A Comparison of Asynchronous and Synchronous Digital Design

Sohum Datta

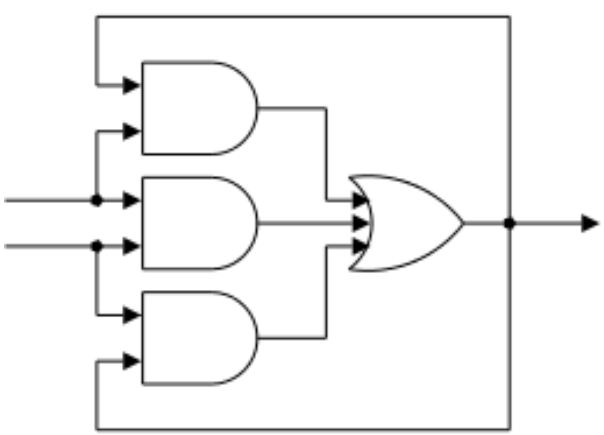
EE241B: Advanced Digital IC

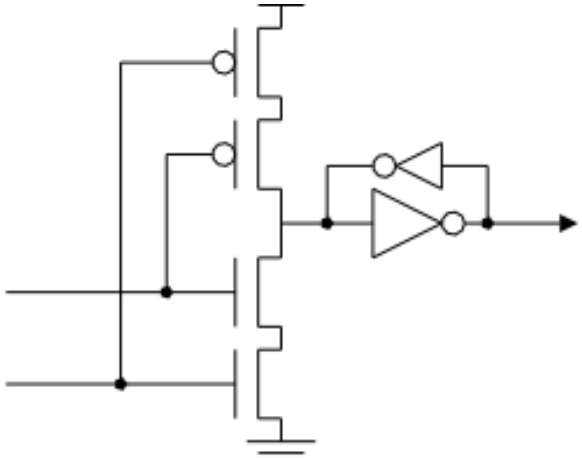
Spring 2017 Project Presentation

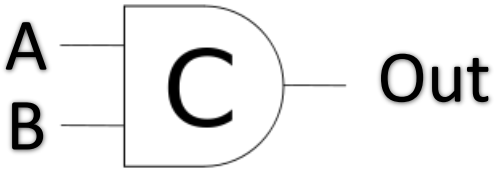
Summary of the two paradigms

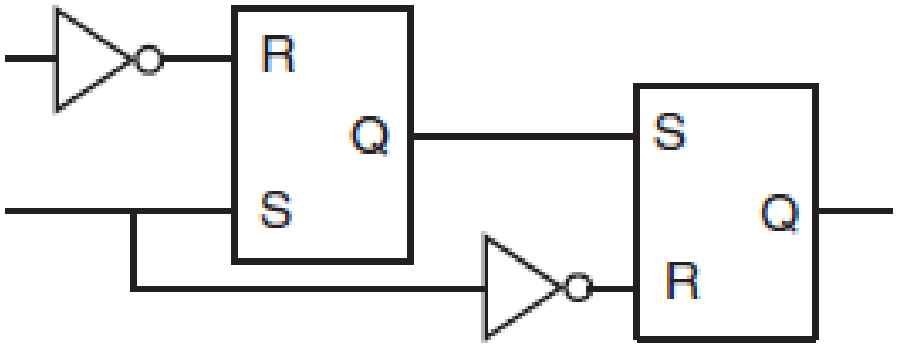
SYNCHRONOUS	ASYNCHRONOUS
+ easier to design and verify	- harder to synthesize, verify and P&R (not many tools ☹)
+ easy robustness guarantees (setup &hold only)	- special handshake (eg. 4-phase) to guard against glitch
- Clock fidelity limit to scalability	+ **automatic** ordering, inherently scalable
- Slack (and power) lost in most paths	+ Timing close to slack-free; power efficient (asymptotically)
- moore's law ending → parallel modules (different clk domains) → interface synchronization	+ seamless integration of asynchronous modules

Basic Gate: Muller “Completion” Element





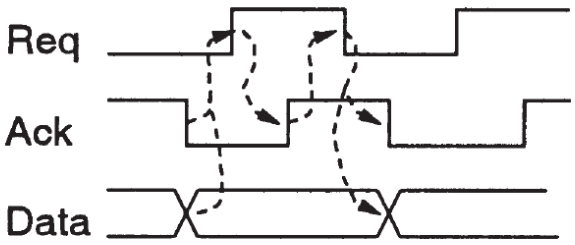




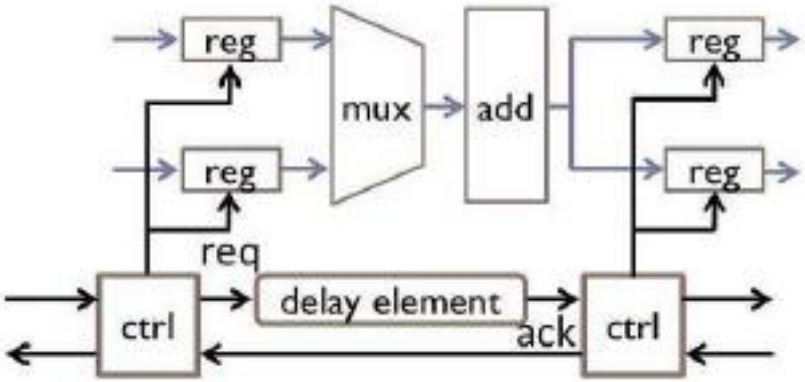
(Synthesized using ***Petrify*** tool from Signal Transition Graph specification)

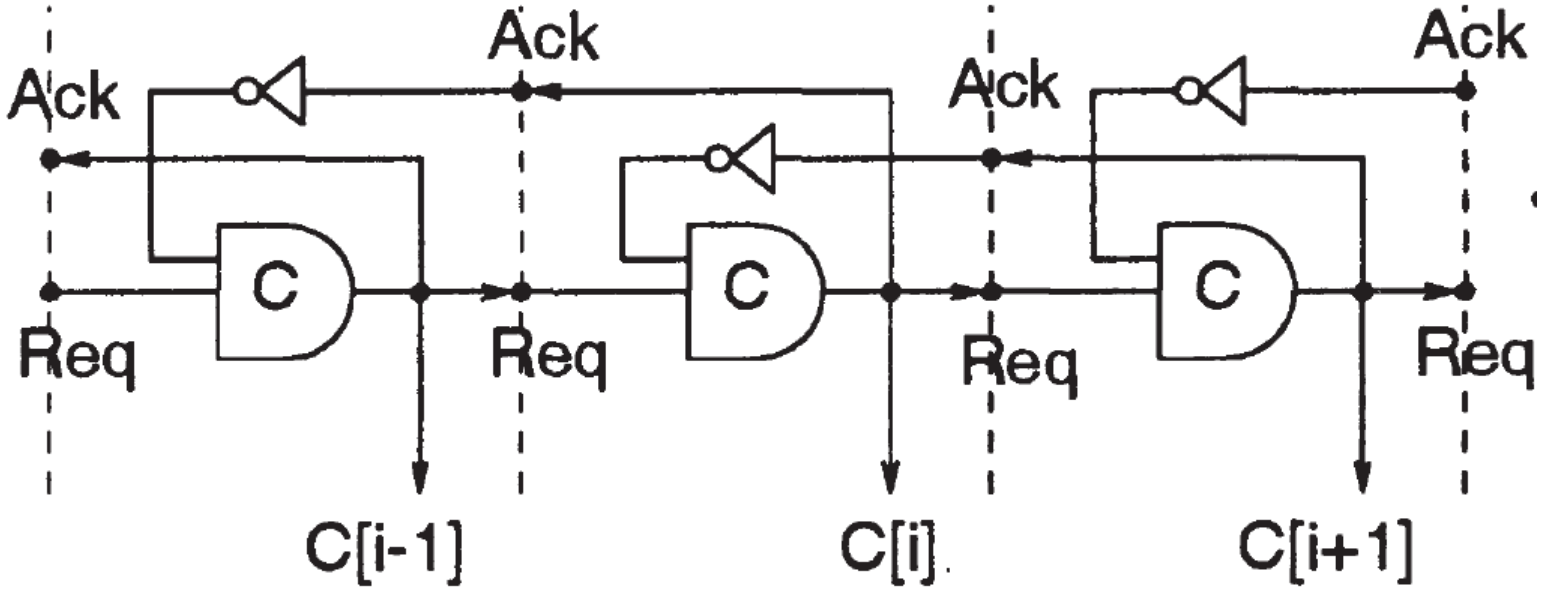
A	B	Out
1	1	1
1	0	No Change
0	1	No Change
0	0	0

Basic building blocks



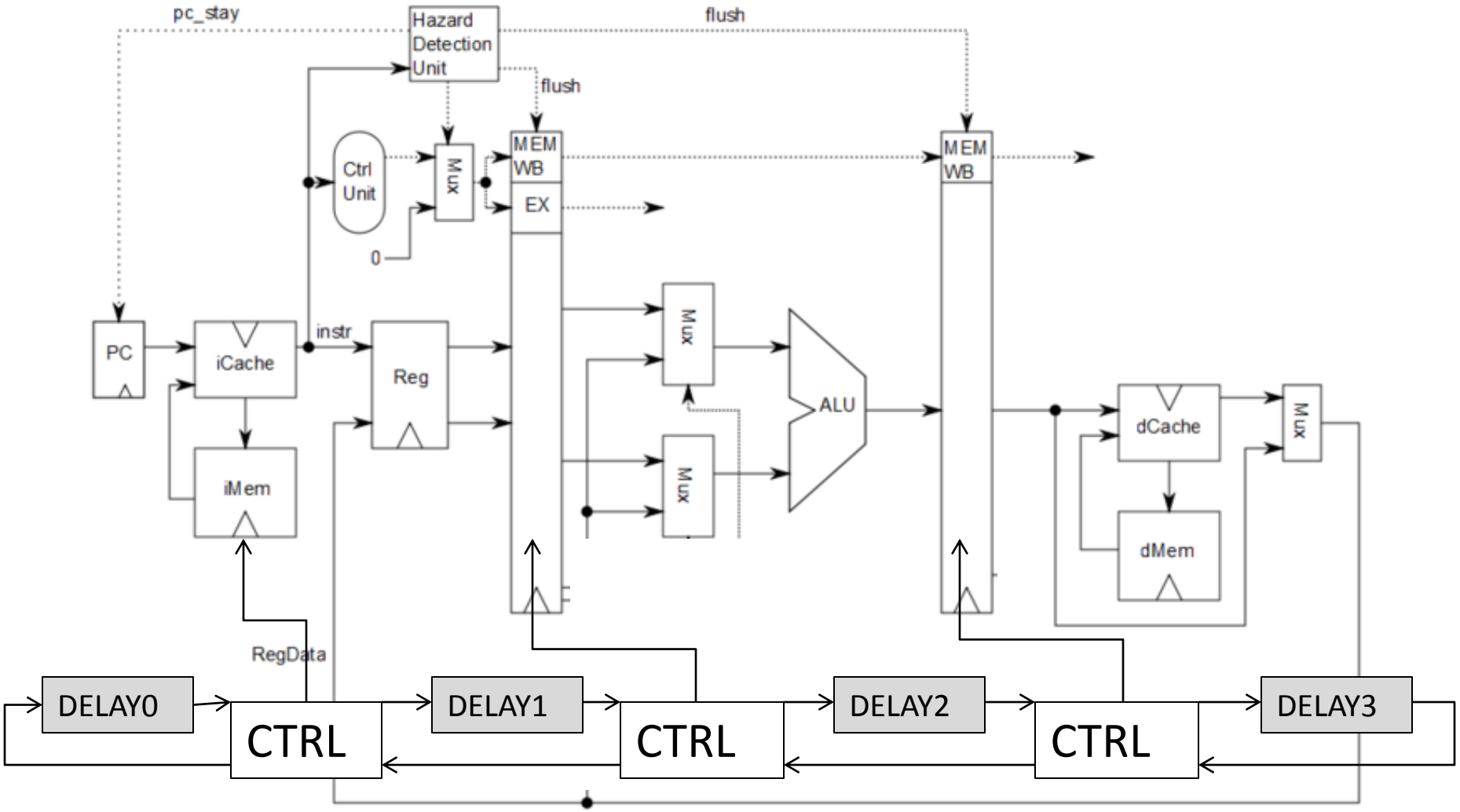
(4-phase bundled handshake)



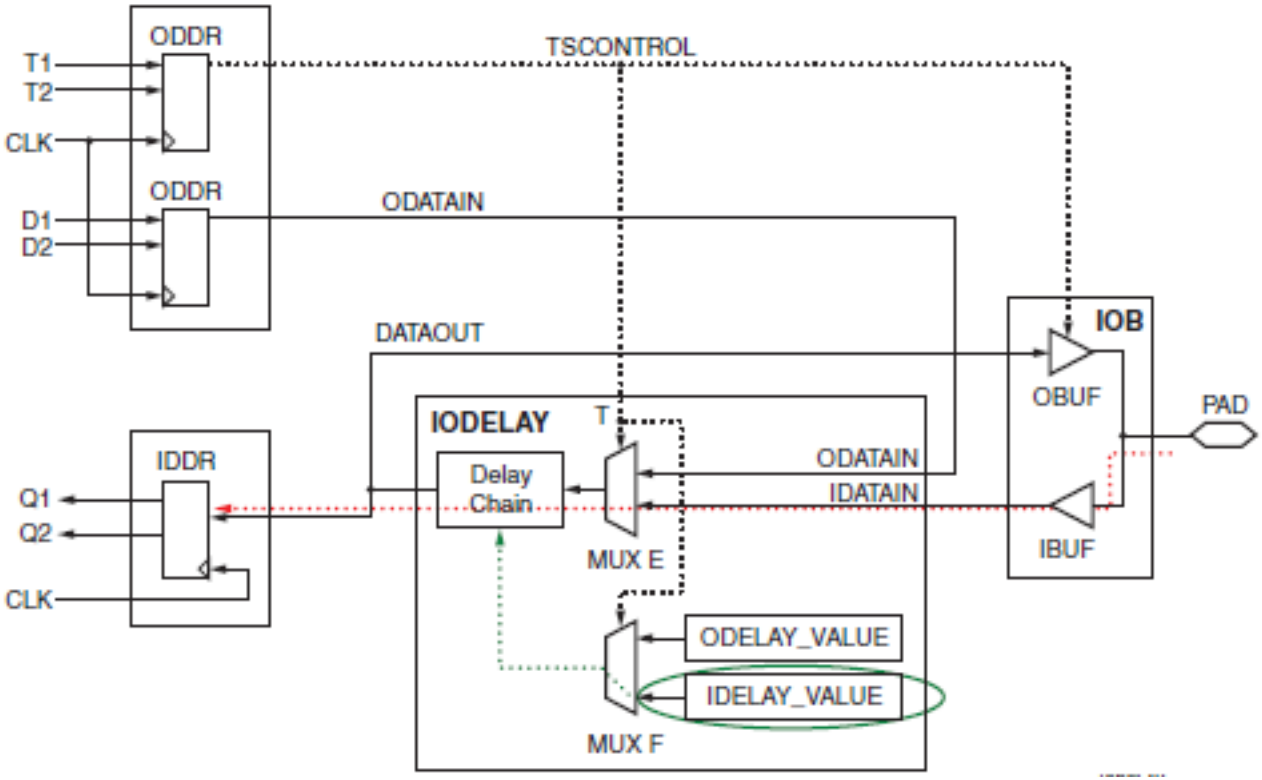


(the Muller pipeline for local “clock” generation)

The 3-stage Asynchronous pipeline



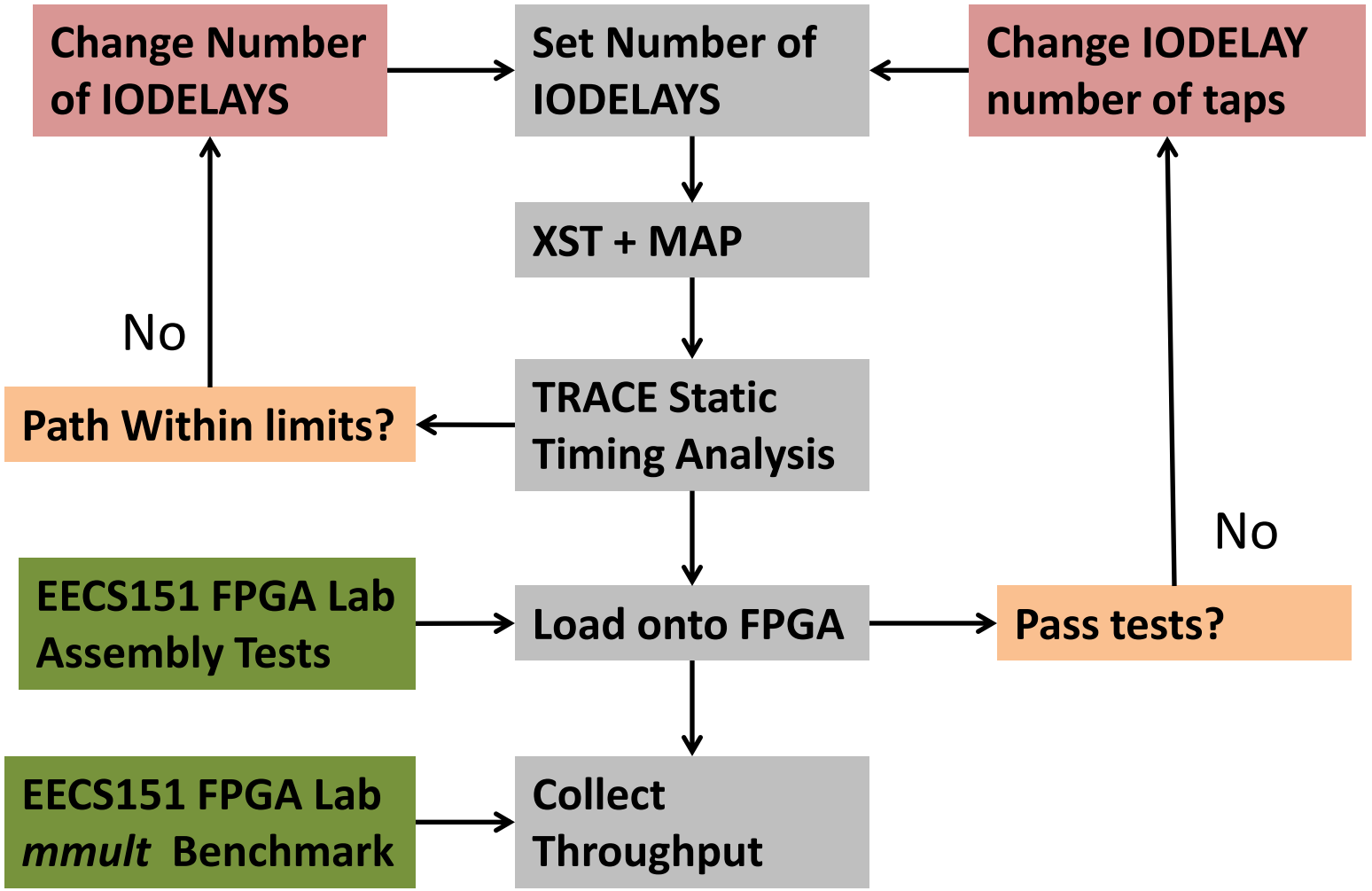
The Virtex-5 IODELAY Primitive



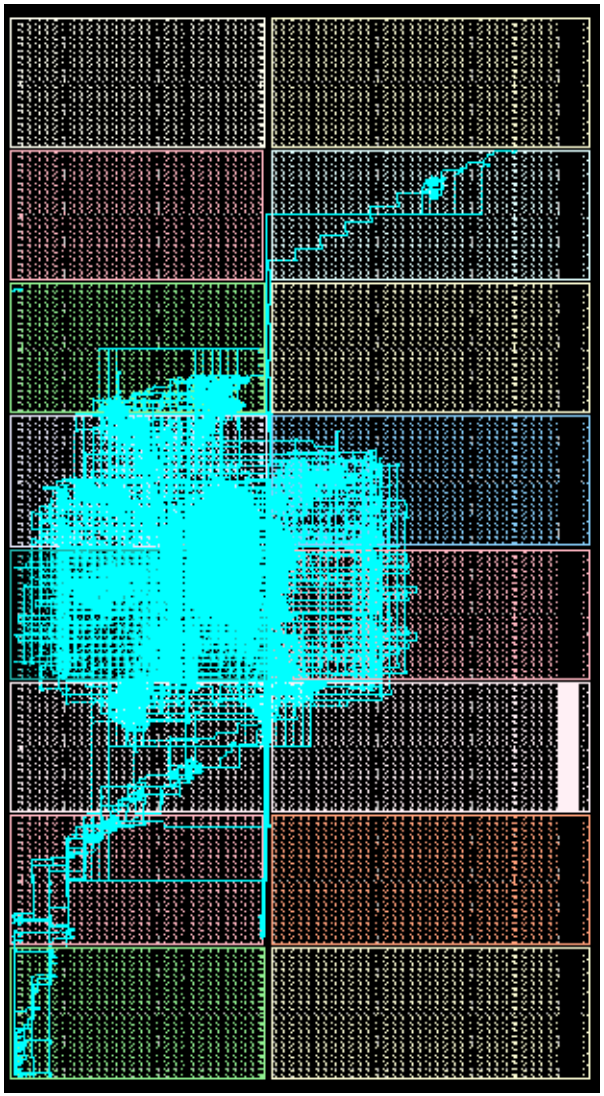
• The Parameter *IDELAY_VALUE* is number of delay taps (0 – 63). Tap delay resolution is 31 ps. at config. freq. of 50 MHz.

• Path delay can be as high as 3.83 ns (3 IODELAYS). Placement (hence delay) depends on number of IODELAYs in the design.

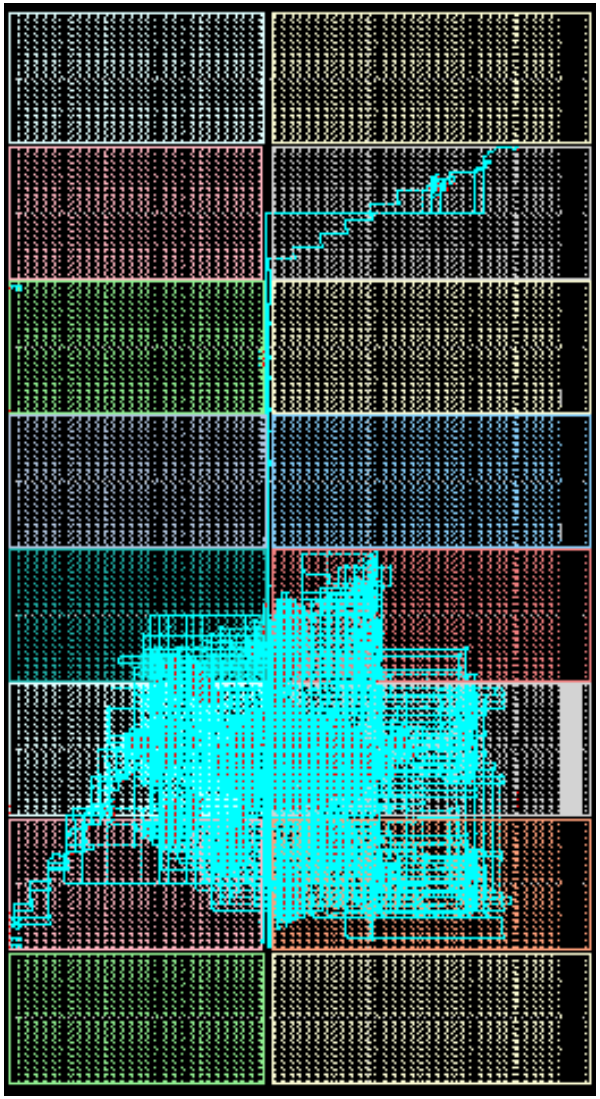
Delay Optimization on Xilinx Virtex 5 FPGA



After P&R on Xilinx Virtex-5

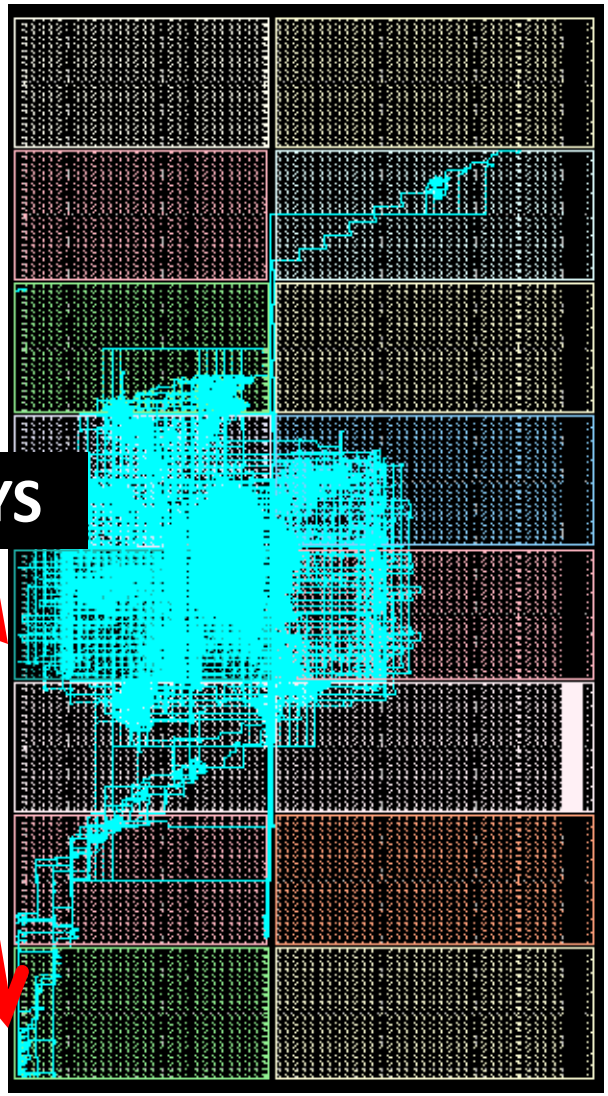


(asynchronous design)

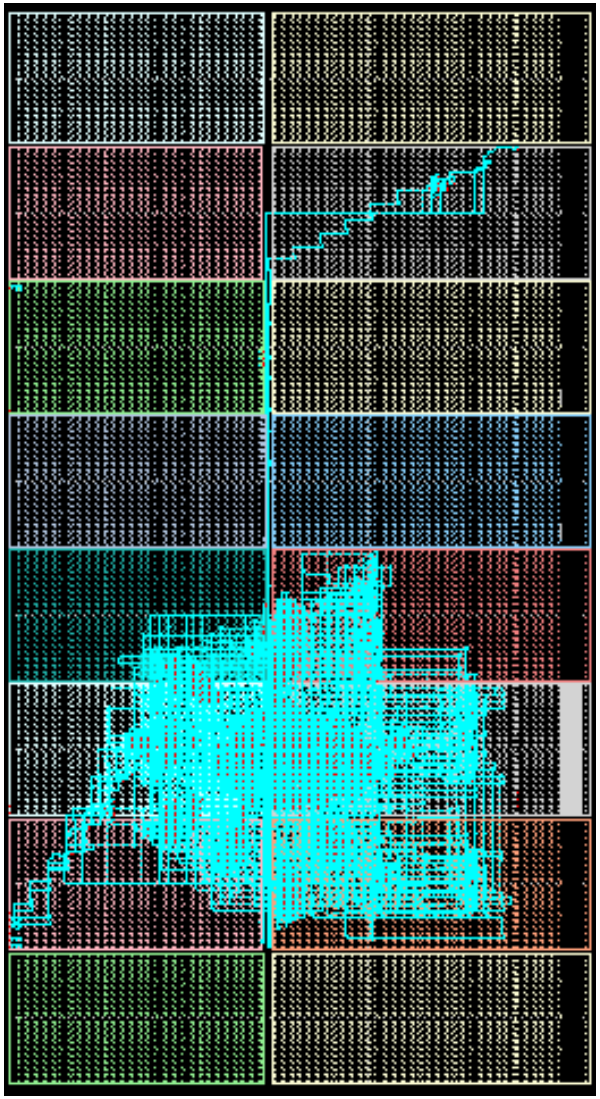


(synchronous design)

After P&R on Xilinx Virtex-5



(asynchronous design)



(synchronous design)

Results of Comparison

Resource Summary	SYNC	ASYNC	Change
SLICE Regs.	1413	1446	+33 (23 %)
SLICE LUTs	2358	2395	+37 (16%)
BUFG/ BUFGCTRL	2	5	+ 3
mmult Throughput	67.4 MIPS	24.1 MIPS	-43.3 MIPS (-64%)

• Throughput reduction as the Muller pipeline is **alternating HIGH and LOW** (about half throughput lost). The rest due to routing and IODELAY overestimates.

Conclusion and Comments

• A **lot of time** was spent optimizing the DELAY variables. These circuits usually require asynchronous primitives **directly on the FPGA board** (eg. Montage)

• The ML505 Development board **does not have supply resistors** to measure runtime power.

- Due to lack of resources, I could not move to another board for power measurements.
- Could have sourced a voltage regulator externally from a breadboard.

• An ASIC tool chain would be a fairer comparison. Using 2 Latches (hence, SLICES) for 1 Muller C gate is very wasteful. Some tools exists (eg. Balsa, Petrify, …) but are hard to setup and time-consuming.

• Have a working asynchronous pipeline! Project challenges indicative why asynchronous **not yet economic**.

Thank you