A Robust and Self-Adaptive Clocking Technique for RSFQ Circuits – The Architecture

Ramy N. Tadros and Peter A. Beerel

University of Southern California (USC) – Los Angeles, CA

USC University of Southern California

USC Viterbi School of Engineering

Motivation

Semiconductor technologies approaching limits → RSFQ is promising for exascale supercomputing → Clock Distribution at high frequencies is a challenge because of uncertainties

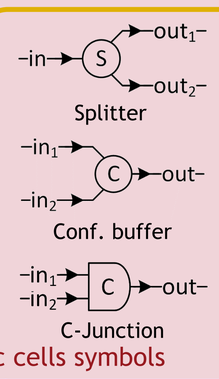
Because of uncertainties: Sync Flow, Async, Hybrid

All could not replace zero-skew clocking → Algorithmic loops are the problem → Processors in literature used custom and manual ways. Not purely synchronous → Clocking is still unresolved !

Clocking is still unresolved ! → We present the hierarchical chains of homogeneous clover-leaves clocking → Robust to unprecedented levels of uncertainties!

Rapid Single-Flux Quantum (RSFQ) Technology

- Amazing potential of 3 orders of magnitude lower power at 1 order of magnitude higher in speed.
- Binary information is presented in short quantized pulses.
- RSFQ basic convention: No levels, an SFQ pulse is logic high.
- All logic cells need a clock → Deep pipelines.
- No simple wires; either matched TLs or active JTLs.
- RSFQ fan-out is strictly one → splitter.
- C-Junction follow an FSM.



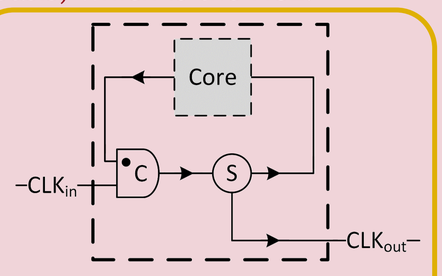
Some RSFQ basic cells symbols

Timing

- Timing RSFQ at very high frequencies with high uncertainty is challenging.
- Processors failed to use zero-skew clock trees.
- Zero-skew clock trees are not the natural solution for RSFQ.
- Algorithmic loops aggravates the problem.
- HCLC achieved up to 93% yield improvement for a 32-gates CSR.

Proposed Technique

- We introduce the Hierarchical Chains of Homogeneous Clover-Leaves Clocking; the (HC)²LC.
- Firs, we propose the architecture basis, which is the HCL.



The Hierarchical Chains' Link (HCL)

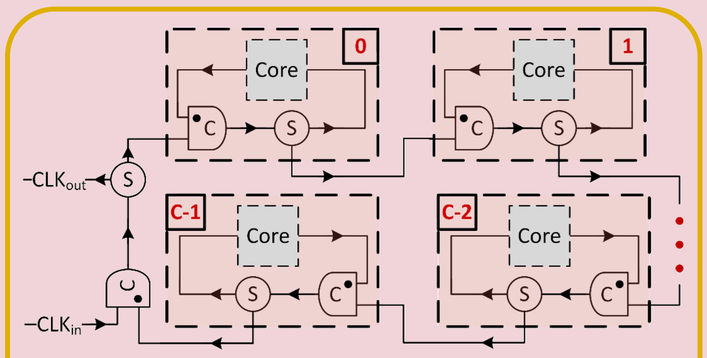
- A core can also be an HCL!
- We connect those links together to form a chain.

$$T_{Chain} = \max\left(T_{in}, (C+1)\cdot\delta_{ov}, \max_i T_{HCL_i}\right)$$

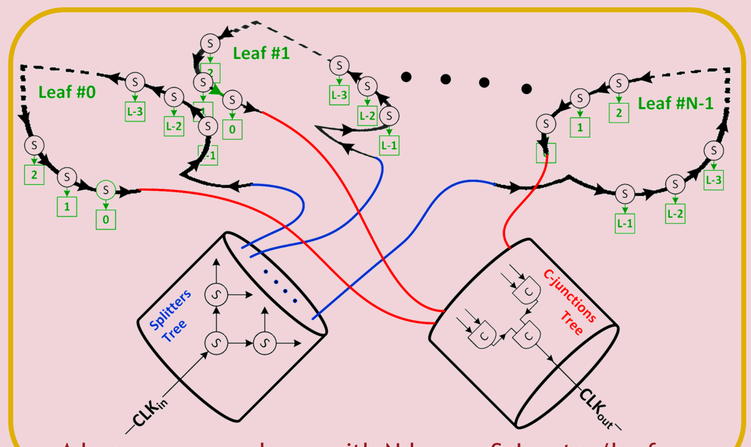
- For a large number of cores, we link each core to an HCL, then chain them and build the hierarchical structure.

- Assuming a periodic signal on CLK_{in} with period T_{in}, then CLK_{out} shall have a period of

$$T_{HCL} = \max(T_{in}, \delta_{ov} + T_{core})$$



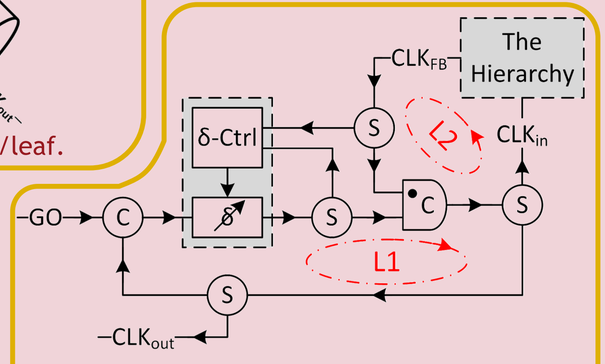
A chain of C HCLs



A homogeneous clover with N leaves & L gates/leaf.

- For the bottom level, the gates per clover/core are divided into L gates per leaf with N leaves per clover. We use counter-flow.

$$\delta_{clover} = \lceil \log_2 N \rceil \cdot \delta_{ov} + L_{max} \cdot \delta_{sp}$$



The top loop structure

- In the top loop, we need a GO pulse for start-up.
- δ_{ctrl} adjusts itself to the hierarchy.
- Always ensures that L1 is slower than any loop in L2.

(HC)²LC Key Properties

- Self-adaptive: The clock speed follows the slowest circuitry.
- The top loop self-adapts to the worst-case delays.
- When the top loop is the slowest, all clock sinks will have a determined skew from the reference point.
- The high resilience towards hold violations is because of the use of counter-flow clocking at the bottom level.
- It exploits spatial correlation between neighboring clock and data circuitry.
- Given a huge level of uncertainty, our clocking follows the circuit's variability and makes less assumptions about the cell delays.

- To evaluate (HC)²LC, we clocked the combinational ISCAS'89 benchmark circuits (see Table I).
- We used the cells characteristics of the MIT-LL 10kA/cm² SFQ fabrication process.
- Each benchmark is implemented twice to show the trade-off between clock speed and area at different parameters configurations.

Cycle Time

- All hierarchical chains share the same cycle time

$$T_{sys} = \max\begin{cases} (\lceil \log_2 N_{max} \rceil + 1)\cdot\delta_{ov} + L_{max}\cdot\delta_{sp} \\ (C_{max}+1)\cdot\delta_{ov} \\ (C_{top}+1)\cdot\delta_{ov} + \delta_{sp} \\ \delta_{cf} + 2\delta_{sp} + \delta_{ov} + \delta_{ctrl} \end{cases}$$

Proof of Concept

- Configure Parameters: (HC)²LC parameters: Number of links per chain (C), number of leaves per clover (N), and leaf length (L).
- Use ABC for synthesis: Levelize. Build Network. Gates to clock-sinks assignment map. Check for setup violations + fix. Re-levelize. Check for hold violations + fix.
- Convert to RSFQ: Convert the graph to behavioral SystemVerilog RSFQ interfaces.
- Cosim Simulation: Generate input combinations and compare to the Verilog benchmark + timing checks.

Table I: (HC)²LC applied to ISCAS'85 benchmark circuits.

Benchmark		c432		c499		c880		c1355		c1908		c2670		c3540		c5315		c6288		c7552	
# of Gates		245		209		285		241		279		718		1073		1073		1661		1783	
Chains per link		3	6	3	6	3	6	3	6	3	6	4	7	3	6	4	6	3	6	6	7
N/L		2/4	4/8	2/4	4/8	2/4	4/8	2/4	4/8	2/4	4/8	4/4	8/8	2/4	4/8	4/4	4/8	2/4	4/8	4/8	8/8
T_{sys} Ovh. %		32.7	106	40.5	141	54.8	140	40.5	141	32.7	106	30.4	116	16.1	80.4	15.9	60.2	32.7	106	44.3	84.2
JJs Ovh.	%	86	67	80	32	84	66	78	61	84	65	74	60	79	62	79	65	88	69	57	55
	%	63	27	63	27	63	27	63	27	63	27	54	26	63	26	54	26	63	26	26	21
Fixes Ovh. %		61.7	58.9	164	78.8	88.7	88.6	145	91.6	90.6	78.0	104	97.0	115	86.4	87.3	86.5	479	184	98.1	97
Latency Ovh. %		61.0	22.0	150	28.6	59.1	27.3	129	35.7	64.0	28.0	63.6	40.9	105	45.9	60	33.3	1417	318	47.8	34.8

- Results show an average performance overhead of 34% with an average area overhead of 58% as well as a higher performance penalty of 108% with a reduced area overhead of 26%.
- These overheads are reasonable given the improved functionality and scalability.
- Previous work used a similar, but limited, clocking structure over a 32-gates CSR, and showed a 93% yield improvement over the zero-skew tree clocking.

Clock-sinks Assignment

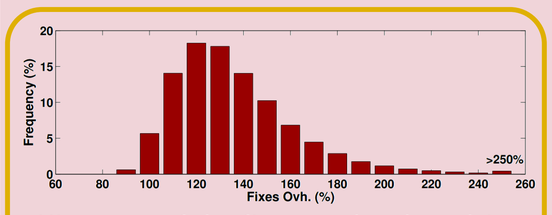
- Data path cannot always be consistently matched to counter-flow clocking.
- This could be optimized to minimize the fixes overheads quantized as follows.
- Number of setup flops, N_{FF}

$$= \max\left(0, \left\lceil \frac{t_i - t_j + \Delta_{max} + \delta_{setup} - T_{sys}}{T_{sys} - \Delta_{FF} - \delta_{setup}} \right\rceil\right)$$

- Number of hold buffers, N_{buf}

$$= \max\left(0, \left\lceil \frac{\delta_{hold} - t_i + t_j - \Delta_{min}}{\delta_{buf}} \right\rceil\right)$$

- The optimization is left for future work.



Histogram of the fixes overhead resulting from random assignment of gates to clock-sinks, using c432 with C=N=L=4.

Acknowledgment

This work was supported in part by the Safe and Secure Operations Office of the Intelligence Advanced Research Projects Activity (IARPA) under contract no. FA8750-15-C-0203-IARPA-BAA-14-03.

Contact

Ramy Tadros {rtadros@usc.edu} and Peter A. Beerel {pabeerel@usc.edu}, Ming-Hsieh department of electrical engineering, Viterbi school of engineering, University of Southern California (USC), Los Angeles, CA.