



# Facts (and Opinions) on Serial Links for ETD

A. Aloisio

University of Naples 'Federico II' and INFN

aloisio@na.infn.it

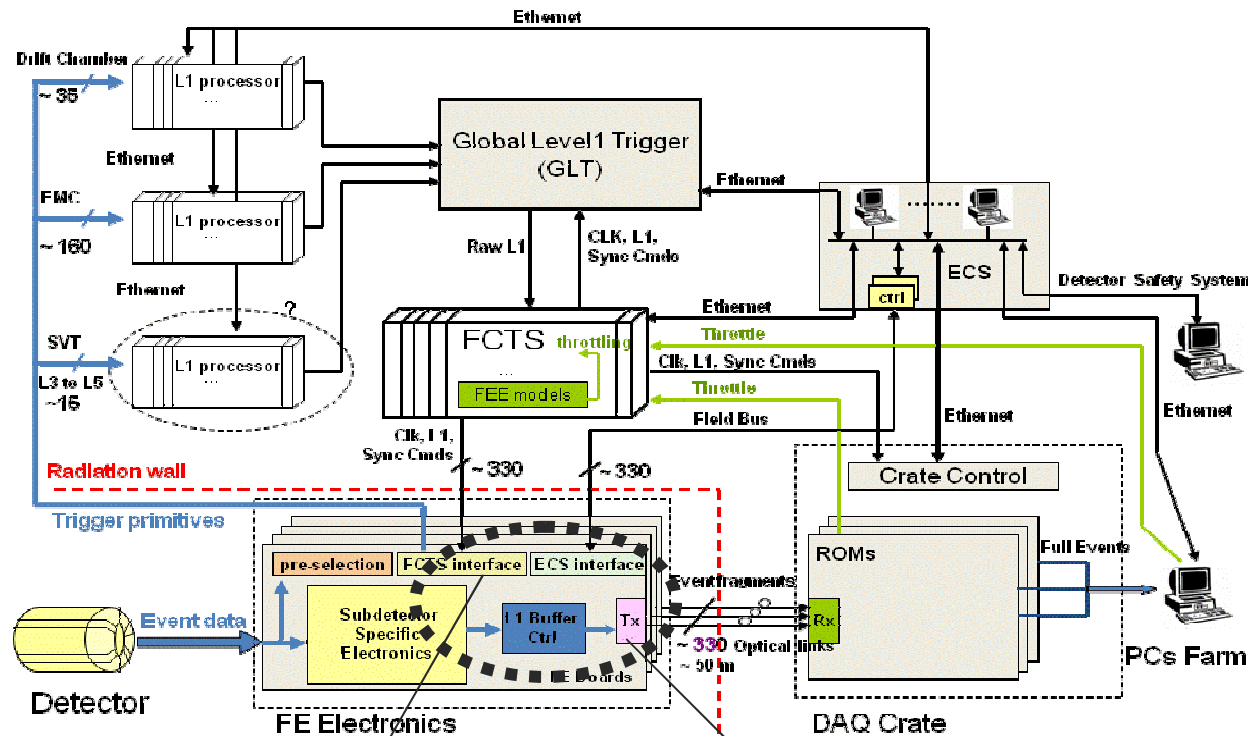


# [ Overview

---

- SerDes in the ETD framework
- Facts:
  - Clock issues
  - Recovered clock issues
  - Fixed latency
  - Rad-tolerance
- Opinions:
  - Specs, Design
  - Test, troubleshooting
  - Deployment, commissioning
- Conclusions

# SerDes in rad environments



FCTS link  
Tight latency requirements  
DS92lv18

Data link  
No tight latency requirements  
DS92lv18 or TLK2711A

- FCTS links:
  - Timing & Clock
  - Commands & Controls
  - config data
- DATA links:
  - Read-out payload



# [ Facts

- *Aime la vérité, mais pardonne à l'erreur*

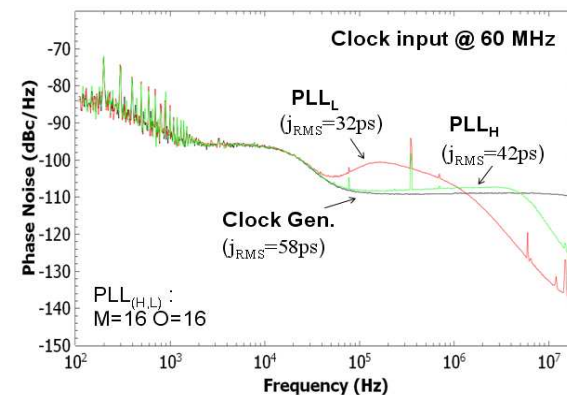
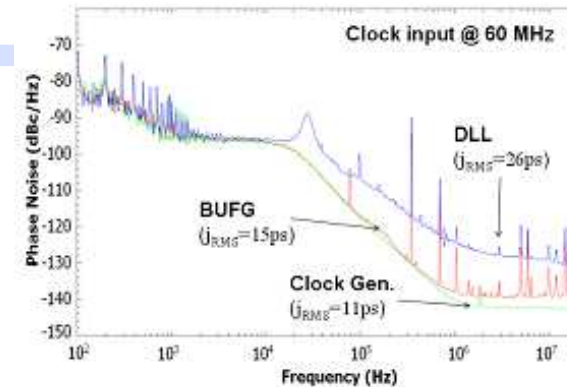
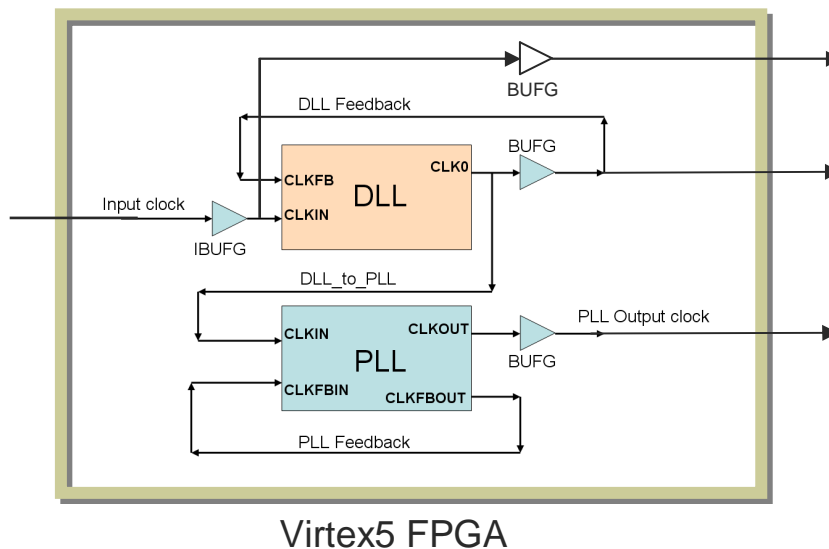
(Love truth, but forgive error)

Voltaire, "Deuxième discours: de la liberté"  
Sept Discours en Vers sur l'Homme (1738)



# Clock issues

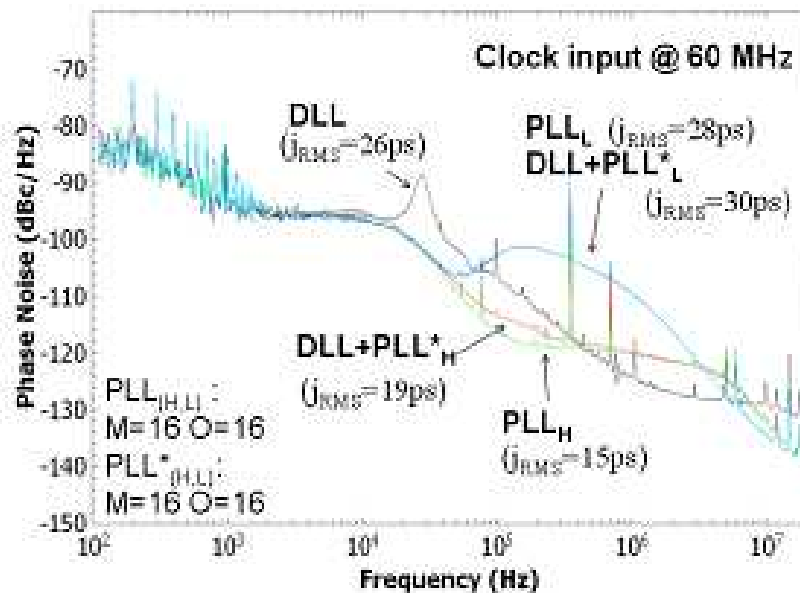
# [ Clock Conditioning



- (Embedded) SerDes requires extremely tight clock specs in terms of RMS jitter  $O(10\text{ps})$
- Signal integrity on clock lines should be carefully analyzed, even on chip
- With light loading and low I/O activity, clock tree can be probed externally
- DLL can nearly double the internal jitter
- PLL can be used as a low-pass jitter filter (1-5 MHz cut-off freq)

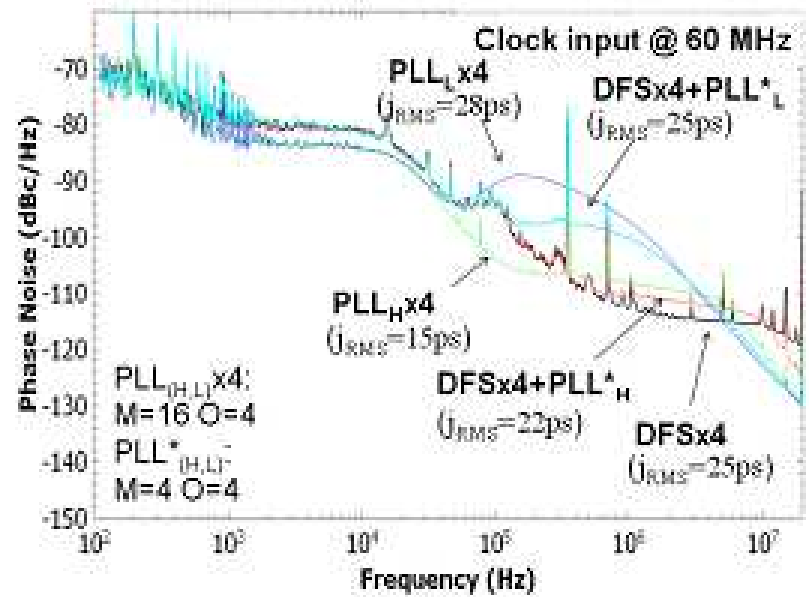


# [ DLL+PLL



- On Virtex5, PLLs can be used as jitter filter after DLLs, with programmable cut-off freq
- PLLs are way less noisier, but no fine clock deskew is allowed
- At high logic switch activity, PLL suffer from substrate noise
- Jitter performance depends upon overall design parameters (power, switching activity, complexity, ...)

# [ Clock multiplication



- Very likely, FPGAs will run at clock frequency multiple of the main clock
- Assuming  $F_{ck} \sim 60\text{MHz}$ , a x4 factor is a reasonable trade-off between power, performance and design margins
- Digital Freq Synth are available on V5, @60MHz performance similar to DLL
- DFS+PLL possible, yet less effective as jitter filter compared to DLL+PLL
- PLL still much quieter than DFS/DLL

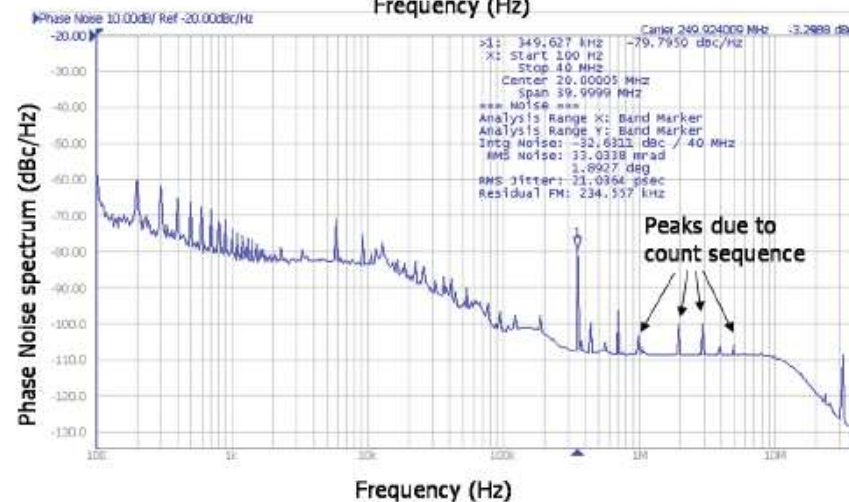
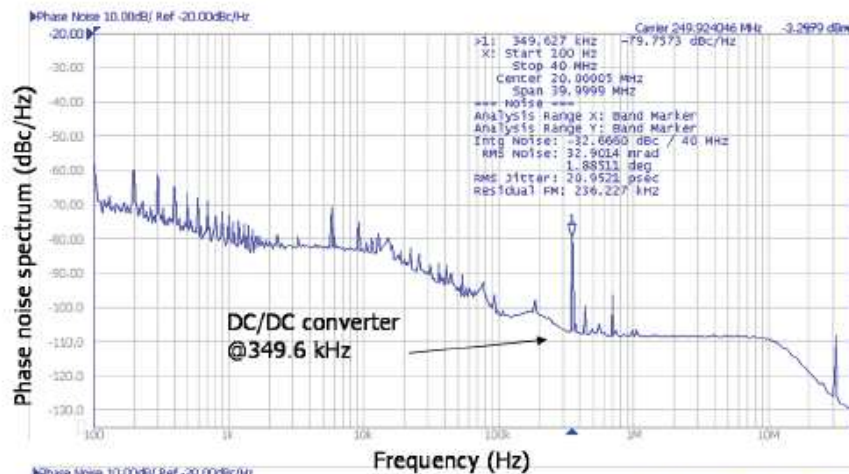




]

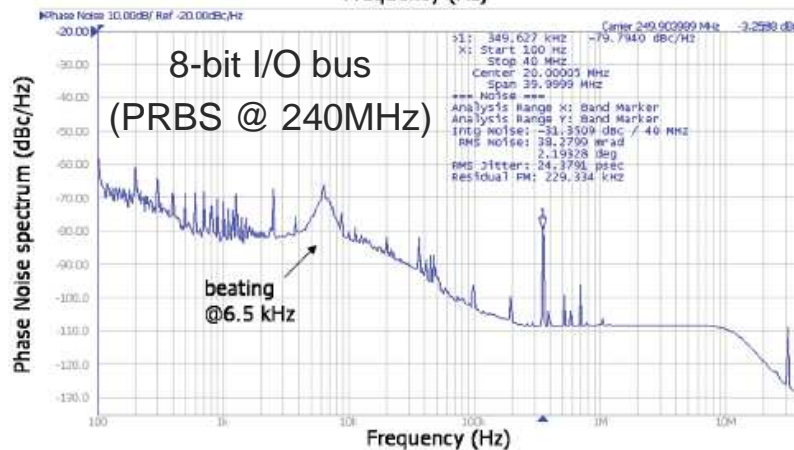
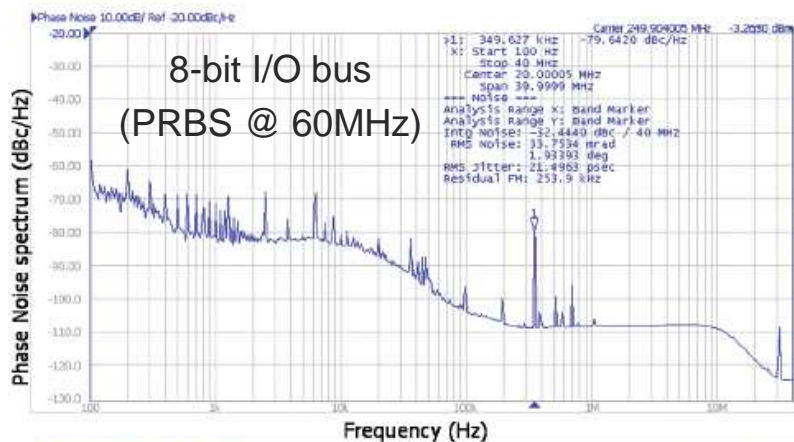
# Recovered Clock Issues

# Recovered Clock Jitter



- Clock recovered by GTP is victim of many aggressors:
  - Power noise
  - Data traffic
  - On-chip xtalk
  - Place-&-Route issues
- Noise comes from both TX and RX ends
- It could change after a new routing or a design revision

# I/O switching

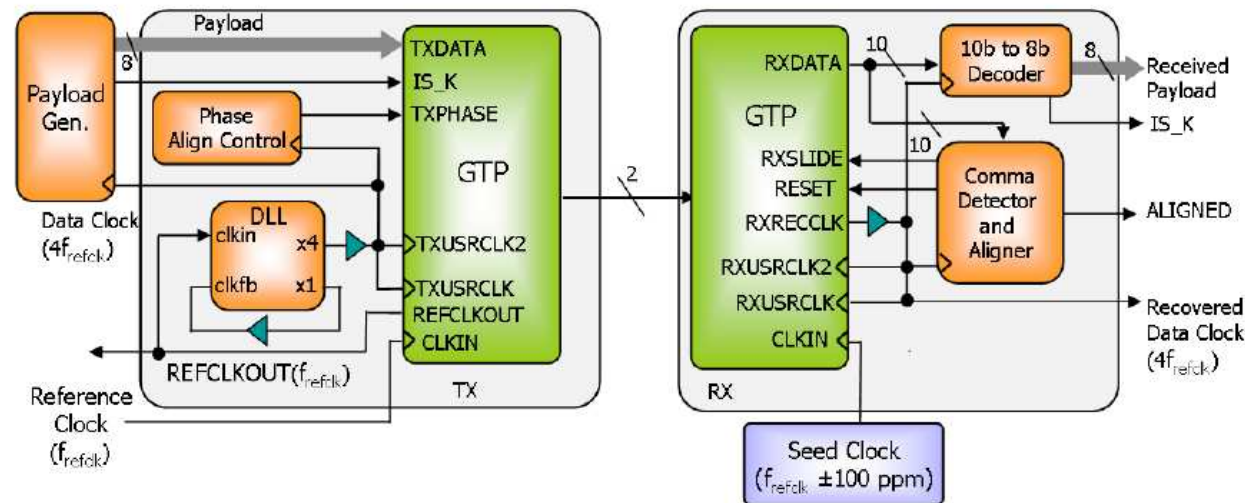


- I/O switching activity also has a large impact on the recovered clock timing jitter
- Huge contributions to jitter may appear at (relatively) low freq because of beating



# Fixed Latency

# [ Fixed Latency Architecture ]



- We have achieved fixed-latency data transfer on V5 and newer Xilinx families
- It is based on a careful design and peculiar hardware resources: VHDL code is not easily portable to other vendors
- Fixed latency and protocol emulation of off-the-shelf SerDes are (very) tightly coupled
- Embedded SerDes offer scalable performance, low power consumption, excellent integration, investment protection against obsolescence



# Rad Tolerance



# [ Radiation Tolerance

- At present, only the DS92LV18 passed successfully our tests (both SEU and TID, > 5kGy(Si) dose)
- TLK2711-A failed at 0.5 kGy(Si), equivalent to ~1 year of operations on the apparatus
- Xilinx FPGA test scheduled in 1 month from now. Encouraging results from SEU simulation in the lab. See Raffaele's talk



# [ Opinions ]

- *Le préjugé est une opinion sans jugement*

(Prejudice is an opinion without judgement)  
Voltaire, "Prejudices" (1764)





# [ Link Design ]

- FPGA and off-the-shelf side of the link need to be finely tuned in order to achieve the best results
- In my opinion, looks quite hard to disentangle the design of the two ends
- A variety of parameters can be defined only by considering the big picture:
  - Clock specs, jitter budget and filtering
  - FPGA power noise, internal switching activity, IO
  - Place-&-Route effects
  - Protocol Emulation, Fixed Latency
- The link design should cover all the aspects of the problem



# [ Is Job Sharing sustainable? ]

- What if teams  $\alpha$  and  $\alpha$  agree to share the design responsibility:
- Two case studies:
  - Split-then-design
  - Design-then-split



# [ Split-then-design

- Teams  $\alpha$  and  $\alpha$  agree on specs and start designing the two ends of the link
- They should wait each other until a link node prototype is available from the partner
- Quite a dead-lock conditions, isn't it? Very likely both teams will develop a test based on loopback
- Unfortunately, loopback is very far from real condition in the field: validation has to wait for the prototypes!
- Each team will have a favorite HW/SW test bench. How long do they take to make their own complete test bench?
- Any change in the FPGA should be validated at system level, including extensive test with different I/O and internal switching activities. Both teams involved ???
- Beware of Routing: in “the ps domain” it could change critical specs



# [ Design-then-split ]

- Teams a and  $\alpha$  agree on specs
- One team (a or  $\alpha$ ) designs the whole link as a FPGA IP core (off-detector) and a plug-in module (on-detector)
- Team a and  $\alpha$  should then use same technology/vendor (VHDL code is not easily portable!)
- IP core validated by team a is then moved to team  $\alpha$
- However, the IP core is embedded in a brand-new 'guest' environment: clock, jitter, IO activity need to be finely tuned
- Every time a major change in the 'guest' FPGA configuration is done, team a and  $\alpha$  should qualify again the design



# [Once In The Field

- Two working nodes not necessarily made a working link ...
- What if elusive errors start appearing at a rate in the  $10^{-5}$  to  $10^{-10}$  range ?
- What if jitter is good in the lab and poor on the apparatus ? (well... it happened with the TTC at CERN ...)
- What if a brand new routing brings brand new issues ?
- Who will be responsible for what ?  
Who will fix it ?  
How ?

# Again a problem of boundary conditions ...



- Frankly, I would not suggest to apply job sharing to high-speed, timing critical link designs
- One team ( $\alpha$ ) should do all the job, design it, deploy it in the field, test it and maintain it
- Keeping the design *monolithic* is the key point: its boundary should include the two nodes and of course the optical layer as a whole
- There are many different ways to do that



# [ What's next

- R&D on rad-tolerance and fixed latency is presently going on, in good shape
- See Raffaele's talk for investigations on SRAM-FPGA
- Next test beam scheduled July 10 (more DS tests, then FPGA) -> just in time to include in the TDR our experience with FPGAs
- New Research Program on FPGA, Optoelectronic, off-the-shelf SerDes already submitted to the Program Advisory Committee of LNS (Catania, Italy), to be discussed June 24
- Deadline for INFN funding requests rapidly approaching: Teams should agree asap how to proceed in 2012



# [Conclusions

- *Le doute n'est pas une condition agréable, mais la certitude est absurde*

(Doubt is not an agreeable condition, but certainty is an absurd one)

Voltaire, Letter to Frederick II of Prussia (1767)





# [ Conclusions

- Risk assessment of critical system components should also include setting the boundary conditions for job sharing
- High speed links fall in such a category: I presented my opinions on this matter based on technical facts and previous design experience
- In the view of TDR completion, ETD community should consider pros and cons of different approaches