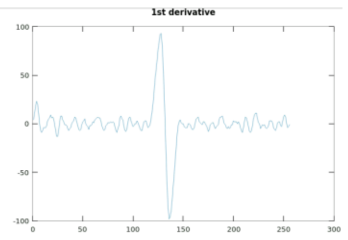


Taken from Intel HLS Reference Manual



```
component int11 peak_finder_adc(uint10 stream_in)
{
    static HLSVar<uint14,3,-3> triangular_stream_buffer;
    triangular_stream_buffer = stream_in;
    static HLSVar<uint14,1,-1> smoothed_stream;
    smoothed_stream = (triangular_stream_buffer.offset(-3) + Token<uint14>(2)*triangular_stream_buffer.offset(-2)
            + Token<uint14>(3)*triangular_stream_buffer.offset(-1) + Token<uint14>(4)*triangular_stream_buffer.offset(0)
            + Token<uint14>(3)*triangular_stream_buffer.offset(+1) + Token<uint14>(2)*triangular_stream_buffer.offset(+2)
            + triangular_stream_buffer.offset(+3))/Token<uint14>(16);
    static HLSVar<uint14> derivative;
    derivative = ( smoothed_stream.offset(1) - smoothed_stream.offset(-1) ) / Token<uint14>(2);
    int11 result = derivative.offset(0).value;
    return result;
}
```





15th Pisa Meeting on Advanced Detectors
La Biodola, Isola d'Elba, May 22-28, 2022