PPrASIC Design review

1. 14/6/00 Official ATLAS Level 1 Calorimeter trigger review - PPrASIC
   1. **People**
      1. Paul Hanke
      2. Tony Gillman
      3. Paul Bright-Thomas
      4. Viraj Perera
   2. **Documentation reviewed**
      1. *Specification:* Specification of the PreProcessor ASIC (PPrASIC) for the ATLAS Level-1 Calorimeter Trigger - July 1999
      2. *Design Guide:* Pre-Processor Asic Design Guide - version 1.0 April 14, 2000
      3. *User manual:* Pre-Processor Asic User and Reference Manual - version 1.0, April 14, 2000
      4. http://wwwasic.ihep.uni-heidelberg.de/atlas/projects/pprasic.html
   3. **Overall comments**
      1. The Specification (which should be the basis for the design as it has got an official ATLAS Preliminary Design Review) is not followed to the letter (features added, others not correctly implemented, features dropped?). A careful check of the correspondance between the design and the Specification should be made, including verification with simulations.
      2. No information on the environment of the chip. Therefore it is impossible to check if the chip will interface correctly (signal levels, speeds). Things normally go wrong on the interface between two chips, so this is a serious omission.
      3. No specifications on I/O timing parameters are given
      4. No system level (or MCM board level) simulations have been made
      5. The design is stuck to one specific type A/D converter (as it inverts the MSB, and has resistors on input)
      6. Resistor on A/D inputs probably not correct (should be external and higher value)
      7. The Verilog code has no headers with a history of modifications and authors.
      8. The Verilog code has hardly any comments on the functioning of the blocks and procedures and is therefore very hard to read and maintain
   4. **General**
      1. It looks like that not all features from the "Specification" (an official document that has had it's official ATLAS Preliminary Design Review) have been implemented (e.g. default values at power-up, CellSumIn) or that other features have slipped in (e.g. general purpose output). Make a table that checks that all the requirements from the "Specification" have been implemented. And also check that no features that were not required have been added.
      2. Make a table that shows for each requirement of the Specification which simulation run has verified the correct functioning of it.
      3. Make clear which options of the Specification are kept. 2.4.1 Leading edge BCID? 4 or 2 channels on one chip (p14 spec).?
      4. What is different in this chip from the FeASIC. Why? What problems have you seen with the FeASIC? Are those corrected? Have all differences extensively been simulated? Errors are likely in those areas.
      5. Enter the design data into the EDMS system

5. **Design Guide**
    1. General
        1. Nice piece of documentation
        2. Figure 3.1 Coding example a bad one with a signal called 'Output'.
    2. Simulation
        1. Does not seem that there is a full list of timing constraints and checks for setup and hold times, both internally and externally
        2. Do simulations check setup and hold time violations?
        3. Have simulations been made with LHC clock and SerialCLK asynchronous to each other?
        4. It would have been much better if the top level simulation model was a board-level simulation including the MCM and other environments. That way system-level errors could be found.
        5. Are all two or four channels simulated? How is it verified that you really test the independent channels and that they don't get all the same data and settings? Especially simulations have to be done with the Daisychaining active.
        6. Has the chip been simulated against software? I.e. are there routines/programs written in C that can be used to drive the simulated chip (and later to test the real chip)?
        7. Are any coverage statistics for the simulations available?
        8. Have all border cases been tested (highest address, lowest address, highest bit, lowest bit etc)?
        9. How is checked that the JTAG or scanpath insertion is done correctly? This is not simulated
        10. Note: the TESTDONE and CHECKLIST files in the CVS repository are all from last year, so it doesn't seem that they have been used.
6. **Specification of the PPrASIC**
    1. Registers
        1. Spec p16. InputControl.2 5 bit only is range 0x00-0x1F to me. In spec is written 0x3F. Where is each field placed?
        2. From section 4 of the specification, it is not clear how many bits the different points of readout are. I don't see the correspondance with the main block diagram, figure 2.
    2. Serial interface
        1. Spec 5.1.1: Daisy chaining not clear (FRAME has to be divided by number of chips). How does the second chip know where his data starts? FRAME is not shifted, so I believe you need seperate FRAME signals for each chip and even then it is not clear what happens with the data. Have several chips in series been simulated?
        2. Spec 5.1.1: I think any engineer connect the following signals for a daisy chain: DataOut -> DataIn and DaisyOut -> DaisyIn. This is wrong, so this is a serious risk when the MCM or other boards are designed. Make sure this will be checked in the reviews of the MCM and other boards.
        3. I don't understand how is synchronised between the async domains of the LHC clock and the SerialClk. Have simulations been made of those two clocks in different domains? May the SerialClk be faster and slower?

      4. Spec, figure 5: only 10 databits shown, not 11 or 13
      5. Spec 5.2: no "hard" reset signal needed?
   3. FIFO
      1. FIFO depth for 11th bit is programmable seperately. Is the total maximum length still OK for all possible variations in cable length, analog electronics and the source for the 11th bit?
   4. FIR filter
      1. Truncates 17-bit result into 10-bit for LUT (2.3 of spec). Is this still workable if low-values are fed in, i.e. doesn't the 10-bit output stay 0?
7. **User and Reference Manual**
   1. General
      1. I think that the User and Reference Manual is a mixture of User documentation and design documentation. And for neither of them it is complete. Appendix A (Internal Protocols) as a description of the internals is very lean... Appendix B (Verilog include file) like appendix A don't belong to a user manual.
      2. Chapters should start at a right (odd numbered) page
      3. "Inputs that can be 'determined' from outside". Preferably write "Inputs that can be 'set'". Are those registers outside the modules or inside?
      4. Why does the Output Summing and BC Mux not have their own chapters describing in detail the functionality? Likely there will be errors in those blocks as they are not described...
      5. No chapter about the environment of the chip. Signal levels, I/O, use as bare chip or packaged, power supplies
      6. Will the chip be usable in case anything goes wrong with the serial loading (i.e. are the default values usable)? Has this been tested?
   2. Introduction
      1. The User and Reference Manual should be the only documentation needed to design a board and to program this chip. Currently it does not contain all the required information.
      2. Really too short. This chapter should describe the whole environment, the functioning (what is output and when etc), the amount of channels etc. It should be about four pages long to be more complete.
      3. What chips are connected to it, required signal levels and speeds.
      4. Describe A/D converter connected to it. Phos4 chip connected to it. What chip is controlling the serial channel, etc.
      5. From Fig 0.1 it is not clear that there are four channels. Unreadable lines to the Cell Summing and BC Mux.Not clear which datawidth is used everywhere. Redo picture and make it larger.
      6. Not clear to where the Serin1 and 2 are connected (does Serin1 only work on channel 1 and 2?, what happens if daisy chain is used?). Expand this.
      7. 9.3.5 ReadOutSource: in figure 0.1 it is not shown that the data can be taken directly from the input and not via the synchronizer.
      8. There is no mentioning of the the GenOut pins (see 9.2.3) in the figure nor in the text
   3. Chapter 1: BC identification

   1. Not clear what really should be the output of the BCID: is it a bunch crossing number (that's what an ID is), or is it an algorithm put on the input data. If the latter, when is the 'value' output and what happens in the other timeslots?
   2. Not clear which filter coefficent is the first one to be used
   3. 1.2.1 Not clear from description in 1.2.1 if FIR result for -150 ns etc are needed and where inputs and outputs on left and right (of the adder part) are going to. It looks like half of the adders and multipliers can be removed. Is figure 1.2 a schematic or is it some mixture of timing information and schematic?
   4. 1.2.1 adders are signed adders?
   5. 1.3.1 Table 1.2 not clear. What does the e.g. mean etc? I think the example part should be taken out and should be a seperate table or example of the register values.
   6. 1.3.1 RegDelayExtBcid is not described. Add description
4. Chapter 2: Readout
   1. 2.2 Header word: isn't it 'dangerous' for fault tolerance that after a ResetBar signal is given, the counters will not start without an TTC synchronous reset?
5. Chapter 3: Histogramming and Rate-Metering
   1. 3.1 "Upon receiving of the 'memory cleared' signal .. the cycle starts from the beginning. How is the 'memory cleared' signal given? Is this a command?
6. Chapter 4: Control logic
   1. Table 4.2 Bit 6 probably means Raw data, channel 2 (and not 1)
   2. 4.1.2 LoadLut needs some example to see how ramp really works (e.g. can it increment also only every third address)
   3. 4.1.2 can the LUT also be written address by address? I think this is a requirement: (spec, section 2.6)
   4. 4.1.2 does the LUT power-up with a normal slope (spec, section 2.6)?
   5. 4.1.2 StopPipeline: second serial interface does not process this command. Why is this so? Makes the design less redundant in case serial channel 1 would fail.
   6. Is there a command to clear the histogramming memory?
   7. Figure 4.4: what are the levels of bits 12-10 of a status word?
   8. From the last paragraph it is not clear if the data comes back as one block or that is it interleaved with the BC data.
7. Chapter 5: Input and Output
   1. 5.1 optmised for AD9042. Have tests been done on this device (I saw a report where two other chips were used, but not this one)? Is a 12-bit A/D, while only 10 bits are used. Which bits are used?
   2. 5.1 The inversion on the MSB in the PPrAsic makes that you can only use this A/D that comes with the MSB inverted. Should there not be a setup register that selects if an inversion is needed, so you can change to another type of A/D easily?

   3. 5.1 The resistor should in fact be at the output of the A/D and not at the input of the receiver. Now it will still drive with it's full dV/dt the capacitance of the track and the input pins/pads capacitance, which may be as high as 10 mA/bit (120 mA total). See section "output loading" (page 13) of the AD9042 datasheet.
   4. 5.1 how is the value of the 330 Ohm series resistor chosen? Have simulations been made? Analog devices used 499 Ohm. "The resistor value was chosen to represent a time constant of ~25% of the data rate at 40 MHz. This reduces slew rate while not appreciably distorting the data waveform". Also makes it that you can't change to another A/D if needed.
   5. 5.1 you know the price of the AD9042 is $27.45 (bare die or 44-pin package only or $150 28-pin ceramic)?
   6. AD9042 is an AC-coupled device. Is that enough, or should it also work on DC signals?
   7. 5.4 only when I read up to here it became clear to me that there are four channels in one PPrAsic (see comments in Introduction).
   8. Edge selection only can work if Tsu+Th < 12.5 nsec. Plus need margins for variations in temperature and jitter. So for example Tsu+Th must be smaller than 9 nsec.
   9. No specification at all on Setup time, Hold time, Clock to output times! How can be verified the chip will interface correctly?

8. Chapter 6: Serial interface
   1. 6.1 "the serial interface uses four datalines". It is not clear to me which are those lines. I see only one DataIn.
   2. 6.1 system of synchronizing so that the serial channel can run at speeds lower than 40 MHz unnecessarily complicates the logic. Probably it will not even be used. May the speed only be lower, or may it also be faster than 40 MHz? Has it been tested thouroughly with different speeds? I suggest to remove the possibility of having a different clock speed.
   3. 6.1 has the daisy chaining been tested? Expand the exact working of it: "the frequency of the Frame signal then has to be divided..." is too short and probably not even true (the second chip should have it's own Frame signal, isn't it?).
   4. 6.1 "processing has to be fast enough to provide a stable output...". This is not precise, how fast should it be?
   5. Figure 6.3 12 bits shown. Should be 13?
   6. 6.3 calculation of readback channel bandwidth is not worst case: does not use daisy chaining. So reading back the histogramming memory may require 2.5 ms.
   7. Not clear how the four channels share the two serial inputs. What is redundant, how can channels be reached?
9. Chapter 7: Frequency detection
   1. What's this Phos4 chip? Should be described in a chapter describing the environment of the chip.
   2. Doesn't seem that the signal is working reliably:
      1. < 0.3 MHz signal is reliable

        2. 0.3 MHz to 20 MHz is reliable
        3. > 20 MHz not reliable. So may show the result the clock is not available! So what is the use?
    3. Is the signal really needed?
    4. How does this frequency detection work (I know it is not easy to implement this)? Has it been simulated?
10. Chapter 8: JTAG interface
    1. This is really not enough information for a Reference Manual. Define the JTAG registers. The length of the registers is not defined, nor the value of the Identification register. Which instruction numbers? What is the length of the chain? What are the "user defined Data Registers"?
    2. How is the JTAG port going to be used? Only for chip testing or also on the MCM?
11. Chapter 9: Registers
    1. 9.2.1 ReadMainEnable: description states that the default is enabled (i.e. 1). However, table 9.1 shows the default value 0.
    2. Are all default values from the table checked with simulations?
    3. Bits are refered as 2-3, 0-6 etc. This may lead to confusion for values (bit 6 is msb isn't it?). Better write 3-2, 6-0 etc.
    4. ReadoutSyncRaw "starts readout of raw data". Does it only start once, or is it more like an enable signal. Same for ReadoutSyncBcid.
    5. Table 9.2 register 17. Both HisEnable and HisSource are on bit 2!. HisOpMode is shown as bits 3-4. I believe this should be 4-5 (cf PPrAsicChannelReg.conf)
    6. 9.2.2 ChannelSelect. References to Channel 0 and Channel 1. In other places of the document always about Channel 1 and Channel 2 is spoken.
    7. 9.2.3 GenOut. Move section one up, so it keeps the same order as the bits in the table.
    8. Should there have been some General input pins to make the chip more flexible? If not, are the GenOut pins really needed then?
    9. p.28, Num=0, Bit 1. Description is wrong. Should probably be DATA bit (and not BCID)
    10. 9.3.2 FIRCoeff1..5 have all defaults of 0. Section 2.6 of the specification requires a simple linear conversion function after power-on. Should follow the specification, is also more robust in case something would be wrong with the chip or environment.
    11. 9.3.2 Sathigh, low. Not really clear that 500 and 50 are decimal values. Maybe write them as hex. Also for other bits. Funny that in 9.3.3 another way is presented: 4 ~ 00 0000 0100.
    12. There doesn't seem to be a Reset bit or reset command that resets the chip and puts all registers in their power-on state. Is really necessary.
    13. 9.3.5 PipeDelayBcid, PipeDelayRaw. Default 10. Spec 2.2 prescribes a default of 16.
    14. 9.3.5 ReadOutSource: in figure 0.1 it is not shown that the data can be taken directly from the input and not via the synchronizer.
12. Chapter 10: Pin Description

   1. Fig.10.1 shows pins that are not described, nor that were in the Specification: TempMeasure.
   2. Fig.10.1 does not have SerDaisyIn2. It is needed to connect to SerOut2, isn't it?
   3. Fig.10.1 the Genout pins are not shown in the layout.
   4. Fig.10.1 what's the "Pulldown" near ExtBcid1..4? Value of it? Should be described in Table 10.4
   5. Fig.10.1 has 110 pins, Table 10.3 110 and Table 10.4 118! The difference seems to come from the Pwr and Gnd pins. Anyway, they all should have their own entry in the table and figure.
   6. Table 10.2 what's the ResetBar signal? Is not described. On which blocks does it act (memories, LUT, registers?). What is the timing of it?
   7. Table 10.2 describe different pad types (IB35P, OB33P, IOA5P, OB93P (?), IB35P pull up. Where is correctly described that the FADC inputs have a series resistor and what is the value? Give schematics of I/O cells.
   8. Table 10.4 why is the power here called Vdd? In other tables it is called Pwr or even 3.3V
   9. Table 10.4 specify which pins use serial resistors on input, or pull-up resistors
   10. What are the signal levels on all pins? Are the inputs 5 Volt tolerant?

13. Chapter 11: Technical Data
    1. Signal toggle rate of 40%. Seems a high number and is likely to be less. Have you simulated this to get a realistic number?
    2. What packaging will be used
    3. Expand this chapter with I/O levels, input capacitance (important for A/D inputs' calculation of resistor value)
14. Appendix A: Internal Protocols
    1. doesn't belong in a Users manual. Is design information
15. Appendix B: Global Verilog Include File
    1. should have a full header describing the filename, description, authors, notes and list of modifications and modification dates. Probably the same is true for all Verilog files.
    2. doesn't belong in a Users manual. Is design information
    3. 'define RAW_WIDTH 'ADC_WIDTH+1 is I believe not correct. Put brackets around it to prevent errors and confusion: e.g. 2*RAW_WIDTH expands to 2* 'ADC_WIDTH+1 = 2*10 +1 is 21, while it should be 2*('ADC_WIDTH+1)=22. Same for others where an operation is performed.
    4. TTC numbers: 12=4: both are "Bunch crossing number"
    5. REG_ALMOST_FULL_WIDTH 7 is described that it has to have the same size as the "width of the derandomizer". DERADATA_WIDTH 11 (common width of derandomizers) is 11<>7? If it has to be the same as another value, make the 'define point at the other value
    6. Definitions of SE_CMD_L etc (page 44, Ctrl word) are not described. Probably brackets missing here as well.
    7. It looks to me that more values should have been calculated out of the others, i.e. probably when you have to change one value, you will have to change others as well for it to work.

    16. Bibliography
        1. Much too short. There are many other notes that are relevant to this document: Specification, bcid note, etc.

8. **Verilog code**
    1. Very nice CVS interface to retrieve files. Also the web coloured Verilog files is a nice system.
    2. Verilog code has no headers with a history of modifications and authors. I know it is in CVS, but that's a bit hidden.
    3. Verilog code has hardly any comments and is therefore very hard to read and maintain
    4. Some files correctly documented: PABcMux,
    5. Most files not at all: PPrAsicGlobalReg.conf doesn't even have the filename in it's file. PaInput, PaSerIntf.v (a really important file, no comments at all).
    6. PABcMux contains certain requirements: (A minimum of one empty bunch crossing after a signal in a given channel.). Is this requirement also documented in the Reference guide and does it still fulfill the Specification?
    7. PPrAsicGlobalReg.conf. Looks like that the ReadoutEnable is only one bit in the field, while in the reference manual (page 27) it is 2.
    8. Defaults in PPrAsicGlobalReg.conf are specified different than those mentioned in the manual (e.g. GenOut=3 vs 0, JetSumMode seems to be NoTrunc and not Trunc9 as default - reg defaults to 6)
    9. PPrAsicGlobalReg.conf: CVS "Adjusted default register values for testability". Is this just for running the Verilog tests or is it for manufacturability? Should it not be set for robustness, i.e. that the chip still gives some usable output even when the serial channel failed or other failures?
    10. PaReg.v. Last version has comments: "Minor corrections of synthesis problems. Small manual updates." But this is file is supposed to be generated automatically by 'makereg'. So next time those 'minor corrections of problems and manual updates" are lost, which will give strange surprises. Repair the generator command 'makereg' so no modifications are necesarry.
    11. PaReg.v (automatically generated) all registers are default 0. But in the configuration file for this register PPrAsicGlobalReg.conf PPrAsic.Global.Reg3 is set default to 6. (Is JesSumMode and Channelselect). How is this possible? Is there a bug in makereg?
    12. PaReg.v and PaChannelReg.v: reset is done via a "always @(posedge Clk or posedge Reset)". Is the reset really edge sensitive? Shouldn't it be a asynchronous reset? It looks to me like a very complicated register that will be used if they really reset on the edge of the signal.
    13. Should the resetbar signal not be synchronised? If the signal gets removed asynchronously, some bits of a state machine or counter may start running, while others are still in the reset state.
    14. PaSerIntfSync.v which "Synchronisation of serial clock and system clock" (but also generates other frame signals etc, does not describe all the requirements on the serial clock and the system clock (speeds, setup times, hold times E.g. how long should the FRAME signal be to let it work correctly? Has something to do with SI_LONGFRAME_WIDTH (in IncPaGlobals.v), but what?

    15. Is PaChannelReadMux.v is it not completed yet?: // READBACK OF REGISTERS HAS TO BE INCLUDED!
    16. Parity calculation for ToCP in PaBcMux.v seems very complicated (is not a simple parity, but function of other paramaters). Would have been better if parity calculation was put at the same place as the selection of the output data.
    17. PaBcMux.v comment: "ToCpParity: is set so as to have odd parity for the bit combination: {ToCpParity,BcMuxFlag,ToCp}; 10 bits.". But it is in fact much more complicated: if RegBypassBcMux=1, then it is not even a parity, but it is the lowest bit of the LUT. This really should be documented in the source file.
    18. Files still being modified on 31 May 2000: Added external BCID processign logic. Modified BC Mux bypass mode. Adjusted default register values for testability. Changes not simple ones, such as extra inputs to PaBcMux: http://wwwasic.kip.uni-heidelberg.de/cgi-bin/cvsweb.cgi/pprasic/verilog/PaBcMux.v.diff?r1=1.14&r2=1.15&hideattic=1
    19. I haven't gone through all Verilog files, but think that the code needs much more in-line documentation.

9. **Things still to Check in Verilog files**
    1. correspondence with URD (p4 spec)
    2. LUT powers up with a linear conversion function (2.6 spec)
    3. power on settings of BCID Decision logic (spec 2.6.1)
    4. Odd parity implemented (spec 2.9)
    5. Check correspondance between Reg bits spec and user manual
    6. Check if LUT RegByPassLut if 1 drops MSBs and not LSBs (1.3.3)
    7. Check how LoadLut works with small values of LutSlope