Ballistic Asynchronous Reversible Computing in Superconducting Circuits

Michael P. Frank, Center for Computing Research

with Rupert Lewis (Quantum Phenomena Dept.)
Contributors to our Reversible Computing research program

- Full group of recent staff at Sandia:
  - Michael Frank (Cognitive & Emerging Computing)
  - Robert Brocato (RF MicroSystems) – now retired
  - David Henry (MESA Hetero-Integration)
  - Rupert Lewis (Quantum Phenomena)
    - Terence “Terry” Michael Bretz-Sullivan
  - Nancy Missert (Nanoscale Sciences) – now retired
  - Matt Wolak (now at Northrop-Grumman)
  - Brian Tierney (Rad Hard CMOS Technology)

- Thanks are also due to the following colleagues & external research collaborators:
  - Karpur Shukla (CMU → Flame U. → Brown U.)
    - Currently in Prof. Jimmy Xu’s Lab for Emerging Techs.
  - Hannah Earley (Cambridge U. → startup)
  - Erik DeBenedictis (Sandia → Zettaflops, LLC)
  - Joseph Friedman (UT Dallas)
  - Kevin Osborn (LPS/JQI)
    - Liuqi Yu, Ryan Clarke, Han Cai
  - Steve Kaplan (independent contractor)
  - Rudro Biswas (Purdue)
    - Dewan Woods & Rishabh Khare
  - Tom Conte (Georgia Tech/CRNCH)
    - Anirudh Jain, Gibran Essa
  - David Guéry-Odelin (Toulouse U.)
  - FAMU-FSU College of Engineering:
    - Sastry Pamidi (ECE Chair) & Jerris Hooker (Instructor)
    - 2019-20 students:
      - Frank Allen, Oscar L. Corces, James Hardy, Fadi Matloob
    - 2020-21 students:
      - Marshal Nachreiner, Samuel Perlman, Donovan Sharp, Jesus Sosa

Thanks are due to Sandia’s LDRD program, DOE’s ASC program, and the DoD/ARO ACI (Advanced Computing Initiative) for their support of this line of research!
Ballistic Asynchronous Reversible Computing in Superconducting Circuits

Background: Why Reversible Computing?
- Relevant classic results in the thermodynamics of computing
  - Recently generalized to quantum case
- Two major types of approaches to reversible computing in superconducting circuits:
  - Adiabatic approaches – Well-developed today.
    - Likharev’s parametric quantron (1977); more recent QFP tech (YNU & collabs.) w. substantial demo chips.
  - Ballistic approaches – Much less mature to date.
    - Fredkin & Toffoli’s early concepts (1978–’81); much more recent work at U. Maryland, Sandia, UC Davis

Review: The relatively new asynchronous ballistic approach to RC in SCE.
- Addresses concerns w instability of the synchronous ballistic approach
- Potential advantages of asynchronous ballistic RC (vs. adiabatic approaches)
- Implementation w. superconducting circuits (BARCS effort).

Focus of this Talk:
- Presenting our recent work on enumerating/classifying possible BARCS functions w. ≤3 ports and ≤2 states.
Can we envision reversible computing as a deterministic elastic interaction process?

**Historical origin of this concept:**
- Fredkin & Toffoli’s *Billiard Ball Model of computation* (“Conservative Logic,” IJTP 1982).
  - Based on elastic collisions between moving objects.
  - Spawned a subfield of “collision-based computing.”
  - Using localized pulses/solitons in various media.

No power-clock driving signals needed!
- Devices operate when data signals arrive.
- The operation energy is carried by the signal itself.
  - Most of the signal energy is preserved in outgoing signals.

However, all (or almost all) of the existing design concepts for ballistic computing invoke implicitly *synchronized* arrivals of ballistically-propagating signals…
- Making this work in reality presents some serious difficulties, however:
  - Unrealistic in practice to assume precise alignment of signal arrival times.
  - Thermal fluctuations & quantum uncertainty, at minimum, are always present.
  - Any relative timing uncertainty leads to chaotic dynamics when signals interact.
  - Exponentially-increasing uncertainties in the dynamical trajectory.
  - Deliberate resynchronization of signals whose timing relationship is uncertain incurs an inevitable energy cost.

Can we come up with a new ballistic model that avoids these problems?
Ballistic Asynchronous Reversible Computing (BARC)

**Problem:** Conservative (dissipationless) dynamical systems generally tend to exhibit chaotic behavior...
- This results from direct nonlinear *interactions* between multiple continuous dynamical degrees of freedom (DOFs), which amplify uncertainties, exponentially compounding them over time...
  - E.g., positions/velocities of ballistically-propagating "balls"
  - Or more generally, any localized, cohesive, momentum-bearing entity: Particles, pulses, quasiparticles, solitons...

**Core insight:** In principle, we can greatly reduce or eliminate this tendency towards dynamical chaos...
- We can do this simply by avoiding any direct interaction between continuous DOFs of different ballistically-propagating entities

Require localized pulses to arrive *asynchronously*—and furthermore, at clearly distinct, *non-overlapping* times
- Device’s dynamical trajectory then becomes *independent* of the precise (absolute and relative) pulse arrival times
  - As a result, timing uncertainty per logic stage can now accumulate only *linearly*, not exponentially!
  - Only relatively occasional re-synchronization will be needed
  - For devices to still be capable of doing logic, they must now maintain an internal discrete (digitally-precise) state variable—a stable (or at least metastable) stationary state, e.g., a ground state of a well

No power-clock signals, unlike in adiabatic designs!
- Devices simply operate whenever data pulses arrive
- The operation energy is carried by the pulse itself
  - Most of the energy is preserved in outgoing pulses
  - Signal restoration can be carried out incrementally

**Goal of current effort at Sandia:** Demonstrate BARC principles in an implementation based on fluxon dynamics in SuperConducting Electronics (SCE) ([BARCS effort](#))
One of our early tasks: Characterize the simplest nontrivial BARC device functionalities, given a few simple design constraints applying to an SCE-based implementation, such as:

- (1) Bits encoded in fluxon polarity; (2) Bounded planar circuit conserving flux; (3) Physical symmetry.

Determined through theoretical hand-analysis that the simplest such function is the **1-Bit, 1-Port Reversible Memory Cell (RM):**

- Due to its simplicity, this was then the preferred target for our subsequent detailed circuit design efforts…

![RM Transition Table](image)

<table>
<thead>
<tr>
<th>Input Syndrome</th>
<th>Output Syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1(+1)</td>
<td>(+1)+1</td>
</tr>
<tr>
<td>+1(−1)</td>
<td>(+1)−1</td>
</tr>
<tr>
<td>−1(+1)</td>
<td>(−1)+1</td>
</tr>
<tr>
<td>−1(−1)</td>
<td>(−1)−1</td>
</tr>
</tbody>
</table>

**Some planar, unbiased, reactive SCE circuit w. a continuous superconducting boundary:**
- Only contains L’s, M’s, C’s, and unshunted JJ’s
- Junctions should mostly be subcritical (avoids $R_N$)
- Conserves total flux, approximately nondissipative

**Desired circuit behavior (NOTE: conserves flux, respects T symmetry & logical reversibility):**
- If polarities are opposite, they are swapped (shown)
- If polarities are identical, input fluxon reflects back out with no change in polarity (not shown)
- *(Deterministic) elastic ‘scattering’* type interaction: Input fluxon kinetic energy is (nearly) preserved in output fluxon
RM—First working (in simulation) implementation!

Erik DeBenedictis: “Try just strapping a JJ across that loop.”
  ◦ This actually works!

“Entrance” JJ sized to = about 5 LJJ unit cells (~1/2 pulse width)
  ◦ I first tried it twice as large, & the fluxons annihilated instead…
    ◦ “If a 15 μA JJ rotates by 2π, maybe ½ that will rotate by 4π” 😊

Loop inductor sized so ±1 SFQ will fit in the loop (but not ±2)
  ◦ JJ is sitting a bit below critical with ±1

WRspice simulations with ±1 fluxon initially in the loop
  ◦ Uses ic parameter, & uic option to .tran command
    ◦ Produces initial ringing due to overly-constricted initial flux
      ◦ Can damp w. small shunt G

Polarity mismatch → Exchange

Polarity match → Reflect (=Exchange)
Resettable version of RM cell—Designed & Fabricated!

Apply current pulse of appropriate sign to flush the stored flux (the pulse here flushes out positive flux)

- To flush either polarity → Do both (±) resets in succession
**barc** tool for enumerating/classifying BARCS device functions

Custom Python program with 16 modules.

Tool is now complete; will be open-sourced.

Layer-cake view of software architecture:
- Modules only import modules from lower-numbered layers.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Module Names &amp; Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td><strong>barc</strong> (top-level program)</td>
</tr>
<tr>
<td>3</td>
<td><strong>deviceType</strong> – Classification of devices with given dimensions.</td>
</tr>
<tr>
<td>2</td>
<td><strong>deviceFunction</strong> – Device with a specific transition function. <strong>stateSet</strong> – Identifies a set of accessible device states.</td>
</tr>
<tr>
<td>1</td>
<td><strong>pulseAlphabet</strong> – Sets of pulse types. <strong>pulseType</strong> – Identifies a specific type of pulse. <strong>state</strong> – Identifies an internal state of a device. <strong>symmetryGroup</strong> – Equivalence class of device functions. <strong>transitionFunction</strong> – Bijective map, input→output syndromes.</td>
</tr>
<tr>
<td>0</td>
<td><strong>characterClass</strong> – Defines a type of signal characters. <strong>deviceDimensions</strong> – Defines size parameters of devices. <strong>dictPermuter</strong> – Used to enumerate transition functions. <strong>signalCharacter</strong> – Identifies I/O event type (pulse type &amp; port). <strong>symmetryTransform</strong> – Invertibly transforms a device function. <strong>syndrome</strong> – An initial or final condition for a device transition. <strong>utilities</strong> – Defines some low-level utility functions.</td>
</tr>
</tbody>
</table>

Symmetry group #38 has 6 functions:
- Function #155.
- Function #340.
- Function #481.
- Function #285.
- Function #365.
- Function #185.

Example: Function #155 = [1]*3(L,R):
- 1(L) → (R)2
- 1(R) → (L)3
- 2(L) → (R)1
- 2(R) → (R)3
- 3(L) → (L)2
- 3(R) → (L)1

Function #155 has the following symmetry properties:
- It is D-dual to function #481
- It is S-dual to function #481
- It is E(1,2)-dual to function #340
- It is E(1,3)-dual to function #185
- It is E(2,3)-dual to function #481
- It R(-1)-transforms to function #365
- It R(1)-transforms to function #285

Example description of a symmetry-equivalence group as output by the **barc** tool.
Symmetry Relations of Interest

The following symmetry relations on BARC functions are considered in this work:

- **Direction-reversal symmetry** $\mathcal{D}$ – Symmetry under exchange of input & output syndromes (involution of transition func.)

- **State-exchange symmetry** $\mathcal{S}$ – Symmetry under an exchange of state labels (and fluxes, for flux-polarized states).

- **Flux-negation symmetry** $\mathcal{F}$ – Symmetry under negation of all (I/O flux & internal state) flux polarities.

- **Moving-flux negation symmetry** $\mathcal{M}$ – Symmetry under negation of all moving (I/O) flux polarities.
  - **Input flux negation symmetry** $\mathcal{I}$ – Symmetry under negation of all input flux polarities.
  - **Output flux negation symmetry** $\mathcal{O}$ – Symmetry under negation of all output flux polarities.

- **Port-relabeling symmetries** $\mathcal{R}_P$ – Symmetry under a particular permutation $P$ of the port labels.
  - **Port exchange symmetry** $\mathcal{E}(p_i, p_j)$ – Symmetry wrt an exchange of labels between a particular pair of ports.
  - **Rotational symmetry** $\mathcal{R}_r$ – Relevant for $n \geq 3$ ports. Symmetry under (planar) rotation of port labels.
  - **Reflection across port axis** $\mathcal{R}_{p_i}$ – Symmetry under reflection of ports on either side of port $p_i$.
  - **Mirror symmetry** $\mathcal{M}_2, \mathcal{M}_3$ – Symmetry under port exchange for a 2-port device, or any reflection for a rotationally symmetric 3-port device.
  - **Complete port symmetry** $\mathcal{R}(n)$ – Symmetry under all possible relabelings of the ports.
Equivalence Groups For the 24 One-Port, Two-State Elements:

2 \cdot 1 \cdot 2 = 4 \text{ I/O syndromes} \Rightarrow 4! = 24 \text{ permutations (raw reversible transition functions).}

**Stateful Reflector**

(State Unused—Not Atomic)

**Configurable Inverter**

(Doesn’t Change State)

**Toggle**

(Doesn’t Use State)

**Toggle & Conditional Invert**

(Neither flux-negation symmetric nor flux-conserving)

**Exchange (RM)**

(Doesn’t Use State)

**Conditional Toggle**

**Type 4**

**Type 5**

(Neither flux-negation symmetric nor flux-conserving)
Two-Port, Two-State, Flux-Polarized Elements

There are $2^3 = 8$ I/O syndromes, thus $8! = 40,320$ raw reversible transition functions.

- But only 96 of them satisfy the flux conservation constraint.
- And only 10 of these are nontrivial primitives satisfying all constraints.

These 10 functions sort into 7 equivalence groups as follows:

<table>
<thead>
<tr>
<th>Self-Symmetry Group Size</th>
<th>Equivalence Group Size</th>
<th>Number of Equiv. Groups</th>
<th>Total # of Raw Trans. Funcs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>TOTALS:</strong></td>
<td></td>
<td><strong>7</strong></td>
<td><strong>10</strong></td>
</tr>
</tbody>
</table>

The corresponding functional behaviors can be described as:

1. Reversible Shift Register (RSR) – More on this one later.
2. Directed Reversible Shift Register (DRSR)
3. Filtering RM Cell (FRM)
4. Directed Filtering RM Cell (DFRM).
5. Polarized Flipping Diode (PFD). – Also has a flux-neutral equivalent.
6. Asymmetric Polarity Filter (APF).

(Osborn & Wustmann ‘22)
Illustrations of 2-port, 2-state, flux-polarized elements:

(Table Rows Shown for $\uparrow$ Initial State Only)

1. Reversible Shift Register (RSR):

   (Implemented by Osborn & Wustmann ‘22)

<table>
<thead>
<tr>
<th>Input syndrome</th>
<th>Output syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\uparrow$A($\uparrow$)</td>
<td>($\uparrow$)B $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$A($\uparrow$)</td>
<td>($\downarrow$)B $\uparrow$</td>
</tr>
<tr>
<td>$\uparrow$B($\uparrow$)</td>
<td>($\uparrow$)A $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$B($\uparrow$)</td>
<td>($\downarrow$)A $\uparrow$</td>
</tr>
</tbody>
</table>

2. Directed Reversible Shift Register (DRSR):

<table>
<thead>
<tr>
<th>Input syndrome</th>
<th>Output syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\uparrow$A($\uparrow$)</td>
<td>($\uparrow$)B $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$A($\uparrow$)</td>
<td>($\downarrow$)B $\uparrow$</td>
</tr>
<tr>
<td>$\uparrow$B($\uparrow$)</td>
<td>($\uparrow$)A $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$B($\uparrow$)</td>
<td>($\downarrow$)A $\downarrow$</td>
</tr>
</tbody>
</table>

3. Filtering RM Cell (FRM):

<table>
<thead>
<tr>
<th>Input syndrome</th>
<th>Output syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\uparrow$A($\uparrow$)</td>
<td>($\uparrow$)B $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$A($\uparrow$)</td>
<td>($\downarrow$)A $\uparrow$</td>
</tr>
<tr>
<td>$\uparrow$B($\uparrow$)</td>
<td>($\uparrow$)A $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow$B($\uparrow$)</td>
<td>($\downarrow$)B $\uparrow$</td>
</tr>
</tbody>
</table>
Illustrations of 2-port, 2-state, flux-polarized elements, cont.:

4. Directed Filtering RM Cell (DFRM):

<table>
<thead>
<tr>
<th>Input syndrome</th>
<th>Output syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑)A(↑)</td>
<td>(↑)B↑</td>
</tr>
<tr>
<td>↓)A(↑)</td>
<td>(↓)A↑</td>
</tr>
<tr>
<td>↑)B(↑)</td>
<td>(↑)A↑</td>
</tr>
<tr>
<td>↓)B(↑)</td>
<td>(↓)B↓</td>
</tr>
</tbody>
</table>

5. Polarized Flipping Diode (PFD):

<table>
<thead>
<tr>
<th>Input syndrome</th>
<th>Output syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑)A(↑)</td>
<td>(↑)A↑</td>
</tr>
<tr>
<td>↓)A(↑)</td>
<td>(↓)B↑</td>
</tr>
<tr>
<td>↑)B(↑)</td>
<td>(↑)B↑</td>
</tr>
<tr>
<td>↓)B(↑)</td>
<td>(↓)A↑</td>
</tr>
</tbody>
</table>

5. Asymmetric Polarity Filter (APF):

<table>
<thead>
<tr>
<th>Input syndrome</th>
<th>Output syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑)A(↑)</td>
<td>(↑)A↑</td>
</tr>
<tr>
<td>↓)A(↑)</td>
<td>(↓)B↑</td>
</tr>
<tr>
<td>↑)B(↑)</td>
<td>(↑)B↑</td>
</tr>
<tr>
<td>↓)B(↑)</td>
<td>(↑)A↓</td>
</tr>
</tbody>
</table>
Two-Port, Two-State, Flux-Neutral Elements

There are \(2^2\)! = 24 raw flux-symmetric transition functions.
1. 14 of these are nontrivial, atomic functional primitives.

These sort into 4 equivalence groups as follows:

<table>
<thead>
<tr>
<th>Self-Symmetry Group Size</th>
<th>Equivalence Group Size</th>
<th>Number of Equiv. Groups</th>
<th>Total # of Raw Trans. Funcs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

TOTALS: 4 14

There are 5 distinct functional behaviors (described in forwards time direction):
1. Alternating Barrier (AB), 2 representations – See next slide.
2. Polarized Flipping Diode (PFD), 2 reps..
3. Variant Polarized Flipping Diode (VPFD), 2 reps..
4. Asymmetric Polarized Flipping Diode (APFD), 4 reps.,
   (and this one is \(D\)-dual to:)
5. Selectable Barrier (SD), 4 reps.
Ex. 2-port, 2-state neutral element: **Alternating Barrier (AB)**

Flux-conserving, flux-negation symmetric element.
- Also has mirror ($\mathcal{M}_2$) symmetry.
- Has two $D, S$ dual representations.

Flux-neutral internal states $\rightarrow$ Doesn’t change fluxon polarity.

State descriptions:
- $S_{WB}^+$: *Positive-wire, negative-barrier*.
  - Transmits positive ($\uparrow$) fluxons, reflects negative ($\downarrow$) fluxons.
- $S_{WB}^-$: *Positive-Barrier, negative-wire*.
  - Reflects positive ($\uparrow$) fluxons, transmits negative ($\downarrow$) fluxons.

Transition function description:
- Fluxons arriving at either port are routed as per the state descriptions above.
- State toggles with every interaction.

<table>
<thead>
<tr>
<th>Input Syndrome</th>
<th>Output Syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\uparrow)p_1(S_{WB}^+)$</td>
<td>$(S_{WB}^+)_p_2)$ $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow)p_1(S_{WB}^+)$</td>
<td>$(S_{WB}^+)_p_1)$ $\downarrow$</td>
</tr>
<tr>
<td>$\uparrow)p_2(S_{WB}^+)$</td>
<td>$(S_{WB}^-)_p_1)$ $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow)p_2(S_{WB}^+)$</td>
<td>$(S_{WB}^-)_p_2)$ $\downarrow$</td>
</tr>
<tr>
<td>$\uparrow)p_1(S_{WB}^-)$</td>
<td>$(S_{WB}^+)_p_1)$ $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow)p_1(S_{WB}^-)$</td>
<td>$(S_{WB}^+)_p_2)$ $\downarrow$</td>
</tr>
<tr>
<td>$\uparrow)p_2(S_{WB}^-)$</td>
<td>$(S_{WB}^-)_p_2)$ $\uparrow$</td>
</tr>
<tr>
<td>$\downarrow)p_2(S_{WB}^-)$</td>
<td>$(S_{WB}^-)_p_1)$ $\downarrow$</td>
</tr>
</tbody>
</table>
Summary of Results for Three-Port, Two-State Elements:
(Still assuming flux conservation & flux negation symmetry)

Devices with flux-polarized states:
- $2 \cdot 3 \cdot 2 = 12$ I/O syndromes
- $12! = 497,001,600$ raw reversible funcs.
- 25,920 of these are flux-conserving.
- 288 of those are flux-negation symmetric.
- 245 of those are atomic (primitives).
- 219 of those use the state non-trivially.
- Sort into 39 equiv. groups as follows:

Devices with flux-neutral states:
- $1 \cdot 3 \cdot 2 = 6$ I/O syndromes (for ↑ inputs)
- $6! = 720$ permutations.
- 653 of them are atomic primitives.
- 600 of those use the state non-trivially.
- Sort into 45 equiv. groups as follows:

### Summary of (3,2) flux-polarized behaviors

<table>
<thead>
<tr>
<th>Equivalence Class Size:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>6</th>
<th>12</th>
<th>Tot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Symmetry Group Size:</td>
<td>12</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>No. of Equivalence Classes:</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>24</td>
<td>4</td>
<td>39</td>
</tr>
<tr>
<td>Total number of Functions:</td>
<td>1</td>
<td>8</td>
<td>18</td>
<td>144</td>
<td>48</td>
<td>219</td>
</tr>
</tbody>
</table>

### Summary of (3,2) flux-neutral behaviors

<table>
<thead>
<tr>
<th>Equivalence Class Size:</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>12</th>
<th>24</th>
<th>Tot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Symmetry Group Size:</td>
<td>12</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>No. of Equivalence Classes:</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>23</td>
<td>11</td>
<td>45</td>
</tr>
<tr>
<td>Total number of Functions:</td>
<td>2</td>
<td>4</td>
<td>54</td>
<td>276</td>
<td>264</td>
<td>600</td>
</tr>
</tbody>
</table>
Illustrations of some 3-port, 2-state flux-neutral elements

Recall there are 45 different non-trivial, atomic functional behaviors (counting $\mathcal{D}$-duals as equivalent).

Of these, only a few exemplar behaviors are illustrated here.

Still seeking implementations of any of these…. 

- Polarized Neutral Toggle Rotary (PNTR)
- Polarized Toggle Controlled Barrier (PTCB)
- Polarized Controlled Flipping Diode (PCFD)
- Polarized Throw Switch, Type A (PTSA)
- Polarized Throw Switch, Type B (PTSB)
- Polarized Knock-twice Toggle Controlled Barrier (PKTCB)

[NOTE: All behaviors shown here are for (+) fluxons only; (−) fluxons interact oppositely with states]
Some Next Steps for the BARCS effort

1. Document classification results more fully (in progress).

2. Finish developing **SCIT** (Superconducting Circuit Innovation Tool) tool to facilitate discovery of circuit-level implementations of BARCS functions.
   - Including training an AI/ML model to quickly solve the inverse (circuit design) problem.

3. Better understand role of physical symmetries in the circuit design of BARCS elements.
   - What, if any, functions are ruled out by the symmetries?
   - Must we consider including additional SCE device types to break the symmetries?

4. Identify a computation-universal set of primitive elements that we also know how to implement!
   - Or, show that this is impossible using the present set of devices.

5. Additional work on fabrication & empirical validation of BARCS circuit designs.

6. Gain a better understanding of the limits of the energy efficiency of this approach.

Clearly, much work along these lines remains to be done!
- We would be very happy to recruit new collaborators
Conclusion

The long-neglected *ballistic* mode of reversible computing has recently attracted renewed interest.

- Classic problems with synchronization & chaotic instability in ballistic computing schemes appear to be resolvable via the asynchronous approach.
- The new method seems to hold some promise for possibly achieving improved energy-delay products and/or more compact circuit designs vs. adiabatic approaches.

Also, note that ballistic approaches are not viable *at all* in CMOS!

- CMOS has nothing like a ballistic flux soliton, & has no nonlinear reactive elements like JJs…
- Thus, we are leveraging unique advantages of superconducting electronics in this approach.

In this paper & talk, we reported our progress on enumerating & classifying the possible BARCS functions…

- Given constraints of full logical reversibility, flux conservation, & flux negation symmetry.

Multiple US-based research groups in superconductor physics & engineering are now making early progress along this line of work…

- We invite additional domestic & international colleagues to join us in investigating this interesting line of research!