Faster and Cheaper Circuits through Self-timing

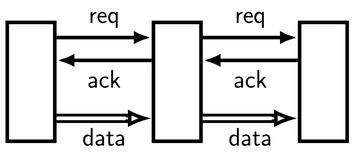
Adrian Wheeldon
a.r.wheeldon2@ncl.ac.uk

Motivation

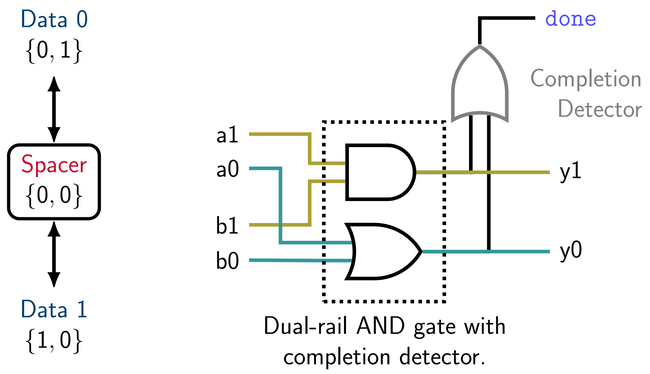
- Process variation on the increase.
  - Smaller silicon feature sizes.
- Subthreshold regime attractive for ultra-low-power.
  - 10x variation in gate delays vs. superthreshold.
- Multi-corner multi-mode analysis required in synchronous.
  - In order to meet highly variable timing.
- Self-timed datapaths adapt to delay variations at runtime.
  - Maximum performance for runtime environment.
  - Less design-time analysis required.

Self-timing & Dual-rail Logic

- No global clock.
  - Blocks exchange data on request/acknowledge.



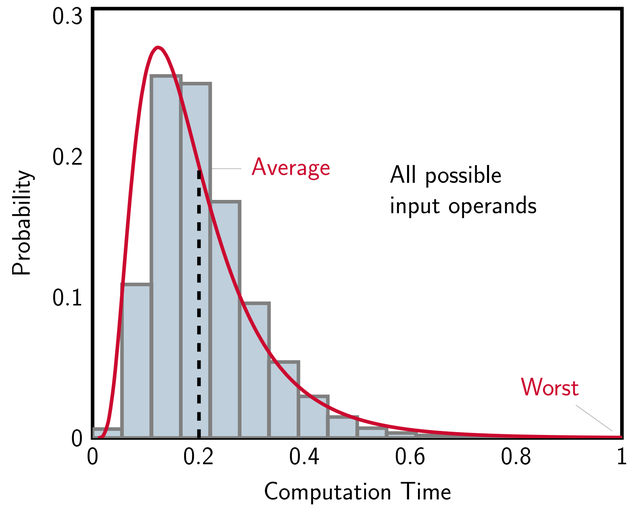
- Each data bit represented on two 'rails'. e.g. {a1,a0}.
  - Data 0: $\{0,1\}$, Data 1: $\{1,0\}$.
  - Spacer: $\{0,0\}$ – separates data temporally.
  - Completion Detector asserts when gate has computed.
  - Performance adapts to gate delays at runtime.



Dual-rail AND gate with completion detector.

Early Completion

- Actual computation time depends on operands.
  - A ripple-carry adder has a log-normal distribution:



  - A comparator shows a negative exponential distribution.
  - Computation is faster than worst-case for most operands.
- A synchronous circuit is idle for some of the clock cycle.
  - Latch answer early → avoid leaking energy.

Results

- Self-timed with early completion vs. synchronous.

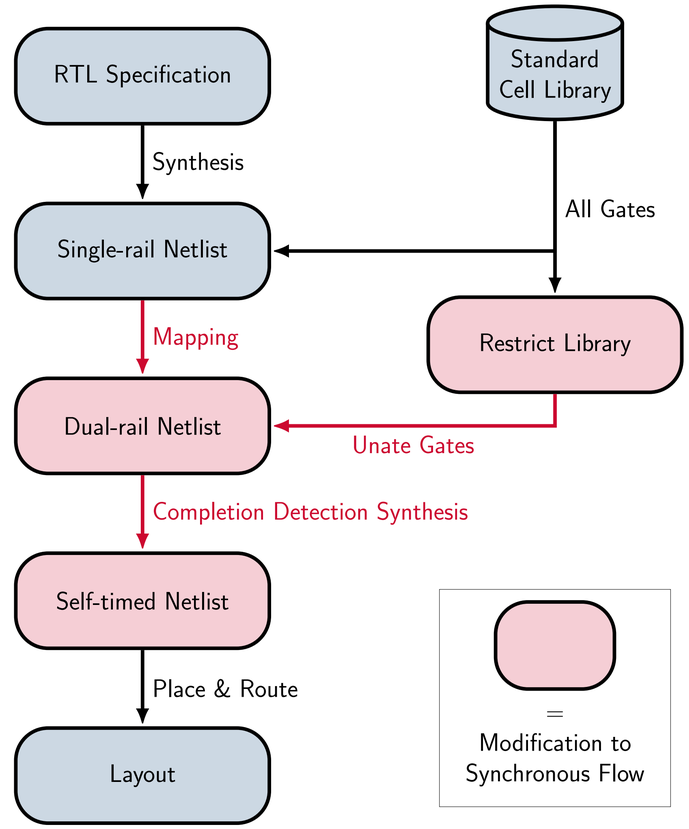
	Average		Worst		
	Delay	Energy	Delay	Energy	Area
32-bit Adder	0.2x	0.1x	1.1x	1.3x	2.3x
32-bit Comparator	0.2x	0.06x	0.9x*	0.6x*	0.9x*

*Less than 1x is achieved by maintaining 1-of-3 encoding at the output.

- Improved average delay and energy due to early completion.
  - Worst-case suffers, but encountered infrequently.
- Duplicated logic and completion detection increase area.
  - But rail swaps can replace inverters for area recovery.

ASIC Design Flow

- Integrates into synchronous design flow.



- Glitch-less logic guaranteed by unate gates and encoding.

Future Work

- More logic = more leakage.
  - How can we efficiently power-gate self-timed circuits?
- Conventionally, each gate requires a completion detector.
  - Can we introduce easy-to-meet timing constraints to reduce the number of completion detectors?
- Automation of single-rail → dual-rail conversion.
- Logical Equivalence Checking of single-rail → dual-rail.