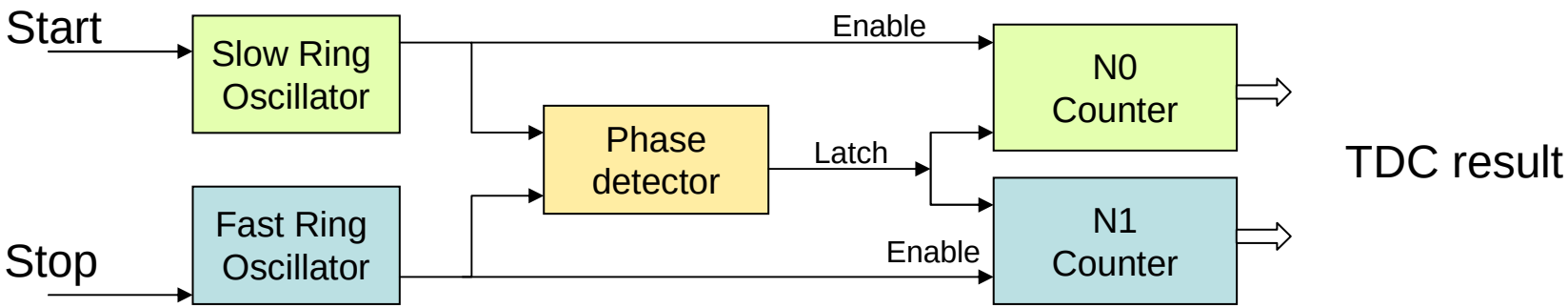
Advanced technique for high accuracy tunable ring oscillator vernier TDC in FPGAs and ASICs

E. Bechetoille, C. Girerd, H. Mathez
Université de Lyon 1, Villeurbanne ;
CNRS/IN2P3, Institut de Physique Nucléaire de Lyon.

Outlines

- Vernier TDC with Ring oscillator principles
- Tuneable ring oscillator architecture
- Test & results
- Limitations of cumulated jitter
- Conclusions

Ring-Oscillator based TDC Architecture

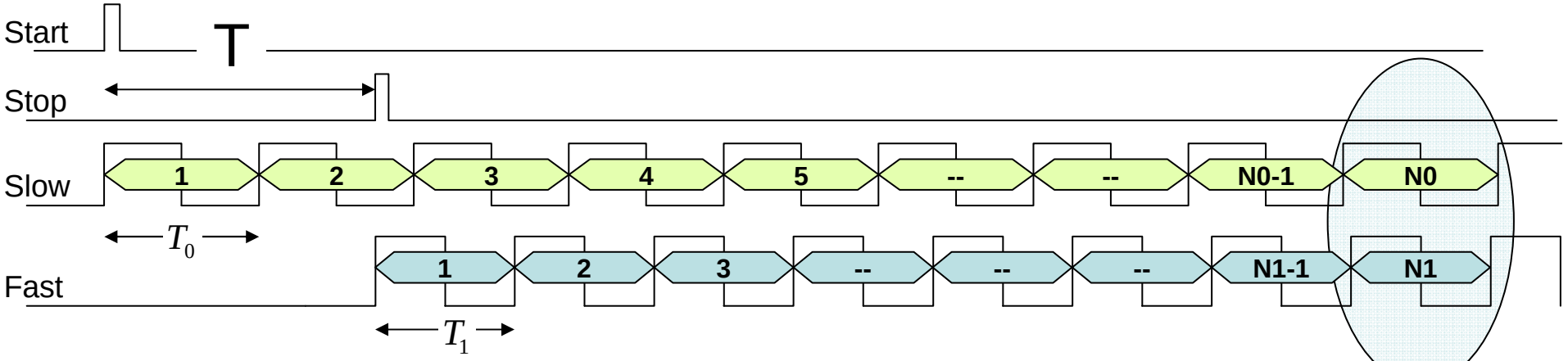


Advantages :

- Simple architecture: Low Area, low consumption, can implemented in standard cells
- The TDC resolution is given by the frequency difference between oscillators
- In theory it can be very small, as small as the frequency difference

-In practice : There are some limitations as jitter …

Ring oscillators Vernier TDC timing



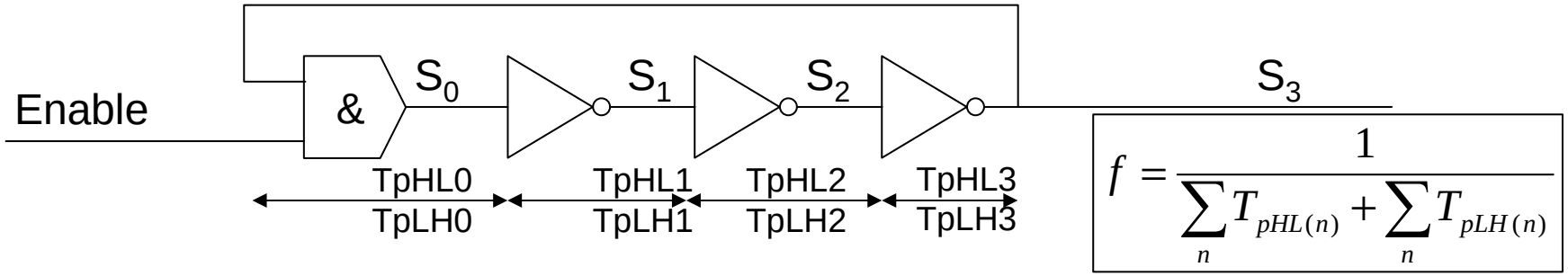
Step 1 The slow oscillator is started on the START signal
Step 2 The fast oscillator is started on the STOP signal
Step 3 At each period the Fast clock get an advance of $\Delta t = (T1-T0)$ over the slow clock
Step 4 The two oscillators are stoped when there are in phase and counters are latched

$$T = (N_0 \cdot T_0) - (N_1 \cdot T_1)$$

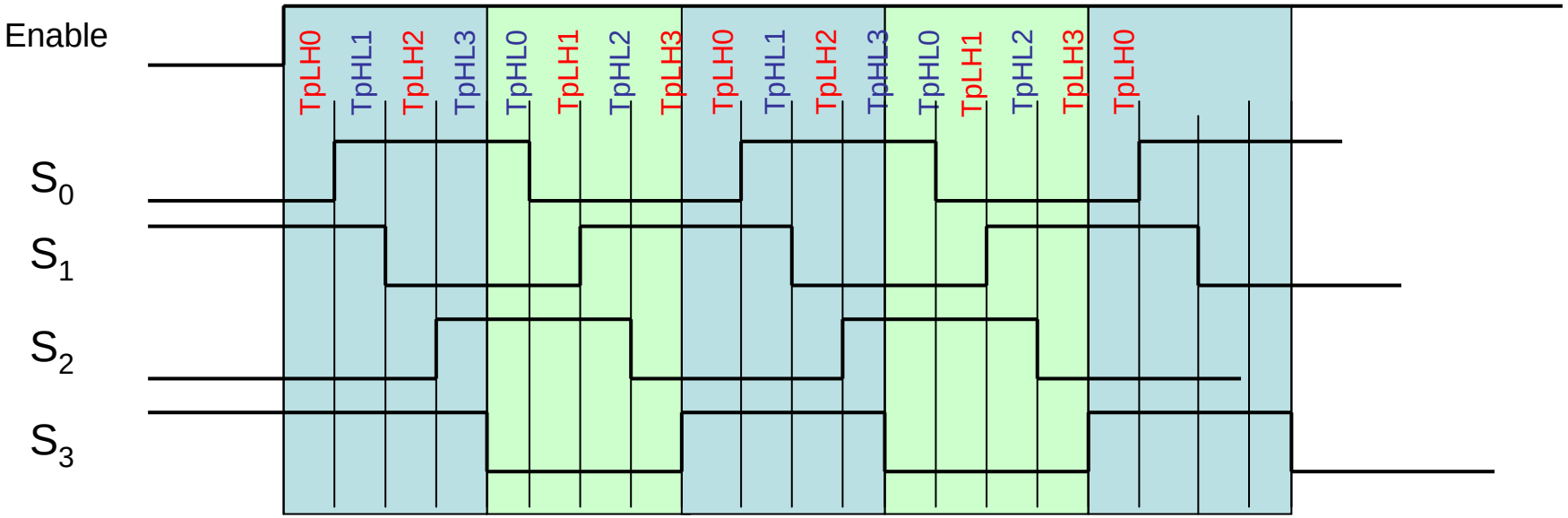
The delay measurement is $$T = T_0 \cdot (N_0 - N_1) + N_1 \cdot \Delta t$$

The résolution is given by: $$T_0 - T_1 = \Delta t$$

Simple Ring oscillator

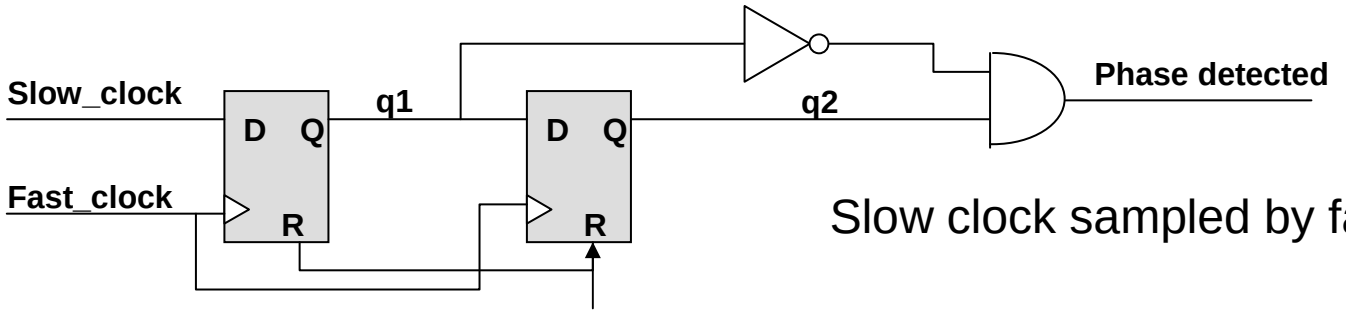


$$f = \frac{1}{\sum_n T_{pHL(n)} + \sum_n T_{pLH(n)}}$$

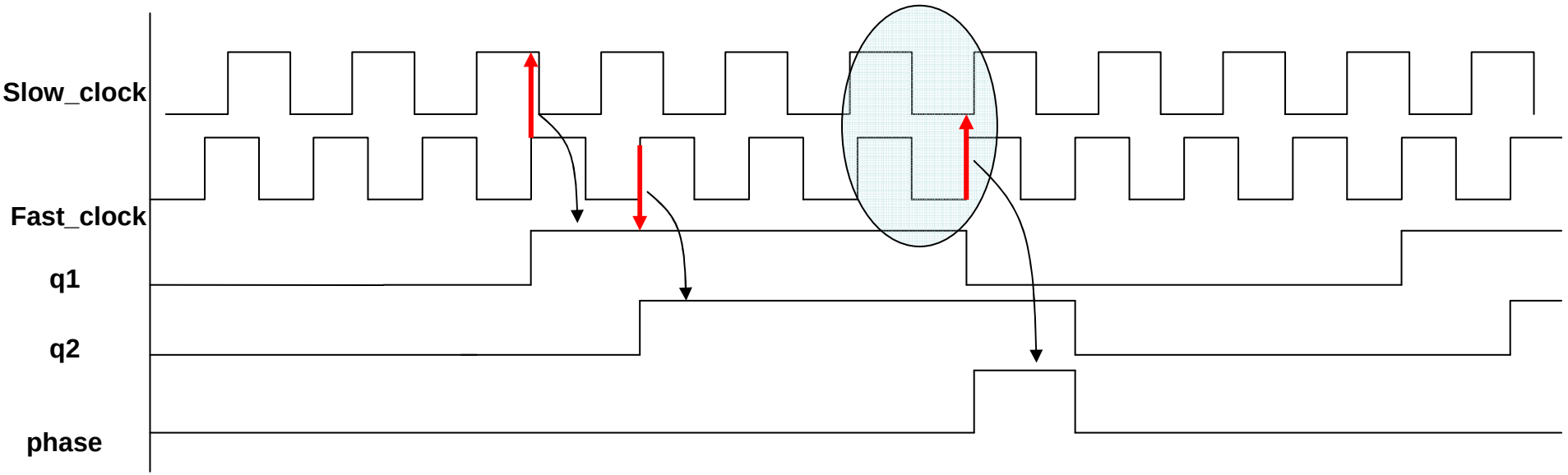


-The number of stages must be odd to allow oscillations
-The number of stages determines the oscillator period
-The OSCILLATOR is gated by and AND for start and stop.

Phase detector



Slow clock sampled by fast clock

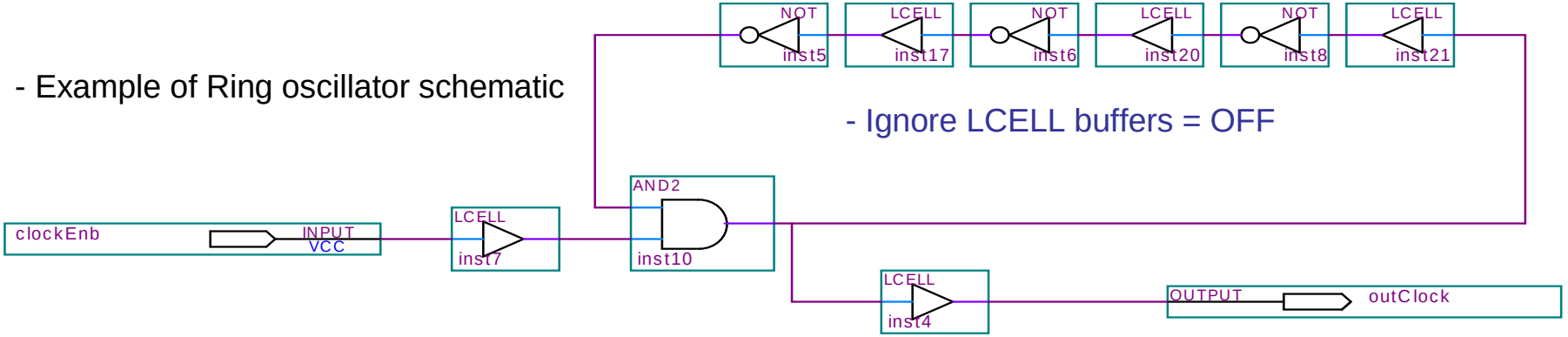


Detects when slow clock previously at state HIGH goes to LOW state

This is the commonly used phase detector in this type of TDC

FPGA IMPLEMENTATION

- Example of Ring oscillator schematic



- For the synthesis tool it is a COMBINATIONAL LOOP which is a bad design practice
- Also without explicit LCELL instantiation the synthesis tool will optimize and reduce the inverter chain to only one inverter

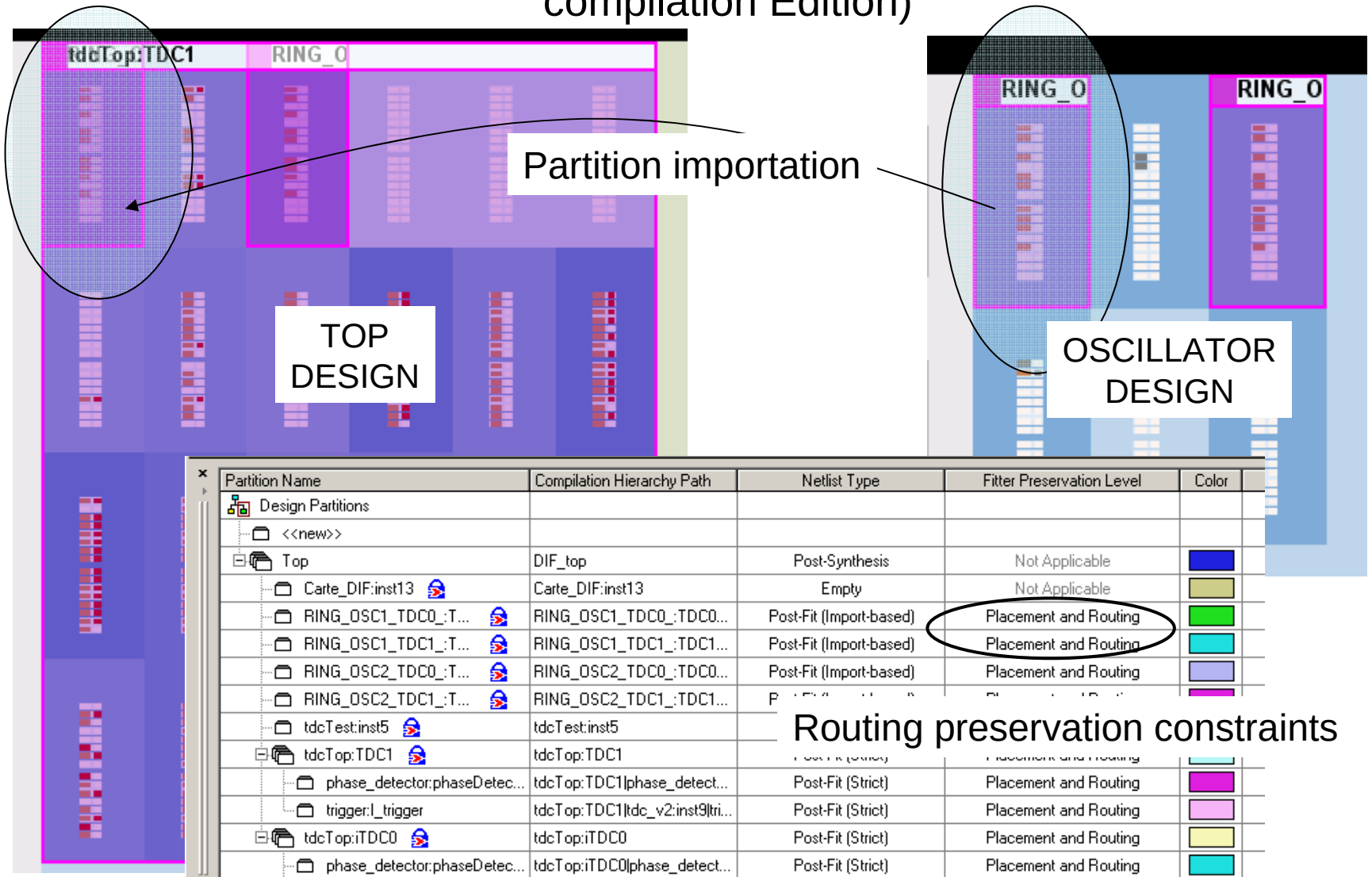
- LCELL buffers prevent the synthesis tool to optimize the design

- A VHDL description is also possible

```
LCELL_0 : cycloneiii_lcell_comb
        generic map (
                dont_touch      =>      "off",
                lut_mask        => "0110100110010110", -- 0x6996
                sum_lutc_input  => "datac",
                lpm_type        => "cycloneiv_lcell_comb"
        )
         port map (
                cin             => cin0,
                combout         => comb0,
                cout            => cout0 ,
                dataa           => fbin,
                datab           => sel(0),
                datac           => sel(1),
                datad           => sel(2)
        );
```

In practice : It is better to synthesize the ring oscillator in open loop in order to allow the timing analysis.

Design methodology (Export/Import of partitation / Regions / Post compilation Edition)



Partition Name	Compilation Hierarchy Path	Netlist Type	Fitter Preservation Level	Color
Design Partitions				
<<new>>				
Top	DIF_top	Post-Synthesis	Not Applicable	
Carte_DIF:inst13	Carte_DIF:inst13	Empty	Not Applicable	
RING_OSC1_TDC0_:T...	RING_OSC1_TDC0_:TDC0...	Post-Fit (Import-based)	Placement and Routing	
RING_OSC1_TDC1_:T...	RING_OSC1_TDC1_:TDC1...	Post-Fit (Import-based)	Placement and Routing	
RING_OSC2_TDC0_:T...	RING_OSC2_TDC0_:TDC0...	Post-Fit (Import-based)	Placement and Routing	
RING_OSC2_TDC1_:T...	RING_OSC2_TDC1_:TDC1...	P...	...	
tdcTest:inst5	tdcTest:inst5			
tdcTop:TDC1	tdcTop:TDC1			
phase_detector:phaseDetec...	tdcTop:TDC1|phase_detect...	Post-Fit (Strict)	Placement and Routing	
trigger:I_trigger	tdcTop:TDC1|tdc_v2:inst9|tri...	Post-Fit (Strict)	Placement and Routing	
tdcTop:iTDC0	tdcTop:iTDC0	Post-Fit (Strict)	Placement and Routing	
phase_detector:phaseDetec...	tdcTop:iTDC0|phase_detect...	Post-Fit (Strict)	Placement and Routing	
trigger:I_trigger_2	tdcTop:iTDC0|tdc_v2:inst9|tr...	Post-Fit (Strict)	Placement and Routing	

Limitations of classic implementation

- This classic architecture has been used successfully in some physics applications see

Implementation of sub-nanoseconds TDC in FPGA: applications to time-of-flight analysis in muon radiography

J. Marteau (IPNL), J. De Bremond D'ars (GR), D. Gibert (GR, IPGP), K. Jourde (IPGP), S. Gardien (IPNL), C. Girerd (IPNL), J.-C. Ianigro(IPNL)

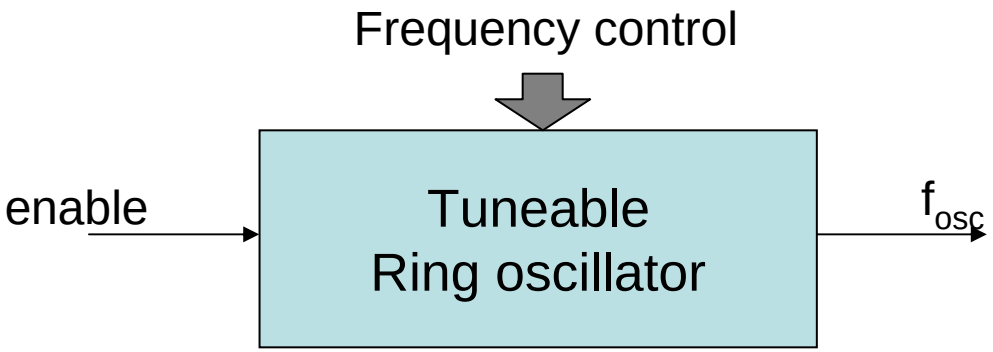
The main problem of classic method is that is very hard to target a specific resolution for the TDC (In fact to have a specific period difference between Slow and Fast oscillator).

Several iterations and manual placement and routing can be necessary to obtain the wanted resolution.

That's the reason why we think to an other technique to control the period difference of ring oscillators.

ADVANCED TECHNIQUE OF TUNEABLE RING OSCILLATORS IN FPGA

TUNEABLE RING OSCILLATORS



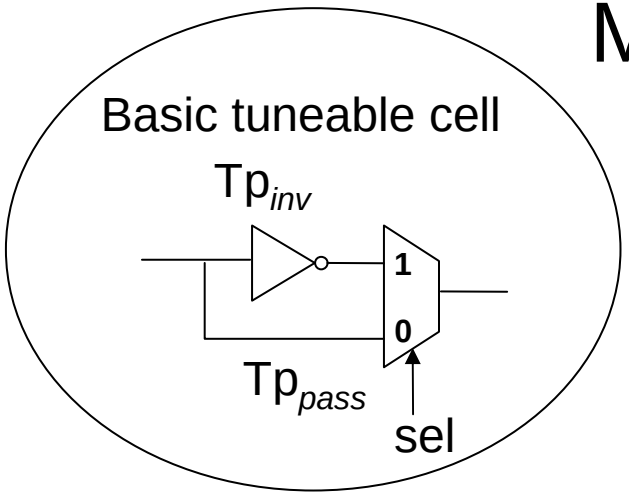
Goal :

- Modifying the frequency by a digital control, on line.
- Being able to target a specific Δt between two oscillators

Techniques:

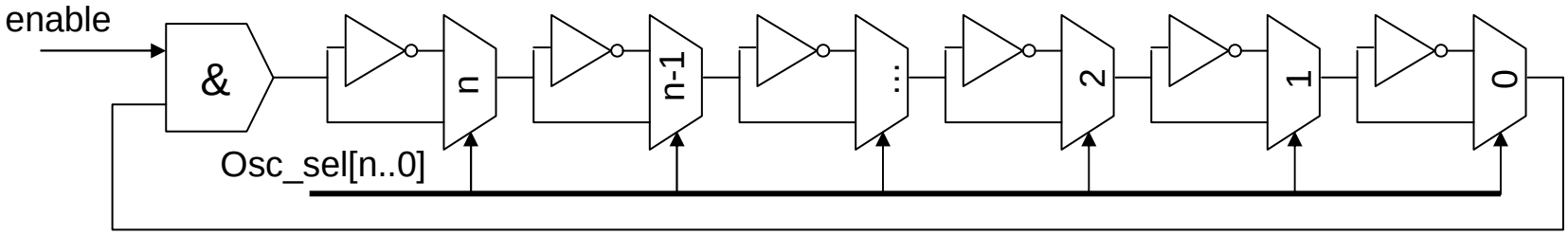
- Using the propagation delay variations of logic cells
- Modifying the path of the signal in the chain
- Preference for structures with low variations

TUNEABLE OSCILLATORS
Moving inverters ©



$$Tp_{pass} \neq Tp_{inv}$$

sel	Tp
0	Tp_{pass}
1	Tp_{inv}

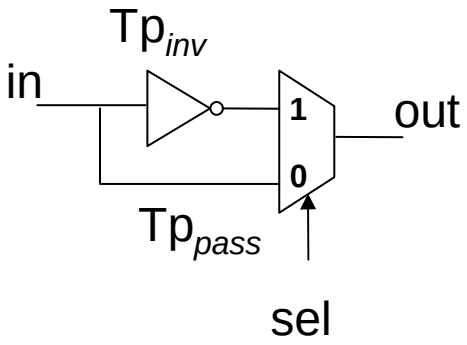


Condition : Somme Cells must be different (e.g. $T_{pass}(n) /= T_{pass}(n+1)$)

- Involuntary due to silicon dispersion.
- Voluntary by choosing inverters of difference strength (propagation delay)

TUNEABLE OSCILLATORS
Moving inverters ©

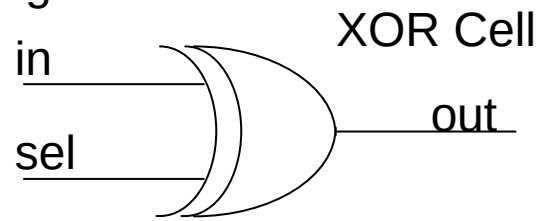
Basic tuneable cell



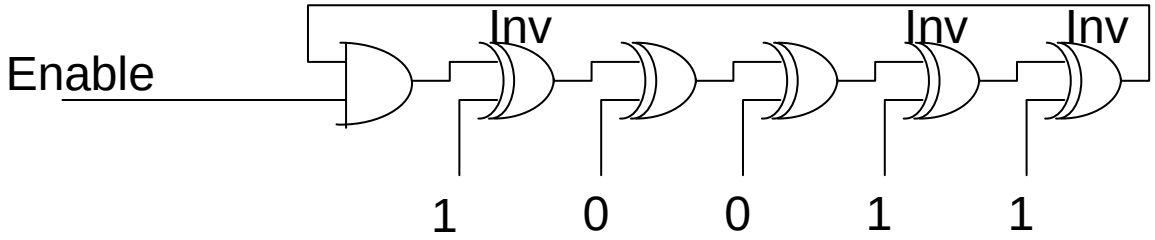
Sel	In	Out
0	0	0
0	1	1
1	0	1
1	1	0

passing

inverter



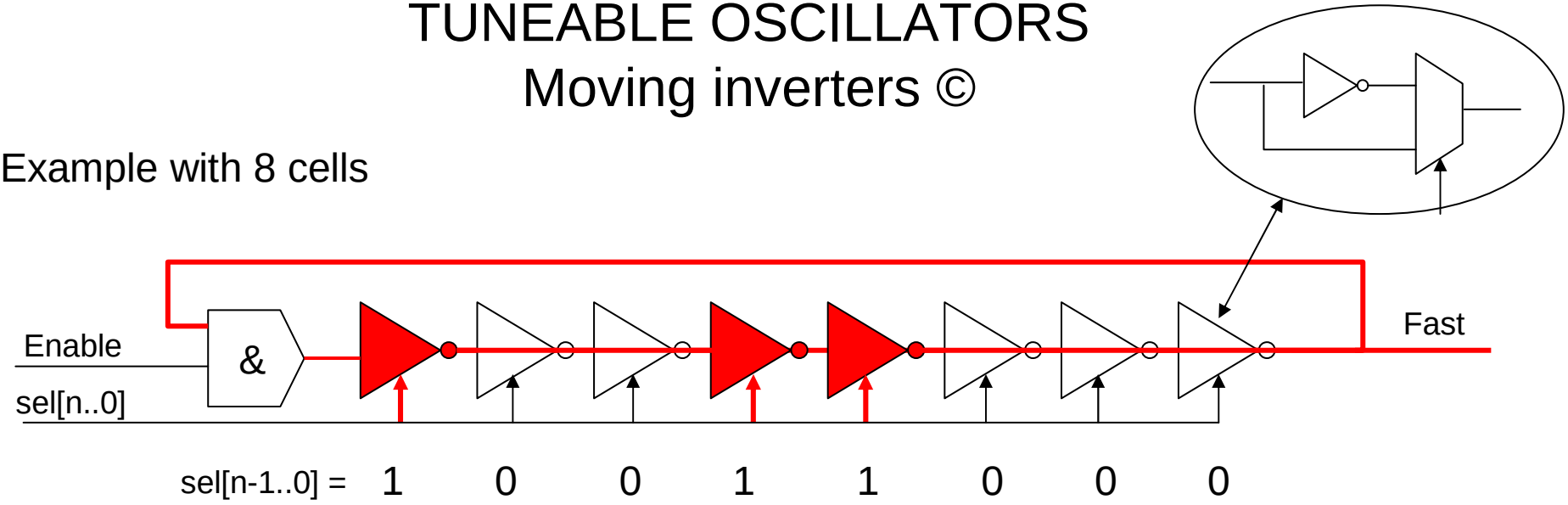
The basic element can be replaced by an XOR cell



The control word can change the position and the number of inverters

TUNEABLE OSCILLATORS
Moving inverters ©

Example with 8 cells



The number of selected inverters must be odd (oscillation condition)

For a chain of n Elements the number of possibilities Y is :

$$Y = \sum_{\substack{p=2k+1 \\ (k\in \mathrm{N},\, p\le n)}} C_n^p$$

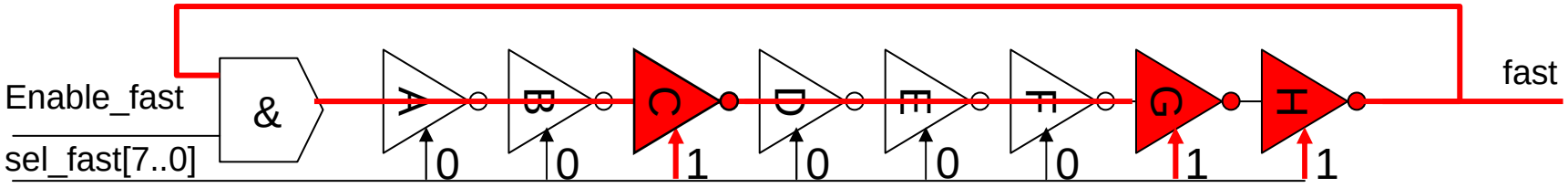
$$\mathrm{Y} = C_8^1 + C_8^3 + C_8^5 + C_8^7 = 128$$

The number of TDCs (combination of 2 oscillators)

$$128^2 = 16384$$

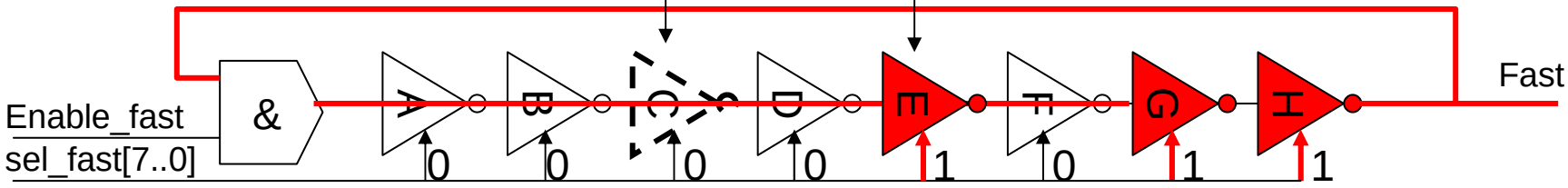
But how to quantify the differences between all these combinations

TUNEABLE OSCILLATORS
Moving inverters ©



$T_1 = Tp_{pass}A + Tp_{pass}B + \mathbf{Tp_{inv}C} + Tp_{pass}D + Tp_{pass}E + Tp_{pass}F + \mathbf{Tp_{inv}G} + \mathbf{Tp_{inv}H}$

We change on bit in the select word



$T_2 = Tp_{pass}A + Tp_{pass}B + \mathbf{Tp_{ass}C} + Tp_{pass}D + \mathbf{Tp_{inv}E} + Tp_{pass}F + \mathbf{Tp_{inv}G} + \mathbf{Tp_{inv}H}$

$$T_2\text{-}T_1 = (\mathbf{T_{pass}C} - \mathbf{T_{pinv}C}) - (\mathbf{T_{pass}E} - T_{inv}E)$$

Conditions to obtain a delay variation : Tpass(i) ≠ Tinv(i) Tpass(i) ≠ Tpass(k)
Tinv(i) ≠ Tinv(k)

TUNEABLE OSCILLATORS
Moving inverters ©

More generally if we have **N** available family of XOR gate
The list of possible differences for one change in one oscillator is **N.(N-1)**

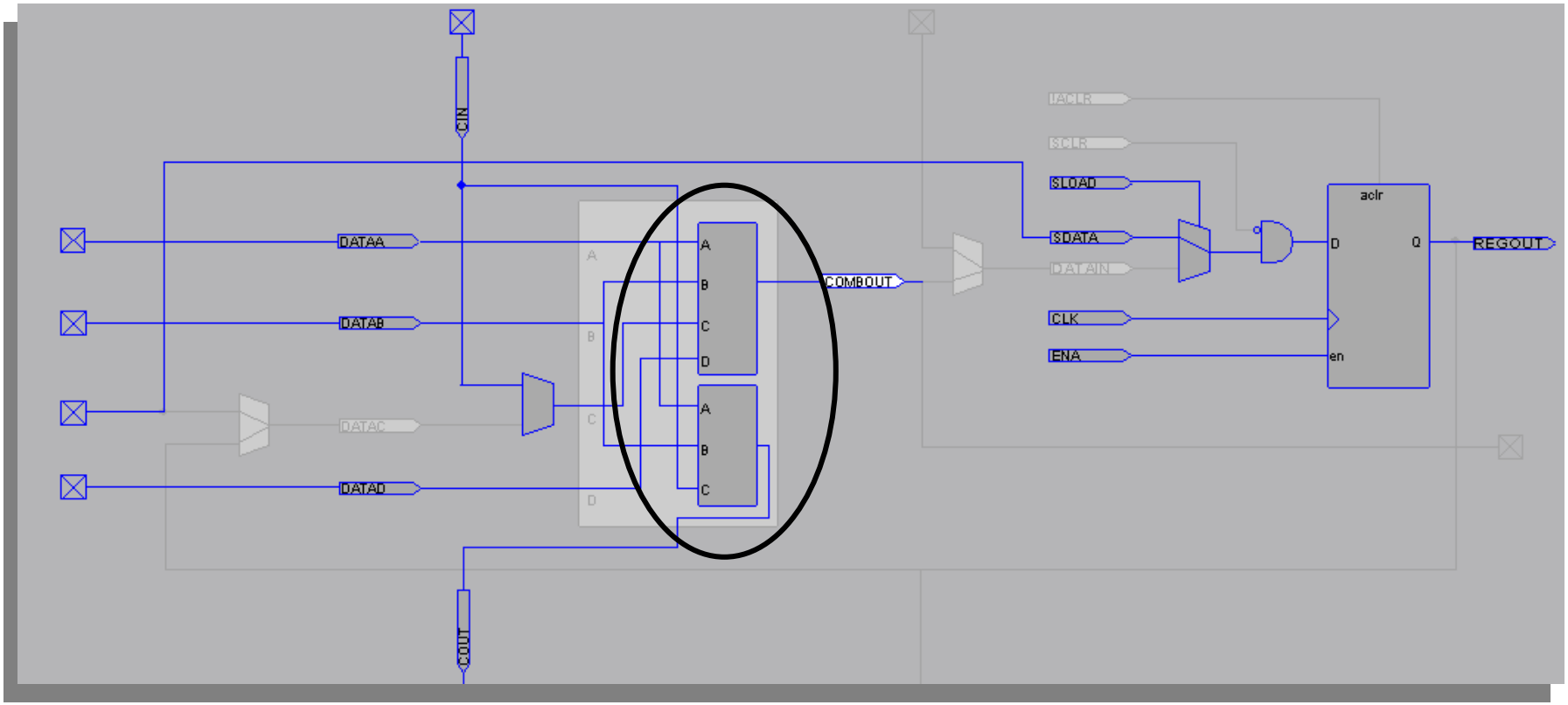
$$(\Delta_{a,b})_{a\neq b\in[1,n]} = (Tpass_a - Tinv_a) - (Tpass_b - Tinv_b)$$

These differences depends one the XOR family (a vs b) and between function passing/inverting in the same family (pass/inv)

CONCLUSION : In order to obtain frequency variations, we must use the maximum of different XOR (propagation delays) in each oscillator

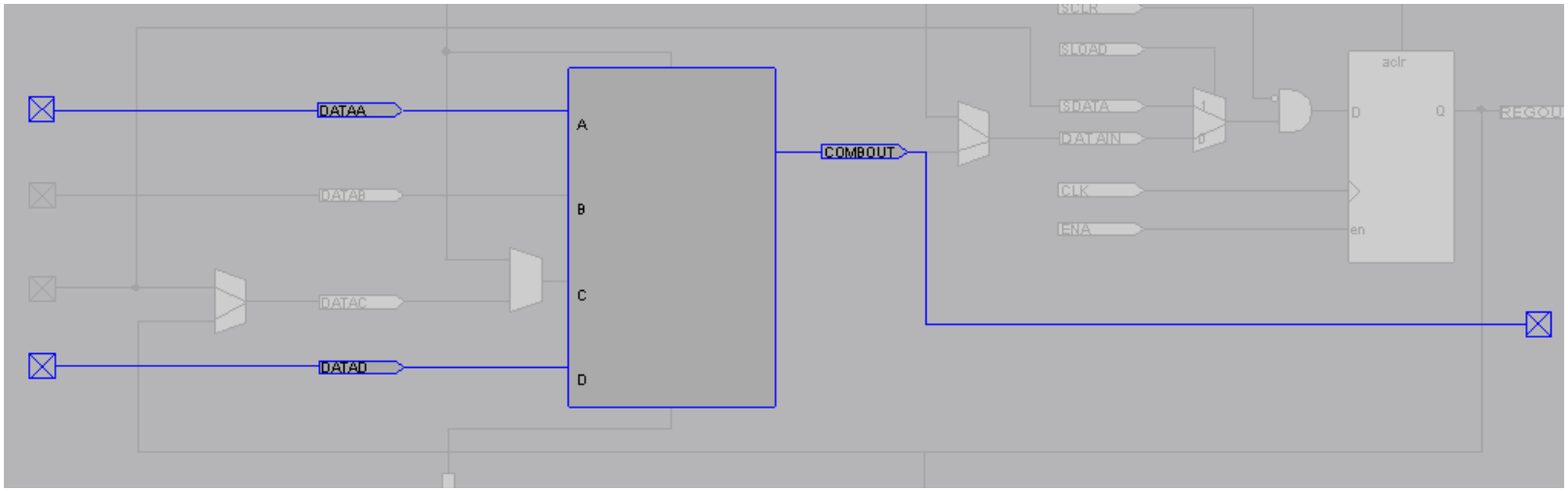
BUT How to implement XOR cells with different propagation delay in a FPGA ?

LCELL STRUCTURE
(ALTERA exemple Cyclone III)



- The combinational parts is Look Up table (65536) possibilities of equations of 4 inputs
- The oscillator stages are implemented in one of fewer LCELLs

Propagation delay characteristics of LCELL in ALTERA FPGA

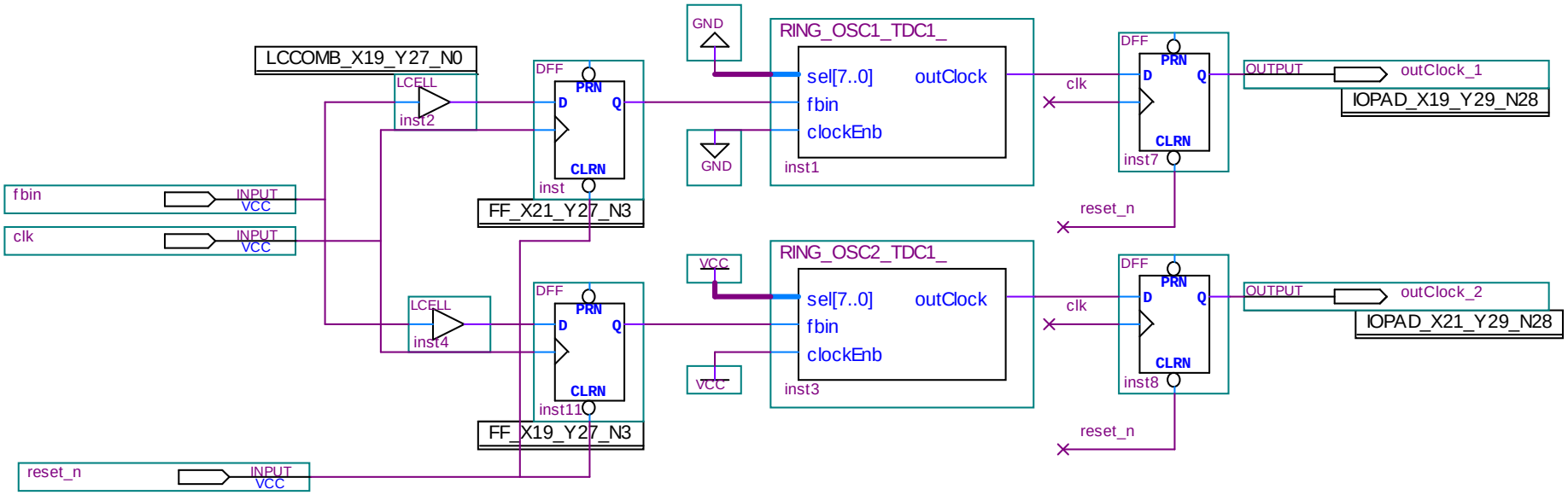


We observed that in the ALTERA FPGA the XOR propagation delays depends on the input which is used. Example with 4 XOR gates using different inputs

Entrée	RR (rising input / Rising output) ps	FF Falling input Falling output ps
A	471	481
B	494	496
C	324	316
D	177	155

Open loop synthesis of ring oscillators

-Allow to specify SDC constraints and control the first synthesis results
-Here a clock constraint is enought to obtain a minimum XOR chain propagation delay



- A partition and a region for each oscillator

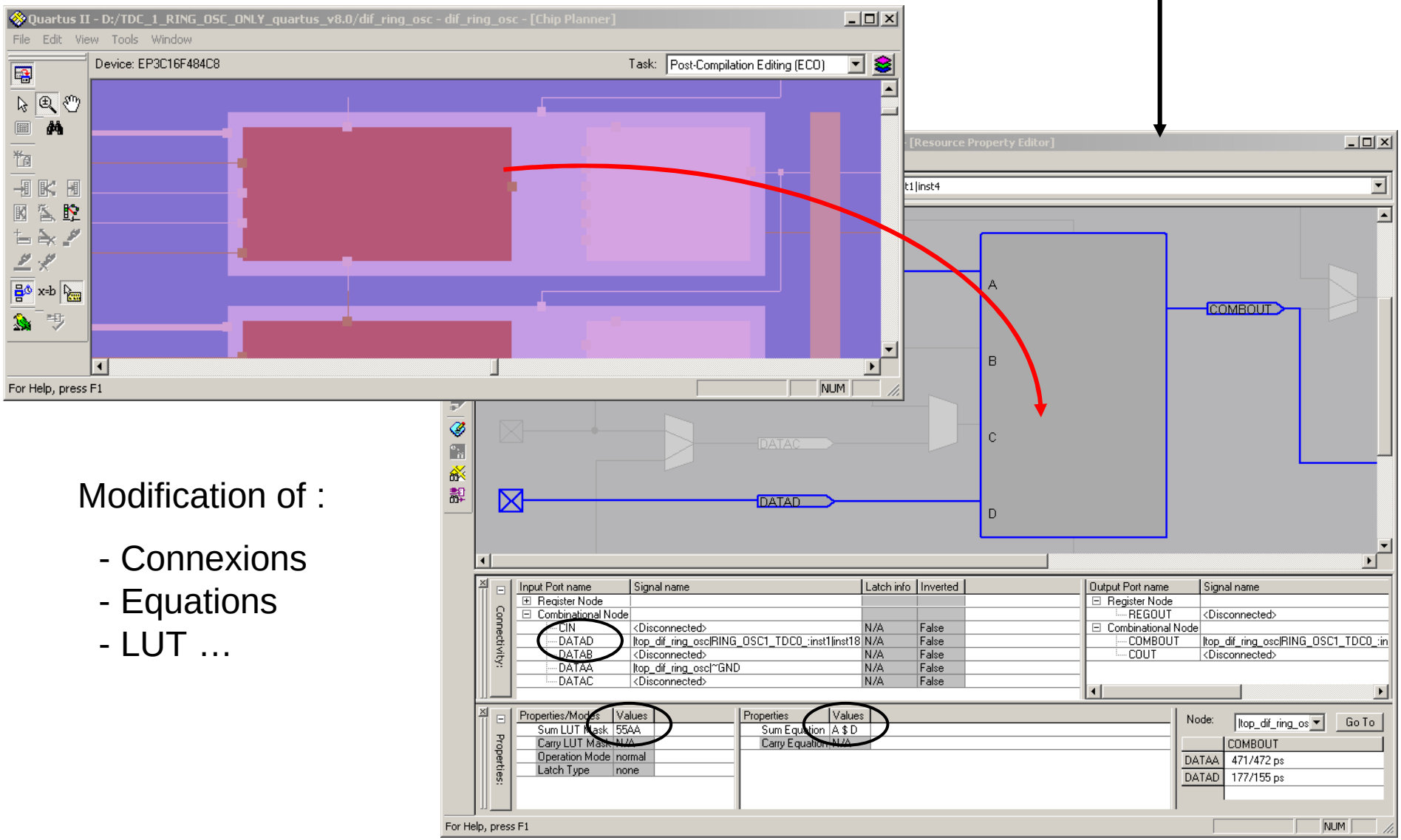
Report path results (Timequest) for open loop synthesis

Total	Incr	RF	Type	Fanout	Location	Element
0	0			1	FF_X2_Y28_N17	inst
0	0	FF	CELL	1	FF_X2_Y28_N17	inst|q
0.53	0.53	FF	IC	1	LCCOMB_X3_Y28_N12	inst1|inst21|datac
0.846	0.316	FF	CELL	1	LCCOMB_X3_Y28_N12	inst1|inst21|combout
1.106	0.26	FF	IC	1	LCCOMB_X3_Y28_N18	**inst1|inst20|datad**
1.274	**0.168**	FR	CELL	1	LCCOMB_X3_Y28_N18	inst1|inst20|combout
1.514	0.24	RR	IC	1	LCCOMB_X3_Y28_N16	**inst1|inst17|datad**
1.691	**0.177**	RR	CELL	1	LCCOMB_X3_Y28_N16	inst1|inst17|combout
1.931	0.24	RR	IC	1	LCCOMB_X3_Y28_N6	**inst1|inst24|datad**
2.108	**0.177**	RR	CELL	1	LCCOMB_X3_Y28_N6	inst1|inst24|combout
2.349	0.241	RR	IC	1	LCCOMB_X3_Y28_N24	**inst1|inst30|datad**
2.526	**0.177**	RR	CELL	1	LCCOMB_X3_Y28_N24	inst1|inst30|combout
2.768	0.242	RR	IC	1	LCCOMB_X3_Y28_N2	**inst1|inst22|datad**
2.945	**0.177**	RR	CELL	1	LCCOMB_X3_Y28_N2	inst1|inst22|combout
3.187	0.242	RR	IC	1	LCCOMB_X3_Y28_N20	**inst1|inst19|datad**
3.364	**0.177**	RR	CELL	1	LCCOMB_X3_Y28_N20	inst1|inst19|combout
3.604	0.24	RR	IC	1	LCCOMB_X3_Y28_N10	**inst1|inst18|datad**
3.781	**0.177**	RR	CELL	1	LCCOMB_X3_Y28_N10	inst1|inst18|combout
4.021	0.24	RR	IC	1	LCCOMB_X3_Y28_N0	**inst1|inst4|datad**
4.198	**0.177**	RR	CELL	1	LCCOMB_X3_Y28_N0	inst1|inst4|combout
4.833	0.635	RR	IC	1	DDIOOUTCELL_X3_Y29_N32	inst7|d
5.278	0.445	RR	CELL	1	DDIOOUTCELL_X3_Y29_N32	inst7

- The synthesis tool choose the fastest LCELL input for all cells (input D / 177 ps)

- the next step is to manually change the input to obtain an XOR chain with a maximum of possible delay adjustment.

Post compilation editing (1) ECO
with **CHIP PLANNER** and **RESSOURCE PROPERTY EDITOR**



Modification of :

- Connexions
- Equations
- LUT …

Example of chain with 8 XORs using inputs B,C,D after Post compilation Edition (input A is used for selection)

Timing report for a chain of 8 XOR. Routing has been manually modified to use 3xC inputs, 3xD inputs and 2xB inputs.

Total	Incr	RF	Type	Fanout	Location	Element
0	0	RR	CELL	1	FF_X4_Y28_N17	inst11|q
0	0			1	FF_X4_Y28_N17	inst11
0.512	0.512	RR	IC	1	LCCOMB_X5_Y28_N12	inst2|inst21|datac
0.836	0.324	RR	CELL	1	LCCOMB_X5_Y28_N12	inst2|inst21|combout
1.283	0.447	RR	IC	1	LCCOMB_X5_Y28_N18	inst2|inst20|datac
1.61	0.327	RR	CELL	1	LCCOMB_X5_Y28_N18	inst2|inst20|combout
1.85	0.24	RR	IC	1	LCCOMB_X5_Y28_N16	inst2|inst17|datad
2.005	0.155	RF	CELL	1	LCCOMB_X5_Y28_N16	inst2|inst17|combout
2.312	0.307	FF	IC	1	LCCOMB_X5_Y28_N6	inst2|inst24|datab
2.806	0.494	FR	CELL	1	LCCOMB_X5_Y28_N6	inst2|inst24|combout
3.263	0.457	RR	IC	1	LCCOMB_X5_Y28_N24	inst2|inst30|datac
3.59	0.327	RR	CELL	1	LCCOMB_X5_Y28_N24	inst2|inst30|combout
3.832	0.242	RR	IC	1	LCCOMB_X5_Y28_N2	inst2|inst22|datad
3.987	0.155	RF	CELL	1	LCCOMB_X5_Y28_N2	inst2|inst22|combout
4.296	0.309	FF	IC	1	LCCOMB_X5_Y28_N20	inst2|inst19|datab
4.79	0.494	FR	CELL	1	LCCOMB_X5_Y28_N20	inst2|inst19|combout
5.242	0.452	RR	IC	1	LCCOMB_X5_Y28_N10	inst2|inst18|datac
5.569	0.327	RR	CELL	1	LCCOMB_X5_Y28_N10	inst2|inst18|combout
5.809	0.24	RR	IC	1	LCCOMB_X5_Y28_N0	inst2|inst4|datad
5.986	0.177	RR	CELL	1	LCCOMB_X5_Y28_N0	inst2|inst4|combout
6.618	0.632	RR	IC	1	DDIOOUTCELL_X5_Y29_N32	inst8|d

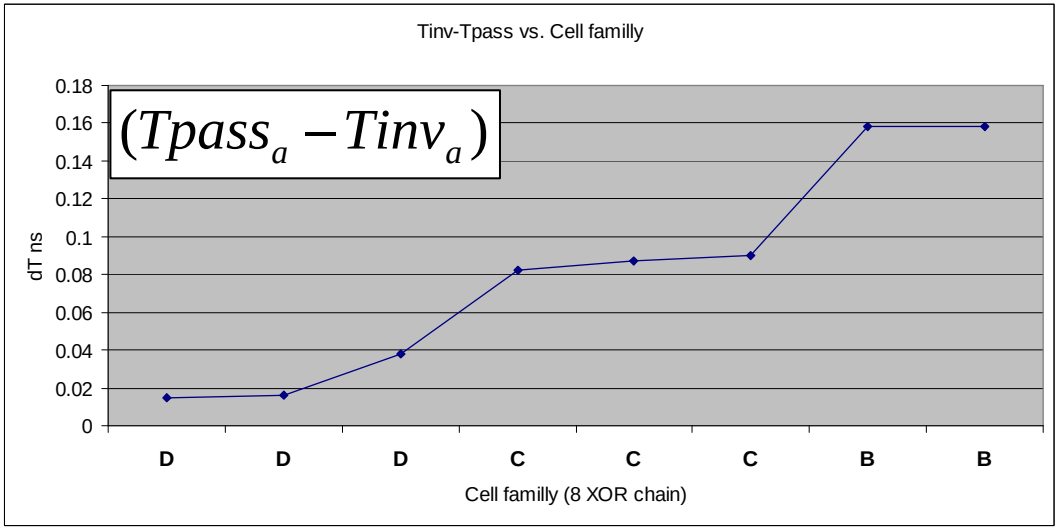
Consequence on period variation in the ring oscillator

As seen before the list of possible variation is in theory given by

$$(\Delta_{a,b})_{a\neq b\in[1,n]} = (Tpass_a - Tinv_a) - (Tpass_b - Tinv_b)$$

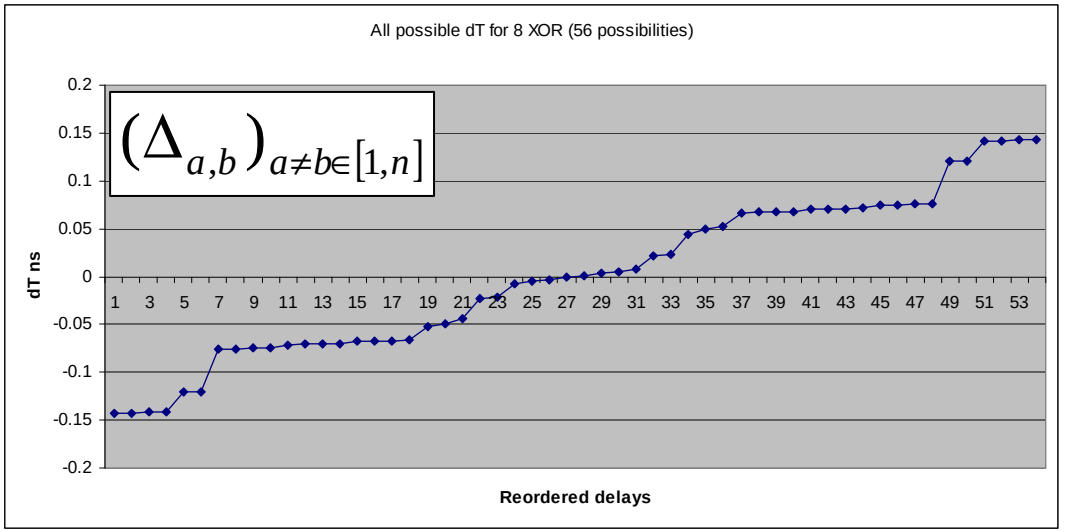
Remark :It is possible to have an estimation using the min/max delay as the timing analyser make a transitions analysis

(T_{max}–T_{min}) for each 8 cells in the chain



Then we calculate all the possible delays

N.(N-1) = 7*8 = 56



We can note the dynamic of variation

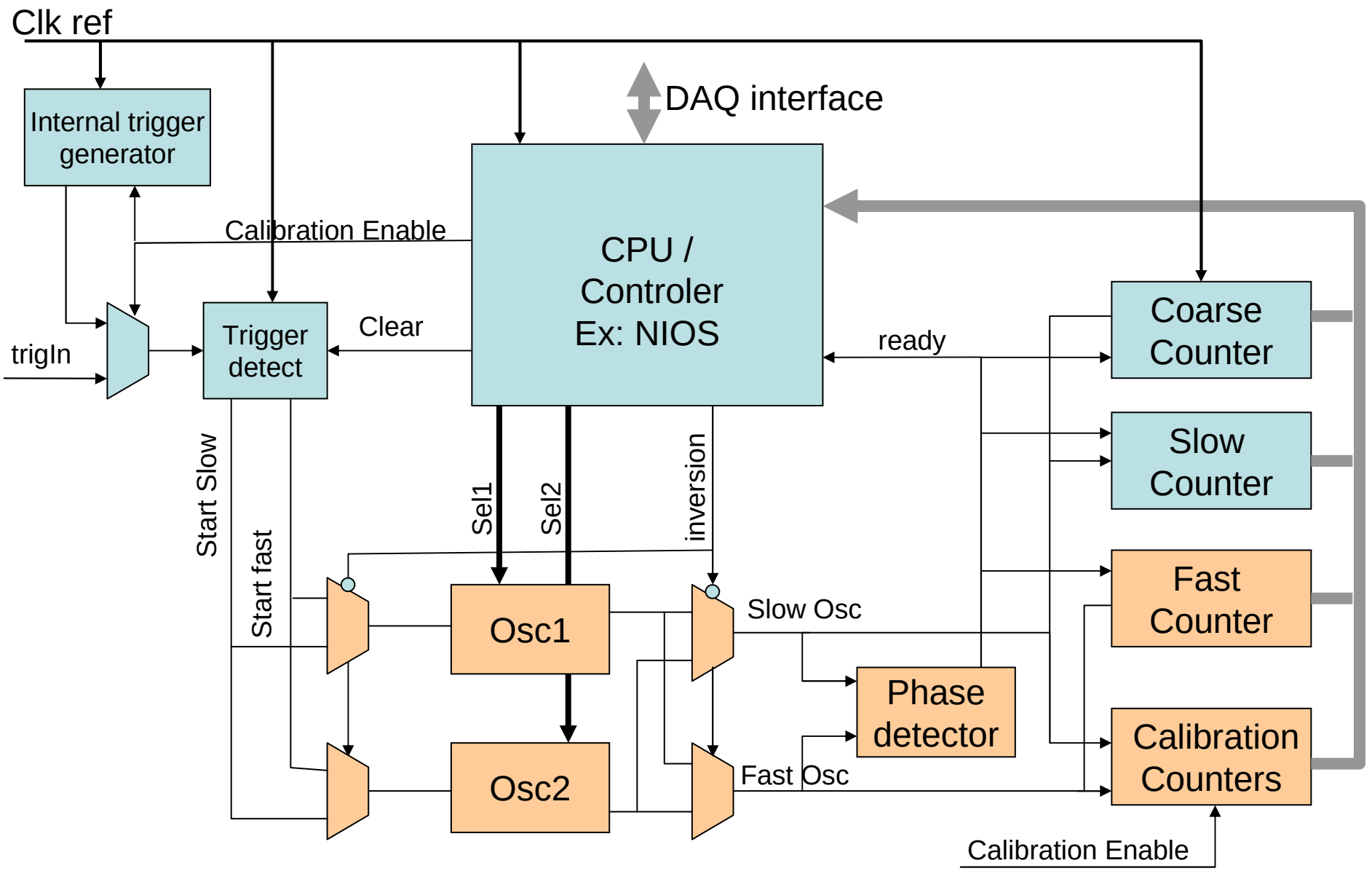
~ 150 ps

and the resolution

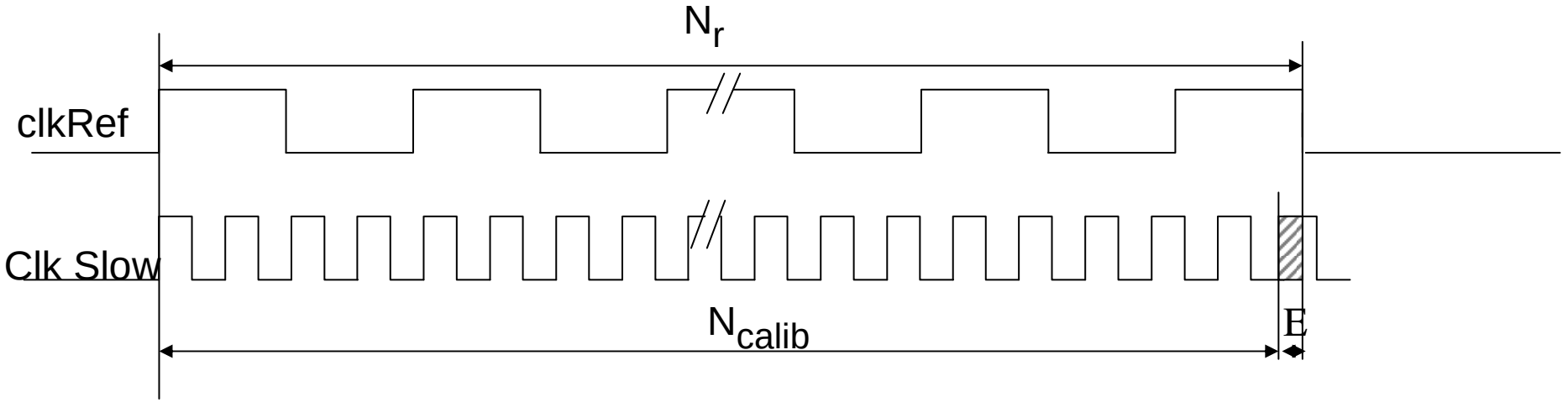
~ few ps

TESTS & RESULTS

TDC bloc diagram



T_{slow} calibration



- Tslow can be calibrate using an external and stable clock reference

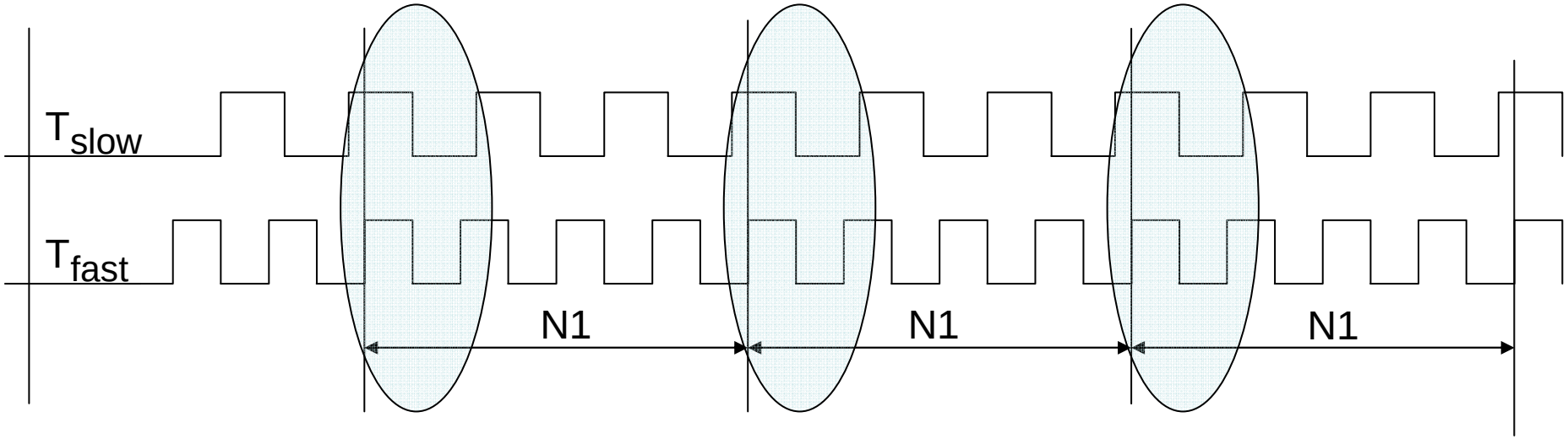
$$N_{calib} \cdot T_{slow} + E = N_{ref} \cdot T_{ref}$$

$$T_{slow} = \frac{T_{ref} \cdot N_{ref}}{N_{calib}} - \left(\frac{E}{N_{calib}} \right)$$

$$Err_{\max} = \frac{T_{slow}}{N_{calib}}$$

- Error on Tslow is minimized by N_{calib}

Calibration of T_{fast} or Δt



- If SLOW and FAST are free running they will periodically reach a minimum phase shift

-The number of clocks periods between this coincidences gives the period difference

Vernier definition gives $$\Delta t = \frac{T_{slow}}{N_1}$$

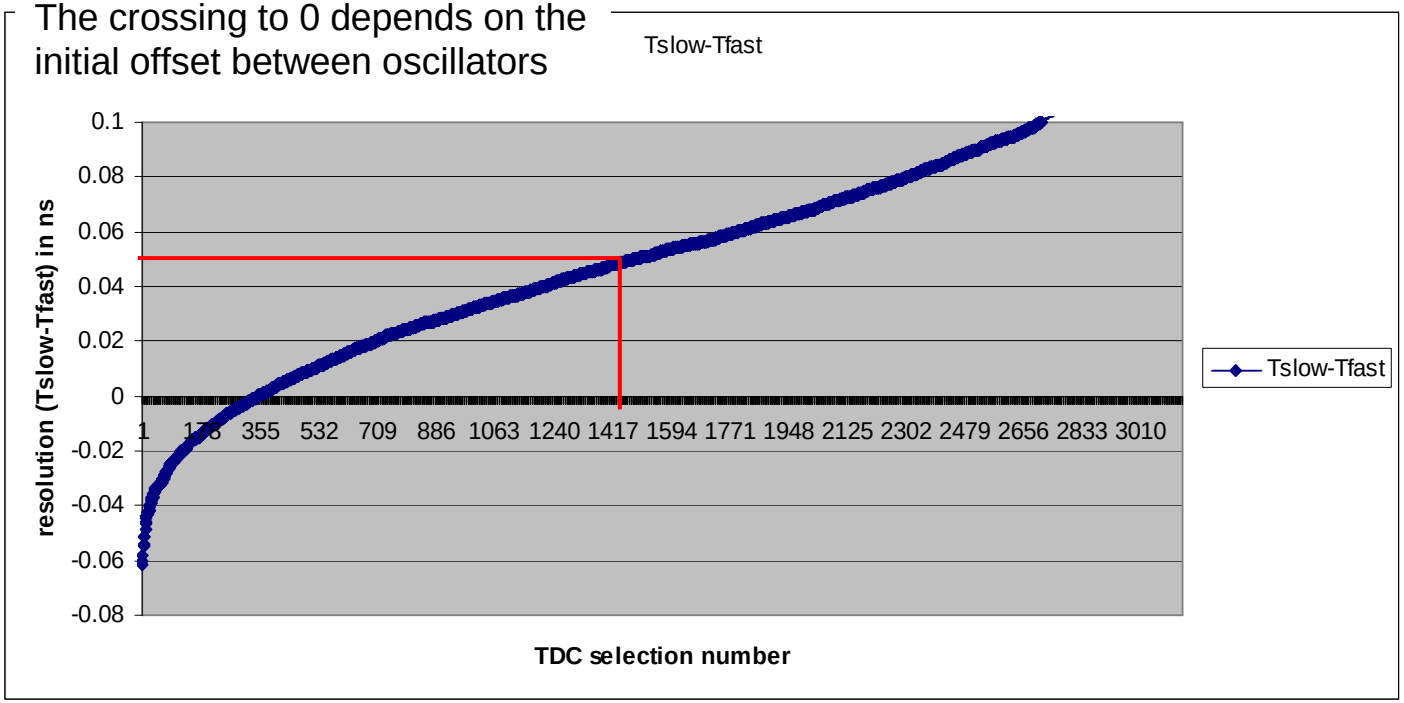
We can also measure directly Tfast with the clock reference as for Tslow

Calibration results on hardware

The calibration consist in sweeping all the combination of slow and fast oscillators (in this case with 3 inverters among 8) when obtained 3146 combinations

Slow	Fast	Res.
162	67	0.05032601
112	138	0.05037702
148	22	0.05039116
208	138	0.05039419
26	164	0.05041987

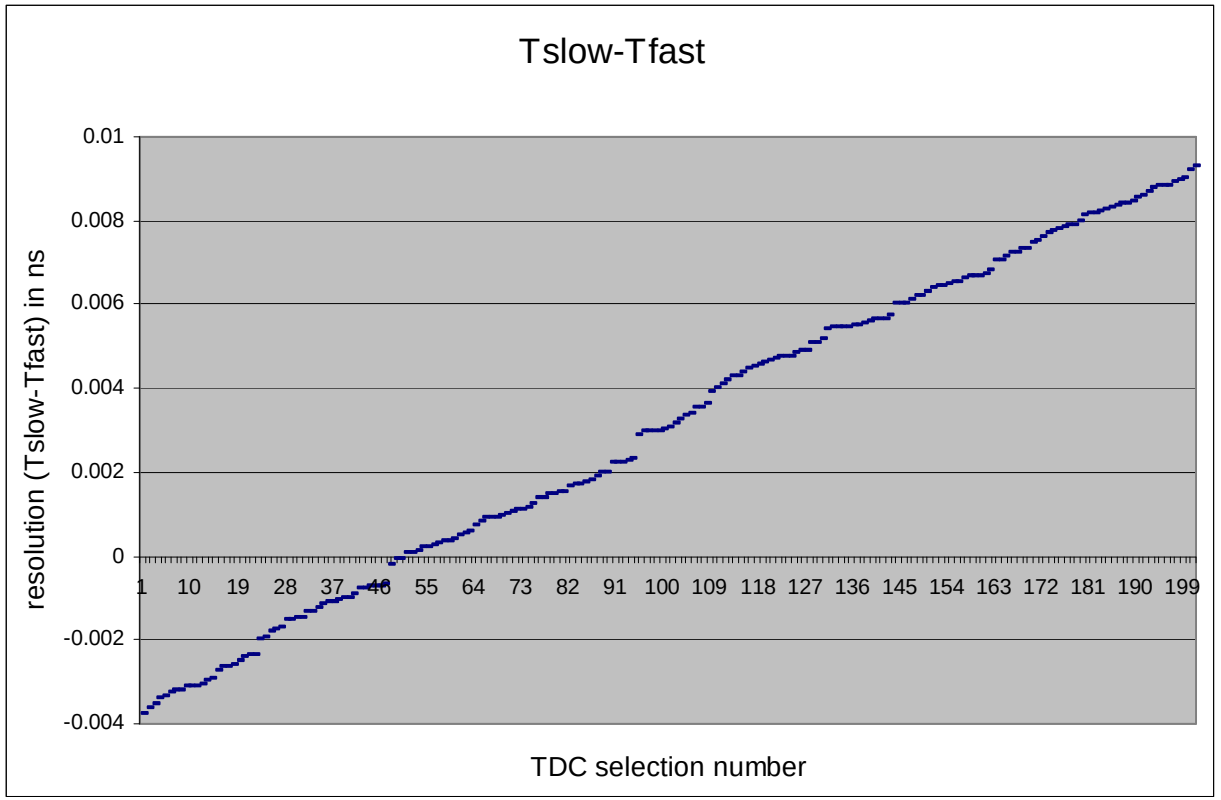
This means that the TDC can be configured with a resolution from 1 ps* to 100 ps.



The calibration result is a list of resolutions with the corresponding TDC selection. We can choose any TDC among these 3146.

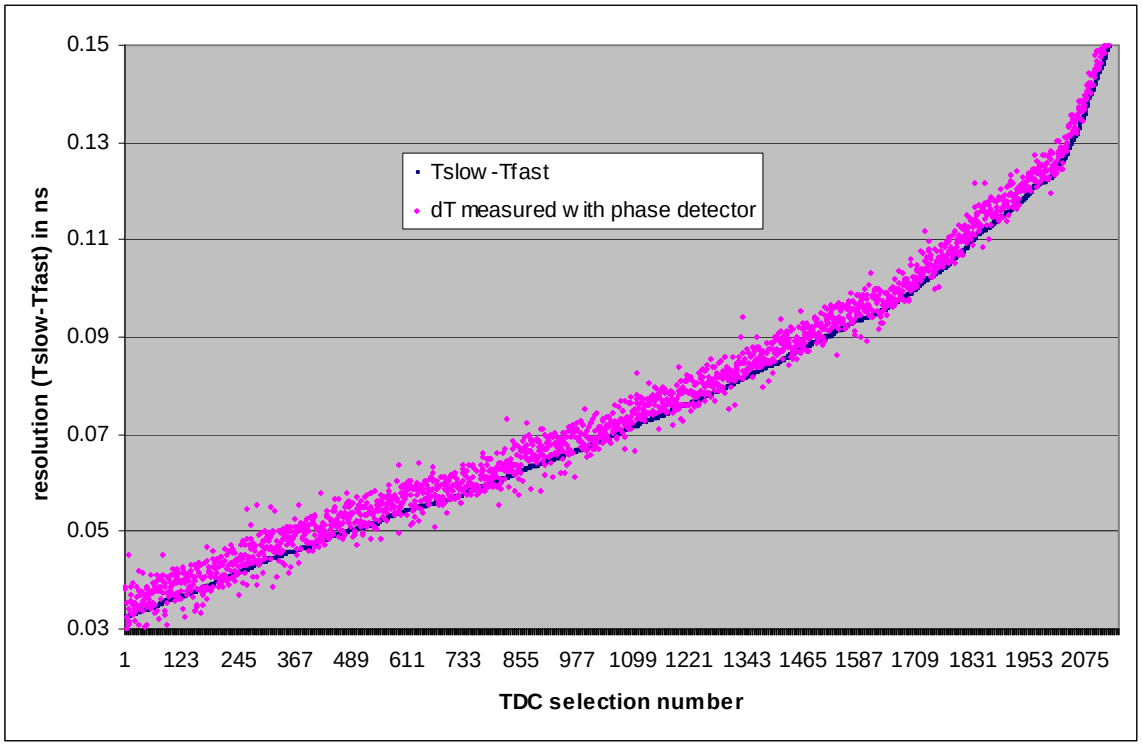
*At least the oscillators difference can be as low as 1 ps but it doesn't guarantee that the will work well at this resolution

Calibration results
minimum frequency differences



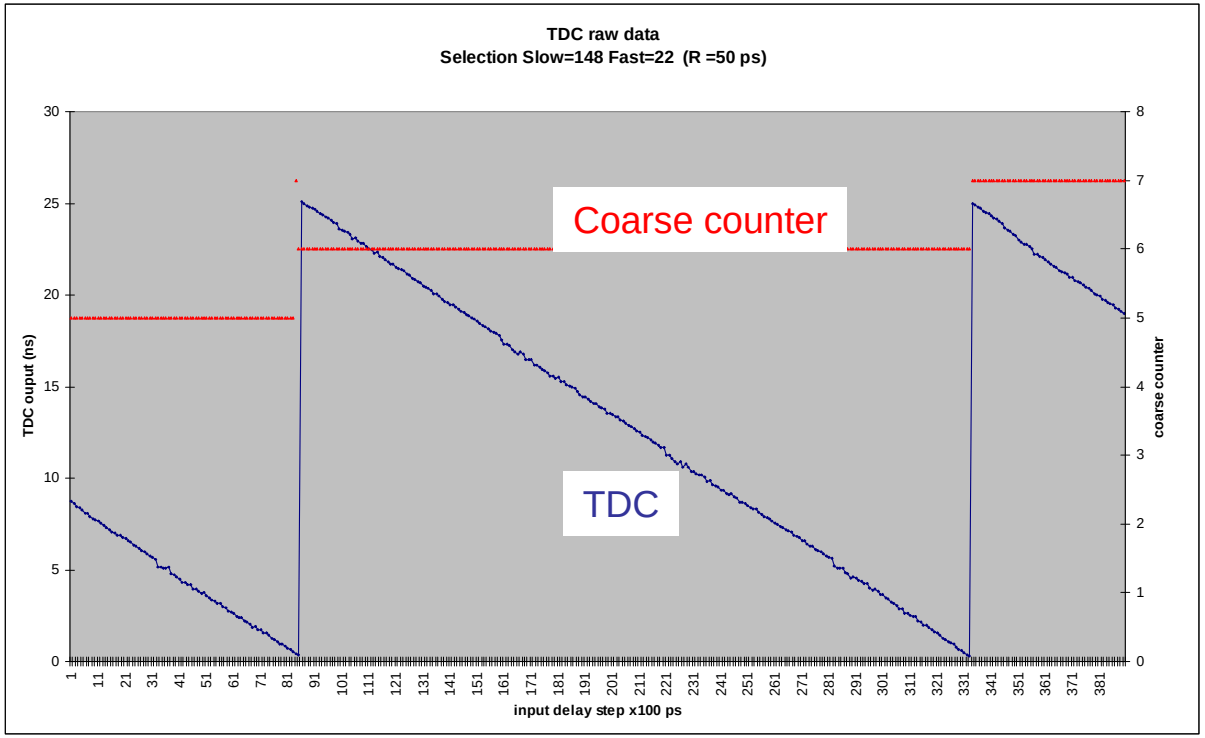
We observed that period differences (between slow and fast oscillator) as low as < 1 ps can be obtained with the moving inverter method in FPGA.

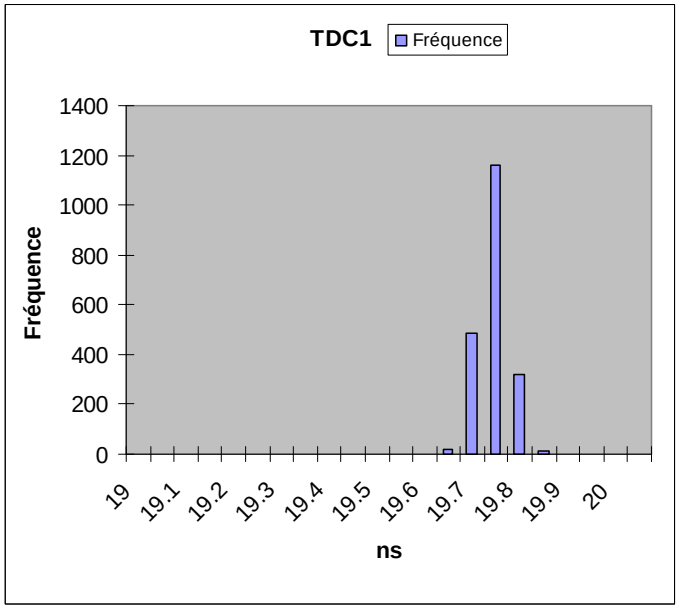
Calibration results
Phase detector measurement



The consistency is checked by comparing the difference of slow and fast oscillator Frequencies (blue curve) with the measure obtained with the phase detector (red curve)

Full range (25 ns) measurement with TDC selection slow=148 fast=22 (Res=50 ps)



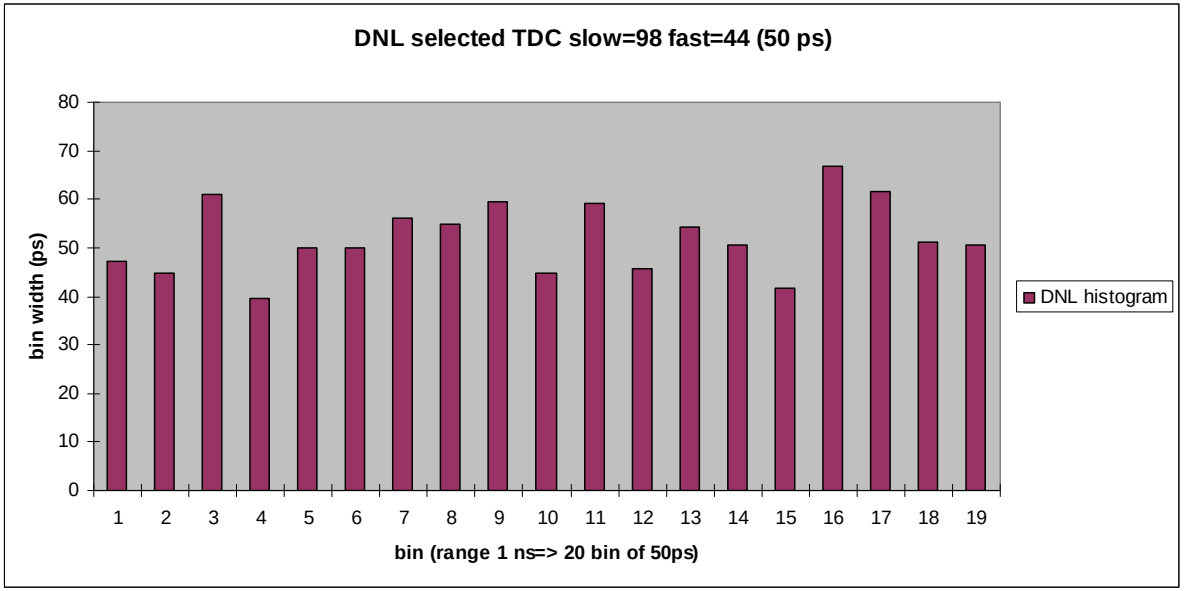


σ =34 ps

TDC measurement between a 40 MHz (25 ns) clock ref (FPGA PLL) and a synchronous trigger delayed by step of 100 ps over a range > 25 ns

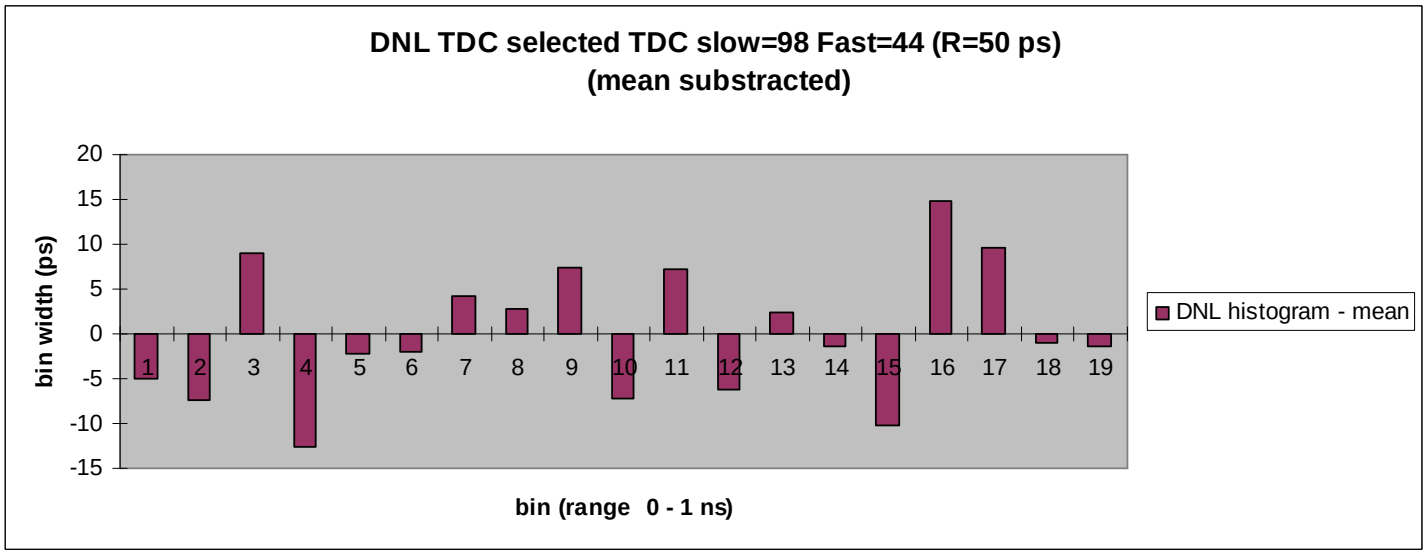
DNL histogram (50 ps TDC selected)

This DNL histogram is obtained by Sweeping the delay of inputs signals from 0 to 1 ns by step of 10 ps (100 measures by step)



Mean of bin width is :
52ps (expected 50ps)

Differential non linearity :
+/- 15 ps max
< 0.3 LSB



Limitations of this architecture
Cumulated Jitter

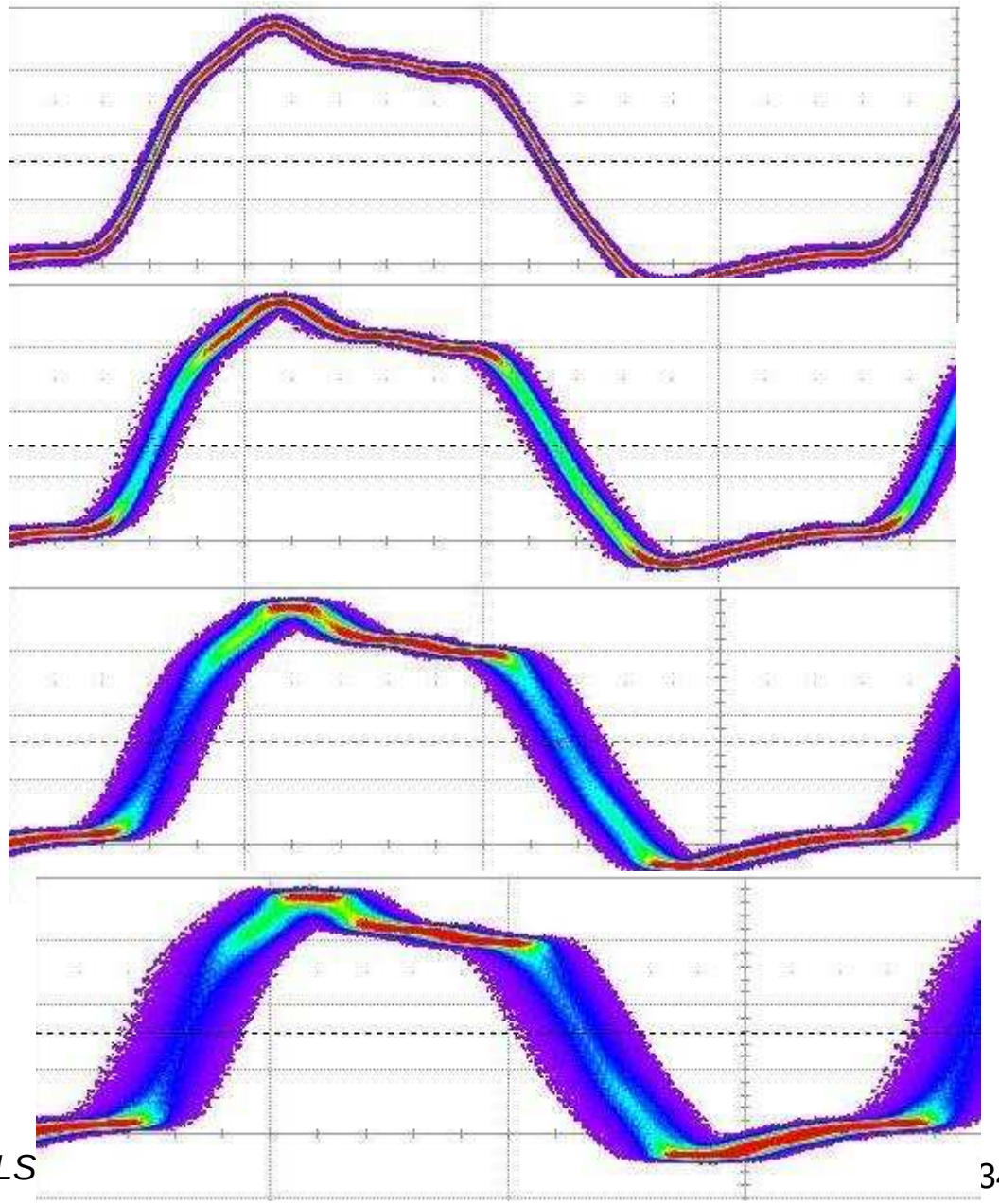
Visualization of the cumulated jitter for a R.O. of T=3.3 ns (7x inverters)
1 ns / div period jitter= 9.4 ps cycTocyc=12.5 ps (40GSps)

0ns

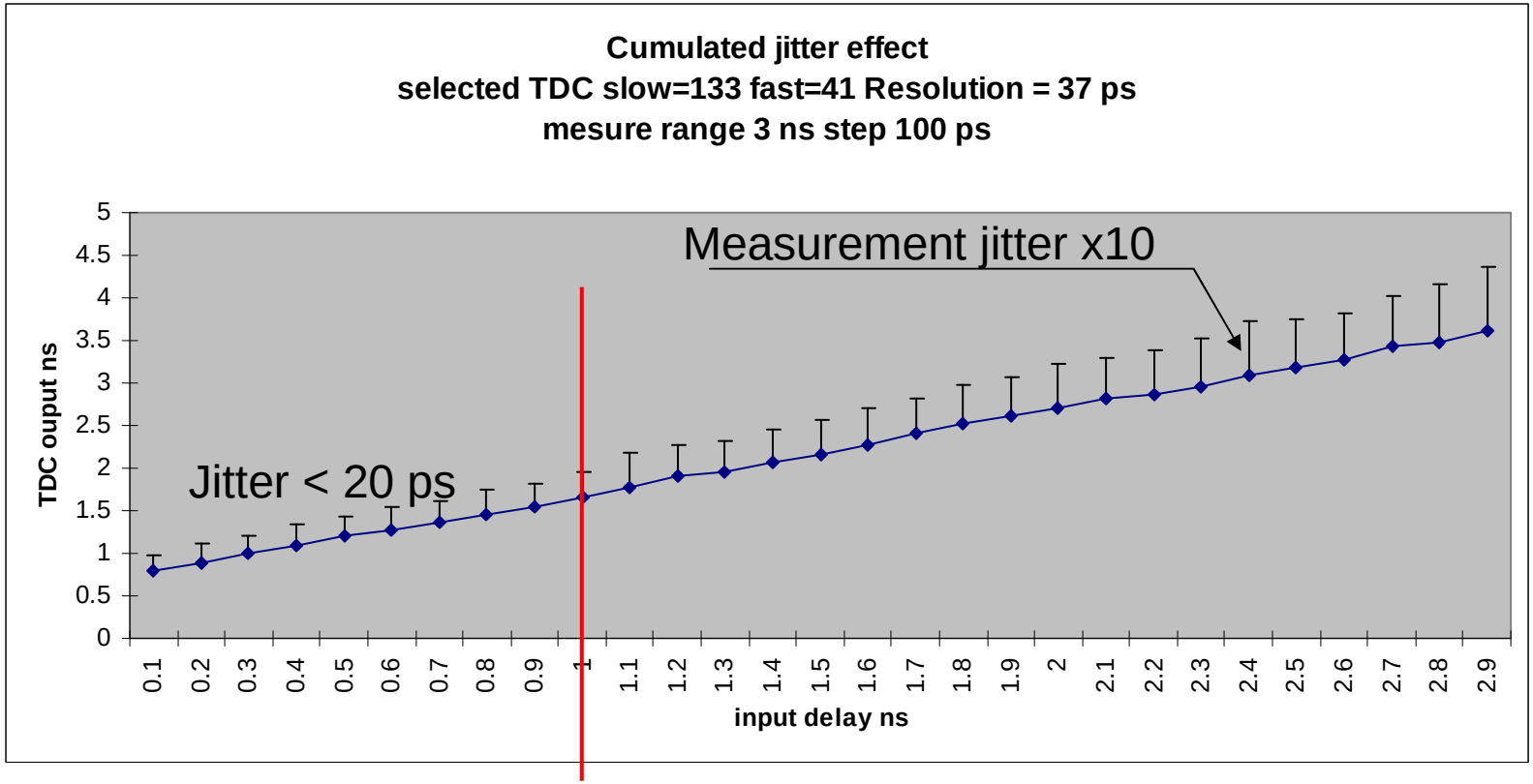
100 ns

200 ns

300 ns

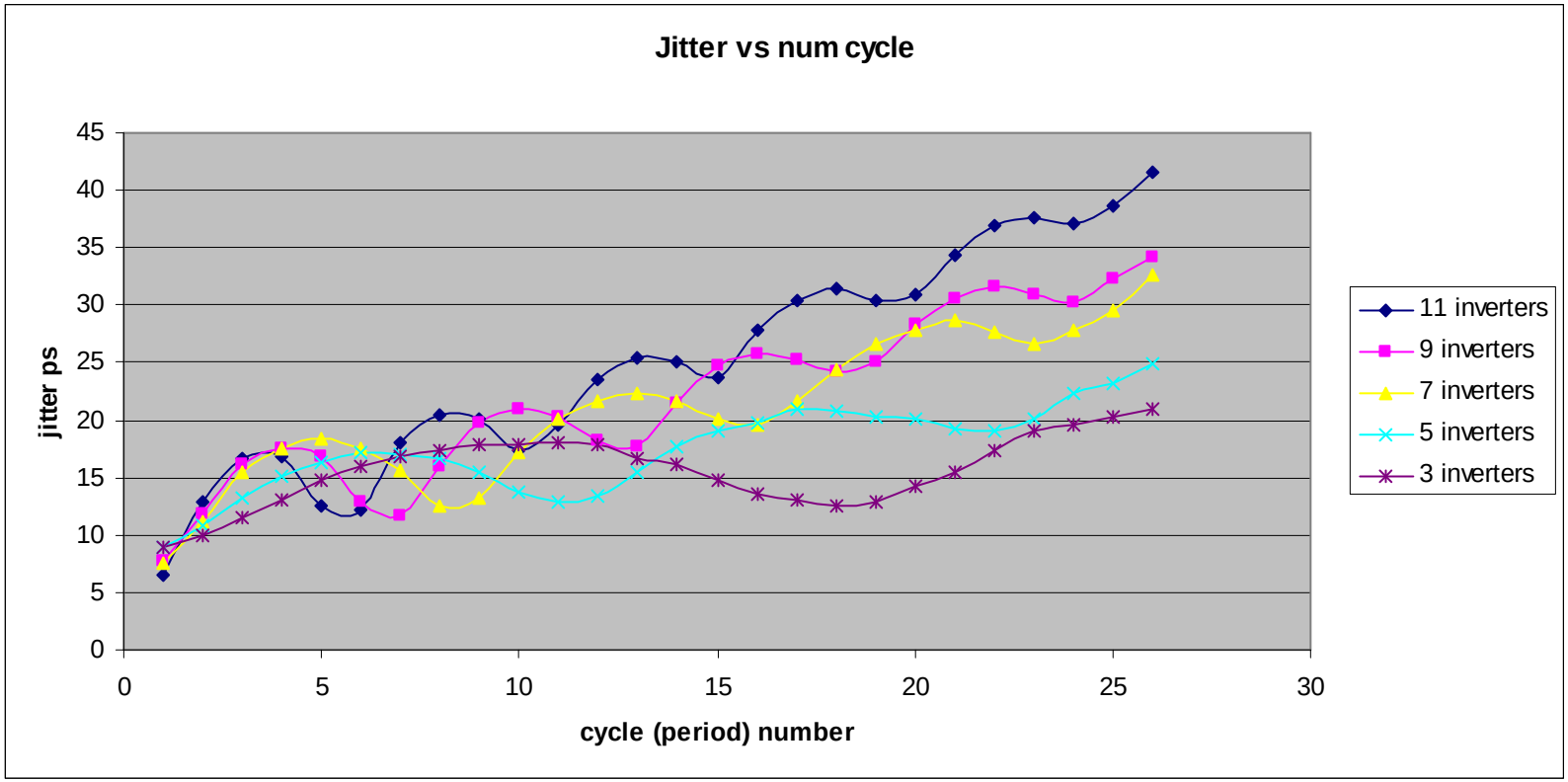


Problems and limitations
e.g. cumulated jitter on a 37 ps TDC selection



We observed that the TDC jitter increases with the delay range.
This means that for low resolutions TDC the delay range must be reduced to preserve a low jitter regarding the resolution.

Cumulated jitter vs. number of cycle periods for different ring oscillators

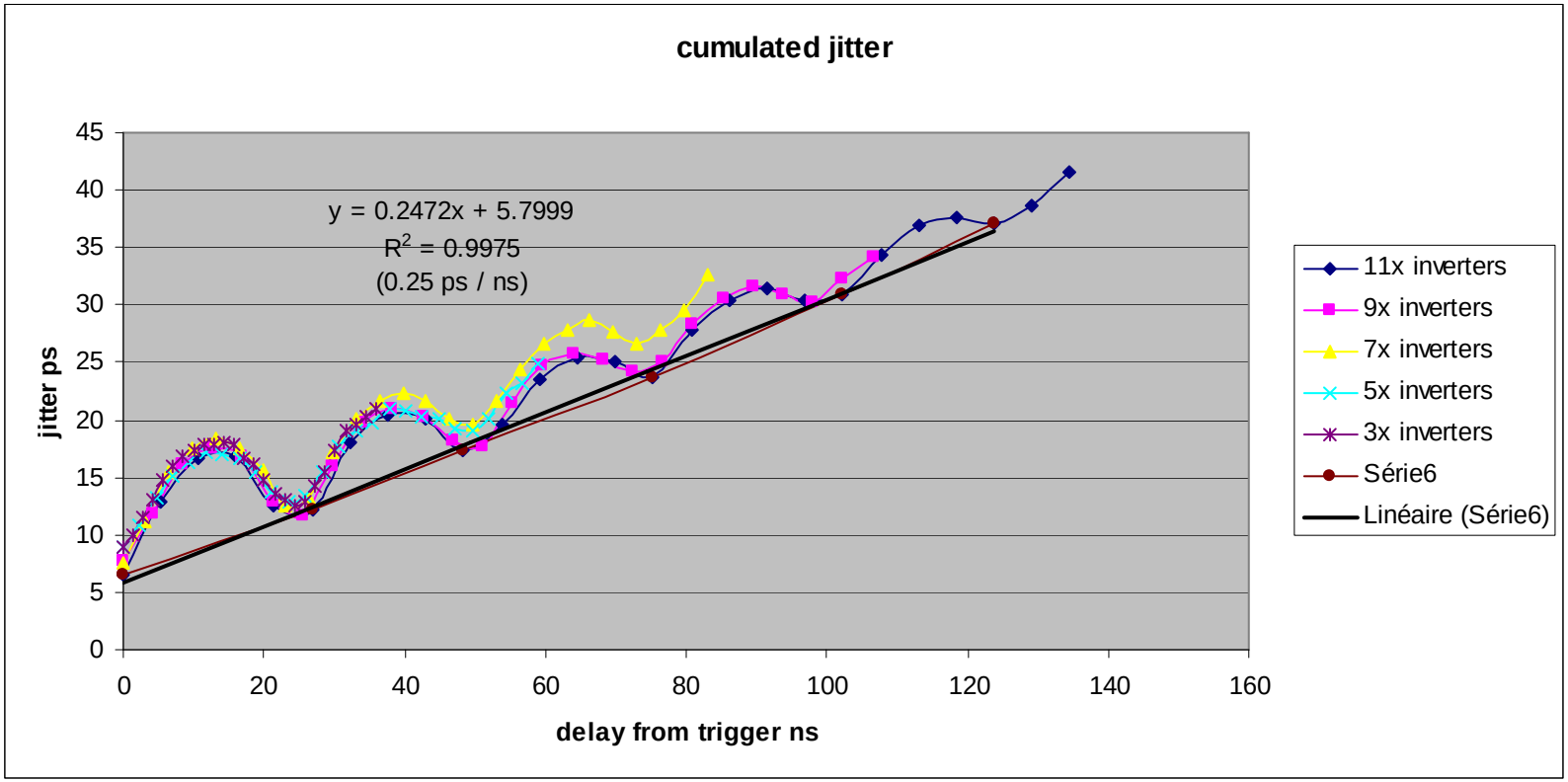


Periods

5,38 ns
4.25 ns
3.32 ns
2.36 ns
1.43 ns

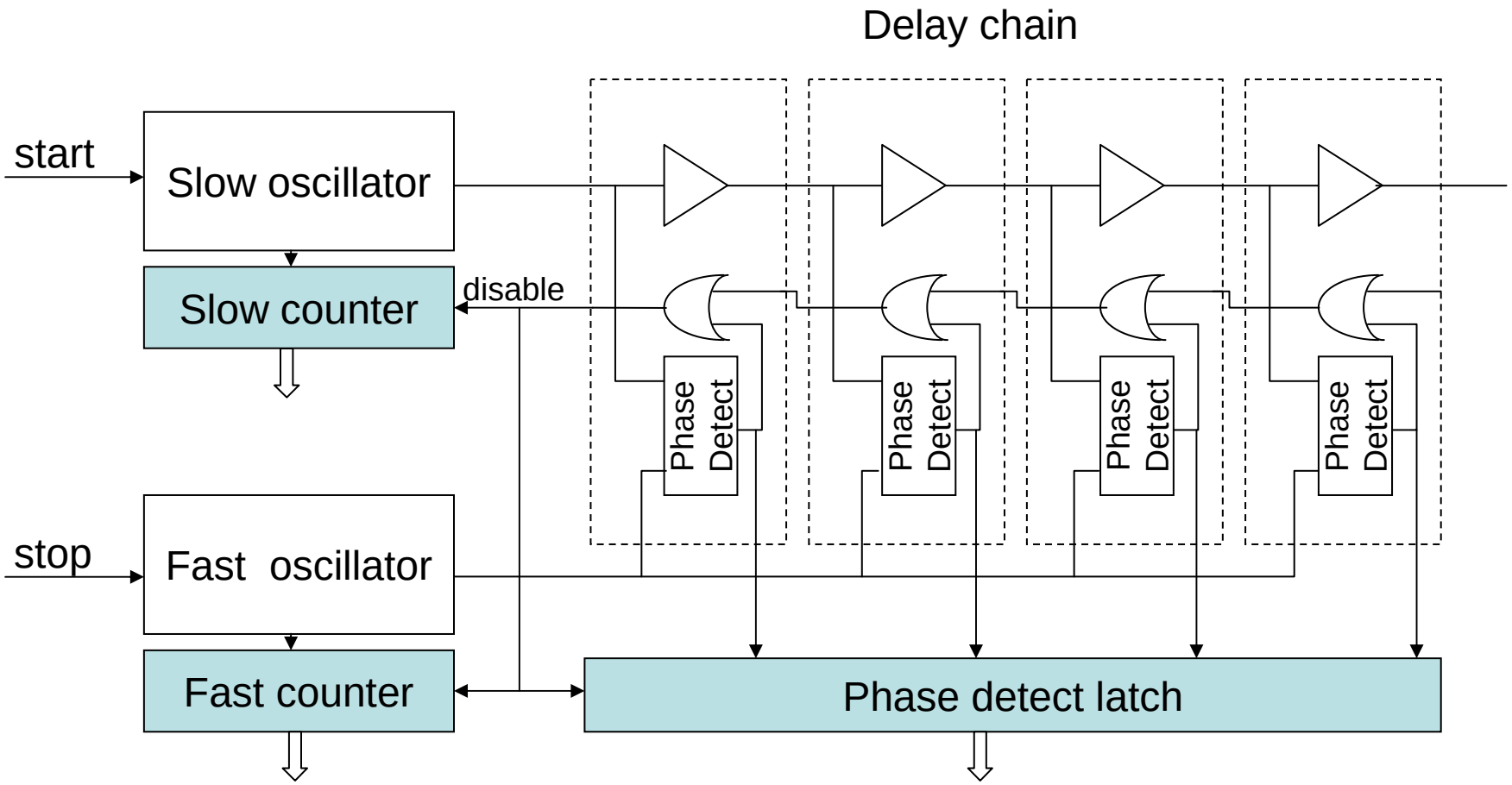
Conclusion : for a same number of cycles the fastest oscillator has the smallest jitter.

Cumulated jitter vs Time



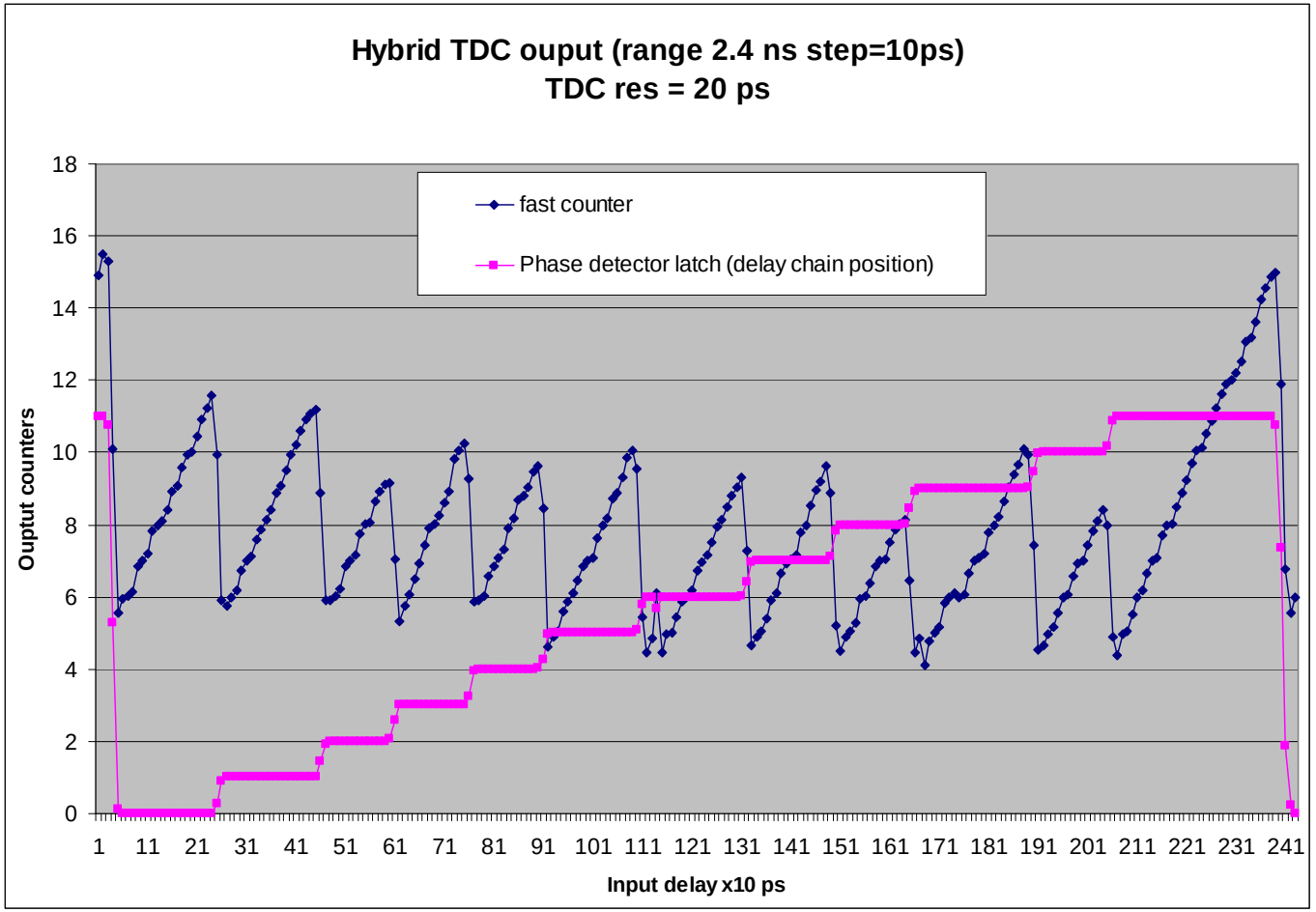
- The cumulmated jitter does not depend on the frequency oscillator
- Cumulated jitter increases linearly with time.
- A periodic component could exists depending on setup configuration (i.e. in our case a 40 MHz oscillator entering in the FPGA)

HYBRID Architecture: Vernier + Delay Chain



Goal : divide the measurement range into small parts
in order to reduce the impact of jitter

Hybrid II adjustable delay Line
(preliminary results)



The maximum oscillation cycles is about 12 (excepted for the widest bin)

Conclusions

- We presented a new technique of tuneable oscillators to be used in vernier TDC or other applications with high sensitivity in frequency adjustment.
- We found that the cumulated jitter prevent us to exploit all the potential of this oscillators
- We started to test an hybrid architecture with delay chain and multiple phase detector stage to reduce the jitter effect
- An ASIC implemention has been sent in foundry, we hope that the jitter effect will lower with a dedicated routing layout.

Références

- FPGA-Based High Area Efficient Time-To-Digital IP Design
- Performance and area tradeoffs in space-qualified FPGA-based time-of-flight systems
- An FPGA wave union TDC for time-of-flight applications
- FPGA-Based Self-Calibrating Time-to-Digital Converter for Time-of-Flight Experiments
- Upgrading of Integration of Time to Digit Converter on a Single FPGA
- Area efficient time to digital converter (TDC) architecture with double ring-oscillator technique on FPGA ...
- FPGA based self calibrating 40 picosecond resolution, wide range Time to Digital Converter
- The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay
- Implementation of High-Resolution Time-to-Digital Converters on two different FPGA devices
- Several Key Issues On implementing delay line Based TDCs using FPGAs