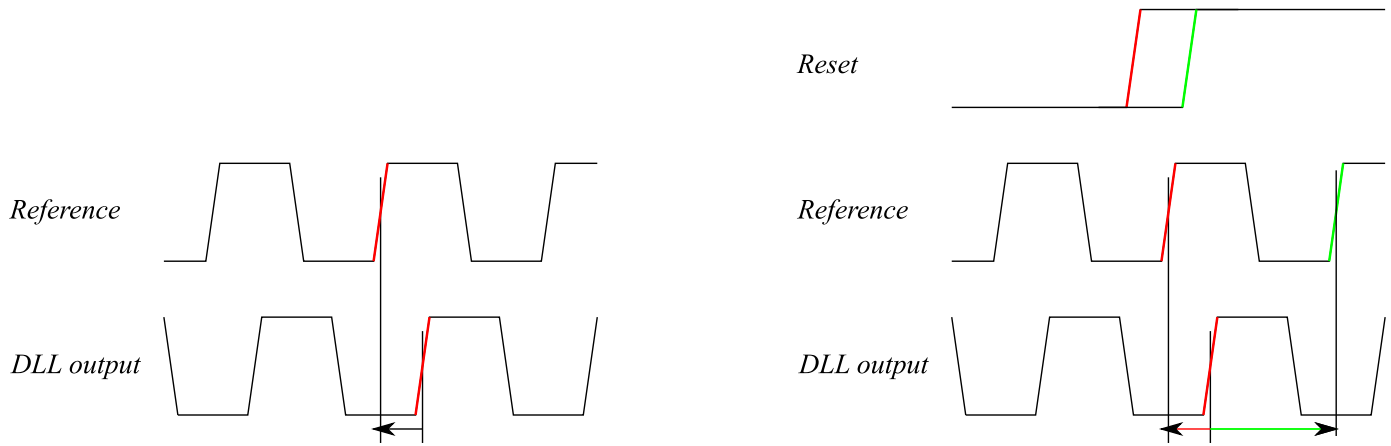
Results of GET4 Tests
Known Bugs and Performance

Holger Flemming

February 11, 2013

Outline

DLL Reset Phase (I)

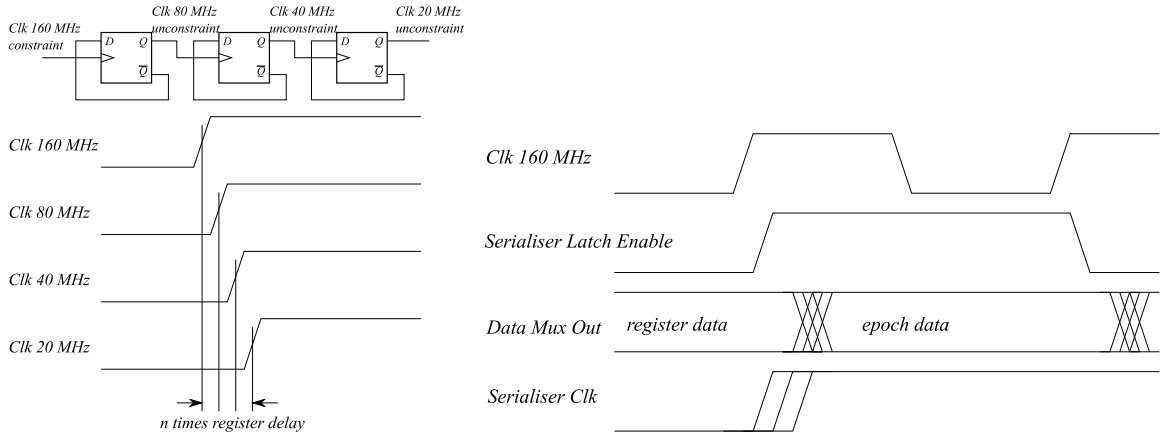


- Left side: Phase detector operation. DLL output comes later $\Rightarrow$ feedback control to faster DLL
- But phase detector has to be initialised to resolve phase ambiguity
  $\Rightarrow$ Reset input of phase detector (rigth side)
- Correct phase detector initialisation depends on phase of reset release
  - Red marked reset: Correct initialisation
  - Green marked reset: Phase detector sees DLL output first $\Rightarrow$ feedback control to slower DLL
    Wrong initialisation!

DLL Reset Phase (II)

- Internal generated DLL reset of GET4 V1.10 unfortunately has wrong release phase
- Solution for GET4 V1.20:
  - Configurable phase shifter for DLL reset release
  - Auto DLL reset (can be disabled!)
  - If DLL is out of lock state:
    1. Generate a 1 μs DLL reset
    2. Wait for 10 μs
    3. If DLL is still out of lock: increment phase shifter configuration and go to first step

24 bit Serialiser Timing



- Internal clock divider for reduced serialiser data rates $\Rightarrow$ inserting clock delays
  Unfortunately generated clocks are not constrained
- Data sources of serialiser are driven by 160 MHz main clock
  - Latch enable register
  - Data multiplexer switching from event data to epoch data
- Serialiser is driven with internally generated delayed clock
- $\Rightarrow$ Data corruption by timing violation of serialiser
- Will be corrected in Version 1.20
  Serialiser will work completely synchronously on 160 MHz clock

Further Modifications

DLL lock indication

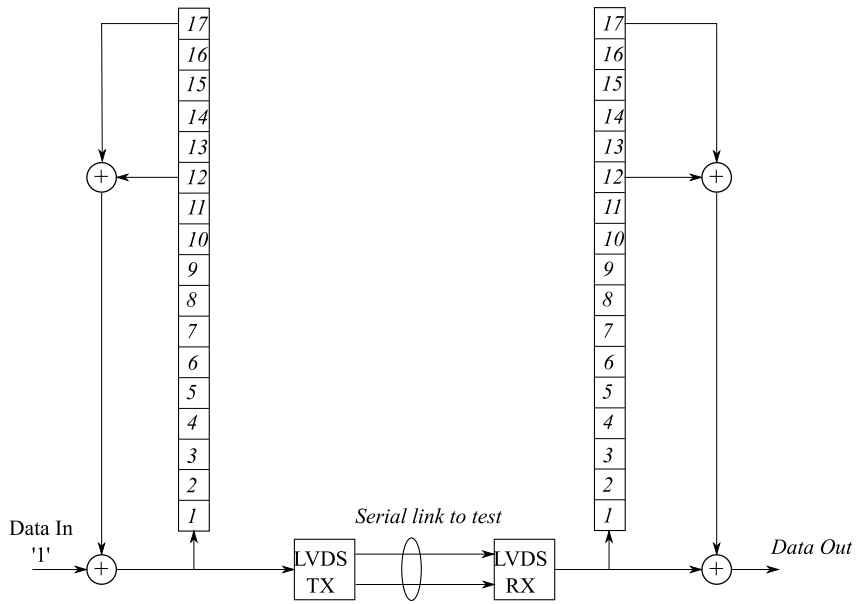
- Internal DLL lock monitor is only available in 32 bit readout mode
- In Version 1.20 an additional error event for DLL lock loss will be implemented in 24 and 32 bit readout mode

Sync flag

- Epoch events contain a sync flag to mark external synchronisation
- In GET4 V 1.10 sync flag is always zero
- Problem is fully understood and will be fixed in version 1.20

Bit Error Rate Test on Serial Downlink

- Till now no option to test bit errors on serial down link
- New in V 1.20: Bit error rate test
  - 17 bit shift register with xor feedback
  - operating as pseudo random code generator
  - At receiver stage another 17 bit shift register is needed
  - Receiver output: bit correct: 1, bit error: 0

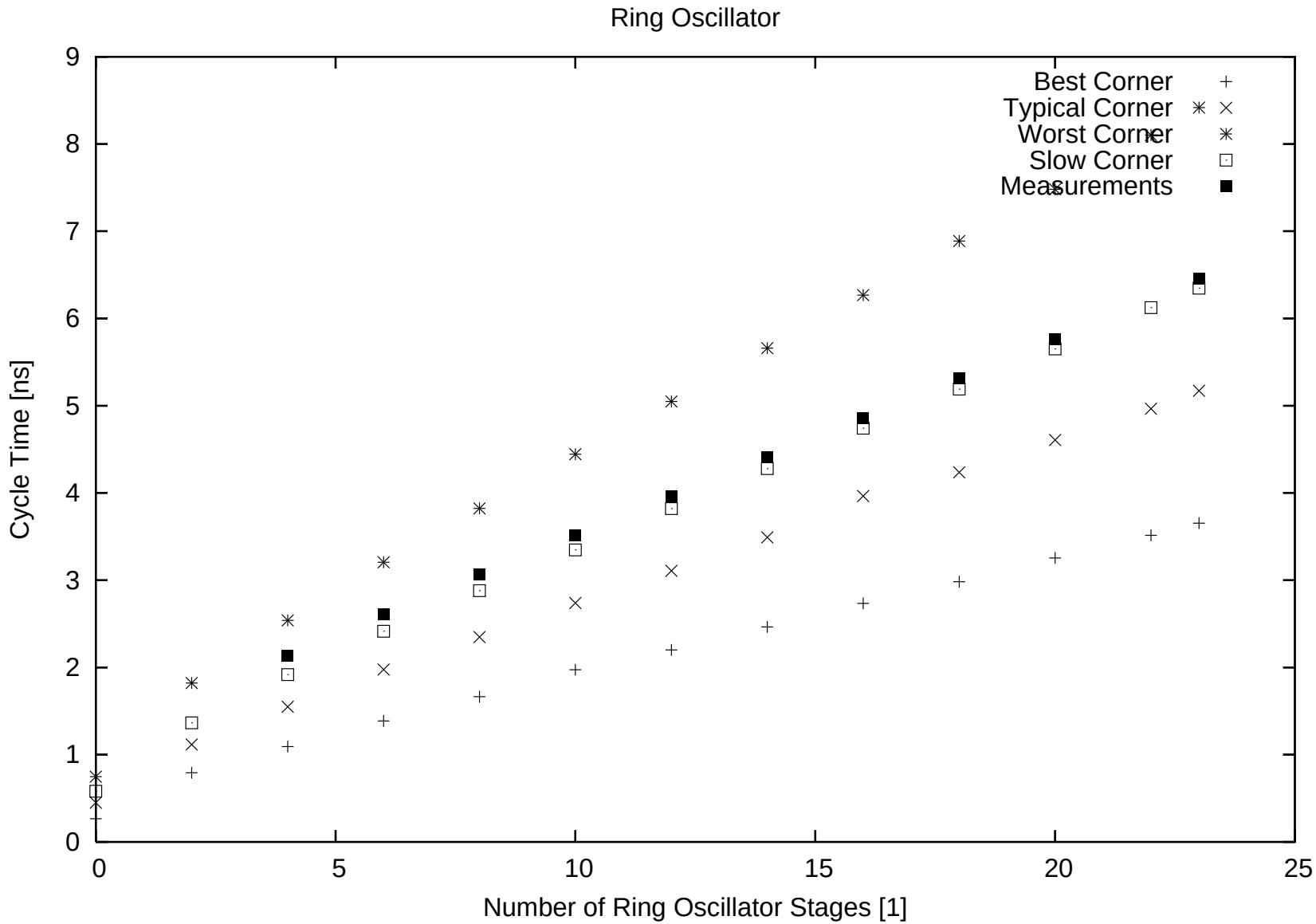


Process Speed Observations

- Observation: DLL of most GET4 V1.10 ASICs do not lock
- Reason: The latest UMC run seems to be slower than previous runs
- Run parameter provided by UMC are in specs
- Ring oscillator measurements are close to slow corner simulations
- Corner simulations:

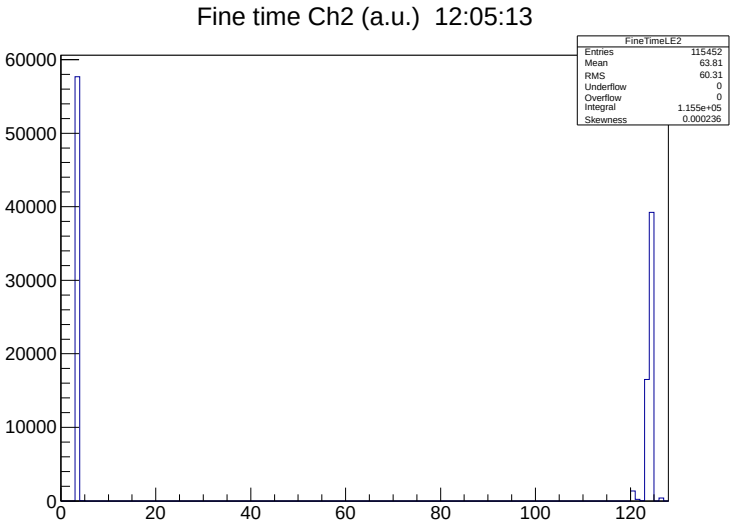
simulated corner	transistor models	temperature	voltage
worst	slow	125°C	1.62 V
slow	slow	30°C	1.80 V
typical	typical	25°C	1.80 V
best	fast	-40°C	1.98 V

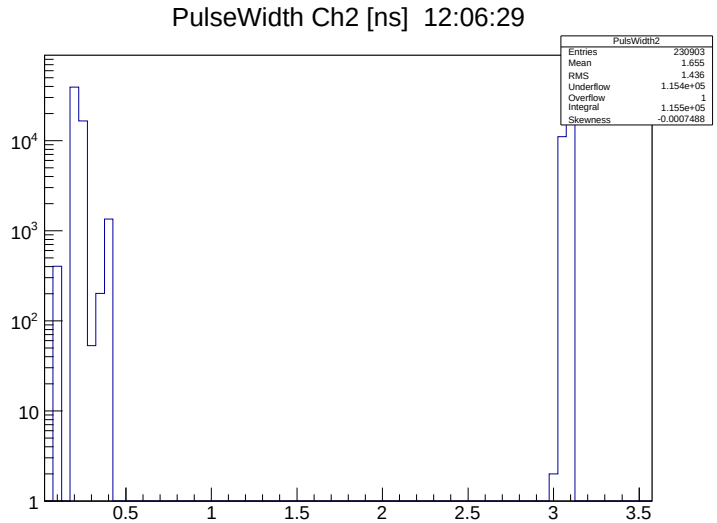
Process Speed – Simulation and Mesurements



Process Speed

Symptoms of unlocked slow DLL





- DLL is to slow
- $\Rightarrow$ New clock edge enters delay chain befor previous edge has passed the whole chain
- Hits close to clock edge are sampled by two clock edges
- $\Rightarrow$ Second pulse width peak with very short pulse width values

Process Speed

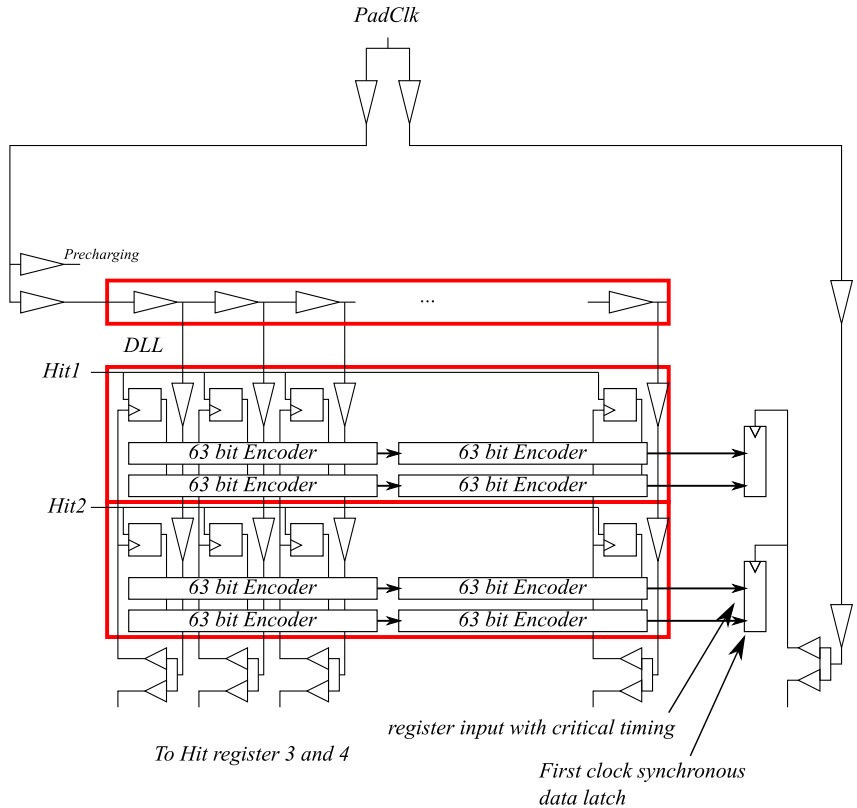
Reasons and work arounds

- Device speed depends on
  1. Process variations
  2. Temperature (cold environment $\Rightarrow$ faster device)
  3. Core voltage (higher voltage $\Rightarrow$ faster device)
- In MPW runs no influence on process parameters
- Maybe in production run we can ask for faster process parameters
- Current problem solved by increase of core voltage (from 1.8 V to 1.9 V)
- Reduced margin in case of warm environment in the experiment
- Influence of radiation damage?

Timing DLL Data Latching

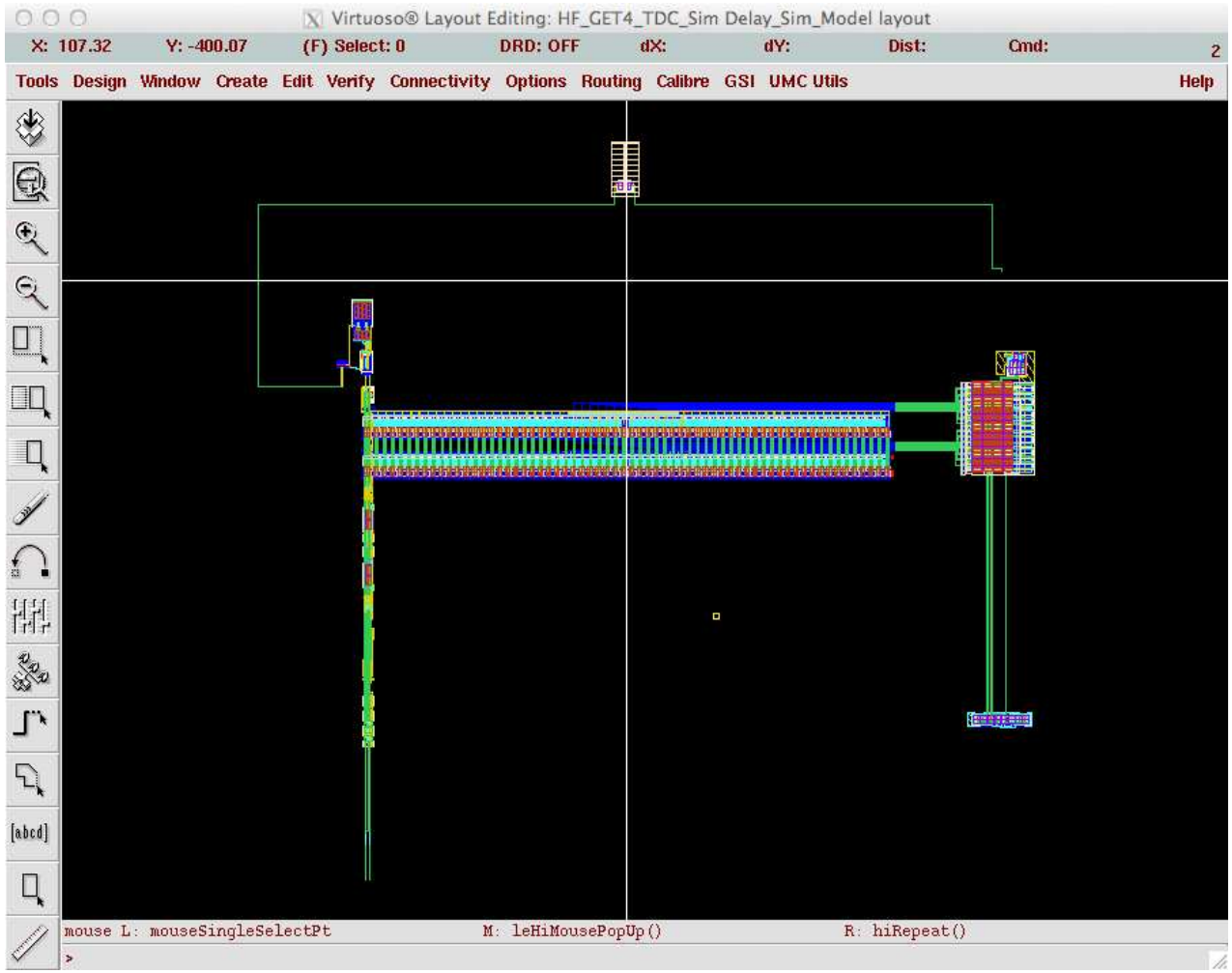
Clock and data path

- Common clock source *PadClk*
- Start of timing path: Hit register
- Clock for start register
  - Clock split and transfer line
  - DLL
  - Bin clock tree
- Combinatorics: Hit position encoder
- End of timing path: Data latch
- Clock for end register



Timing DLL Data Latching

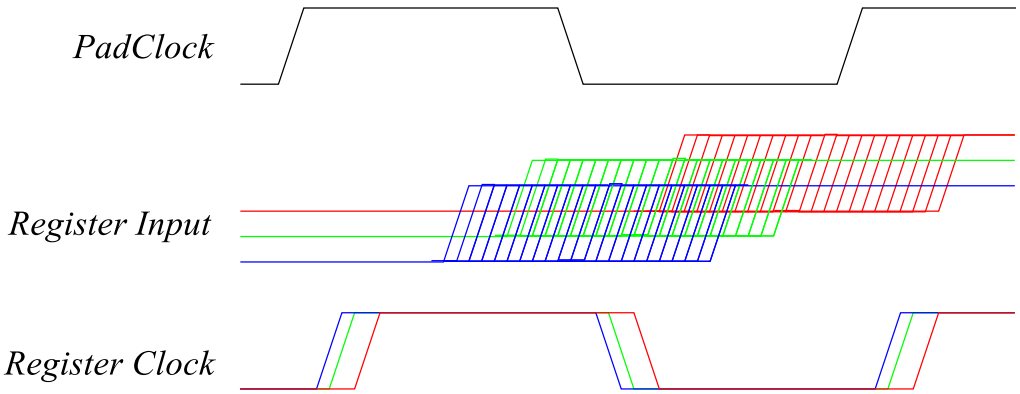
Layout exception for simulation



- Extraction of TDC core layout
- Simulation model from parasitic extraction

Timing DLL Data Latching

Simulation results



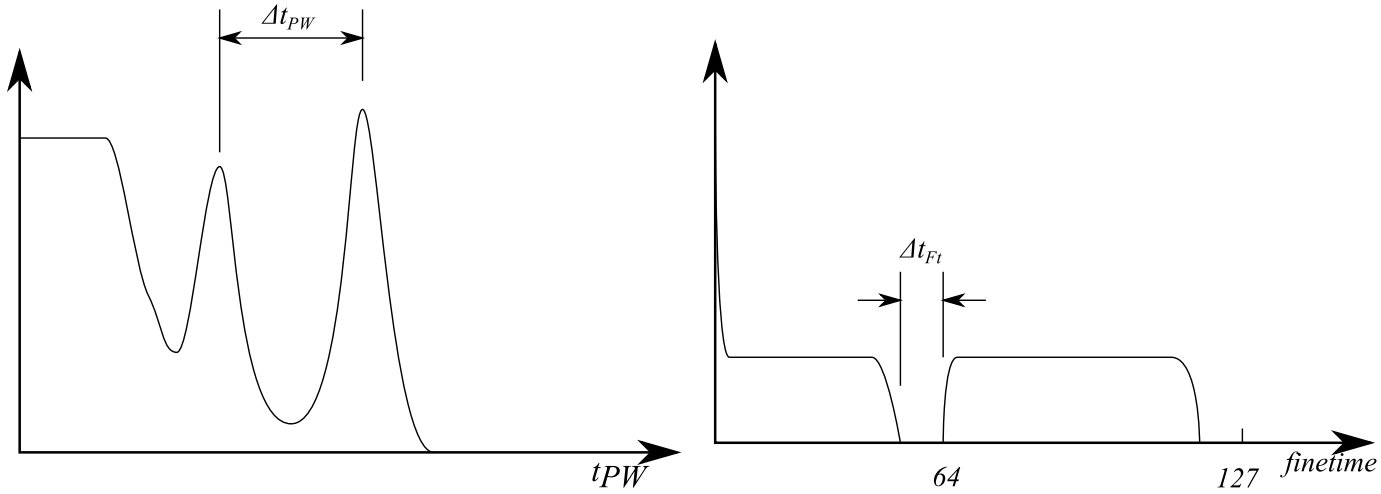
- Corner effect on register input timing stronger than on register clock
- Simulated slack:

Corner	Slack	Setup time t_{Setup}
best	$t_{Sl} = 1.78\ \mathrm{ns} - t_{Setup}$	$\approx 220\ \mathrm{ps}$
typical	$t_{Sl} = 1.088\ \mathrm{ns} - t_{Setup}$	$\approx 276\ \mathrm{ps}$
slow	$t_{Sl} = 500\ \mathrm{ps} - t_{Setup}$	$\approx 320\ \mathrm{ps}$
worst	$t_{Sl} = -356\ \mathrm{ps} - t_{Setup}$	$\approx 390\ \mathrm{ps}$

- setup violation in worst corner

Timing DLL Data Latching

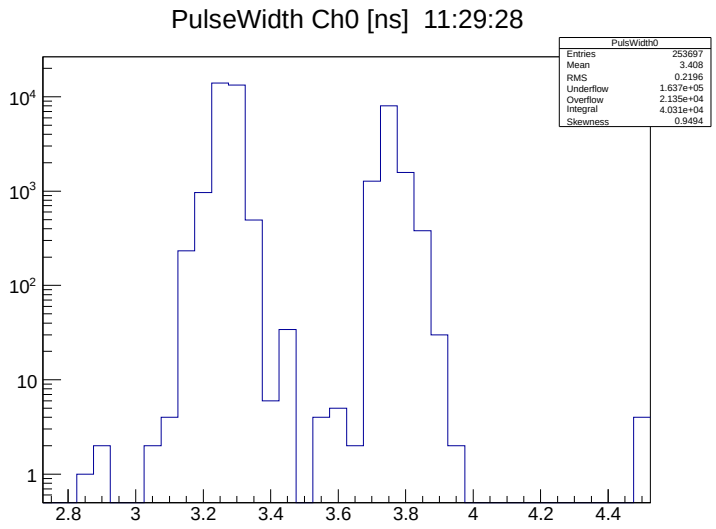
Measurement, method

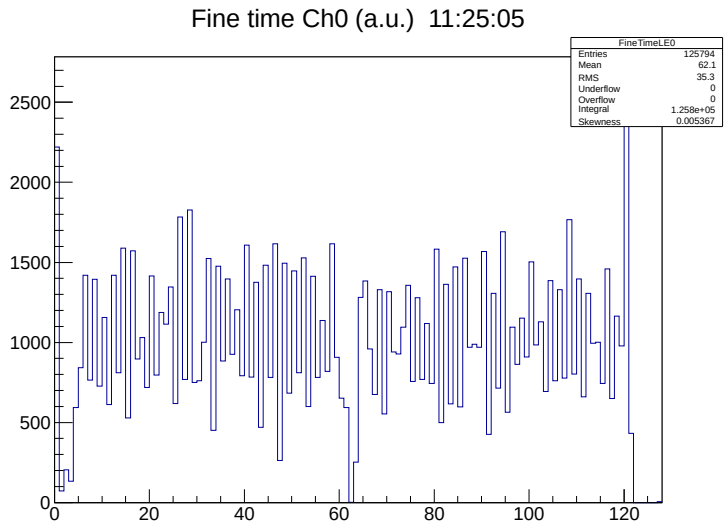


- Reducing DLL speed by reducing core voltage
  - DLL no longer locks
  - Two rising edges in delay chain
    $\Rightarrow$ double peak in pulse width spectrum
- Effects of slower DLL:
  - Increasing of Bin size: $\tau_{Bin} = \frac{6.4ns}{128 - \Delta t_{PW}}$
  - Hit data of Bin 63 arrive later: $\delta_{63} = 64 \cdot \tau_{Bin} - 3.2\text{ns}$
    $\Rightarrow$If slack gets negative a gap in finetime spectrum appears
- Slack at normal conditions: $t_{SL} = \delta_{63} - \Delta t_{FT} \cdot \tau_{Bin}$

Timing DLL Data Latching

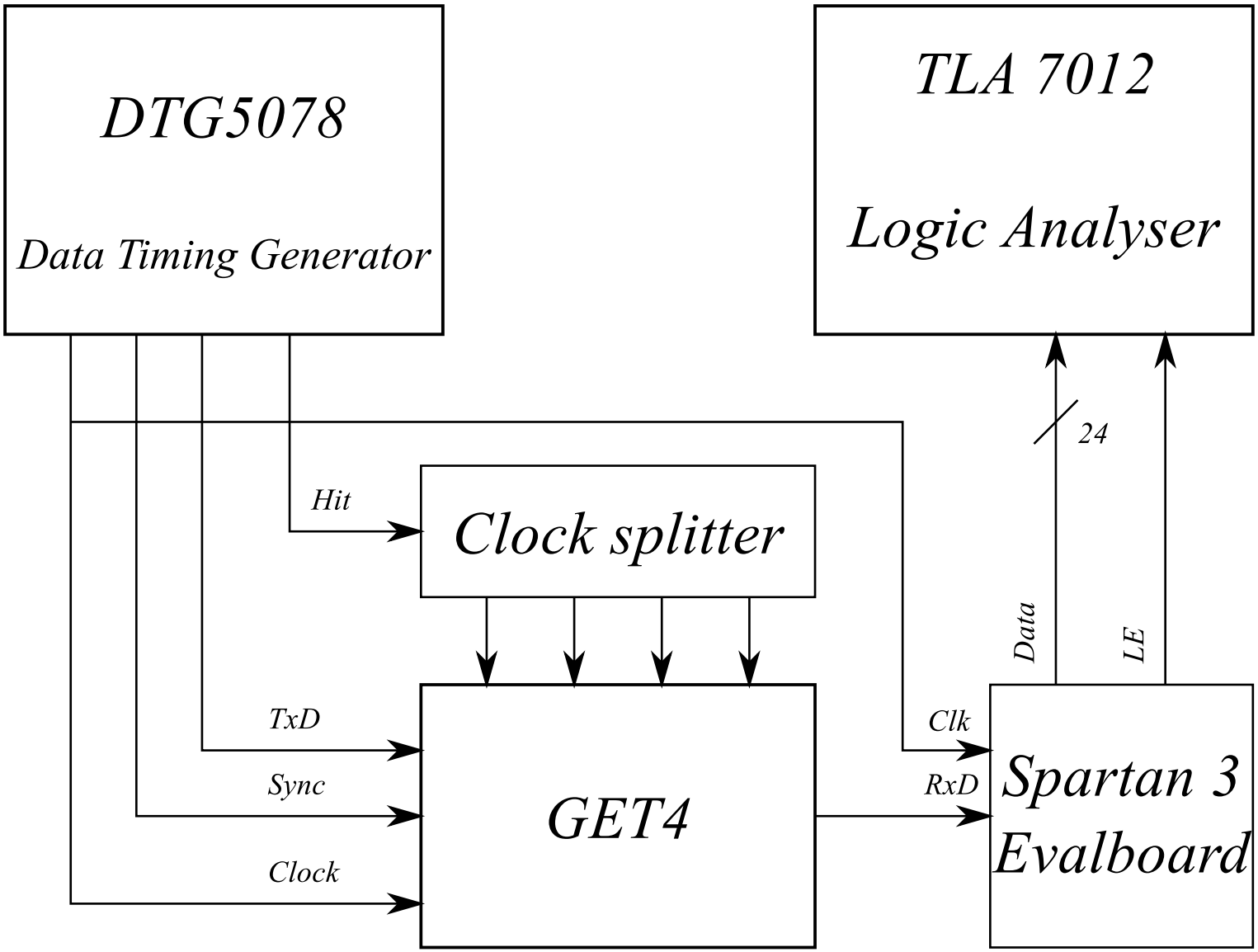
Measurement, results



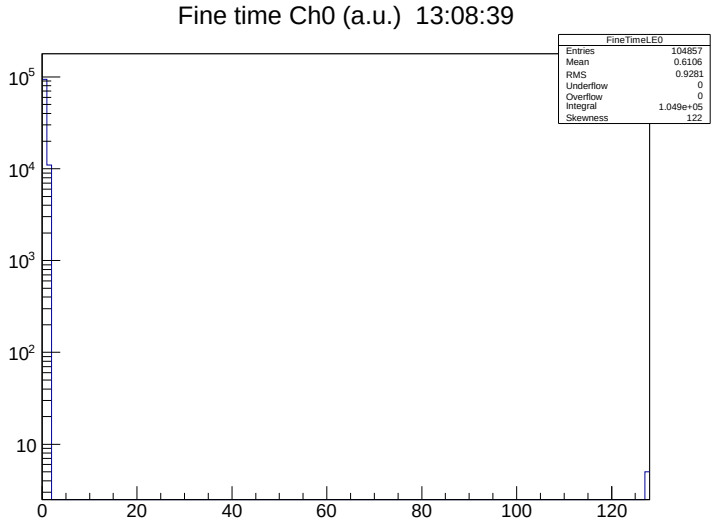


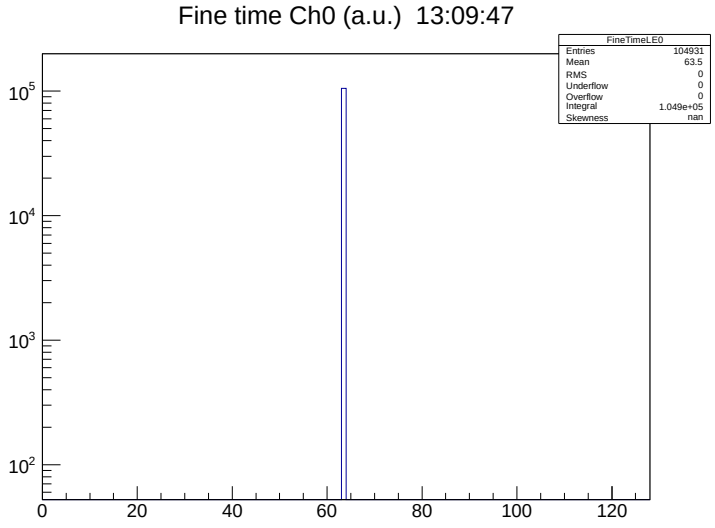
- $\Delta t_{PW} = 9.5 \text{ bins} \Rightarrow \tau_{Bin} = 54 \text{ ps}$
- $\delta_{63} = 257 \text{ ps}$
- $\Delta t_{FT} = 4 \text{ bins} \Rightarrow t_{SL} = 41 \text{ ps}$
- Timing is at the edge and will be configurable shifted in GET4 V1.20

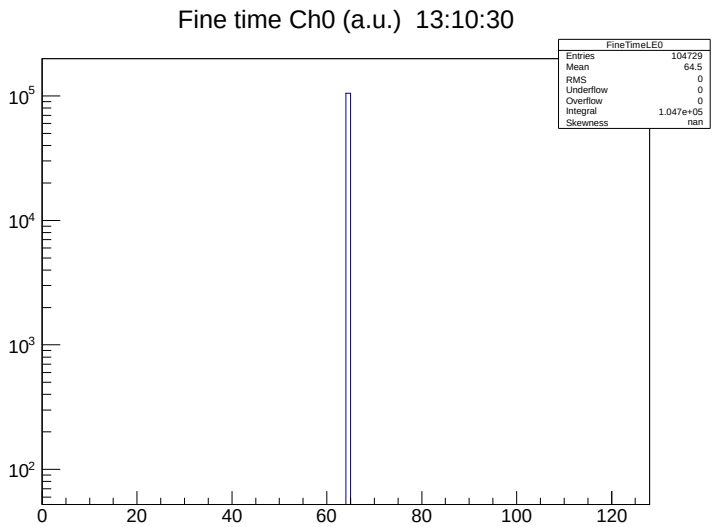
Test setup

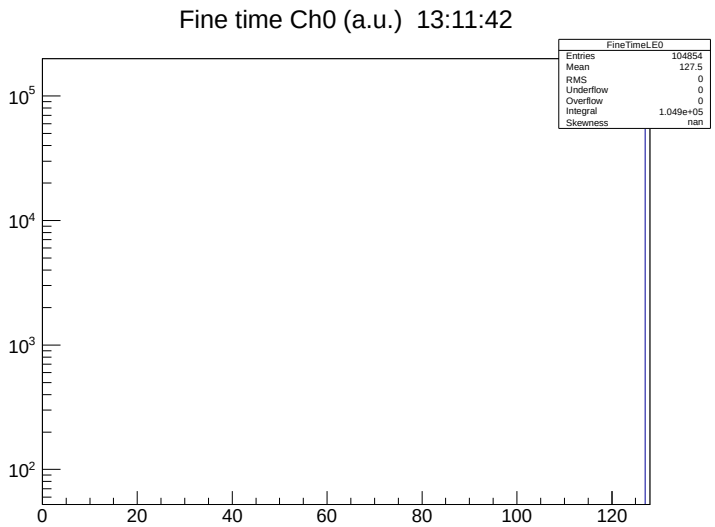

DTG5078
Data Timing Generator
TLA 7012
Logic Analyser
24
Hit
Clock splitter
Data
LE
TxD
Sync
Clock
GET4
Clk
RxD
Spartan 3
Evalboard

Finetime spectra with correlated Hits

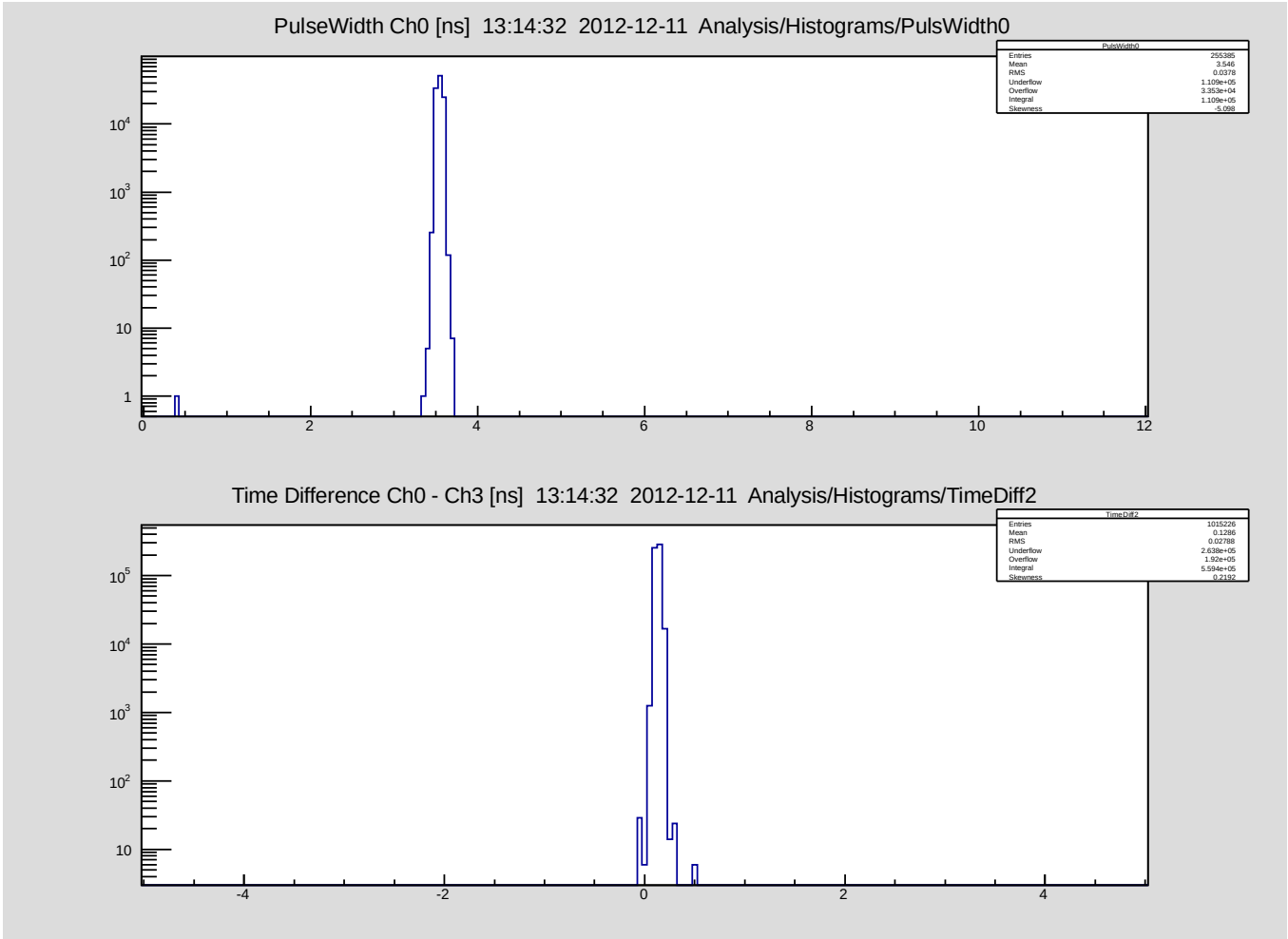






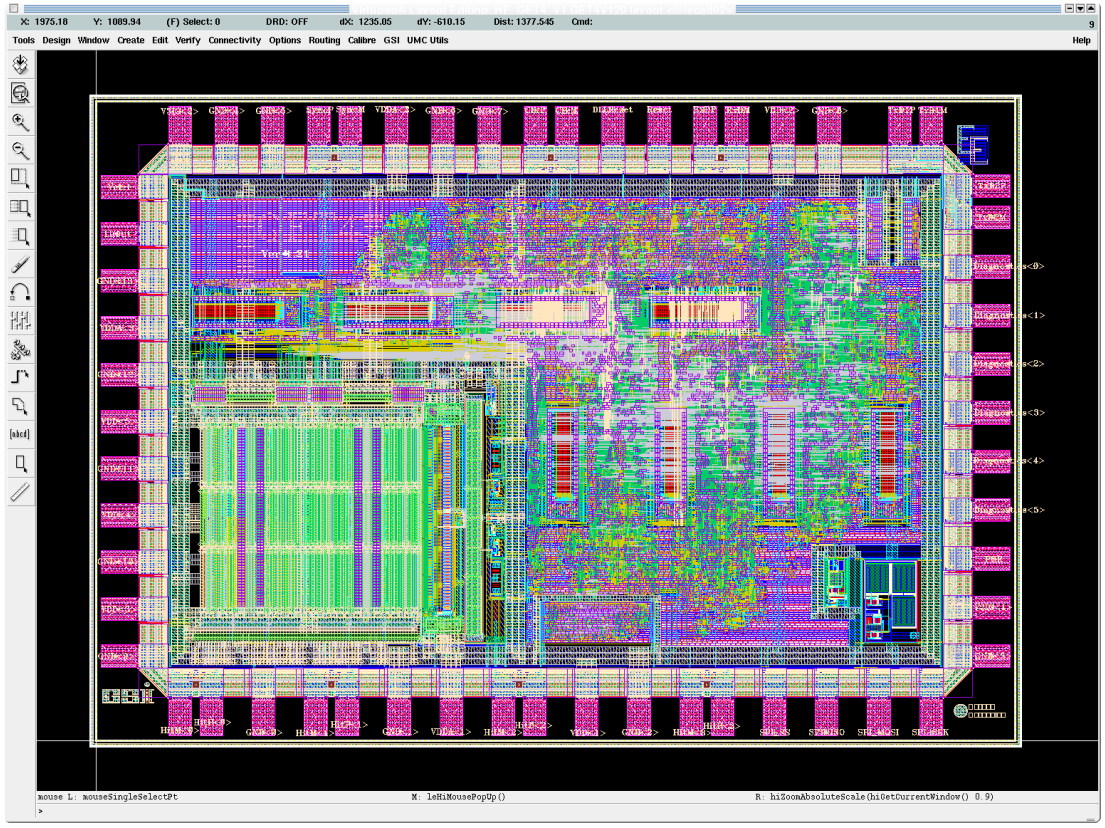


Time resolution



- Internal test pattern generator used to get uncorrelated hits
- Pulse width: 37.8 ps RMS / 26.7 ps RMS Uncorrelated
- Time Difference (Ch0 and Ch3): 27.9 ps RMS / 19.7 ps RMS Uncorrelated

Outlook



- A second iteration of GET4 was designed: GET4 V1.20
  - All known bugs are corrected
  - Pin and functional compatible with GET4 V1.10
  - Same test pcbs and setup can be used for test
- Submitted on Feb. 4th