Design and Optimization of Quantum Electronic Circuits

Giovanni De Micheli
What is cool out there?

▲ Making chips faster by cooling them
  ▼ CryoCMOS
  ▼ Superconducting single-quantum gates
  ▼ Quantum computing

▲ We set the operating conditions to beat the RC
Superconducting electronics

Classical computing

High operational speed
Dataflows operate in the 50-100 GHz range

Gates require clocking to operate
Pipelined logic has to be balanced

Refrigeration (4 K) energy offsets power efficiency
Refrigeration technology is slowly improving
Quantum computing technologies

QC leverages superposition and entanglement
Support algorithms with lower complexity

A wide array of realization technologies
Superconductors, silicon, optics

Scaling and noise are still issues
Qubit count and coherence time are limited

Refrigeration to tenth of mK for noise reasons
Interfacing to host is complex

(c) Giovanni De Micheli
Superconductors and JJs

- A superconductor is a material where
  - Resistivity vanishes and magnetic flux field is expelled
  - When below a critical temperature $T_c$

- Josephson junction - JJ
  - Superconducting wire interrupted by thin insulator
  - Nb/AlO$_x$/Nb or Al/AlO$_x$/Al

- Supercurrent $I_s = I_c \sin(\phi)$, where $\phi$ is the phase difference of electrodes wave functions

- Critical current: $I_c$ – max current in SC state

[Source: Dwave]
Some history

- IBM ran a large SCE program until 1983
  - The project failed because of the *latching* issue
- Several Japanese companies followed IBM’s path
  - They improved the *materials* but still had latching issues

\[ I_b \] is slightly below \[ I_c \]
\[ I_in \] makes the JJ be resistive
A fast voltage transition 0 \(\rightarrow\) 1
More difficult to reset JJ -- latching
Slower transition 1 \(\rightarrow\) 0

(c) Giovanni De Micheli
Likharev radically changed to *pulse-based* signalling

- **Single-flux quantum logic** SFQ
- Input is short pulse $\int V(t)dt = \phi_0 = h/2e = 2.07 \text{ mV ps}$
- Load is *overdamped*
- An output pulse is generated

[Likharev et al., TAS, 91] (c) Giovanni De Micheli
Fanout and Splitters

Output pulse has fanout equal to 1

Splitter circuit – splitter trees

Splitter trees needed to handle high fanout

(c) Giovanni De Micheli
The S/R latch or SQUID

- Two persistent states
  - Corresponding to direction of current in loop $J_1 - L_1 - J_2$
  - Flux quantum
- Input sets to 1 – Clock resets to 0

(c) Giovanni De Micheli
Principles of combinational cells

- Signal encoding: 1 = pulse -- 0 = no pulse
- Cell has two stable states:
  - Direction of current flow in a loop
- Operation:
  - Inputs are evaluated when clock is present
  - Output is issued at end of clock
  - Clock resets the state

(c) Giovanni De Micheli
Example of OR and AND

Combinational cells are more complex than registers

[Likharev – Stony Brook]

(c) Giovanni De Micheli
Most variants are related to biasing the gates:

- **RSFQ** – Resistive/Rapid SFQ, replace R with JJ [Likharev91]

- **ERSFQ** – Energy-Efficient SFQ, replace bias resistors with limiting JJ and L [Mukhanov 11]

- **eSFQ** – Merge bias and clock [Mukhanov 11]

- **LV SFQ** – Low-voltage bias (1mV) [Tanaka 13]

- **DSFQ** – Dynamic (Self-resetting) SFQ [Krylov 20]

Adiabatic variants

- **RQL** – Reciprocal Quantum Logic [Herr 11]

- **AQFP** – Adiabatic Quantum Flux Parametron [Takeuchi 13]

The SFQ variants (c) Giovanni De Micheli
Achievements

**Fig. 4. Twiddle factor ROM sub-block diagram.** The proposed ROM is designed in a serial-in-parallel-out manner to reduce the I/O pads.

**Fig. 5. Data buffer sub-block diagram.** Two types of registers are introduced in this design: 9-bit shift registers and 9-to-7 bit round-down shift registers. The Twiddle Factor ROM must output the corresponding twiddle factor for every butterfly operation as shown in Fig. 1. Therefore, the twiddle factors stored in the memory are in the order of $W_0$, $W_2$, $W_1$, $W_3$. The number of butterfly operations of each processing stage is counted so that the correct twiddle factor corresponding to the current butterfly operation can be output. Additionally, this ROM is designed in a serial-in-parallel-out manner to further reduce the I/O pads.

**IV. IMPLEMENTATION OF A 7-BIT 8-POINT SFQ FFT**

In a previous design, a 4-bit 8-point FFT was implemented due to the limitation of the chip area [14]. The design cannot cope with bit extension for carry operations, which limits the accuracy of the FFT. In this study, we designed a low hardware-cost FFT processor with an extendable data bit-length. Specifically, in the previous design, a shuffling circuit was introduced to sort the data, which consists of four n-bit shift registers and two multiplexers. In this design manner, not only extra hardware costs but also extra execution time for data write operations are required. To resolve this problem, we proposed a new data buffer structure that can sort data more efficiently, as shown in Fig. 5 and explained in section III. Thanks to this new structure, the proposed design can reduce the hardware cost by about 40% compared to the previous design, and the data bit-length is extended to 7-bit.

The design of a physical layout of the SFQ FFT processor has been accomplished using a 1.0 µm 9-layer process. **Fig. 6. Die micrograph of the designed SFQ FFT processor.** Design components: high-speed clock generator, data buffer, SFQ-DC interface, input/output register, twiddle factor ROM, butterfly ALU.

**A. Experimental Test Results**

Multiple chips have been tested under low temperature at 4.2K in liquid helium. Low-speed test signals are input to trigger the on-chip high-speed clock generator, which can generate a serial of clock pulses around 50GHz depending on the bias voltage supply. The input data is first written in an input register at a low frequency of 100kHz. After on-chip high-speed operation, the output is stored in an output register for low-speed readout (100kHz). The write-in and read-out operation is conducted from room temperature testing devices.

Correct operation of the FFT processor is verified using many sine wave sampling codes with an amplitude of 1 and a period of 8. **Fig. 7 shows an example test result.** In the output waveform, a transition/flip of the voltage level represents an SFQ signal. For every clock signal transition, if the data signal flips at the same time, the output becomes “1,” otherwise the output remains “0.” The upper 6 waveforms represent the low-speed write-in and read-out clock signals together, real and imaginary part of input $x(0)$ and real part of input $x(1)$ as the imaginary part of input $x(1)$.
Achievements

Synopsys SuperTools Superconducting Electronics Phase 2A Program

First Fully Automated Superconducting Microcontroller Design Demonstration with Fusion Compiler

The First Fully-Automated ERSFQ Microcontroller Circuit including CTS, Splitter Insertion, Power Delivery and PTL Routing – all desired features for SCE Technology automation

Fully synthesized AMD 2901 [Amaru et al., 2021]

(c) Giovanni De Micheli
Some ballpark numbers

<table>
<thead>
<tr>
<th>Logic</th>
<th>Clock Freq. [GHz]</th>
<th>$E_{\text{bit}} / I_c \Phi_0$</th>
<th>Typical $I_c$ [mA]</th>
<th>EDP [aJ·ps]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>$\sim 10^5$</td>
</tr>
<tr>
<td>RSFQ [1]</td>
<td>50</td>
<td>19</td>
<td>150</td>
<td>120</td>
</tr>
<tr>
<td>eSFQ [2]</td>
<td>20</td>
<td>0.8</td>
<td>150</td>
<td>12</td>
</tr>
<tr>
<td>RQL [3]</td>
<td>10</td>
<td>0.33</td>
<td>150</td>
<td>10</td>
</tr>
<tr>
<td>LV-RSFQ [4]</td>
<td>20</td>
<td>3.5</td>
<td>150</td>
<td>54</td>
</tr>
<tr>
<td>AQFP [5]</td>
<td>5</td>
<td>0.0083</td>
<td>50</td>
<td>0.086</td>
</tr>
<tr>
<td>Quantum limit</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$5.3 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Input is a current
Direction of current encodes 1/0
Circuit has two loops
  Either one is conducting – state 1/0
Input evaluated when clock is present
  May force a gradual transition of conducting loop
  Creates a corresponding output signal

[Goto 56]

[Yoshikawa, Ayala et al. Yokohama National University]

(c) Giovanni De Micheli
AQFP buffer and derivatives

▲ Buffer
  ▼ Output follows input

▲ Inverter
  ▼ Buffer where output coil is reversed

▲ Majority gate
  ▼ Three buffers in parallel with an inductive coupler

▲ Extensions:
  ▼ Logic gates NPN-equivalent to majority
  ▼ AND-2 and OR-2 by creating “constant buffers”

(c) Giovanni De Micheli
Adiabatic Quantum Flux Parametron (AQFP)
Logic design principles

▲ Balancing:
  ▷ When considering a logic network graph, all paths from inputs to any node have same length

▲ Fanout:
  ▷ All multiple fanout points need signal splitting

▲ Logic optimization:
  ▷ Logic should be optimized by majority logic, especially when majority is native abstraction

(c) Giovanni De Micheli
Path balancing for SFQ

Assumption: splitters are not clocked in SFQ

[Katam, Pedram, TAS18]
Dynamic programming for SFQ

Assumption: splitters are not clocked in SFQ

PBMap (based on AIG representation in ABC)

- Given a node and its cuts
- Best local solution minimizes the sum of cost of the children nodes plus cost of the inserted buffers
- Proceed bottom up

For tree topologies gives optimum solution

For DAGs, it is a heuristic

Area-retiming can refine solution

[Pasandi, Pedram, ICCAD 19]
"Splitter insertion for AQFP"

- A splitter is a clocked buffer with a passive fork
  - Inserting splitters imbalances a circuit
  - Require additional buffers

A novel algorithm will be presented at DAC 2022

[Cai et al., ICCD 19]
Majority logic synthesis

△ Exploit redundancy of majority function
△ Boolean substitution

(c) Giovanni De Micheli
What is in for physical design?

- A new set of interesting problems
  - Inductors are the limit in placement
- Wiring can be done by *passive or Josephson* transmission lines
  - Relatively less critical for speed (compared to CMOS)
- Clock routing is complicated
  - Because of reaching out to all gates
- Adiabatic realizations are even more complex
  - Power and clock routing
Conclusions

▲ There is strong evidence that SCE circuits can provide us with:
   ▼ Faster computing
   ▼ Higher energy-efficient computation
▲ Current limitations are:
   ▼ Scalability to larger realizations
   ▼ Lack of EDA design tools and open libraries
   ▼ Cooling systems
▲ The technical difficulties are surmountable and new opportunities are in reach

(c) Giovanni De Micheli
Thank you
Warm congratulations to Ricardo Reis

Recipient of the Lifetime Achievement Award