ABCN’: modifying the pipeline

- Pipeline entry modes have been ripped out (DATATAKINGMODE, PULSETESTMODE, etc)
- Chip input ports are also different from ABC130* (14 parallel ports @ 320 MHz, vs 256 parallel ports @ 40 MHz)
- Deserializer in the pipeline entry block parallelizes the 320 MHz inputs into a 14*8 = 112 wide port at 40 MHz for ease of processing later
- The deserializer looks for rising edges on chessData[0] to synchronize with the frame so that L0IDs can be assigned

```
pipeLineEntry RegIn_dut(
      .chessData(CHESS_DATA),
      .BCID(BCID),
      .chess_clk(BC),
      .hrdrstb(syncRstb & EvtBufRstb),
      .mode(TMmode),
      .pulse(digitalTestPulse),
      .TestPattern1(TestPattern1),
      .TestPattern2(TestPattern2),
      .ThreeBCSel_mode(ThreeBCSel_mode),
      .lock(1'b1), // useless
      .switch(testPatternEnable), // useless
      .pipeLine(masked_dIn) //
    );
```

- What is the purpose of the “even” and “odd” memory banks in the L0L1 buffer?

```
mem256x64  SRAM_256x64_P1_G1(
    .CLOCK_in(CLK),
    .writeB(PipeW1),
    .readB(~PipeR1E),
    .addr_in(PipeADR1[`PIPE_ADDR_WIDTH-2:1]),
    .Page_sel_B(PipeADR1[`PIPE_ADDR_WIDTH-1]),
    .data_in(DataPIn[111:64]),
    .data_out(PipeData1[111:64]),
    .data_m(PipeData1_m1)
);
```

```
always @(posedge CLOCK_in)
    begin
    //***WriteReg    <=  #0.1 (~writeB);
    ReadReg <= #0.1 (~readB);
    AddrReg[6:0] <= #0.1 addr_in[6:0];
    AddrReg[7] <= #0.1 Page_sel_B;
    DataInReg    <= #0.1 data_in;
    end
// WRITE and READ operations
always @(negedge CLOCK_in)
    begin
    // WRITE operation
    //***if (WriteReg)
    if (~writeB)
      begin
         mem[AddrReg] <= #0.1 DataInReg;
    //***WriteReg <= 1'b0;
      end
end
```

Next steps

- Need to understand ClusterFinder so that we can connect the new pipeline to the outputs
- ABC130 spec says that the FastClusterFinder finds hits of 2 or more consecutive strips
- Is this relevant to us?
- How can we transform the CHESS II data into a form that ITSDAQ will understand?
- Is it necessary to write a new DIO block for the ITSDAQ?

Next steps

- Also need to understand the ITSDAQ firmware
- eg. what is the difference between dio_f2v_std, dio_f2v_drv?
- Which ports on the Nexys Video FMC connector go where on the ABC130/ABC130*/ABC130N'?
- Is a driver chip between the ITSDAQ and the ABCN' necessary? What is the role of the ABC130 driver chip that connects to the Atlys ITSDAQ?