State Access for RSFQ Test and Analysis

Marly Roncken, Ebelechukwu Esimai, Vivek Ramanathan, Warren Hunt, Jr., Ivan Sutherland
Asynchronous Research Center at Portland State University and The University of Texas at Austin
(mroncken@pdx.edu)

Examples and Simulation Results

ASC 2022
Invited poster Digital I
1EPo2E-01

MOTIVATION

The Need:

- **General, affordable test strategy for RSFQ circuits [2]** to access states and control actions for:
  1. flexible initialization
  2. (functional) test and analysis
  3. at-speed test and debug

The Problem:

- Too many RSFQ states to afford test access to all
- Too little RSFQ test control of asynchronous actions
- Destructive RSFQ readouts create moving test targets

SOLUTION

- **Limit test access:** by avoiding information-free states*
  (* states that are the same before and after a computation)
- **Fix test targets:** by using nondestructive readouts
- **Adopt solutions from CMOS:**
  scan chains for external access to states [5]
  MrGO for external control of asynchronous actions [4]
  - *go* — start / continue
  - *nogo* — stop / halt
  - scan chain delivers *go-nogo* — per action

REFERENCES

[1] E. Dijkstra, "Guarded Commands, Nondeterminacy and Formal Derivation of Programs," *CACM 18(8)* 1975.

[2] K. Likharev, V. Semenov, "RSFQ Logic/Memory Family: A New Josephson-Junction Technology for Sub-Terahertz-Clock-Frequency Digital Systems," *TAS 1(1) 1991.*

[3] P. Patra, S. Polonsky, and D. Fussel, "Delay Insensitive Logic for RSFQ Superconductor Technology," *ASYNC 1997.*

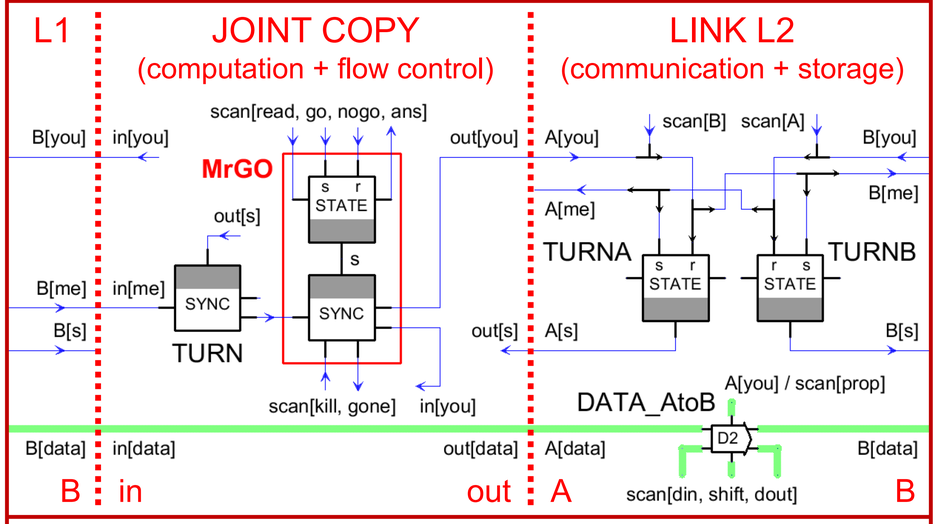
[4] M. Roncken, S. Mettala Gilla, H. Park, N. Jamadagni, C. Cowan, and I. Sutherland, "Naturalized Communication and Testing," *ASYNC 2015.*

[5] I. Sutherland, Q. Morgan, W. Hunt, Jr., V. Ramanathan, and M. Roncken, "An IEEE-compatible JTAG Test-Access-Port Controller for RSFQ Logic and Systems," ***ASC 2022 presentation 2EOr1B-05 (TUESDAY).***

DESIGN APPROACH

The Link-Joint model [4]:

- **Design:** a network of Joints connected by (one-to-one) Links
- **Partition:** Joints **act** on **states** stored in Links
- **Protocol:**
  synchronous: a clock updates all Link states globally
  asynchronous: the Joints take turns updating a Link they share



RSFQ gate specifications — after Dijkstra [1]

STATE

- uses nondestructive readout
- delivers *go-nogo* + protocol state
- provides fixed scan targets

STATE (read, set, reset, ans, s1)
set → s1:=true ; set:=false
reset → s1:=false ; reset:=false
read → ans:=s1 ; read:=false

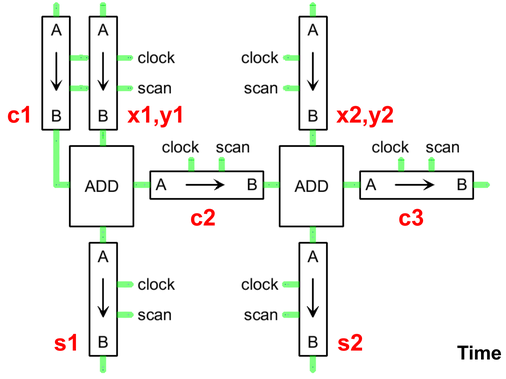
SYNC

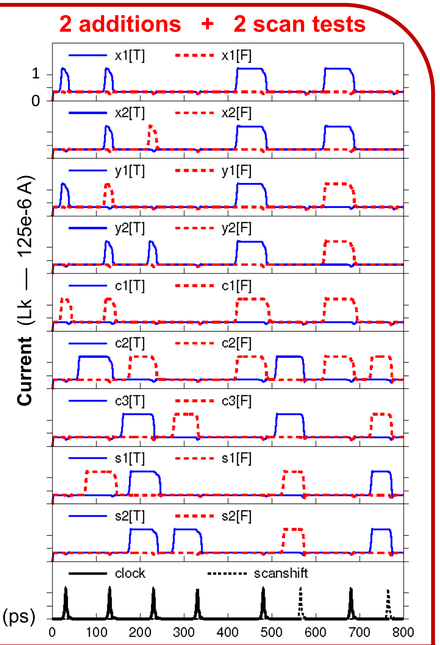
- ANDs multiple STATE deliveries
- can kill further propagation (SYNC in MrGO)

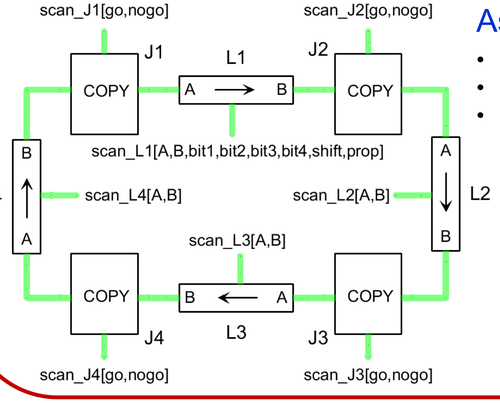
SYNC (in, kill, gone, ansA, ansB, s1)
s2 : bool
in → s2:=true ; in:=false
kill → gone:=s2 ; s2:=false ; kill:=false
s1 ∧ s2 → ansA:=true ; ansB:=true ; s2:=false

Clocked pipelined adder

- Dual-rail ADD — after Patra [3]
- D2 latches in the Links have clock + scan
- (information-free) states in ADD have neither
- **Simulated: Time x Current**
  2x (2-bit) pipelined addition (0-400 ps)
  2x (1-bit) scan test per ADD (400-800 ps)







Asynchronous ring FIFO

- Partial scan of data (in Link L1 only)
- Partial scan of MrGO, TURNA, TURNB
- **Simulated: Occupancy x Throughput**