



DLS using on-chip CPU

“Harder, better, faster, stronger”?

Application examples

Gaël Faggion
EUFANET Workshop 2009
Toulouse



Two DLS cases using on-chip CPU

Case study 1:

A mobile music application processor

Case study 2:

An audio DSP

Conclusion, Q&A

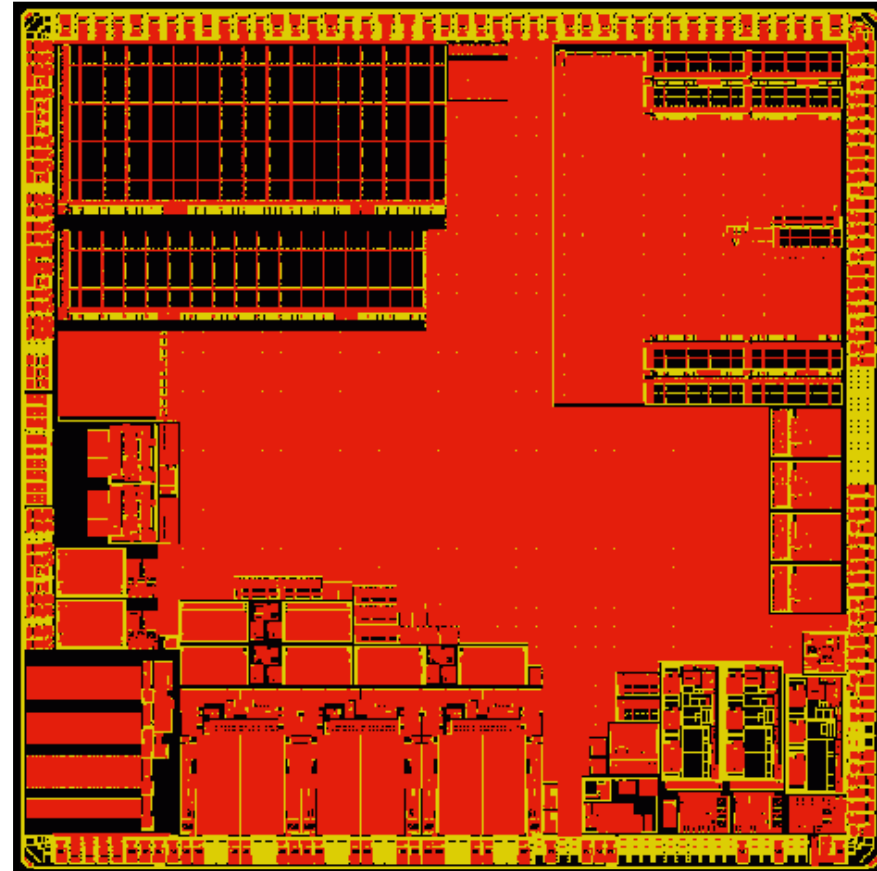


Case study 1:

**A mobile music application
processor**

Problem description

- ▶ New IC, 90 nm process
- ▶ Contains ARM926 CPU
- ▶ **Speed issue**: Too slow
- ▶ 240 MHz, spec = **300 MHz**

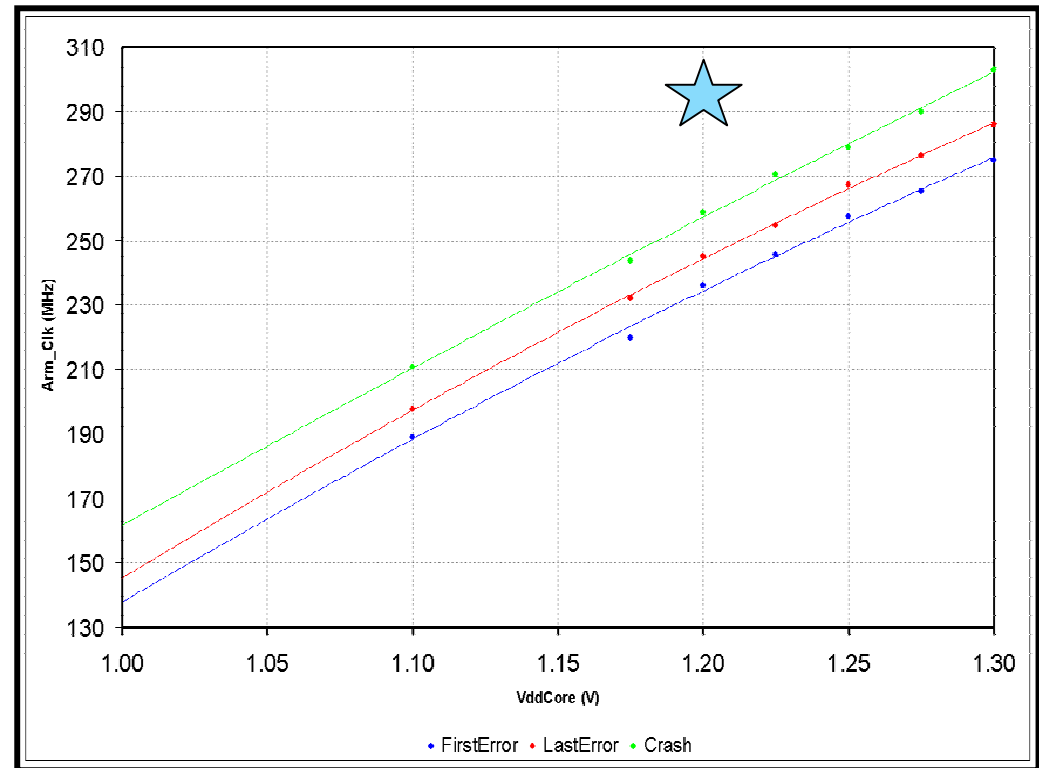


▶ Problem: Where is the critical path?

Use LADA to find it, then TRE to analyze timings?

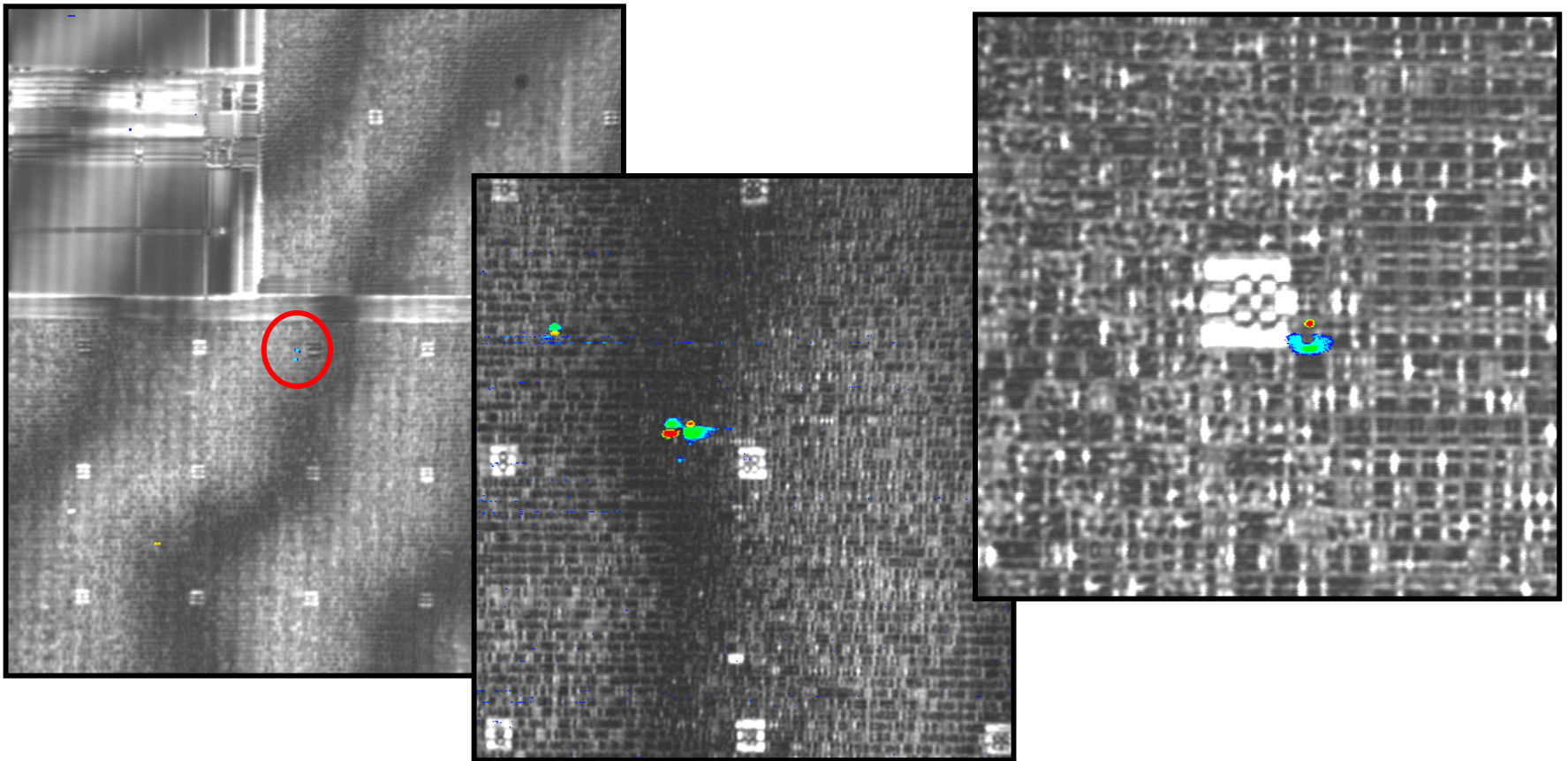
FA setup

- ▶ Must create a suitable pattern for FA!
- ▶ Test program made with application engineers
- ▶ Test program:
 - Write memory byte
 - Read memory byte
 - Compare to expected
 - Set output of chip pass/fail
 - Repeat

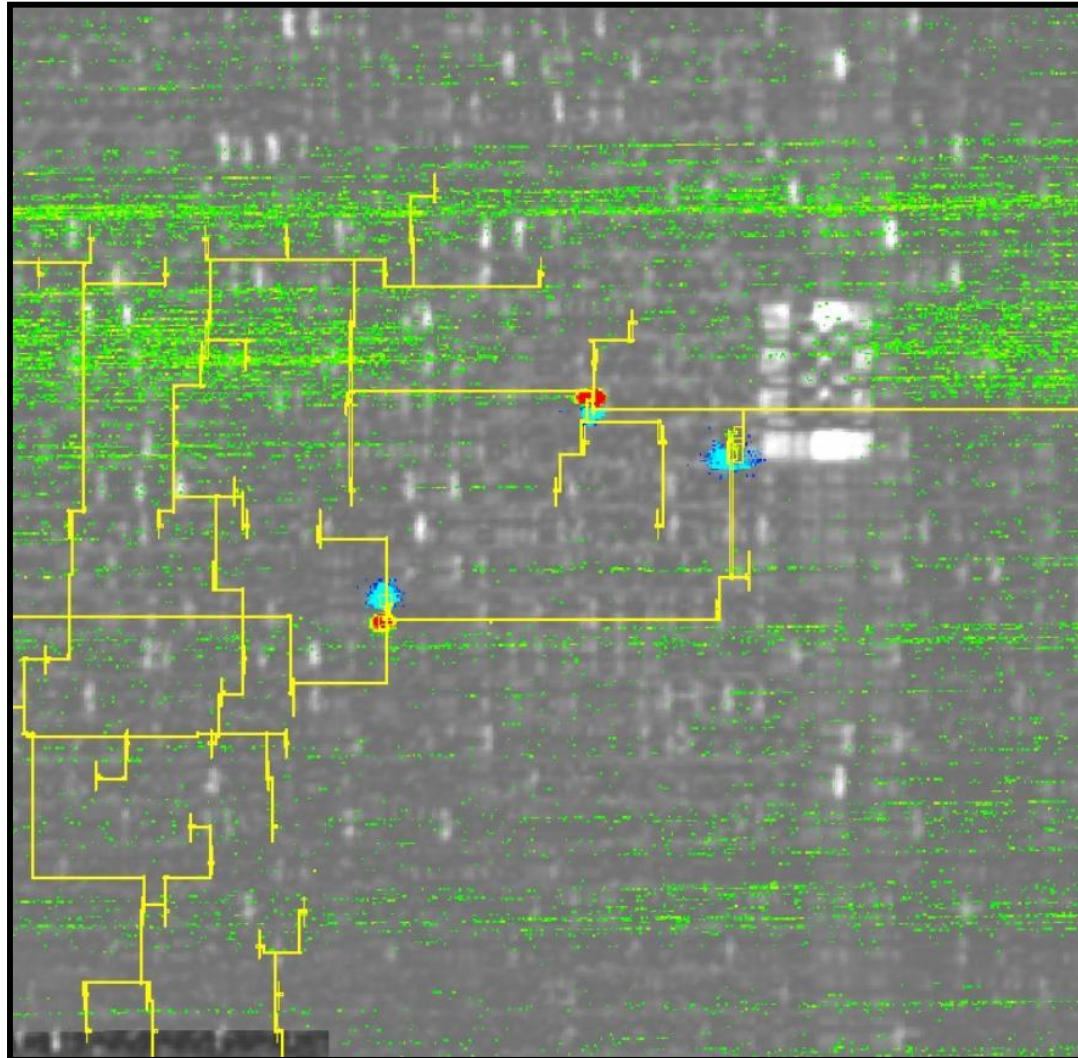


FA investigation: Laser Scanning (1/2)

Setting the device to fail around 50% of the time, we are then able to see sensitive areas where the laser makes the device fails more or less.

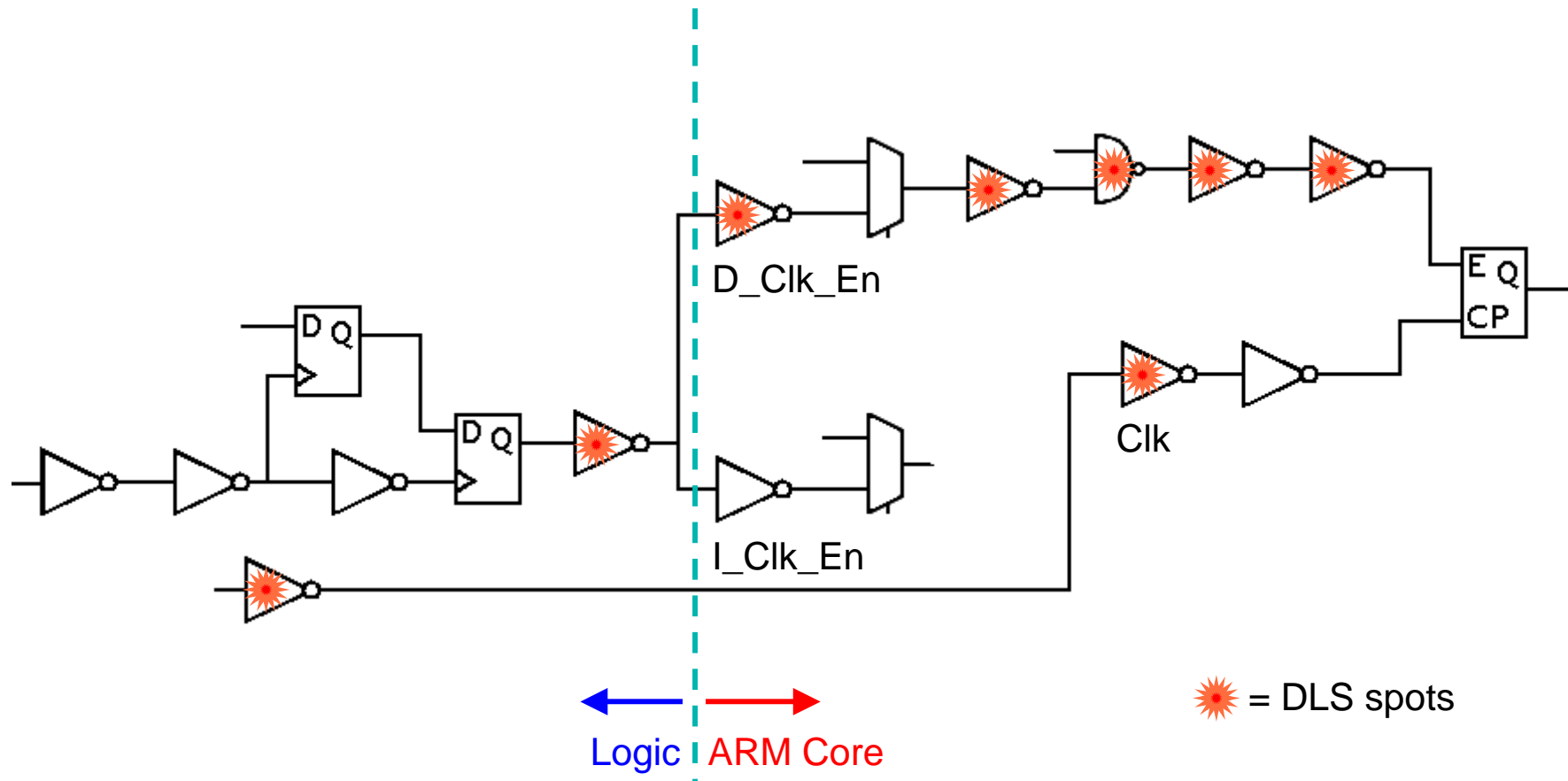


FA investigation: Laser Scanning (2/2)



Failing path identification

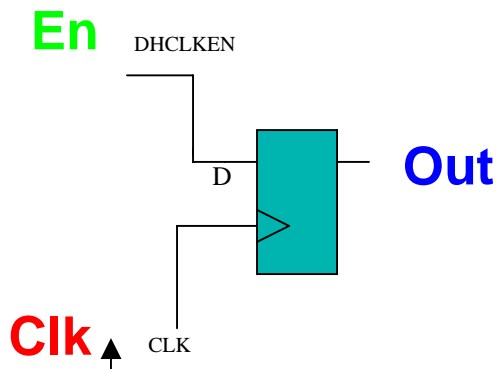
These spots were identified as part two different paths, the Data Clock Enable and the Clock path.



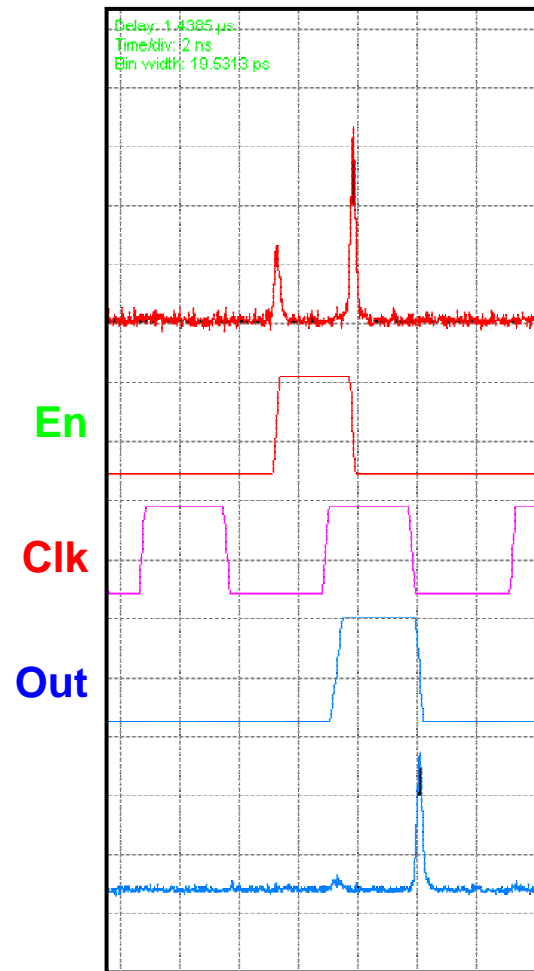
Time Resolved Emission (TRE)

Using the Emiscope we were able to confirm a design timing issue.

Final latch:



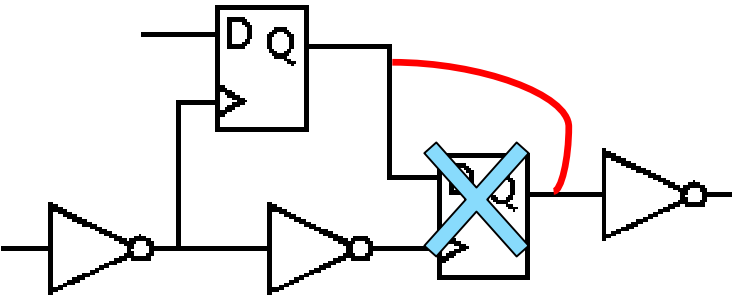
Pass @ 1.30v **160** MHz



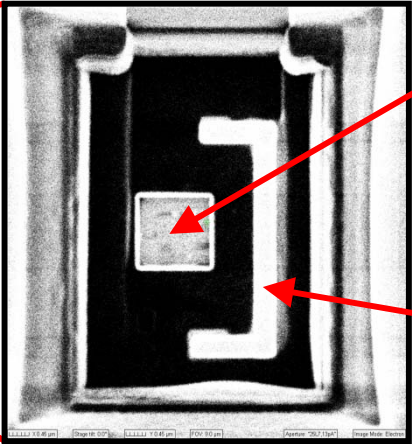
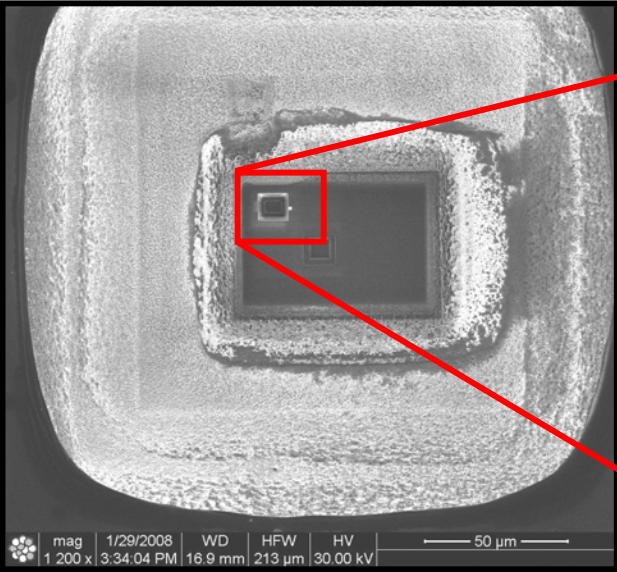
Fail @ 1.30v **290** MHz



Focus Ion Beam (FIB) modification



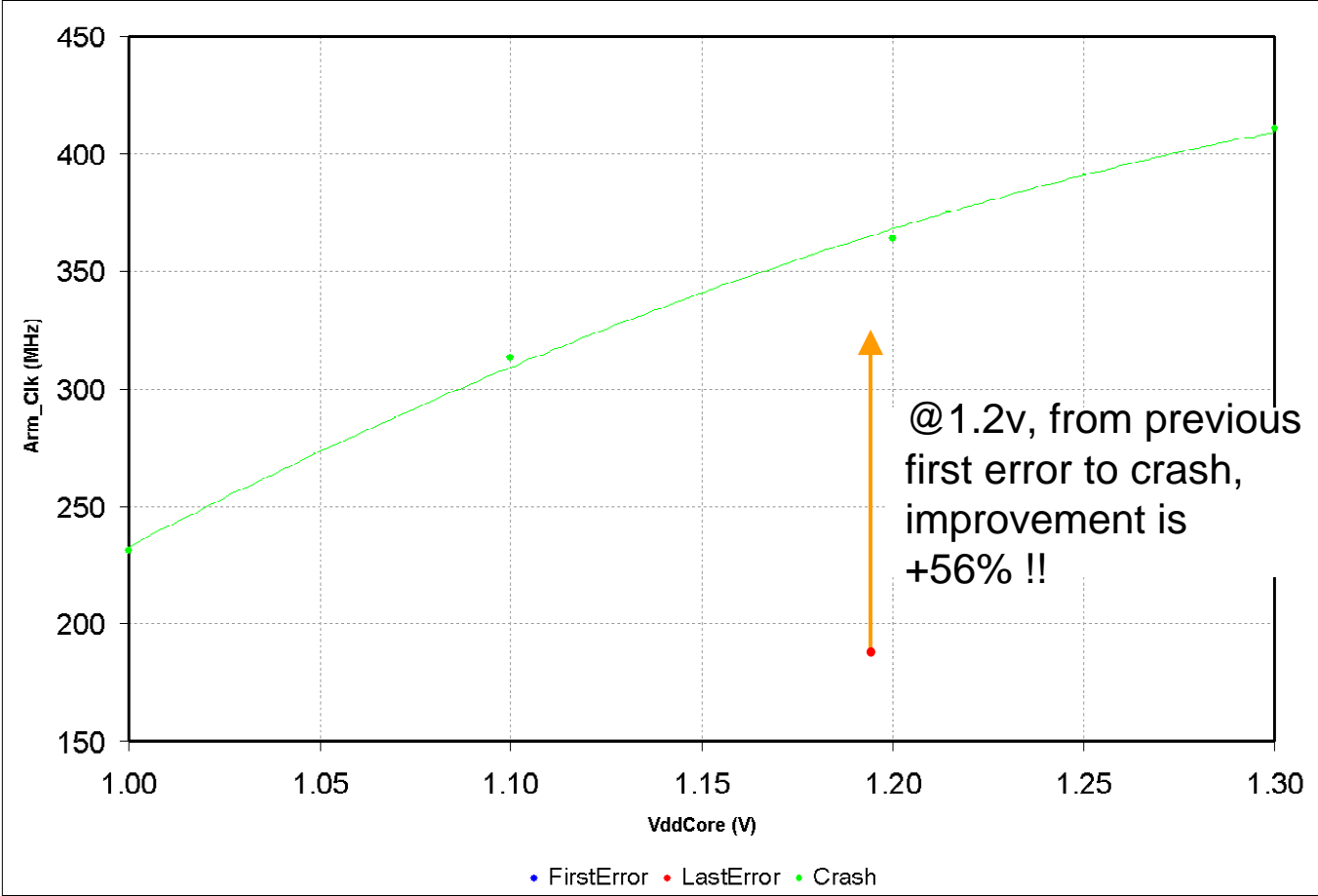
Proposed modification to circumvent setup violation



Flip-Flop output destroyed

Input directly connected to output

FA Solution: Electrical validation



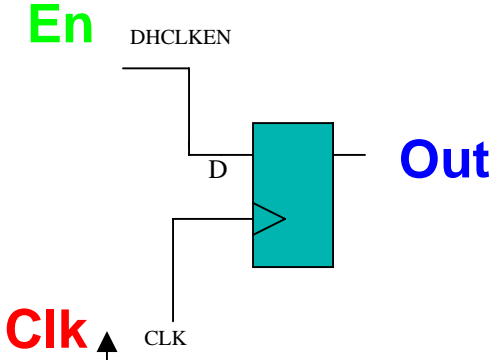
Success! Now, speed specification is reached



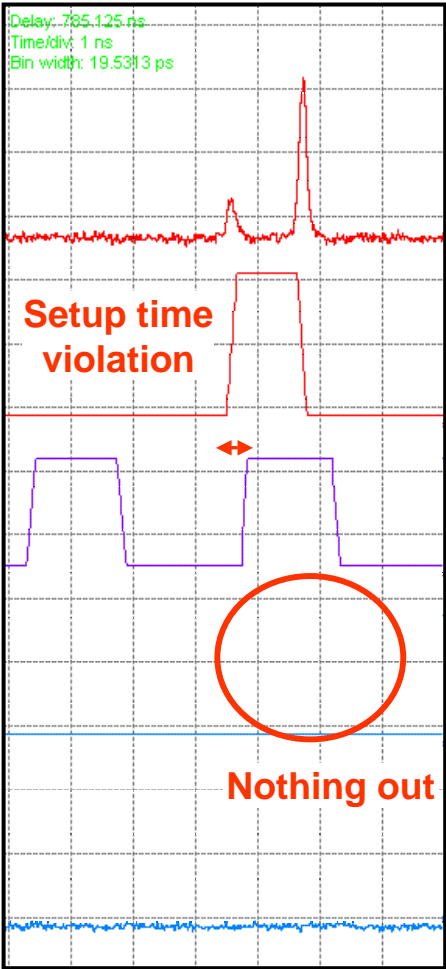
FA Solution: TRE validation

- ▶ Extra verification with Emiscope
- ▶ Shows that fix works as expected

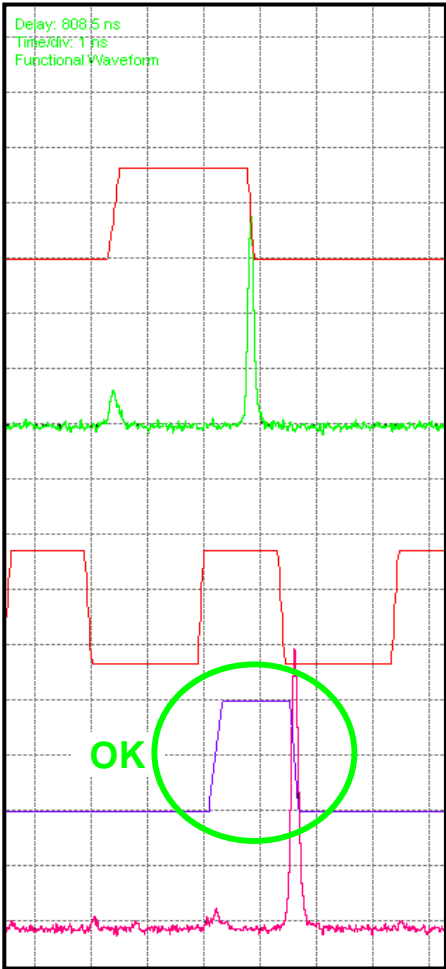
Final latch:



Original @ 1.30v 290 MHz



Fibed @ 1.30v 290 MHz

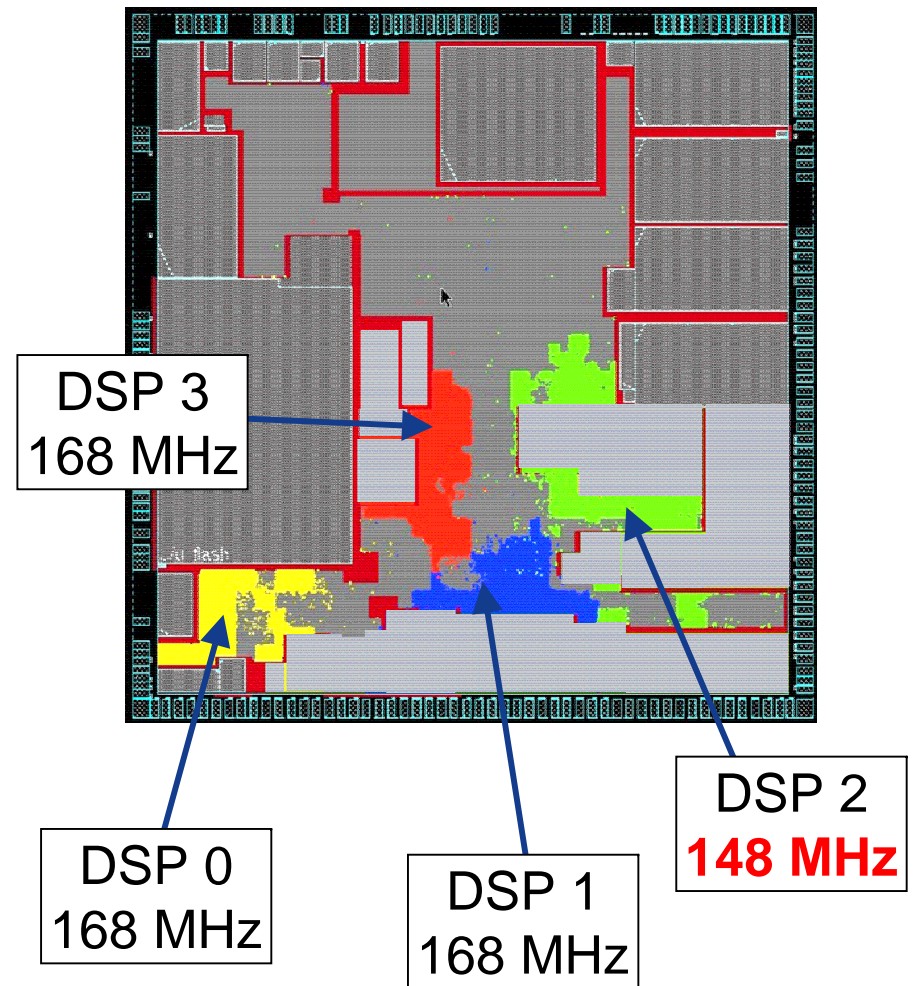


Case study 2 :

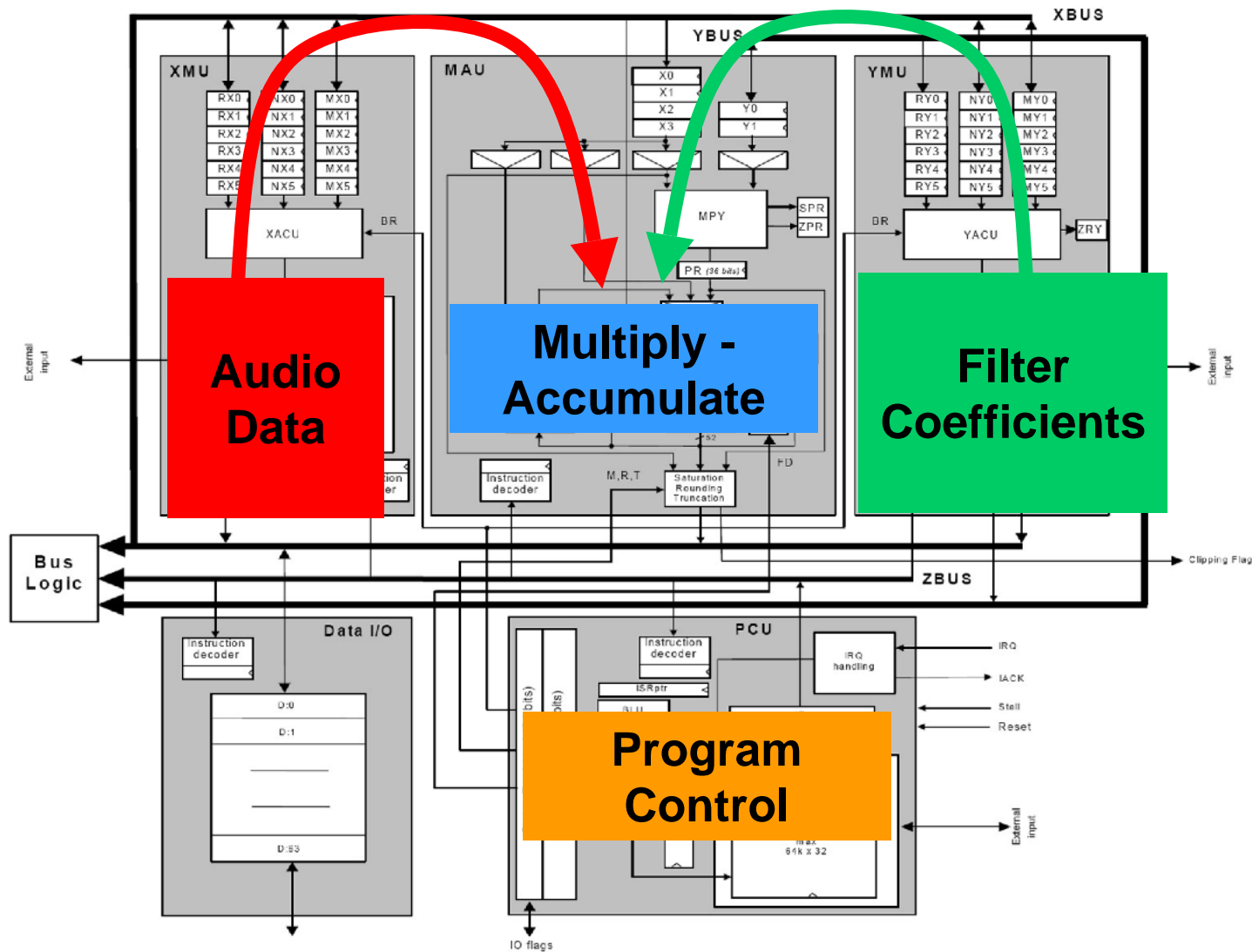
An audio DSP

Problem description

- ▶ Audio DSP using 0.15 μ CMOS technology
- ▶ Contains: 4x DSP Cores
- ▶ DSPs speed must be 125 MHz worst case (1.65V, 125°C, Slow)
=> Design pushes speed capability of process
- ▶ 'First silicon' evaluation: **Chip fails speed**
- ▶ 168 MHz typ. expected, 148 MHz on silicon



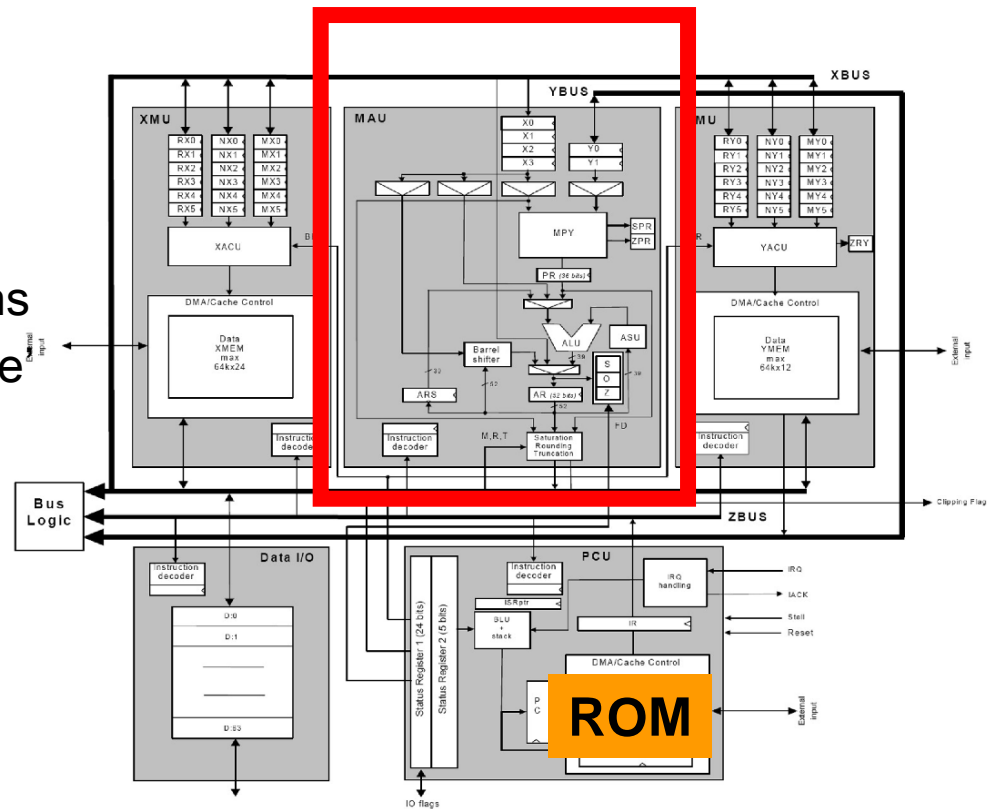
The DSP 2 core diagram



Homing in : Software Debug investigation

- ▶ DSP Software engineer investigated...
- ▶ Complication :
 - DSP 2 runs only from ROM
 - ➔ Can not choose instructions
 - Can not just 'step through code'
 - It **must be run at full speed**
- ▶ Can choose data processed.
 - e.g. add 0+0 ➔ no activity

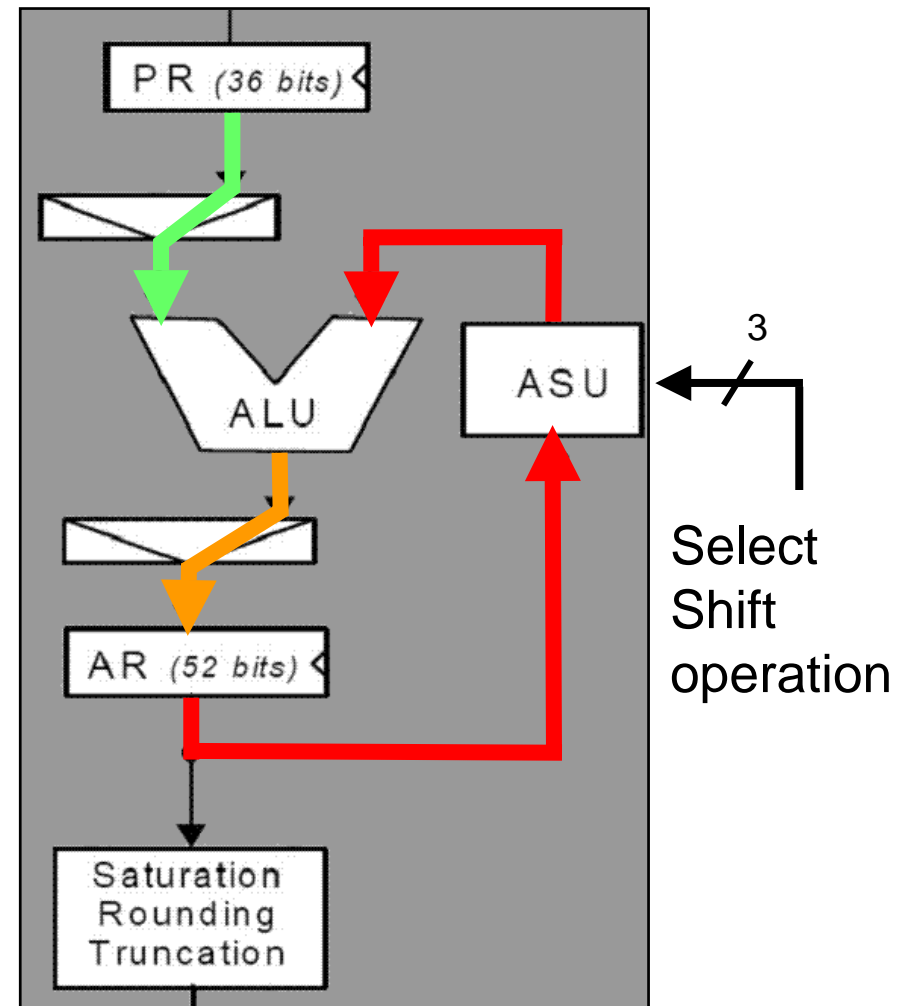
Investigation Points to MAU



➔ Identified failing part of code

Homing in: Software Debug investigation

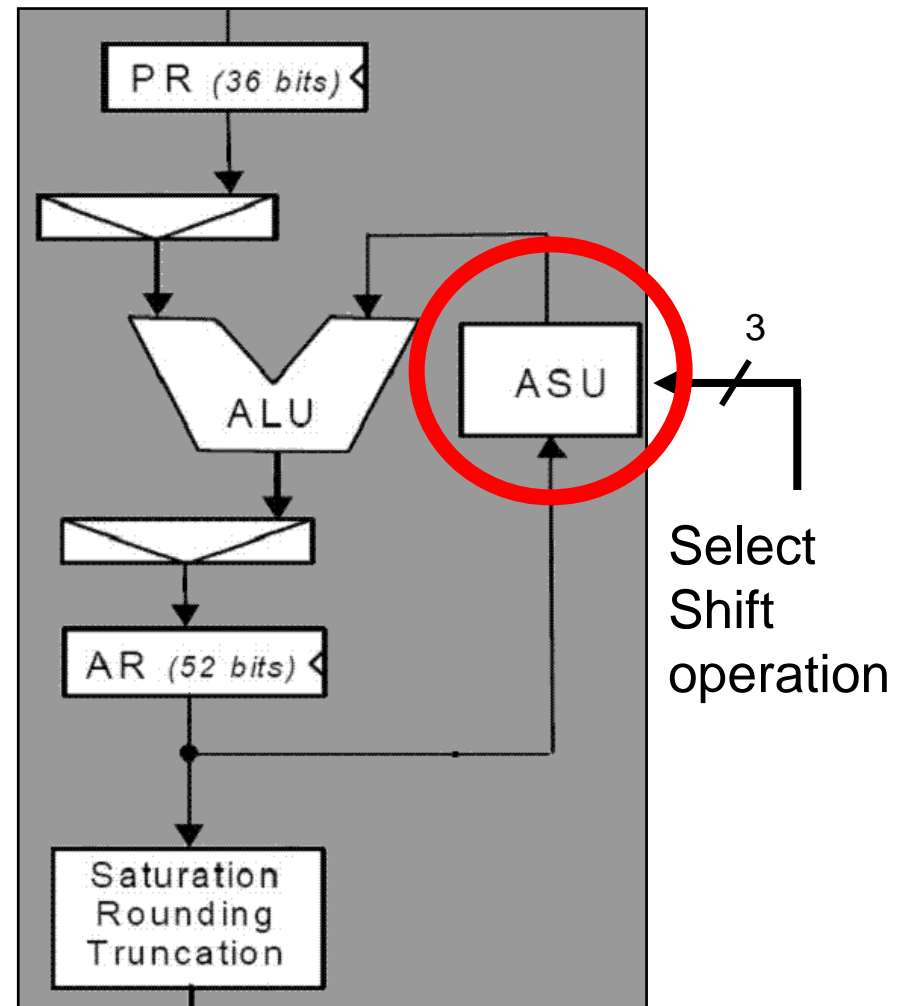
- ▶ DSP Software engineer investigated...
- ▶ Localized to 57 instructions
- ▶ 1 clock cycle does:
 - Take accumulator, bit shift
 - Add result to product register
 - Store again in accumulator



MAU - Simplified

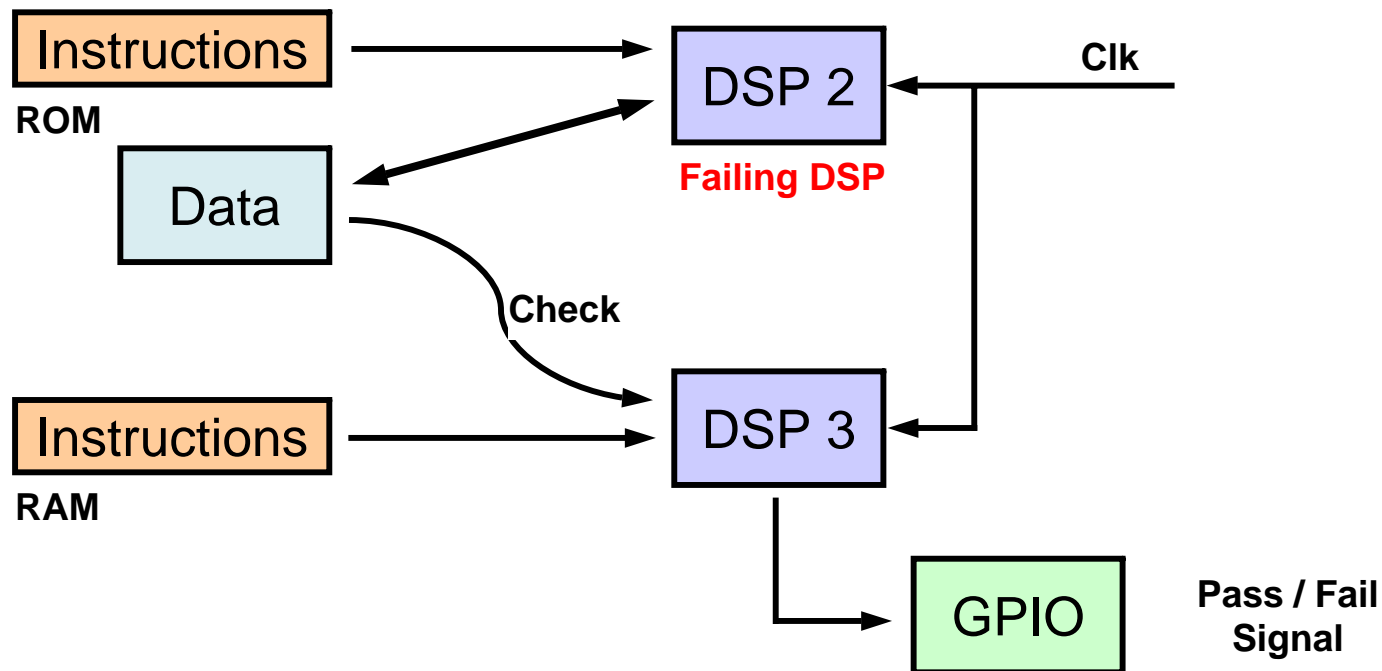
Homing in : Software Debug investigation

- ▶ DSP Software engineer investigated...
- ▶ Localized to 57 instructions
- ▶ 1 clock cycle does :
 - Take accumulator, bit shift
 - Add result to product register
 - Store again in accumulator
- ▶ If shift unit unused, then no faults
→ Suspect shift unit
- ▶ Different data → Different speed
- ▶ Different speed → Multiple faults

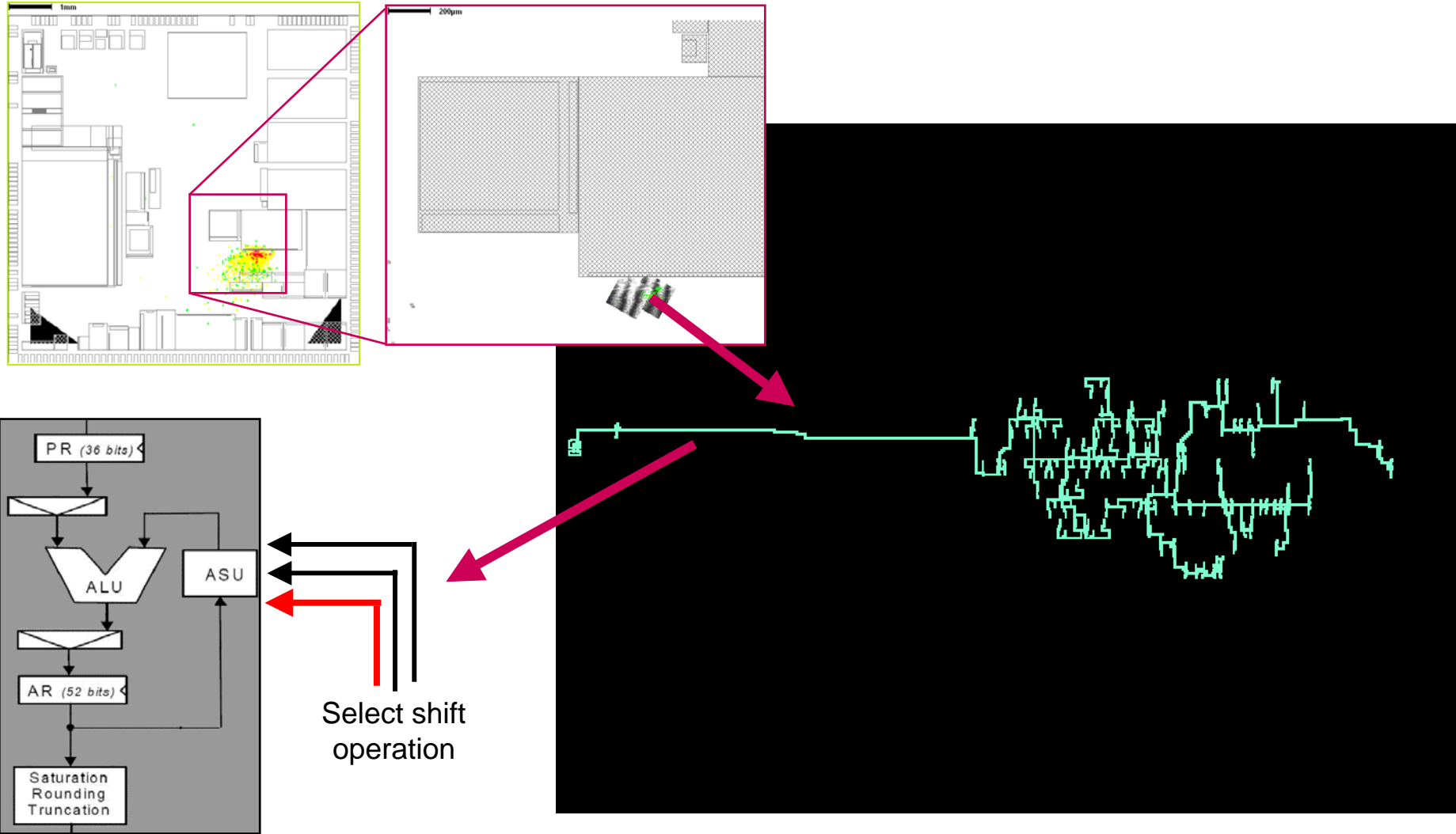


A challenging FA setup

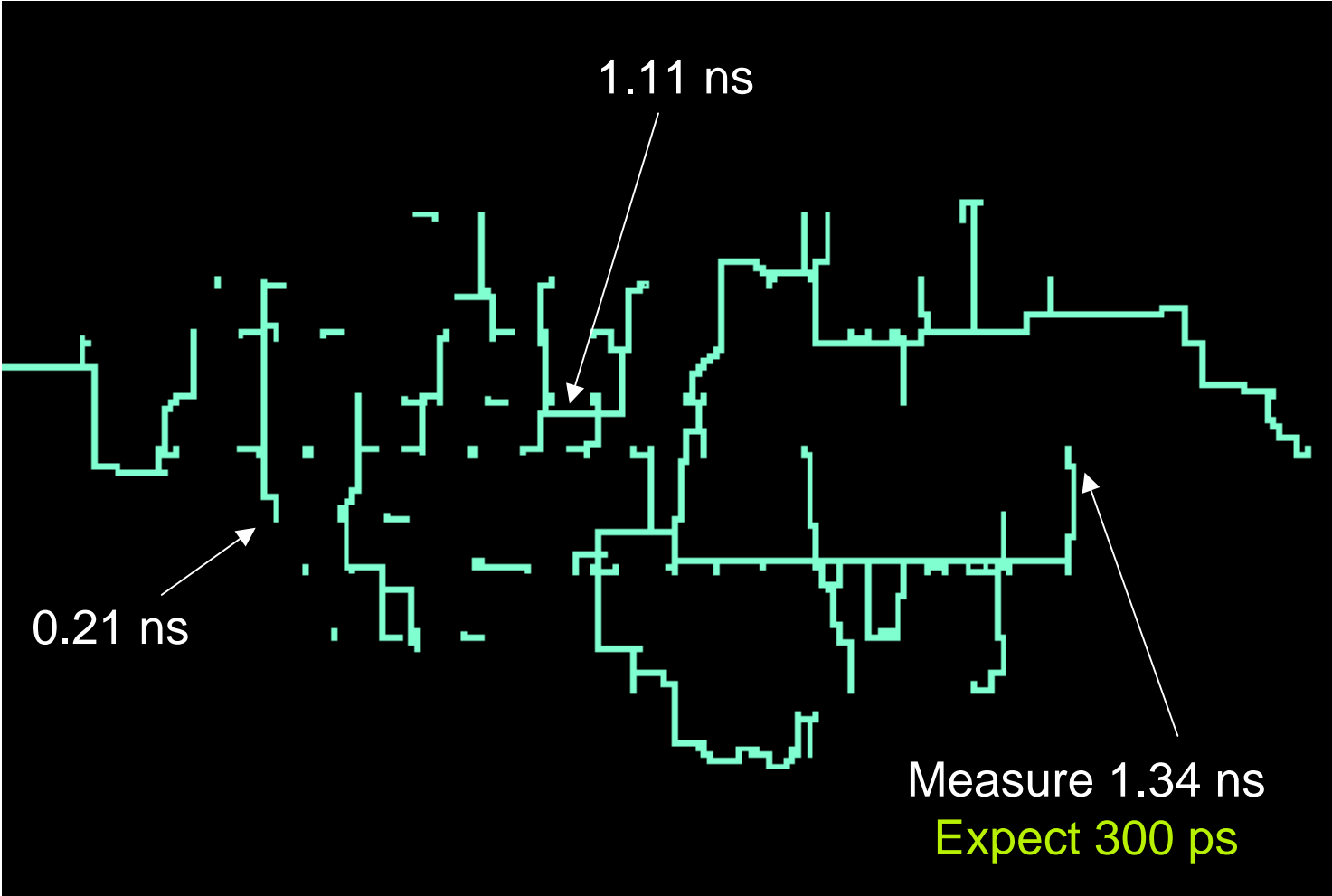
- ▶ At this stage there were still more than 10 000 suspect gates
- ▶ We had to create a suitable program for FA, to characterize the failure and localize the critical path



Detailed localisation by Laser Scanning



Detailed timing measurements → Root Cause



Root cause: Routing creates timing problem

This net is routed through 7 of these cells

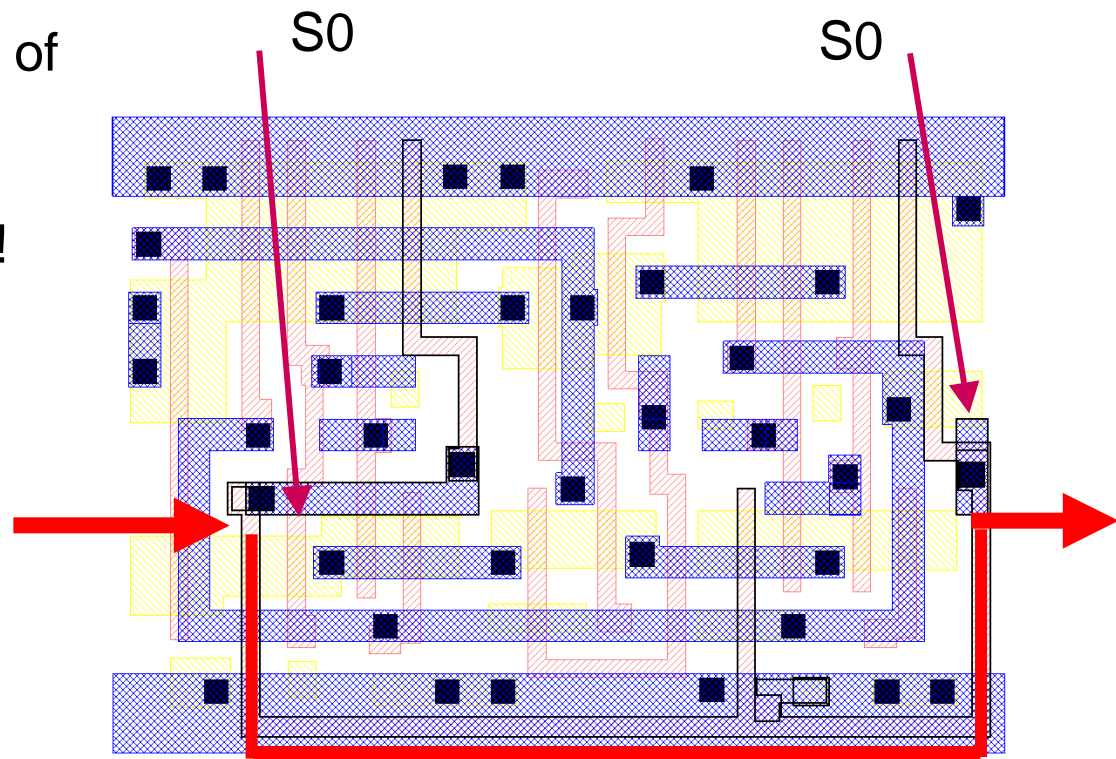
Signal routed through Poly !

Resistance 7x500 Ohm

Total load : 500 fF

RC : 1750 ps

Margin : 300 ps



Conclusion, Q&A

Conclusion

Cons

- ▶ Harder !
 - You must program the device
 - Setup are quite difficult
- ▶ Needs close cooperation
 - FA lacks application knowledge

Pros

- ▶ Better !
 - Can run full functional mode
- ▶ Faster !
 - Much faster than scan test (i.e. for a memory)
 - Decrease test time
 - Decrease acquisition time
 - Can reach very high speeds

Acknowledgements

- Frank Zachariasse
- Michiel Klaarwater
- Frank Zegers
- Stefan Eichenberger
- Maggie Larragy
- Patrick Renaud
- Arno Smit
- Johan van Ekeren
- Jan van Hassel
- Bob Knoppers
- Hildebrand Tigelaar
- Alexander van Luijpen
- Durk Pieter Vogel

... Plus many other contributors!



Q&A ...

Two cases have been shown where we used on-chip CPU, but in both cases there was no alternative !!...

If you have the choice, would you use on-chip CPU ?

