Superconductor-based computer design and modeling tools

Masamitsu Tanaka[1], Teruo Tanimoto[2], and Koji Inoue[2]

Nagoya University[1] and Kyushu University[2]

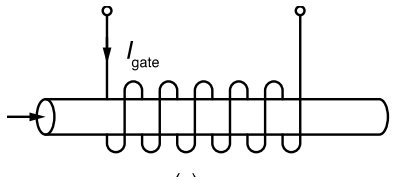
Outline

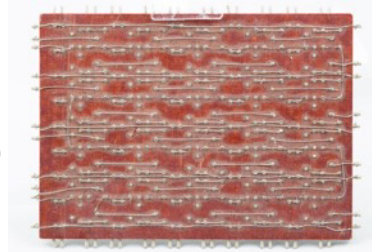
- Basic of SFQ Technology: Masamitsu Tanaka
- Impact of Practical Designs on Area/Power/Performance: Teruo Tanimoto
- Architectural Challenge on SFQ-based Computing: Koji Inoue

Basic of SFQ Technology

History of Superconductor Digital Circuits

- **1955 Cryotron**
  - Use thermal superconductor-normal transition.
  - Bulky device. Slow.

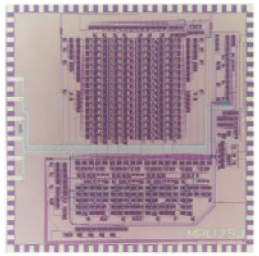




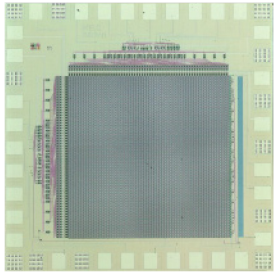
Cryotron ring oscillator

Photo: Computer History Museum

- **1966 Latching logic**
  - Use Josephson devices and voltage levels.
  - Fast operation up to 1 GHz a.c.

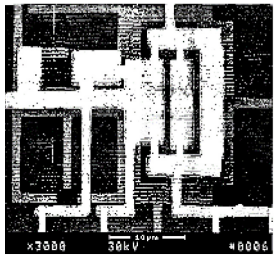


4-bit MPU
Kotani et al., ISSCC1988



4K-bit RAM
Nagasawa et al., IEEE TAS 1995

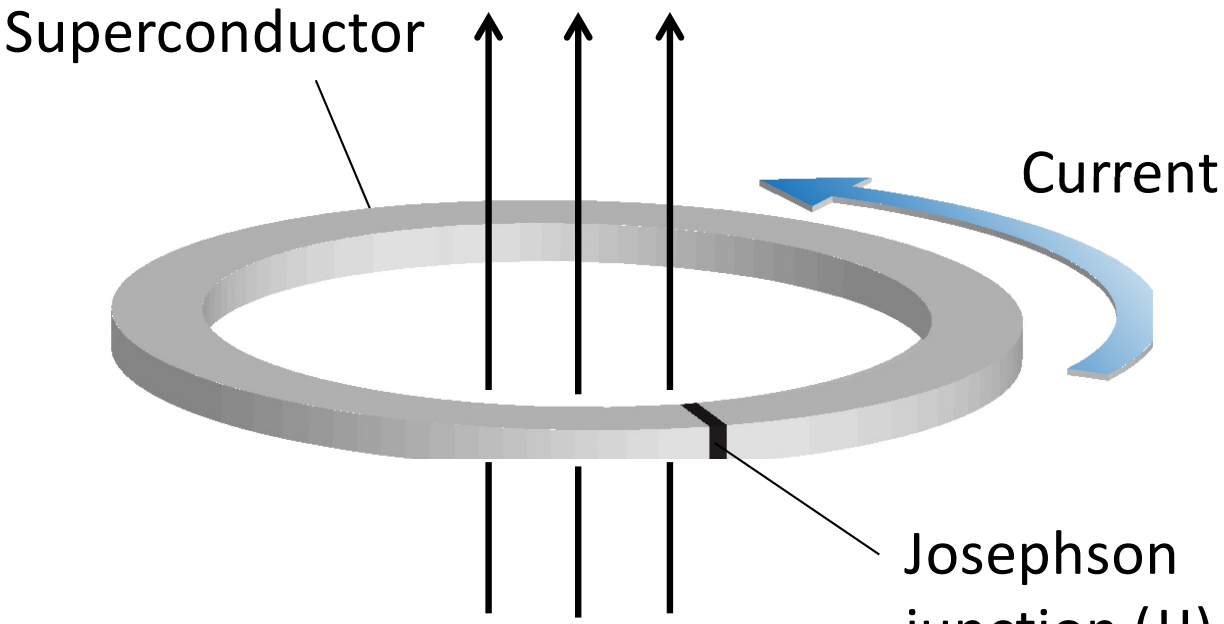
- **1976 Phase-mode / 1991 rapid single-flux-quantum (RSFQ) logic**
  - Use single flux quantum (SFQ) in a superconductor loop.
  - Ultrafast (100+ GHz) & low power.



770-GHz TFF
Chen et al., IEEE TAS 1999

- **2010– Energy-efficient families & new devices**
  - ERSFQ, adiabatic flux parametron, nanocryotron, π junctions, etc.

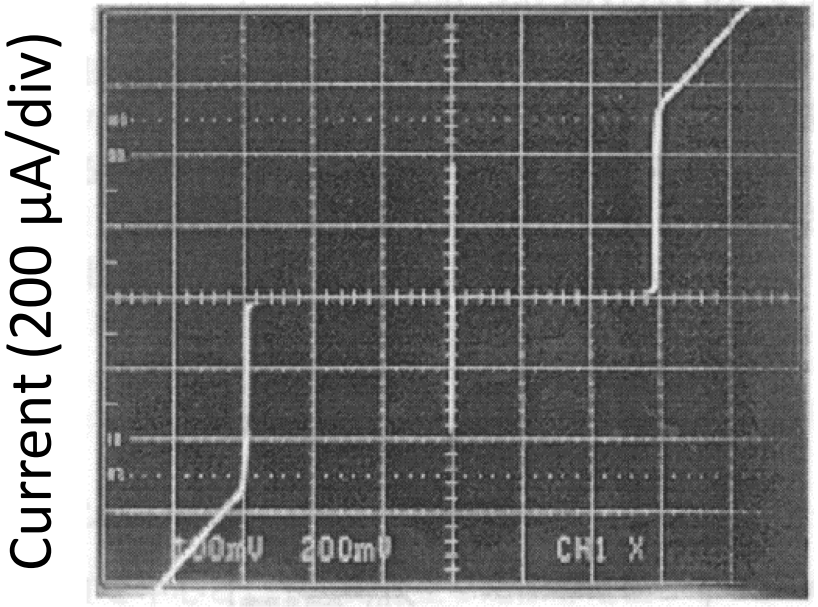
ABC of SFQ: Fundamental

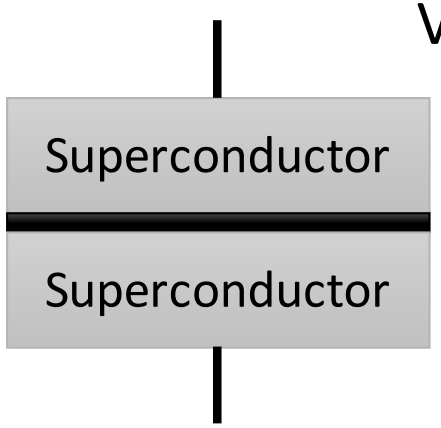


$\Phi = n\,\Phi_0$

$\Phi_0 = 2.068 \times 10^{-15}$ Wb

Magnetic flux quantization

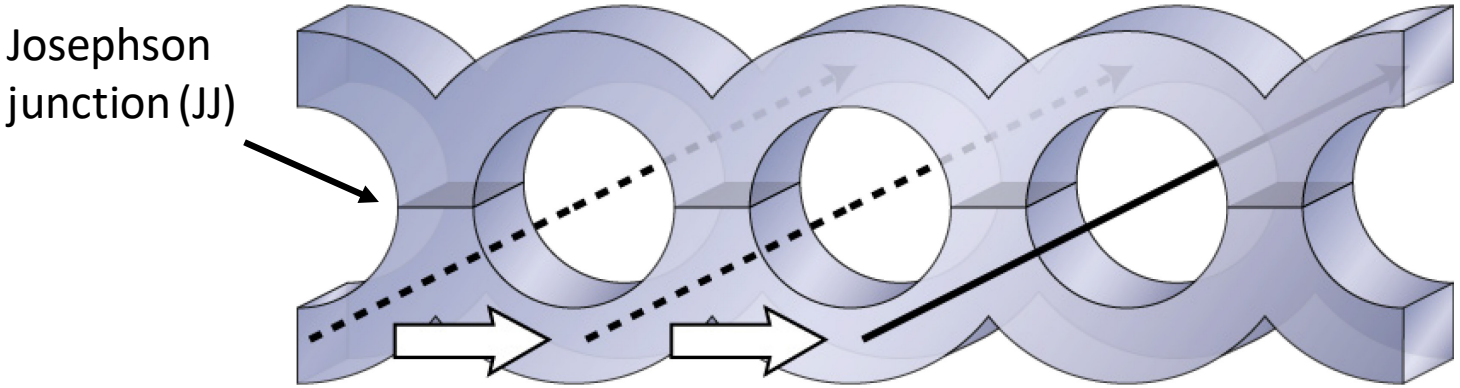




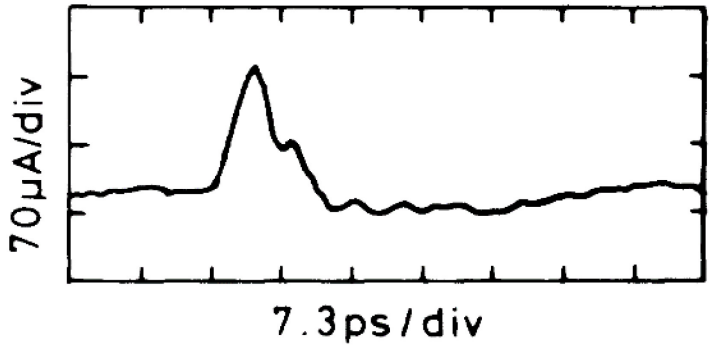
Josephson junction

Propagation of SFQ Signal

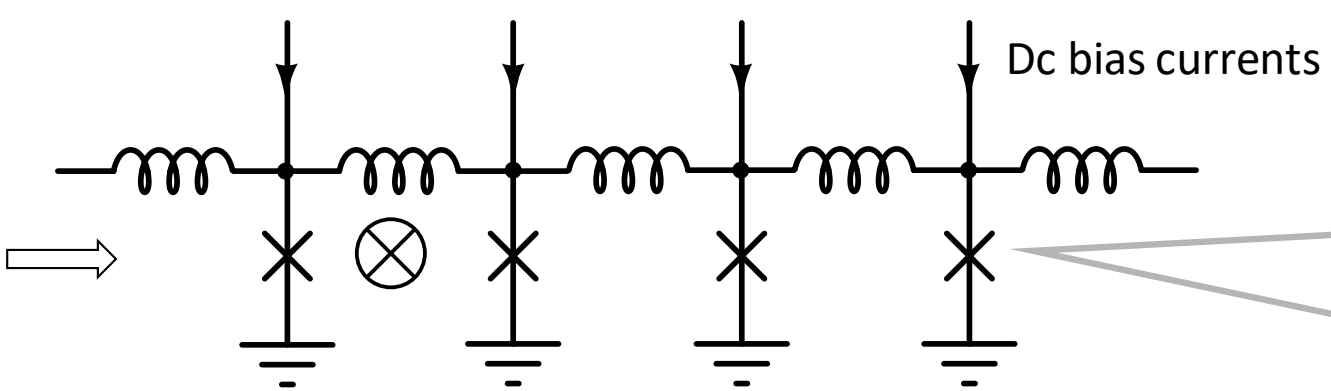
- Josephson transmission line (JTL)

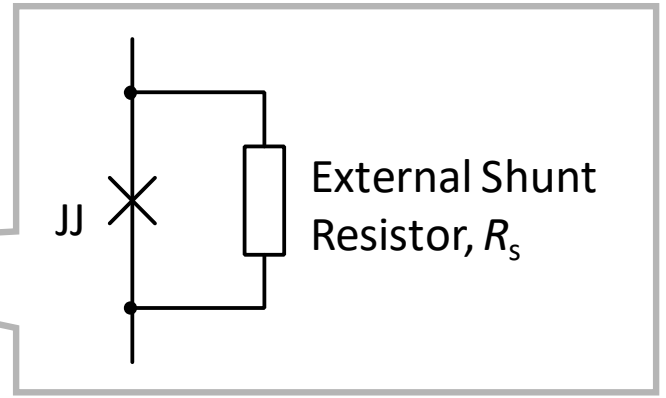


Impulse-shape voltage
only when JJ switches.
→ **Energy consumption**



A. Fujimaki et al., PRL 1987

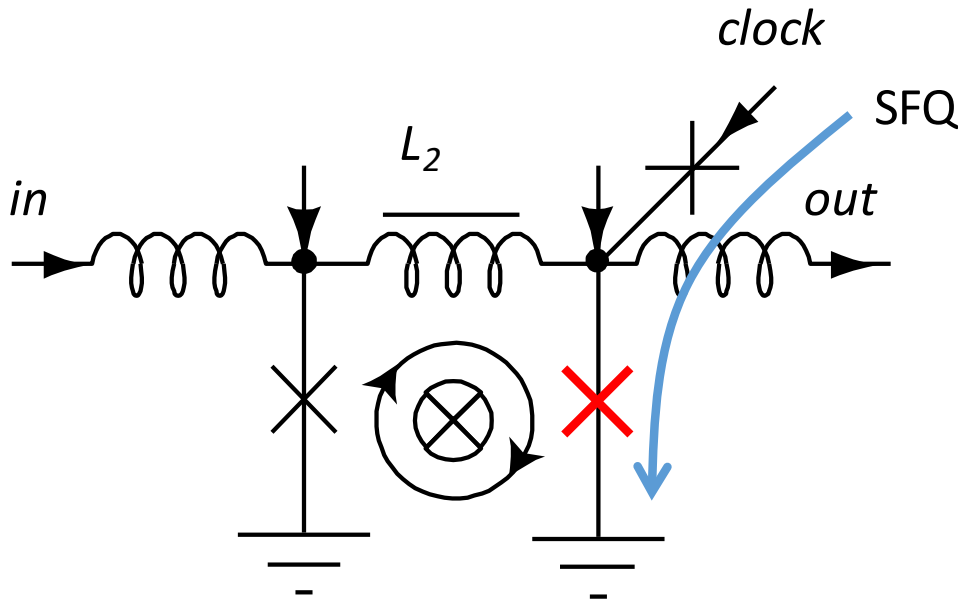




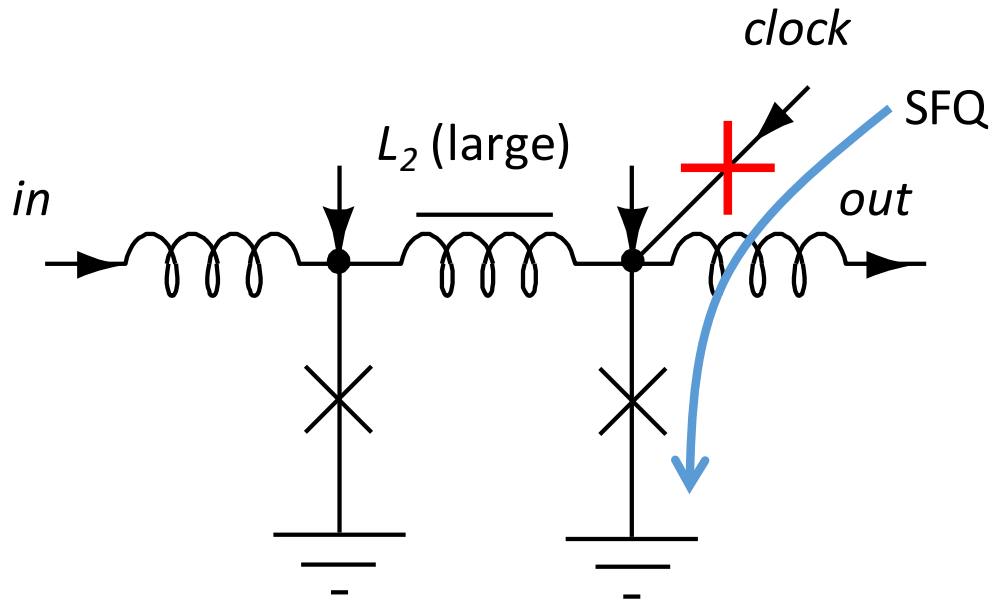
After K. K. Likharev and V. K. Semenov *IEEE Trans. Appl. Supercond.* **1** (1991).

SFQ Flip-flop

- A superconductor loop with large inductance can hold an SFQ.
- Clock signal triggers the stored SFQ and generates output.



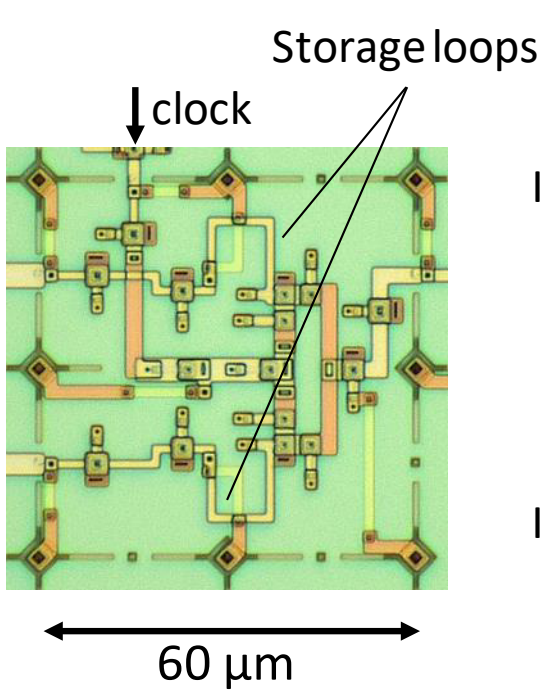
SFQ stored in the loop ➔ "1"

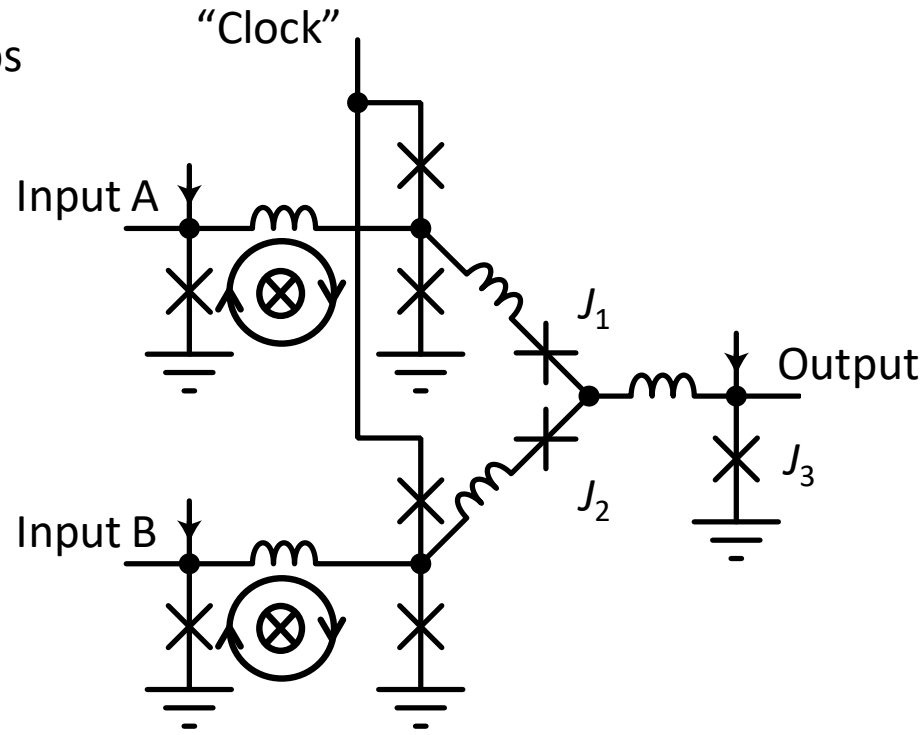


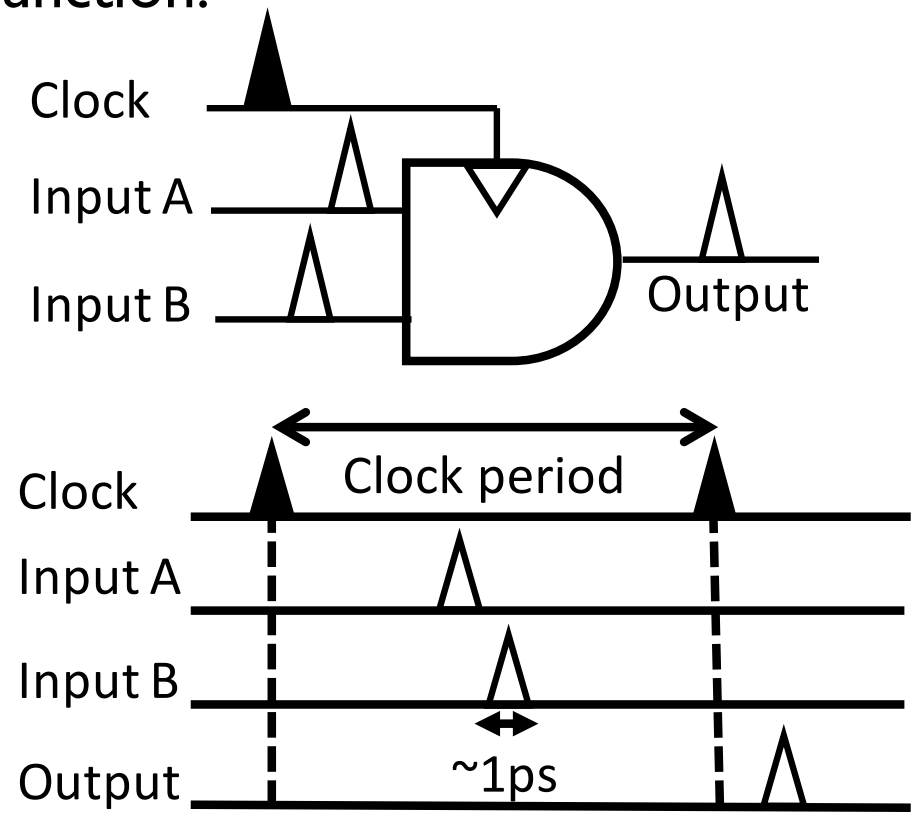
No SFQ stored ➔ "0"

SFQ Logic Gate (AND)

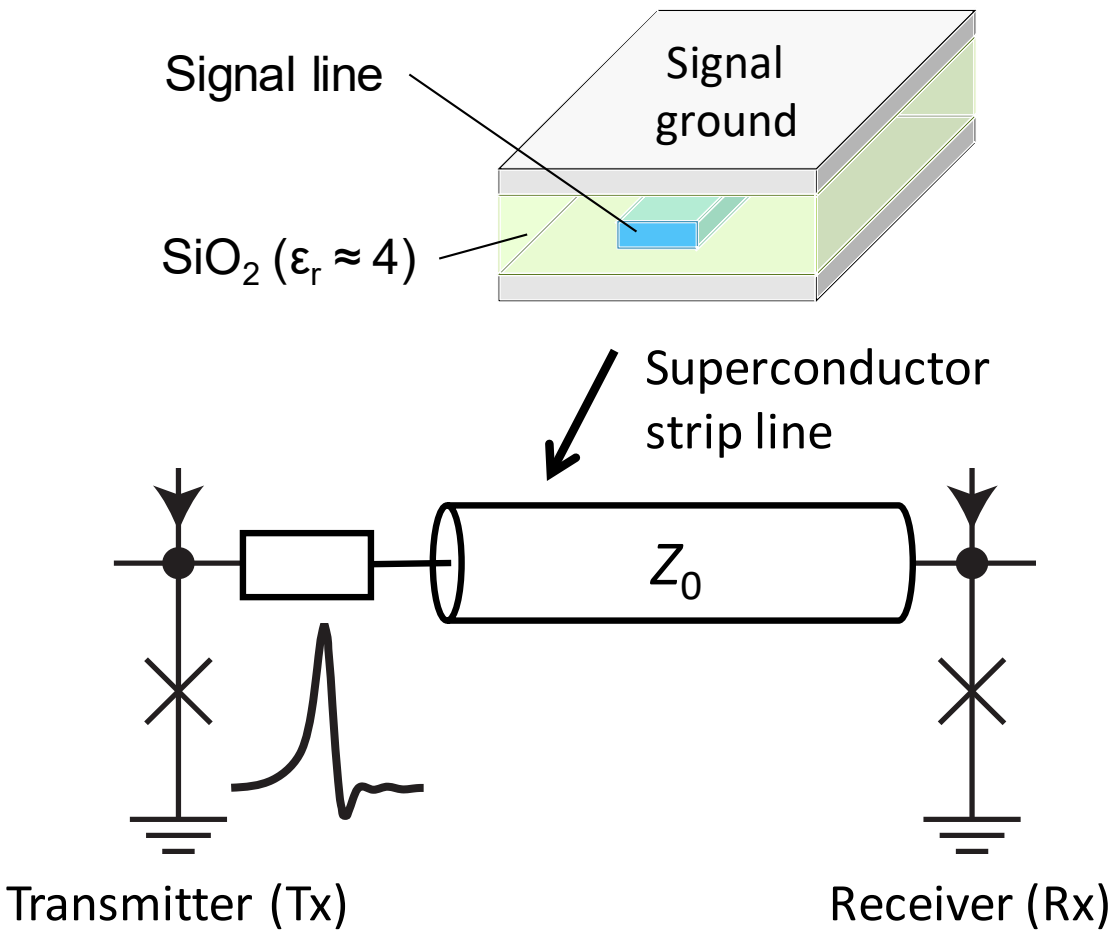
- Use "Clock" as a timing reference for synchronization.
- Every logic gate is clocked gate and has the latch function.



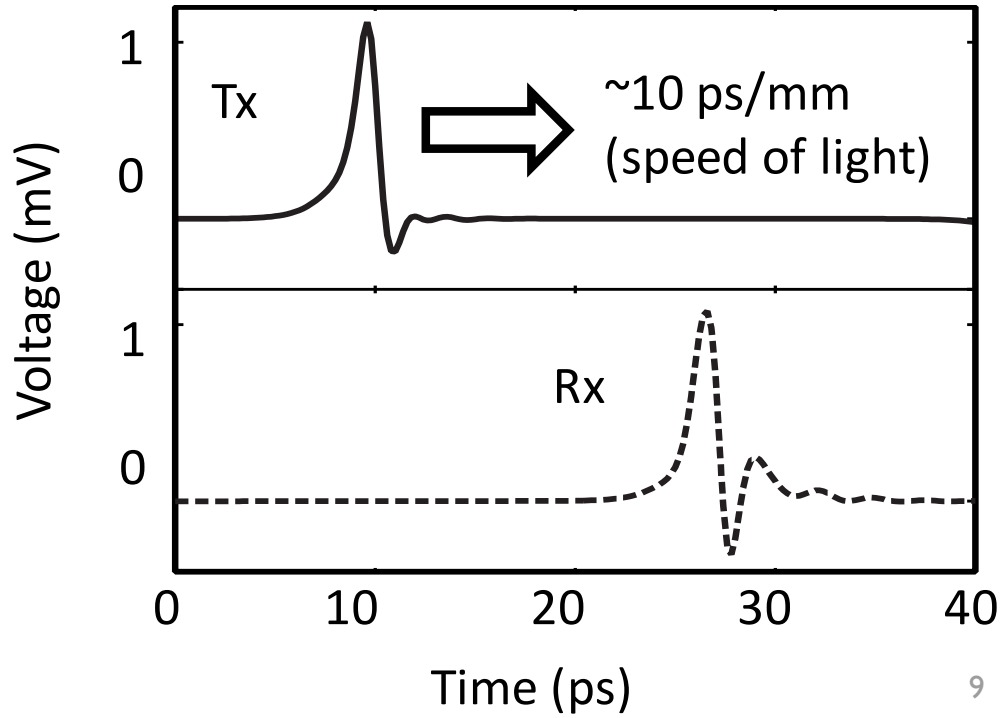




Superconductor Waveguide



- ✓ Release from recharge process
- ✓ Signal propagation at the speed of light
- ✓ Small dispersion
- ✓ Energy-efficient, small-jitter interconnects



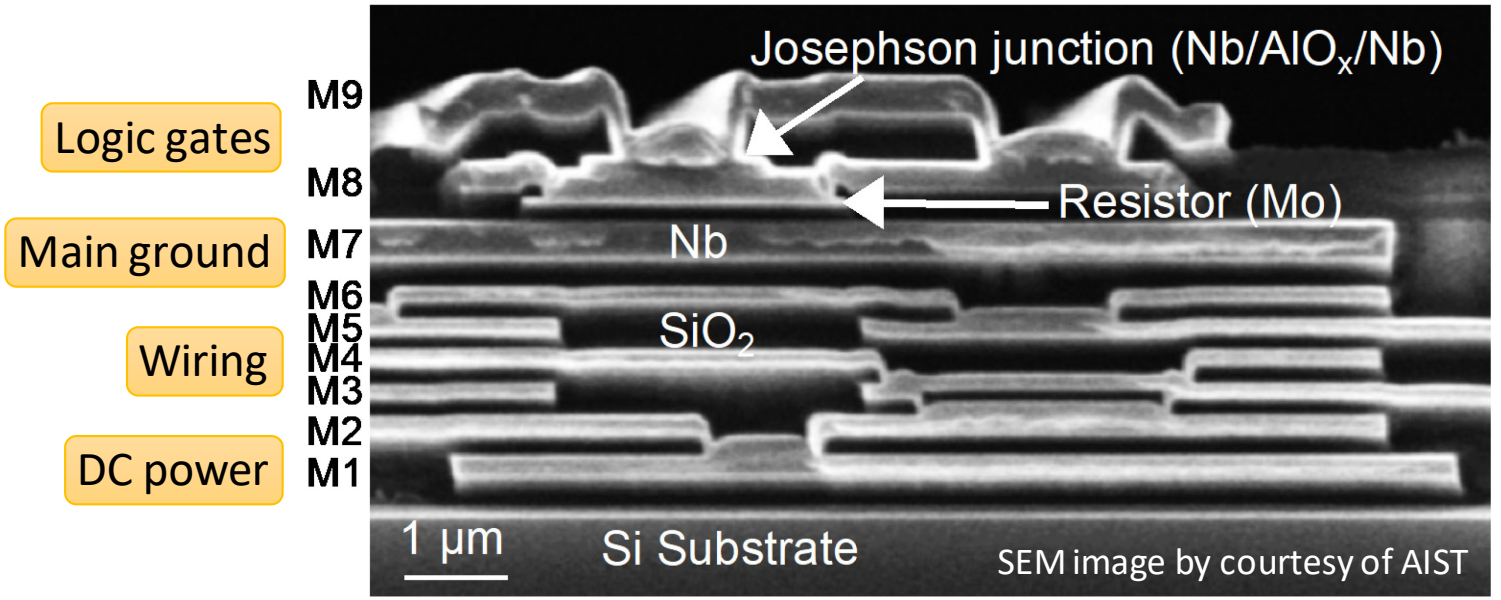
After S. V. Polonsky et al, *IEEE Trans. Appl. Supercond.* **3** (1993) 2598.

Fabrication Process

- 3–10 layer process is under development in Japan, US, and China.

AIST Advanced Process, Japan

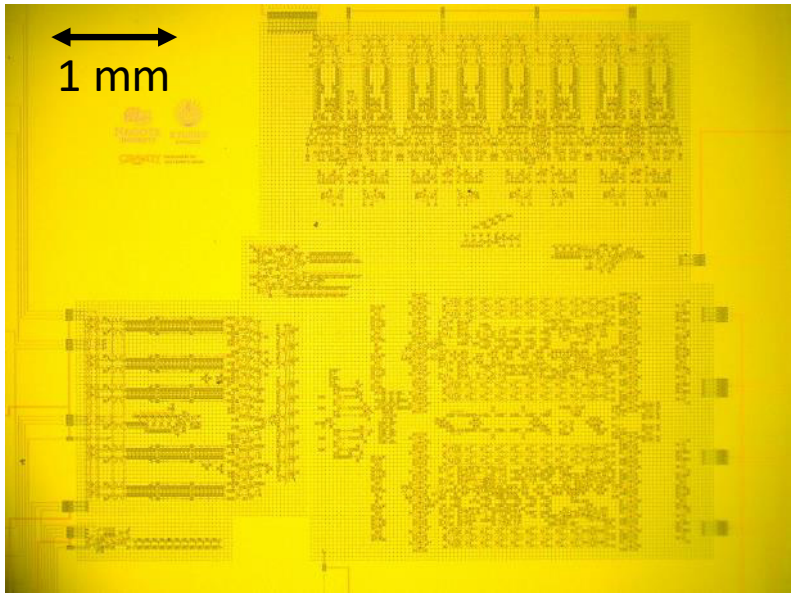
1-μm sq. JJ, Nb 9-layer + Mo



S. Nagasawa et al. *IEICE* **E97-C** (2014) 132-140.

32-GHz, 6.5-mW SFQ MPU

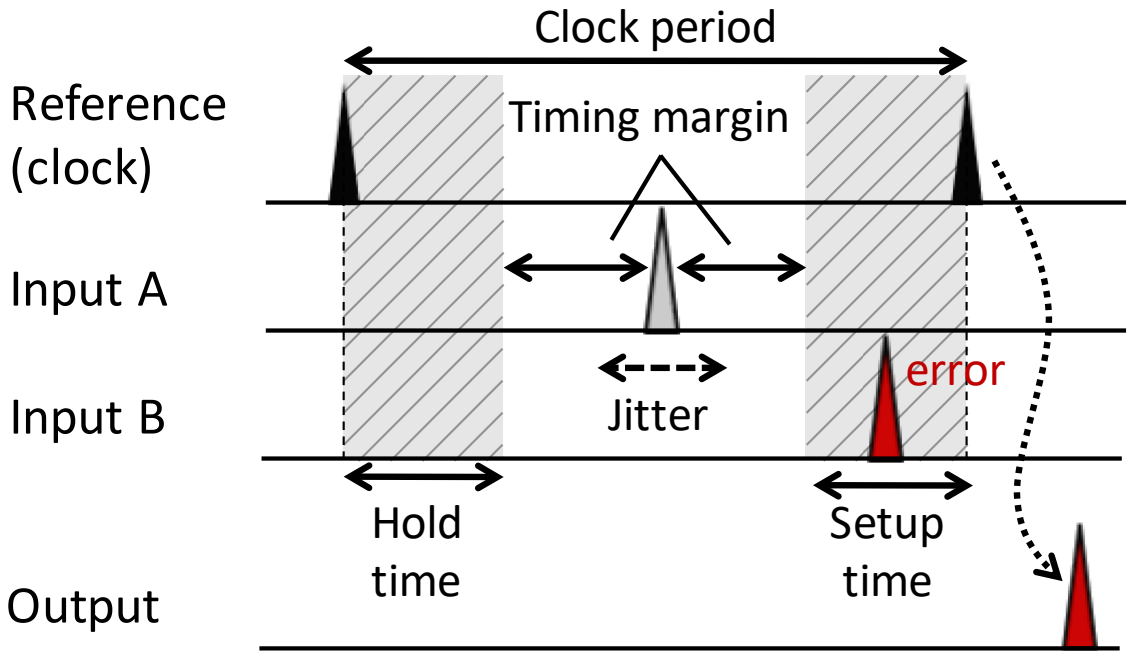
25,403 JJs, 4.1 x 5.3 mm^2

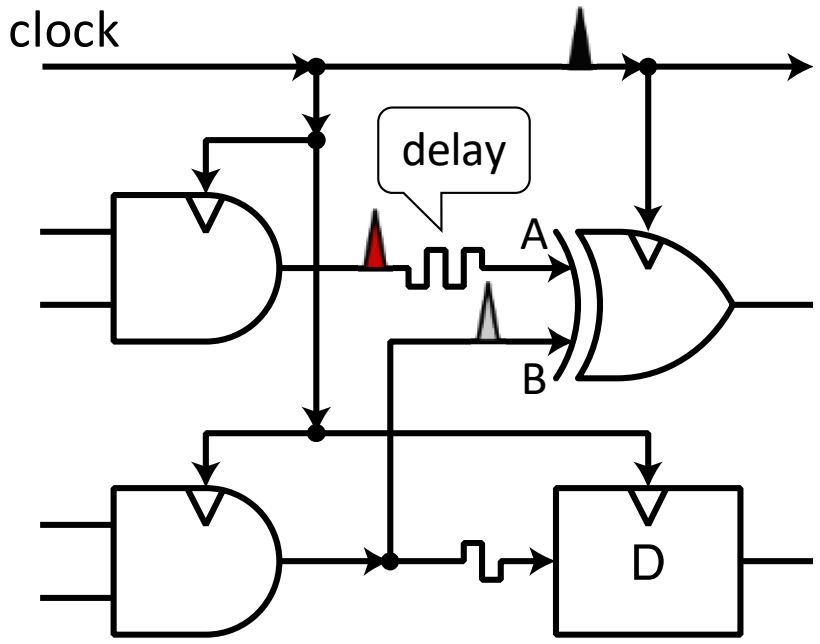


K. Ishida et al., VLSI 2020

Difficulty in Timing Design

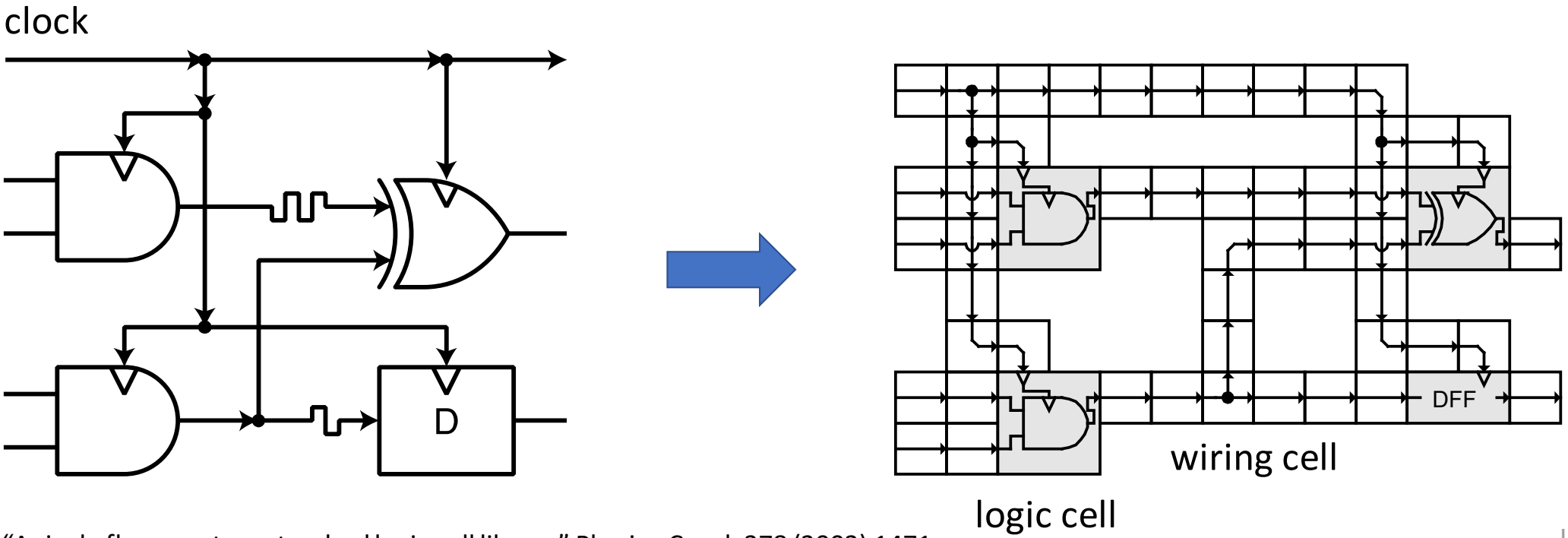
- *At every gate*, we must control timing carefully. We need splitters for fan-outs.
- Timing is fluctuated by bias voltage, thermal noise, fabrication spreads, etc.
- Wiring delay is comparable to gate delay. Speed of light is *only* ~100 μm/ps.





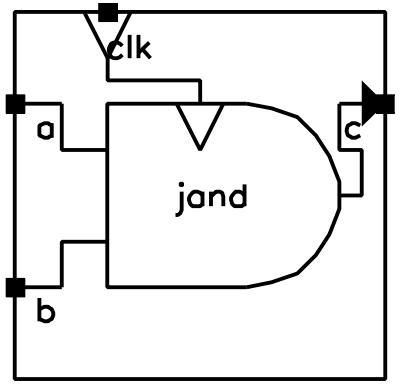
Cell-Based Design & Timing Adjustment

- CONNECT: Standard cell library specialized for SFQ circuits
  - Clocked logic gates and special gates, such as non-destructive readout gate
  - Wiring element (pulse splitters, delay elements, passive transmission lines)

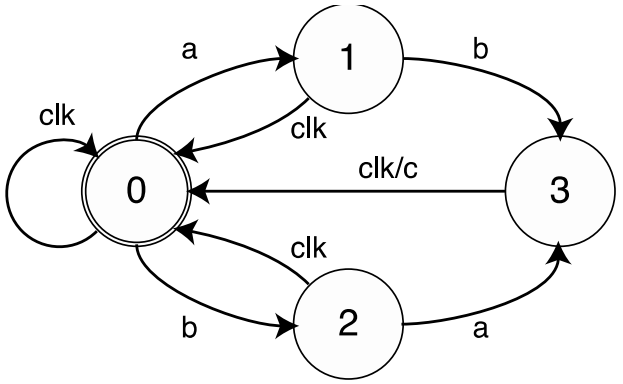


S. Yorozu et al., "A single flux quantum standard logic cell library," Physica C, vol. 378 (2002) 1471.

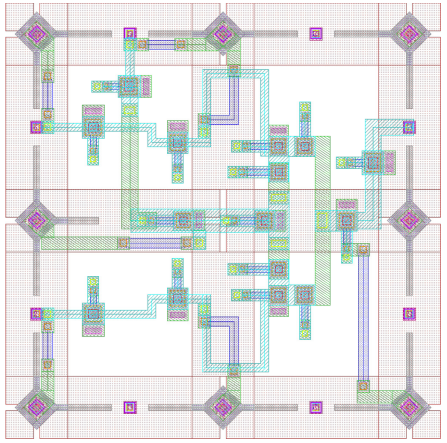
CONNECT Standard Cell



Symbol



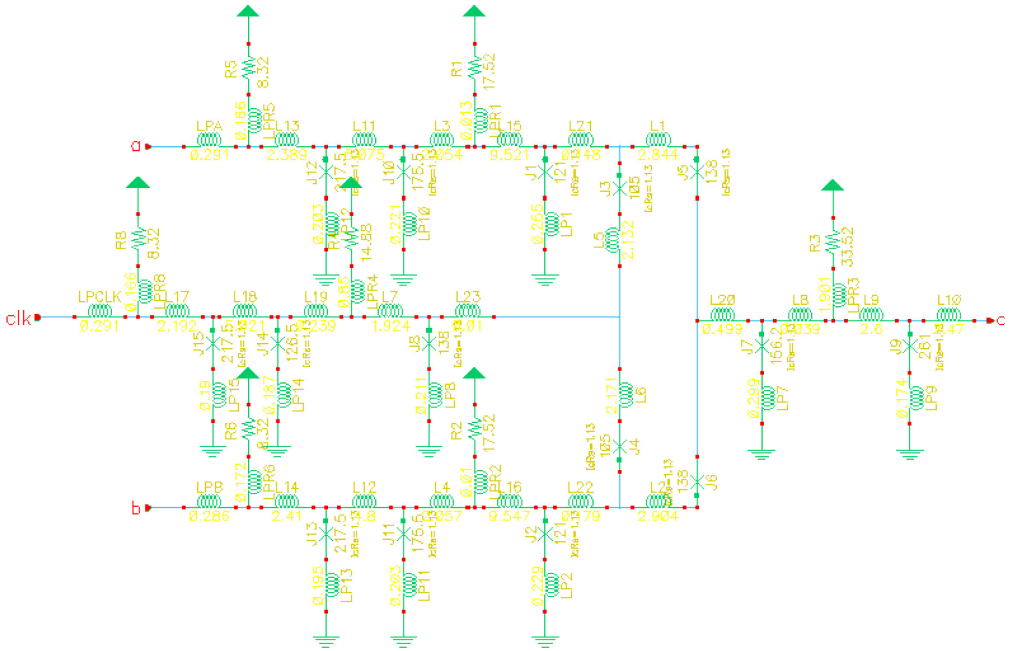
Logical behavior (Verilog)



Mask layout

```
parameter BV70 = 1.75 ;
parameter  CLK__C__1 = 12.1 ;
parameter  CLK__CLK__1 = 9.8 ;
parameter  CLK__A__1 = 4.3 ;
parameter  B__CLK__1 = -2.7 ;
parameter  CLK__B__1 = 4.4 ;
parameter  A__CLK__1 = -2.7 ;
:
```

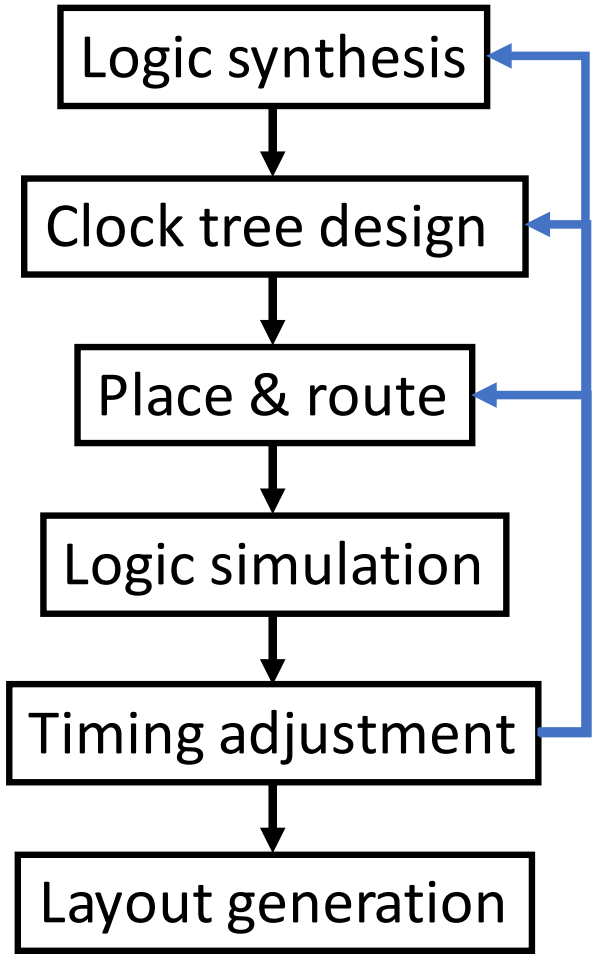
Timing parameters with bias dependence (Verilog)

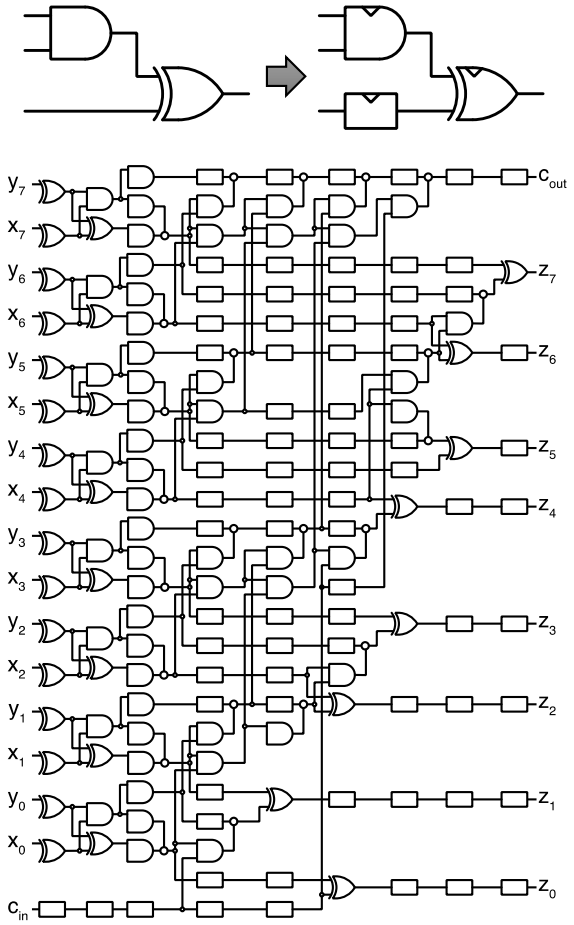


Schematic w/ extracted parameters for SPICE-base simulator

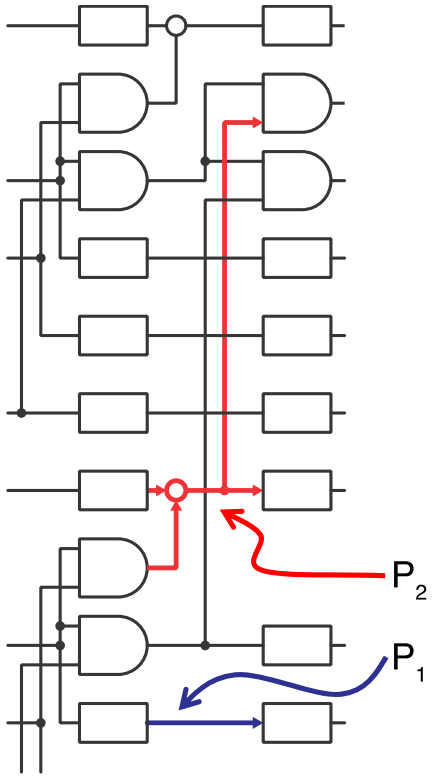
→ You can access area, timing, JJ count, bias current, etc.

Design Flow

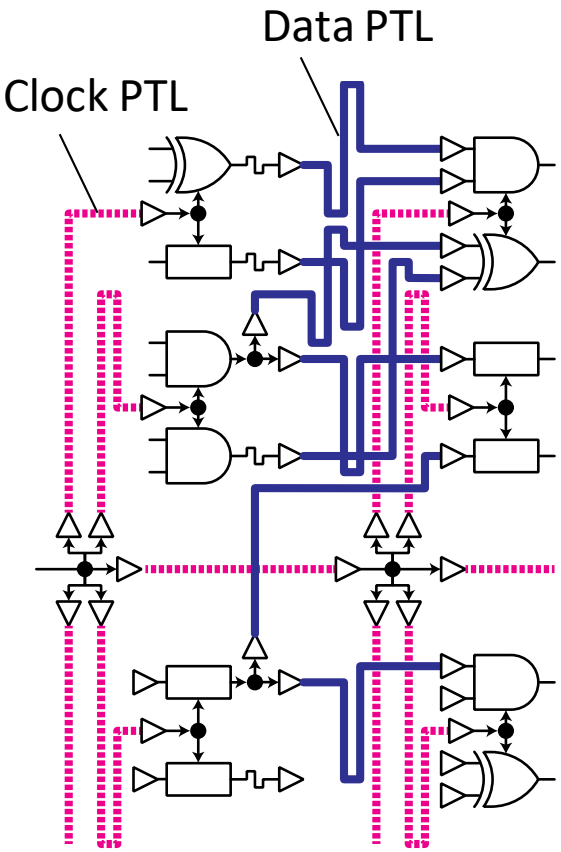




Schematic design

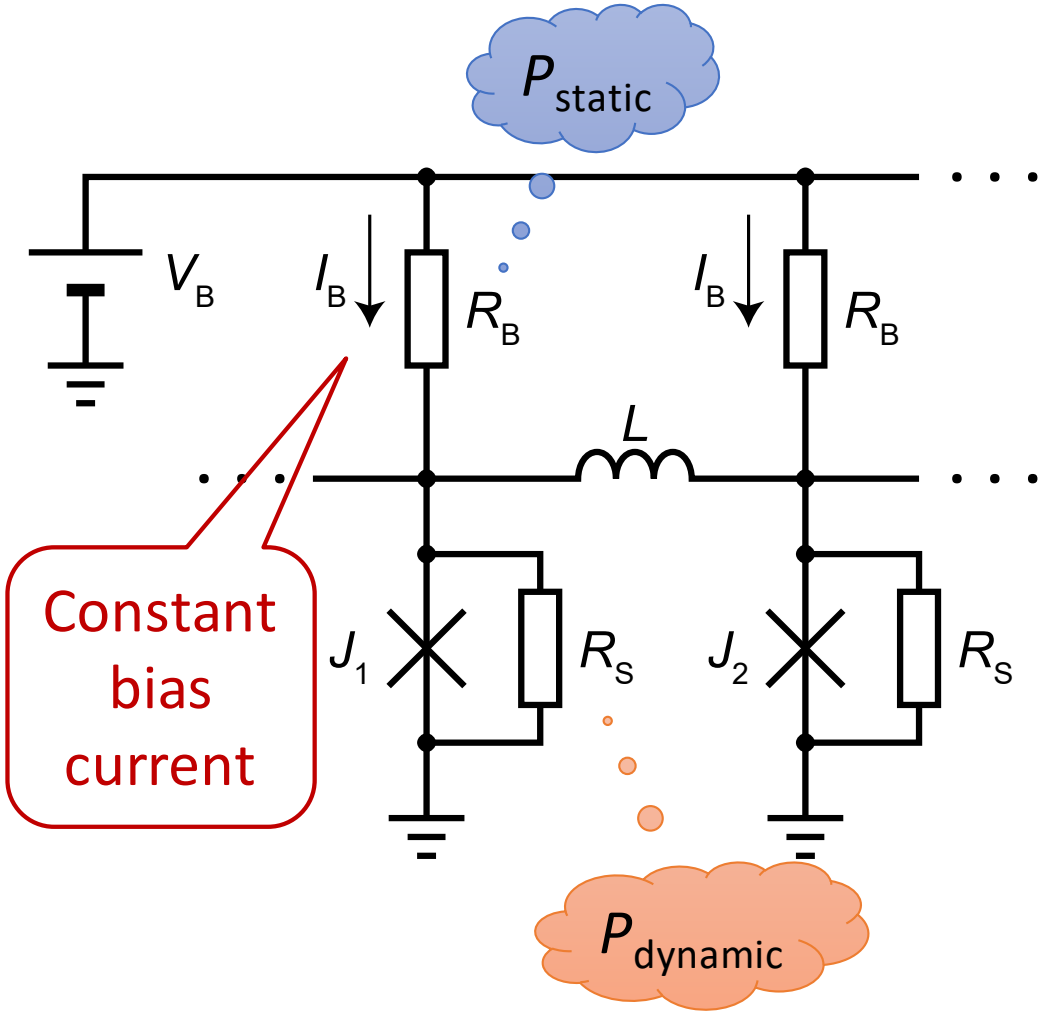


Delay tuning



Place & Route

Power Consumption in SFQ Circuits



Static power (R_B)

$P_{static} = V_B^2/R_B \approx 0.7\, I_c\, V_B$

In typical D flip-flop:

$$\sum \frac{V_B^2}{R_{Bi}} = 1.8\ \mu W$$

Dynamic power (R_S)

$P_{dynamic} = \alpha\, f\, I_c\, \Phi_0$

$$f\Phi_0 \sum \alpha_i I_{ci} = 36\ nW$$

α: switching activity

f: operating frequency

Φ_0: flux quantum (= $h/2e$)

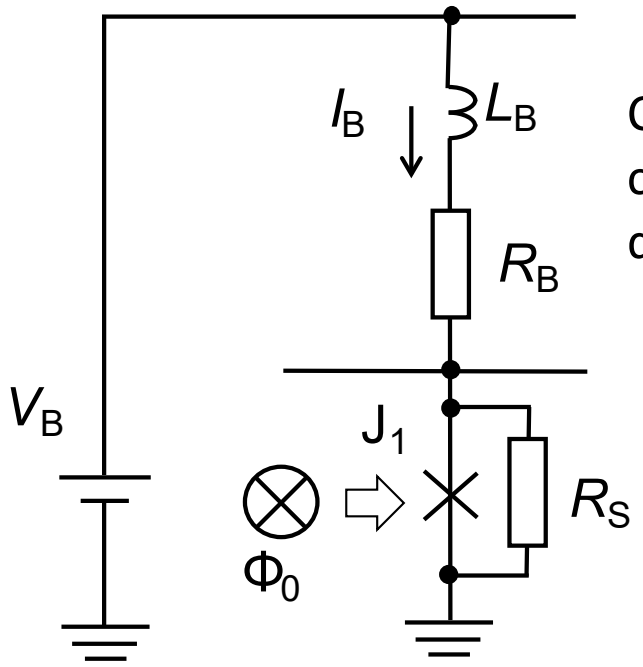
How to Reduce Power?

- **Reduce currents – trade-off to noise tolerance**
  - Use small Josephson junctions (depends on fabrication process)
  - Use π-shifted Josephson junctions "half-flux-quantum logic" [1]
- **Lower voltages**
  - Use small *R* and large *L* "LR-bias SFQ" [2, 3]
  - Drive by low voltages "LV-RSFQ" [4]
  - Use JJs for limiting currents "ERSFQ" [5] / "eSFQ" [6]
- **Others: AC-powered circuits**
  - RQL: reciprocal quantum logic [7]
  - AQFP: adiabatic quantum flux parametron [8]

[1] T. Kamiya *IEICE* **E101-C** (2018). [2] A.V. Rylyakov *IEEE TAS* **7** (1997). [3] N. Yoshikawa *SUST* **12** (1999). [4] M. Tanaka *JJAP* **51** (2012). [5] D.E. Kirichenko *IEEE TAS* **21** (2011). [6] M.H. Volkmann *SUST* **26** (2013). [7] Q.P. Herr *JAP* **109** (2011). [8] Takeuchi *JAP* **115** (2014).

Energy-Efficient SFQ Circuits

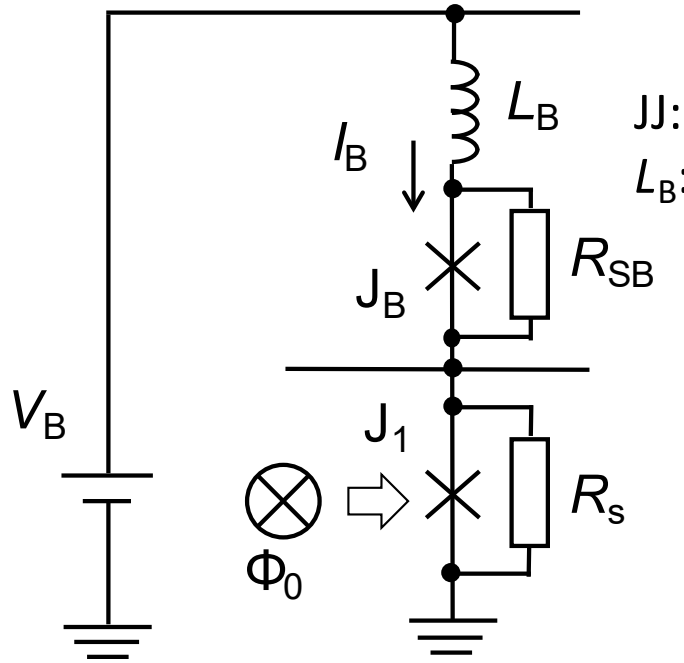
LR-bias/LV-RSFQ



Constant-currents or constant-voltage driving

N. Yoshikawa and Y. Kato *SUST* **12** (1999) 918.
M. Tanaka et al. *JJAP* **51** (2012) 053102

ERSFQ



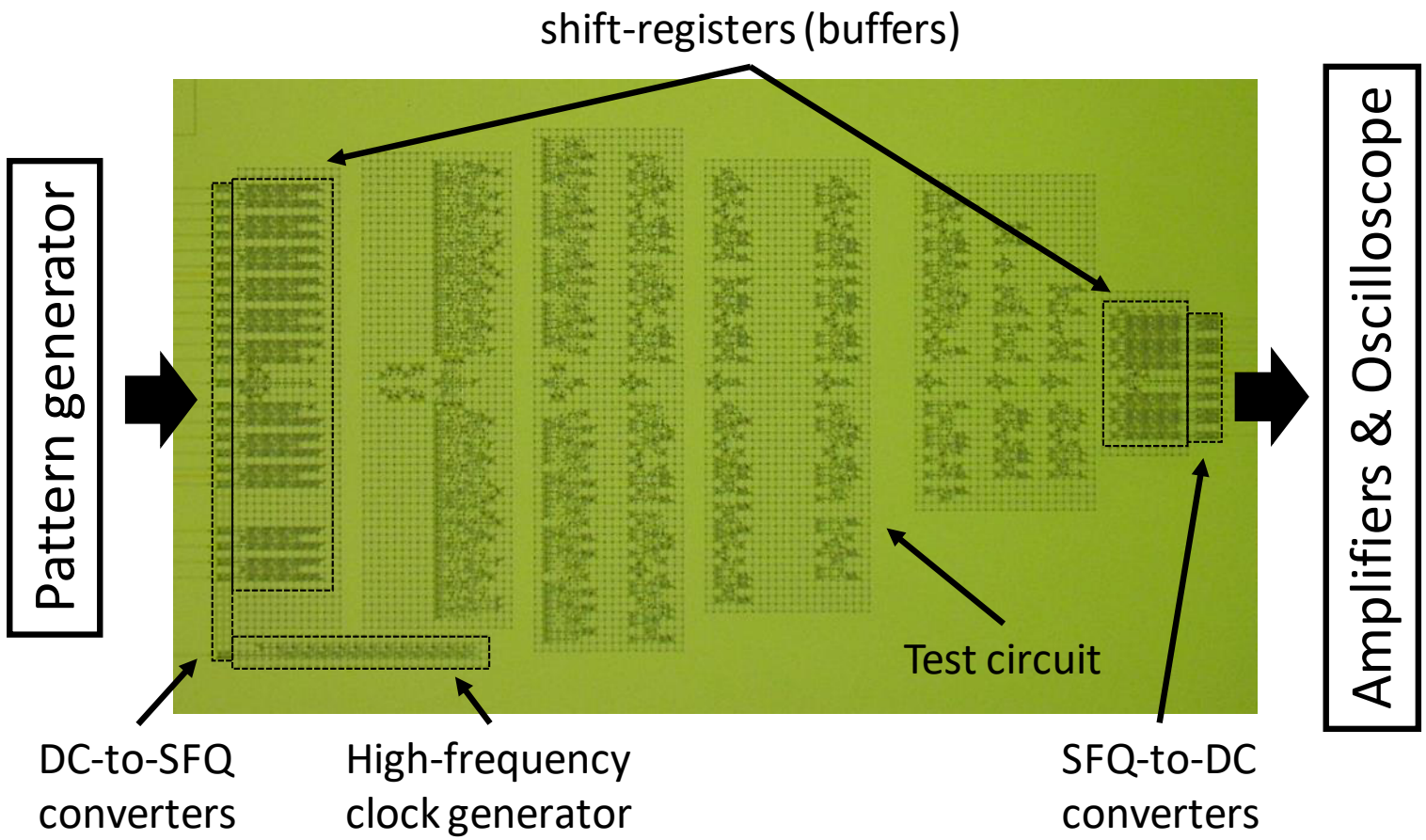
JJ: current limiter
L_B: smoothing

D. E. Kirichenko et al. *IEEE TAS* **21** (2011) 776.

Pros and Cons

	Pros	Cons
RSFQ	• Ultrafast operation beyond 100 GHz • Clocked gates (inherit latch function) • Well-established cell library	• Large static power • Large DC bias currents • Challenging clock distribution
LV-RSFQ	• Similar to RSFQ design • Power reduction up to 1/10	• Static power is still dominant • More complex timing design
ERSFQ	• No static power • Frequency is almost same as RSFQ	• Smaller integration density (doubles JJ count and requires large inductors)
RQL	• No static power • Use of small-amplitude AC currents that also play the role of clock signal	• Difficulty in multi-phase RF design (power splitters, skew control, etc.) • Scalability limit by magnetic couplings
AQFP	• Ultralow energy operation near physical limits (w/ use of reversible gates) • Small-amplitude AC driven	• Limited frequency due to adiabatic operation • Limited wiring length (PTLs not available) • Scalability limit by magnetic couplings

Chip Measurement



Impact of Practical Designs on Area/Power/Performance

What happens on SFQ circuit in practical

- Design policy **for high perf. and process variation tolerance**
  - Gate-level pipelining: explain later from the architectural viewpoint
    - Feeds clock pulse to each gate to design as a synchronous logic
  - Equivalent length wiring: next slide
    - Conservative approach but important for successful demonstration currently
- Area and power estimation model: mainly counting #JJs

Area

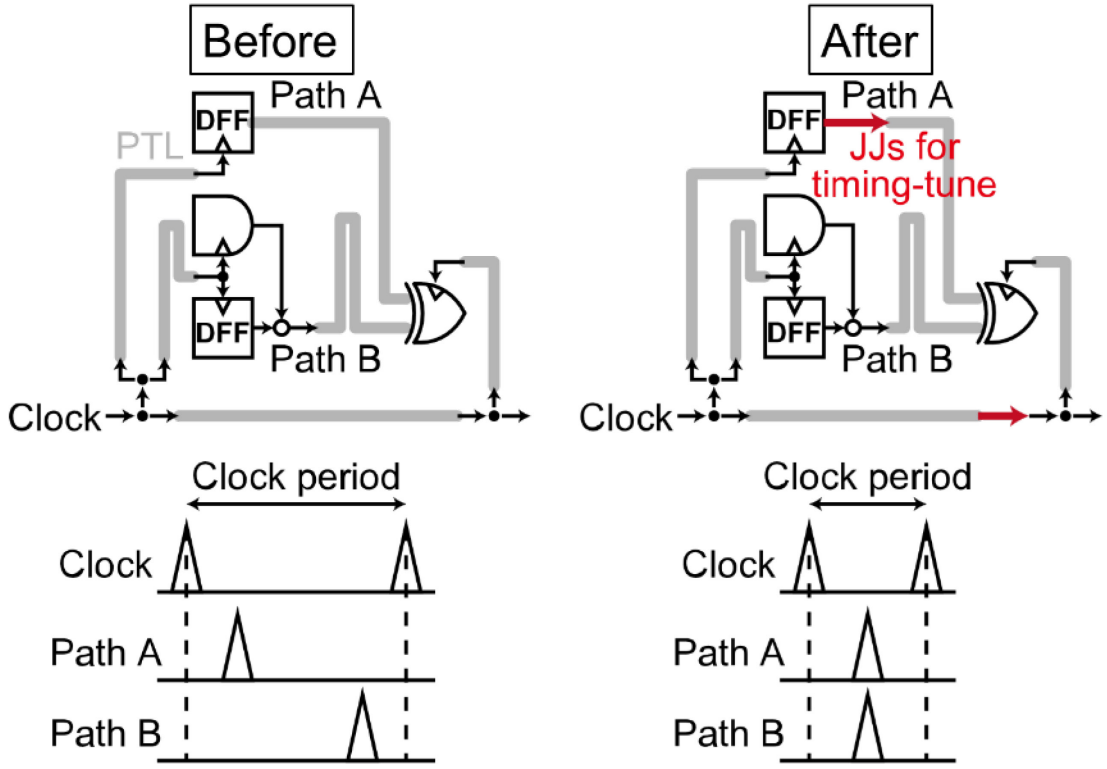
o Each gate consists of a few JJs (3-4)
o Wires
 v PTL: clock tree and data path
 v JTL (=JJ): timing adjustment

Power

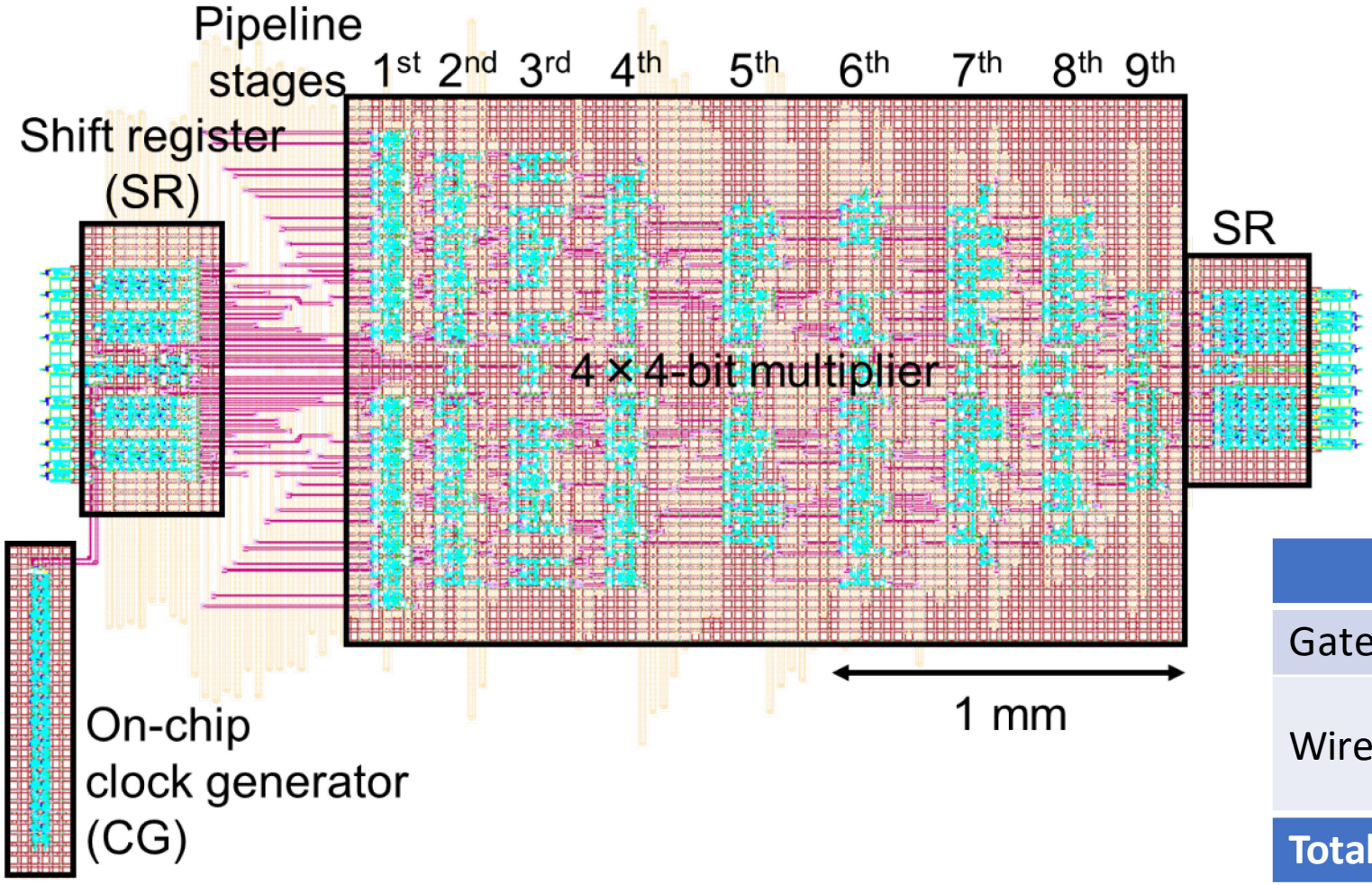
o Most JJs switch every cycle due to clock pulse
o Estimation methodology is common among SFQ device family

Impact of equivalent length (timing) wiring

- Insert JTL (work as delay cell) to the shorter path (A)
  - Currently the most time consuming part of our custom layout by hand
- Effect and impact
  - contributes to achieve higher frequency by timing adjustment of pulse arrival from path (A) and (B)
  - Shortcoming is increase in #JJs, which impacts on both area and power consumption



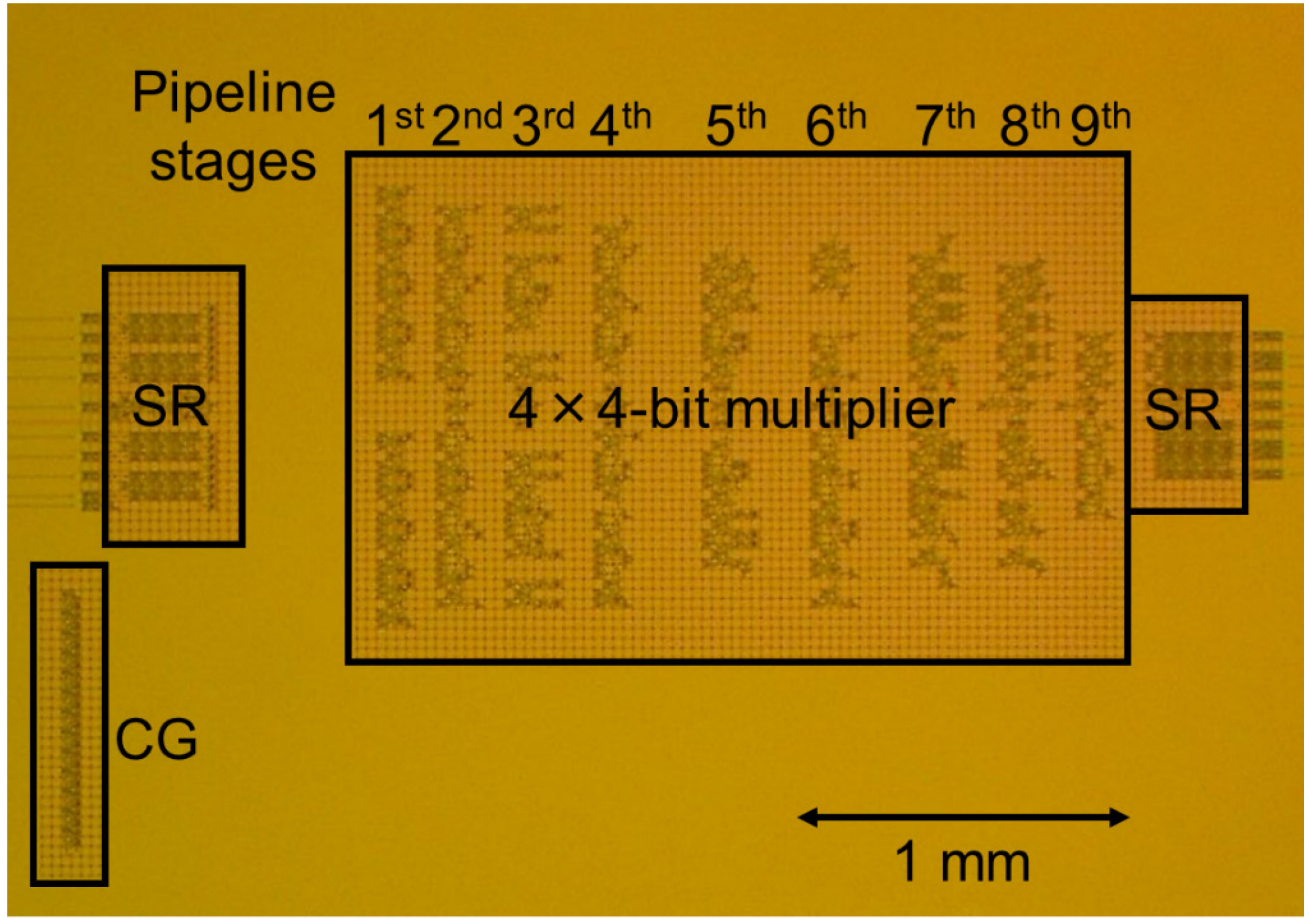
The case for 4 × 4-bit LV-RSFQ Multiplier

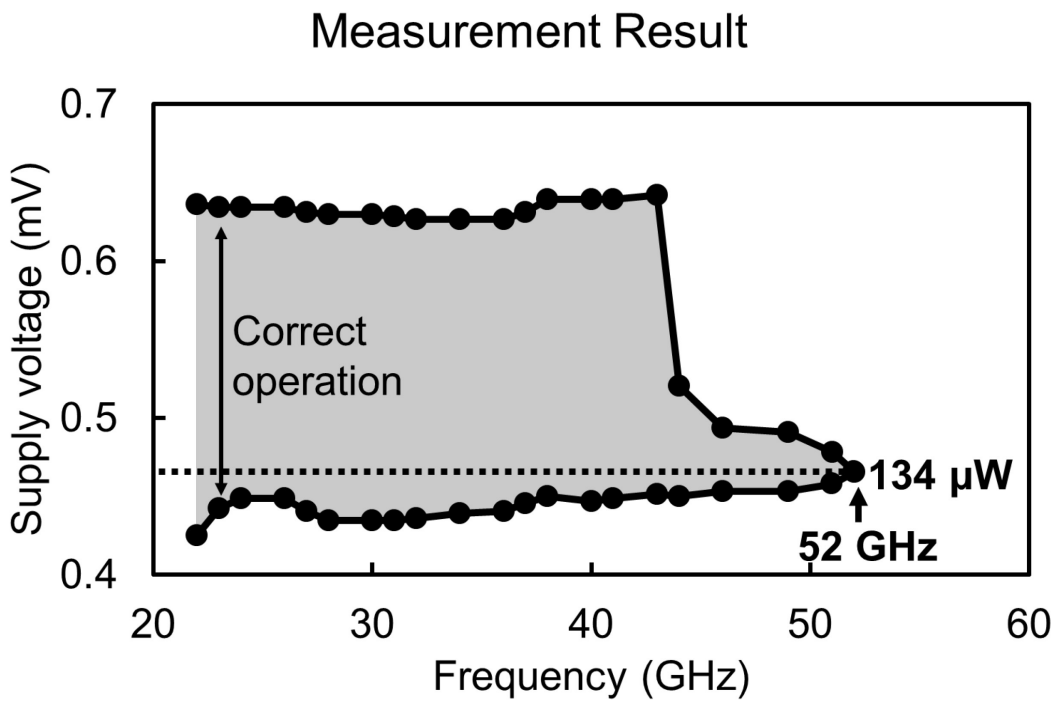


		#JJs
Gates		1185 (38.6%)
Wires	in total	1872 (61.4%)
	for timing tuning	464 (15.2%)
Total		**3067**

[I. Nagaoka+, IEEE Trans. on Appl. Supercond.] "Demonstration of a 52-GHz Bit-Parallel Multiplier Using Low-Voltage Rapid Single-Flux-Quantum Logic"

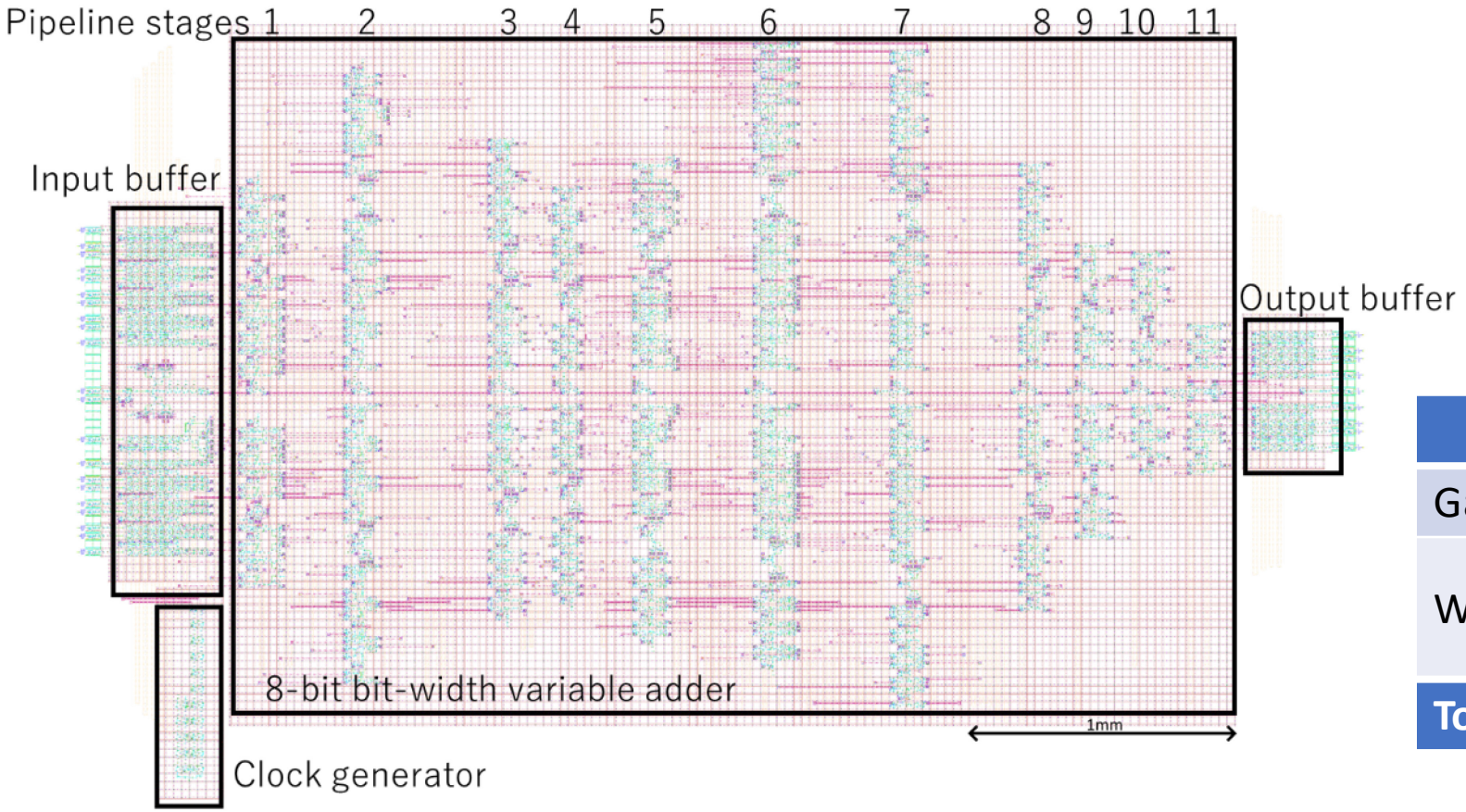
The case for 4 × 4-bit LV-RSFQ Multiplier





[I. Nagaoka+, IEEE Trans. on Appl. Supercond.] "Demonstration of a 52-GHz Bit-Parallel Multiplier Using Low-Voltage Rapid Single-Flux-Quantum Logic"

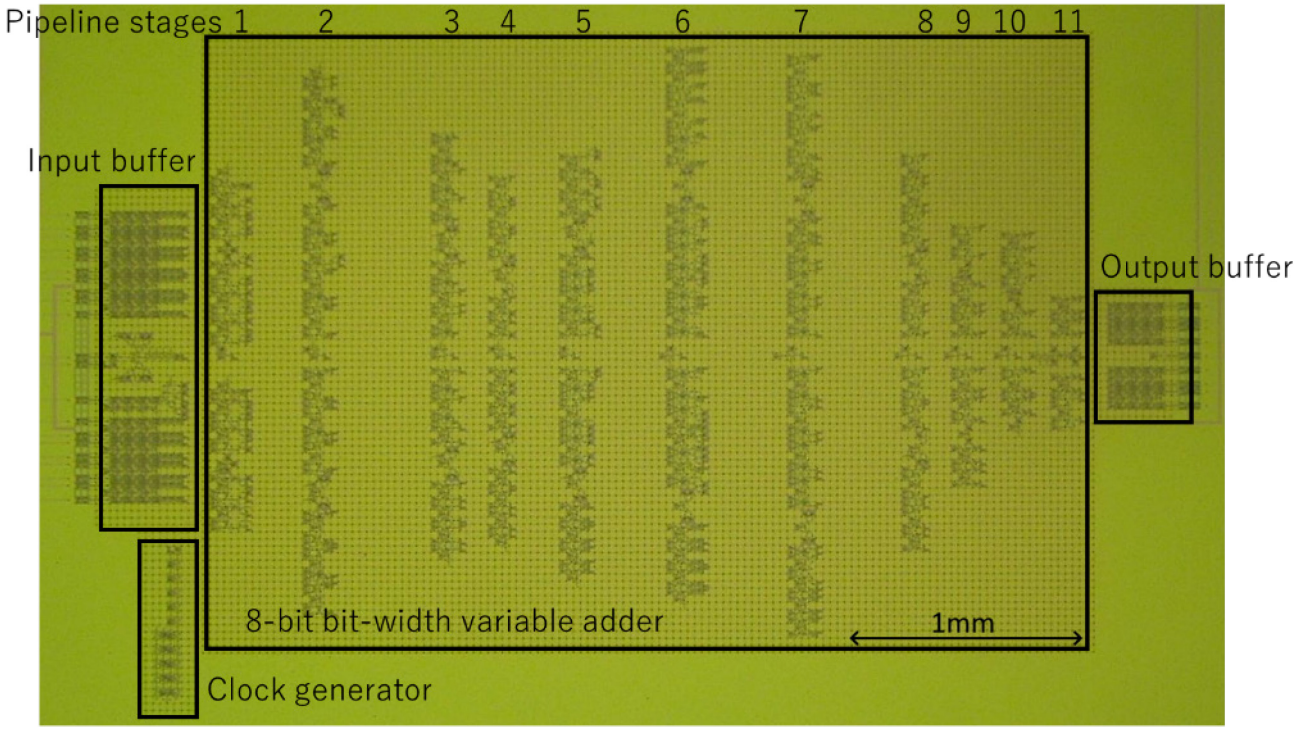
The case for Bit-width Variable Adder

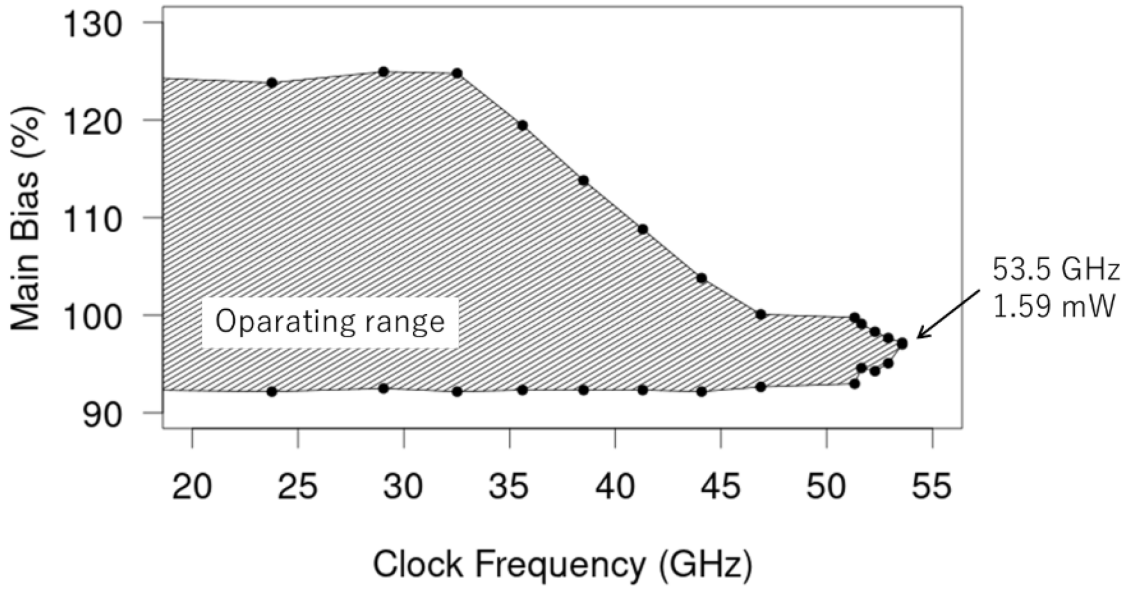


		#JJs
Gates		3116 (38.5%)
Wires	in total	4975 (61.5%)
	for timing tuning	974 (12.0%)
Total		**8091**

[I. Ishikawa+, ISCAS22] "Design of Variable Bit-Width Arithmetic Unit Using Single Flux Quantum Device"

The case for Bit-width Variable Adder





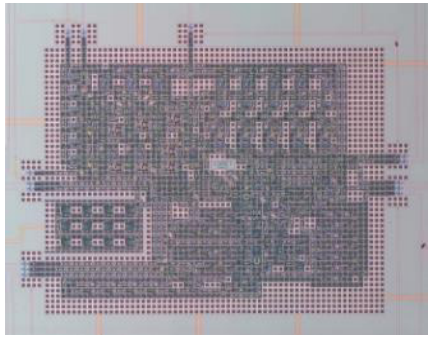
[I. Ishikawa+, ISCAS22] "Design of Variable Bit-Width Arithmetic Unit Using Single Flux Quantum Device"

Future directions of SFQ circuit design

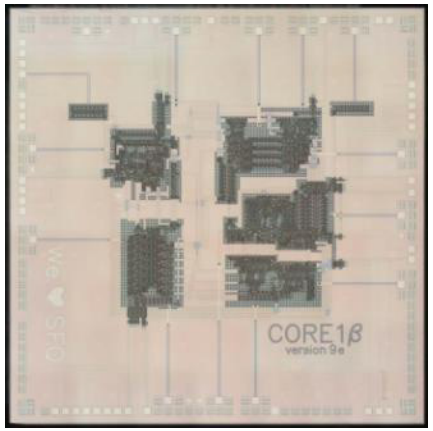
- Fidelity improvement of fabrication
  - Relaxes the conservativeness (i.e., timing margin and adjustment)
  - Fabrications for production will make it mature
    (but much investment will be necessary actually)
- Computer assisted layout
  - Variation-aware design verification
  - Reduce human effort to wiring both clock tree and data path
- Asynchronous logic (similar to combinational logic in CMOS)
  - Potential for higher performance and #JJs reduction
  - Partly exploited already (OR operation by "merger")
  - Hard to verify the design (as for an asynchronous CMOS logic)

Architectural Challenge on SFQ-based Computing

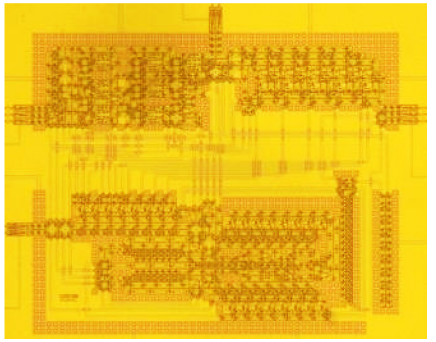
OLD Challenge in RSFQ-based Computer #1
~ SFQ microprocessor designs @ 2003-2016 ~



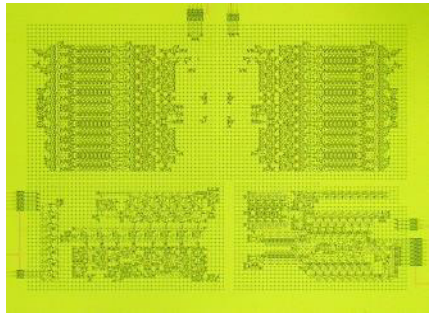
CORE1α v5 (2003)
4999 JJs, 15 GHz
167 MIPS, 1.6 mW



CORE1β v9e (2006)
10955 JJs, 25 GHz
1400 MOPS, 3.3 mW

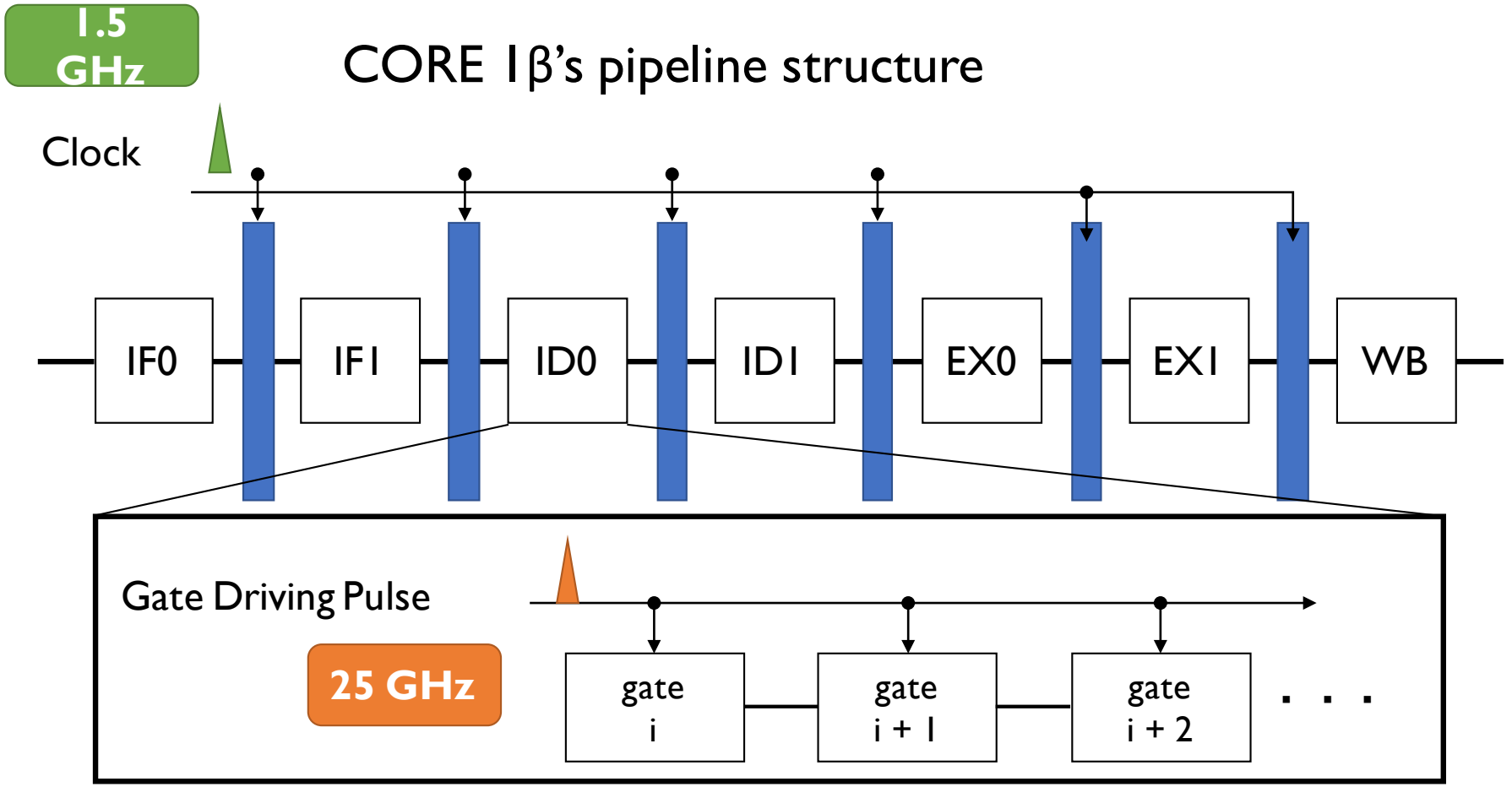


CORE100 (2015)
3073 JJs, 100 GHz
800 MIPS, 1.0 mW



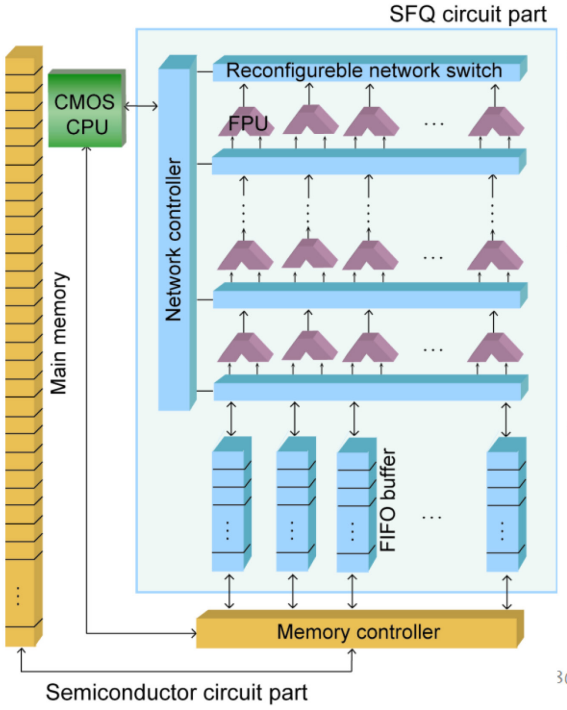
CORE e2 v5h (2016)
10603 JJs, 50 GHz
333 MIPS, 2.5 mW

OLD Challenge in RSFQ-based Computer #1
~ SFQ microprocessor designs @ 2003-2016 ~



OLD Challenge in RSFQ-based Computer #2
~ SFQ Reconfigurable Data-Path @ 2006-2012 ~

Large-Scale Reconfigurable Data-Path for SFQ : Architecture Level



- 1K FPUs operate at 80 GHz
- Re-configurable operand network
- Much simple organization for SFQ design (No feedback loops)
- Make a good balance between "Parallel Exe. Vs. Sequential Exe."

How To Exploit A Number of FPUs: Algorithm Level

$$\frac{2\pi^{5/2}\exp\left(-\frac{ab}{a+b}(\mathbf{A}-\mathbf{B})^2\right)\exp\left(-\frac{cd}{c+d}(\mathbf{C}-\mathbf{D})^2\right)}{(a+b)(c+d)\sqrt{a+b+c+d}}F_m(T)$$

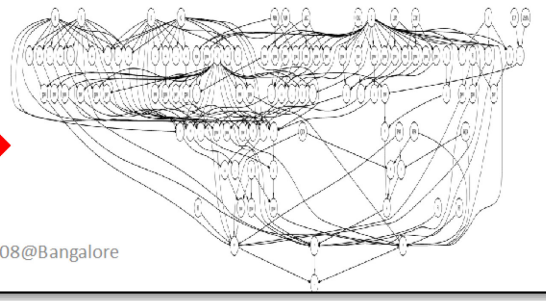
Computation of molecular orbital

while (I < 1000):

tei(4,4,4,4)=(((3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*f(0,t))/(p**2*q**2)+(4*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*PQx*(QCx+QDx)*(3+2*q*QCx*QDx)*f(1,t))/(p*q*(p+q))(4*(PAx+PBx)*(3+2*p*PAx*PBx)*PQx*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*f(1,t))/(p*q*(p+q))(8*(PAx+PBx)*(3+2*p*PAx*PBx)*(QCx+QDx)*(3+2*q*QCx*QDx)*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p*q*(p+q)**2)+(2*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3+q*(QCx**2+4*QCx*QDx+QDx**2)))*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p*q**2*(p+q)**2)+(2*(3+p*(PAx**2+4*PAx*PBx+PBx**2)))*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p**2*q*(p+q)**2)+(4*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*PQx*(QCx+QDx)*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(q*(p+q)**3)\+(8*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*PQx*(QCx+QDx)*(3+2*q*QCx*QDx)*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(p*(p+q)**3)(8*(PAx+PBx)*(3+2*p*PAx*PBx)*PQx*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(q*(p+q)**3)(4*(PAx+PBx)*PQx*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(p*(p+q)**3)+((3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t))))/(q**2*(p+q)**4)(8*(PAx+PBx)*(3+2*p*PAx*PBx)*(QCx+QDx)*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t))))/(q*(p+q)**4)(8*(PAx+PBx)*(QCx+QDx)*(3+2*q*QCx*QDx)*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t))))/(p*(p+q)**4)+(4*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t))))/(p*q*(p+q)**4)+(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t))))/(p*q*(p+q)**4)+(8*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*PQx*(QCx+QDx)*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t))))/(p*q*(p+q)**5)+(8*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*PQx*(QCx+QDx)*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t))))/(p*q*(p+q)**5)+(4*PQx*q*(QCx+QDx)*(3+2*q*QCx*QDx)*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t))))/(p*(p+q)**5)(8*(PAx+PBx)*PQx*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t)))/(p*q*(p+q)**5)(8*p**3*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))+8*p**3*PQx**6*q**3*f(6,t)))/(q*(p+q)**6)+(2*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(15*(p+q)**3*f(3,t)30*p*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))+8*p**3*PQx**6*q**3*f(6,t)))/(p*q)**6)

→ 787 MUL, 261 ADD, 69 FUNC

I ++:

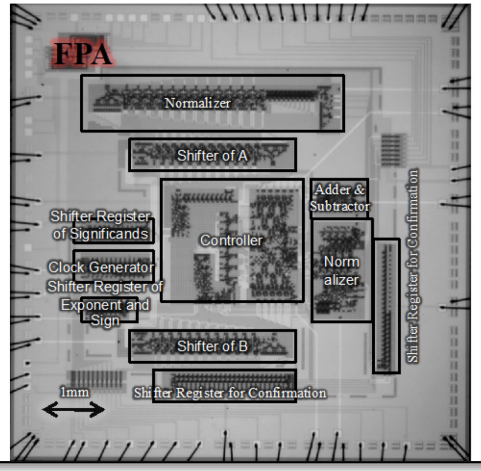


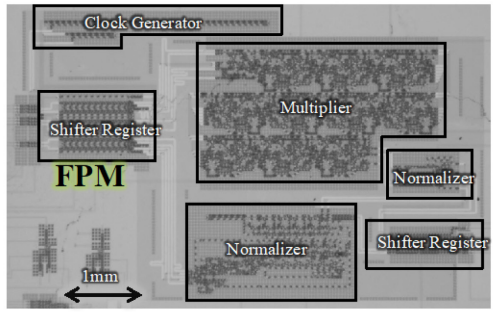
OLD Challenge in RSFQ-based Computer #2
~ SFQ Reconfigurable Data-Path @ 2006-2012 ~

Summary of Half-Precision FPUs

	Floating Point Adder	Floating Point Multiplier
# of JJs	11700	11044
Size (mm^2)	6.76 x 4.96	6.22 x 3.78
Minimum interval (clocks)	12 ($n_f + 1$)	
Latency (clocks)	23 ($2 n_f + 1$)	

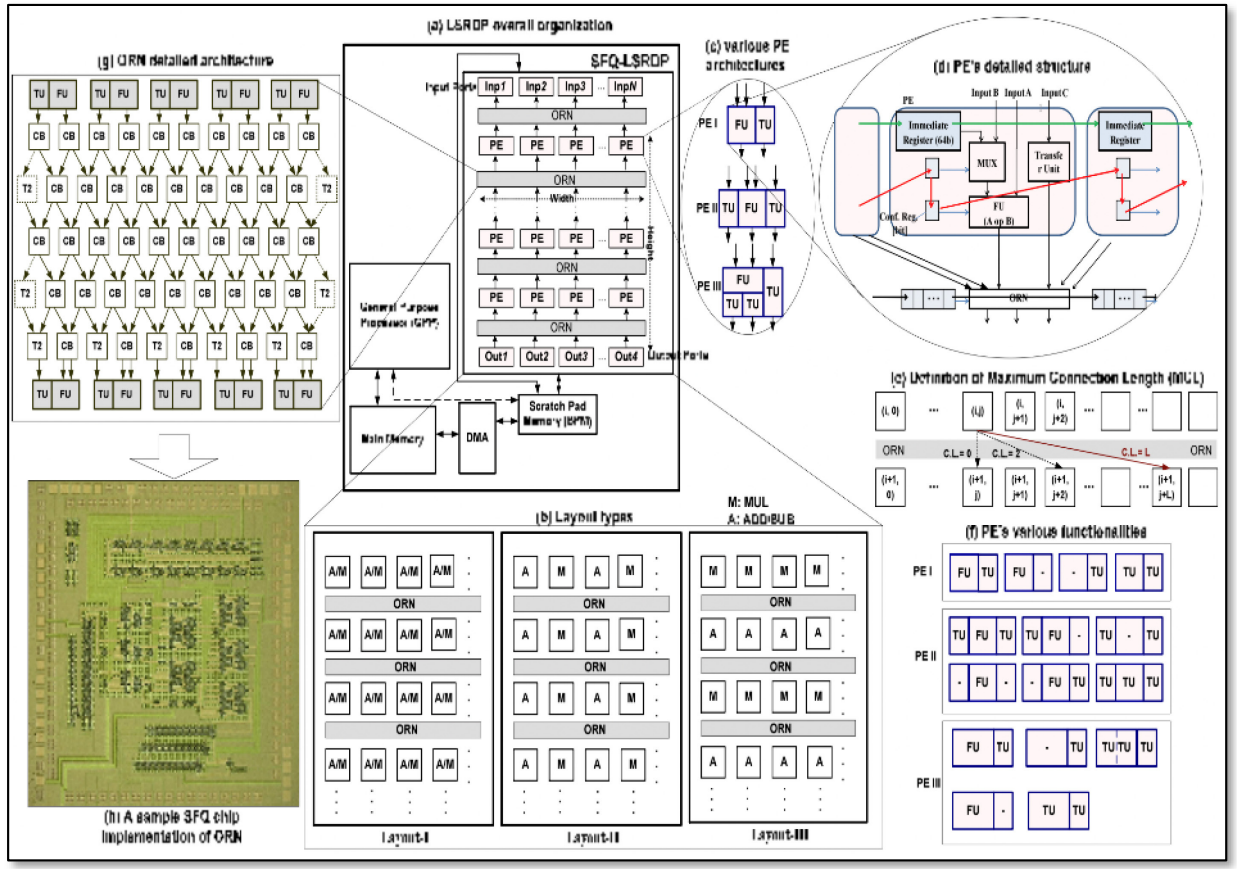
n_f : bit length of fraction part





CONNECT
cooperated with SRL, NiCT, NU & YNU

N. Yoshikawa, "RSFQ Project in Japan," 5th FLUXONICS RSFQ workshop, 2008.

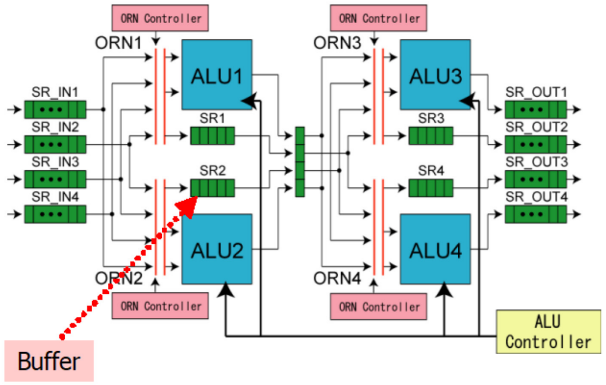


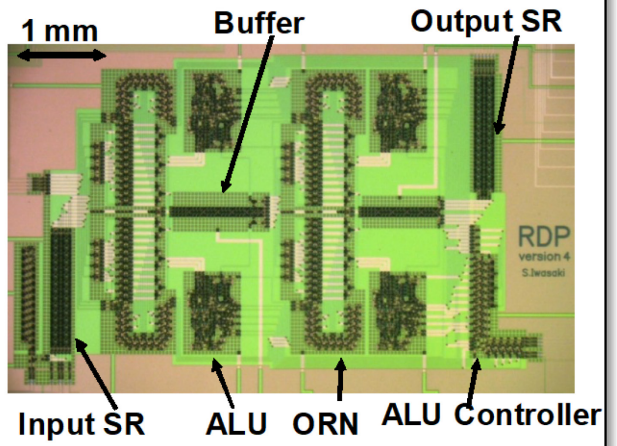
F. Mehdipour et al., "Mapping scientific applications on a large-scale data-path accelerator implemented by single-flux quantum (SFQ) circuits," DATE 2010.

OLD Challenge in RSFQ-based Computer #2
~ SFQ Reconfigurable Data-Path @ 2006-2012 ~

Design of 2x2 SFQ-RDP

- 11 pipeline stages
- Designed frequency : 25 GHz
- InSR & OutSR length : 16-bits
- Data length: 7-bits
- Bias current: 1.27 A
- Circuit area : 5.90 x 3.68 mm^2
- 10839 JJs





	SFQ-RDP
Application	HPC (MO)
Data Reuse (on-chip)	Low
Operation	Bit-Serial FP
On-chip network	Complex & Frexible
On-chip memory	Simple input/output buf
Optimization mainly focused	DFG mapping and routing

Panel Discussion @ ISLPED 2008

What we learned…

+ Stream processing sounds suitable for SFQ logics (no feedback loops)

- Bit-Serial designs significantly degrade the computation performance
- Memory wall problem becomes critical
- Complex on-chip communications consume a lot of JJs

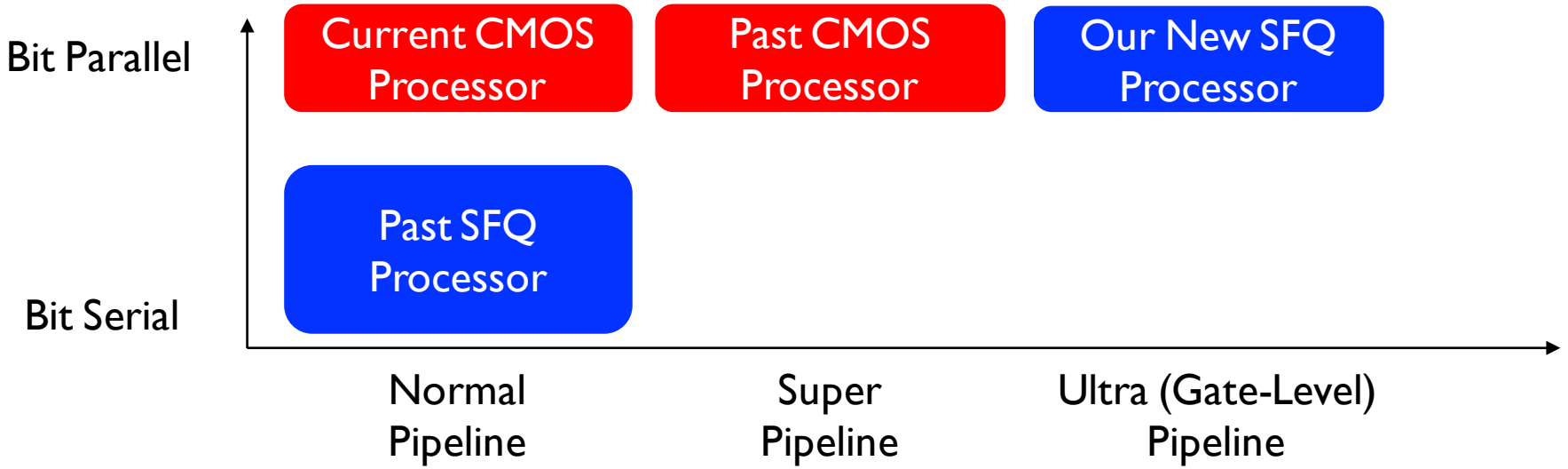
Revisiting Microarchitecture for RSFQ

Pitfall

Bit-serial operation is suitable for RSFQ designs!

Our Approach

Bit-parallel operation + Gate-level deep pipelining

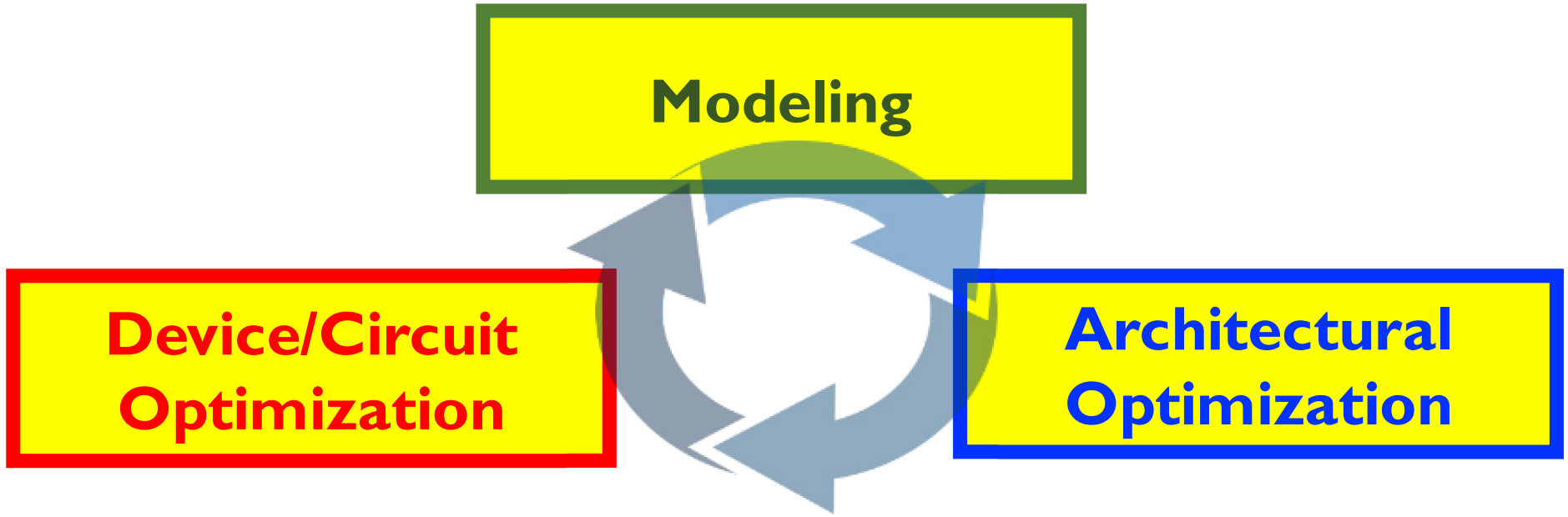


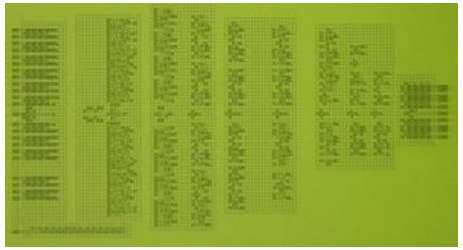
[Our starting point]

“Hey Koji, you should chill your head before dipping your chip in liquid helium at 4 kelvins!”

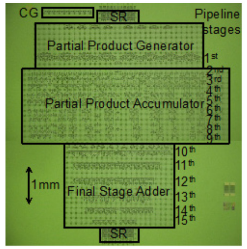
RECENT Challenges

Cross-Layer Interaction

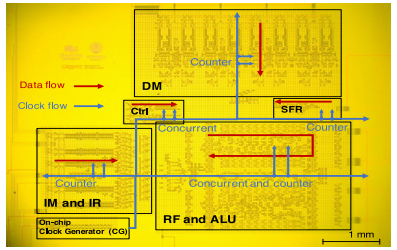




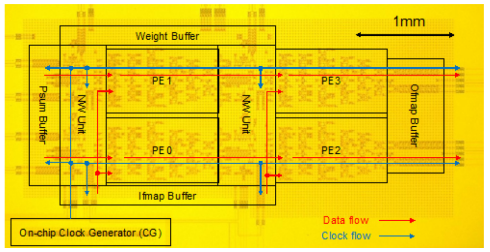
56GHz 1.6mW ALU
ISLPED'17 Design Contest
Honorable Mention



48GHz 5.6mW Multiplier
ISSCC'19
SilkRoad Award

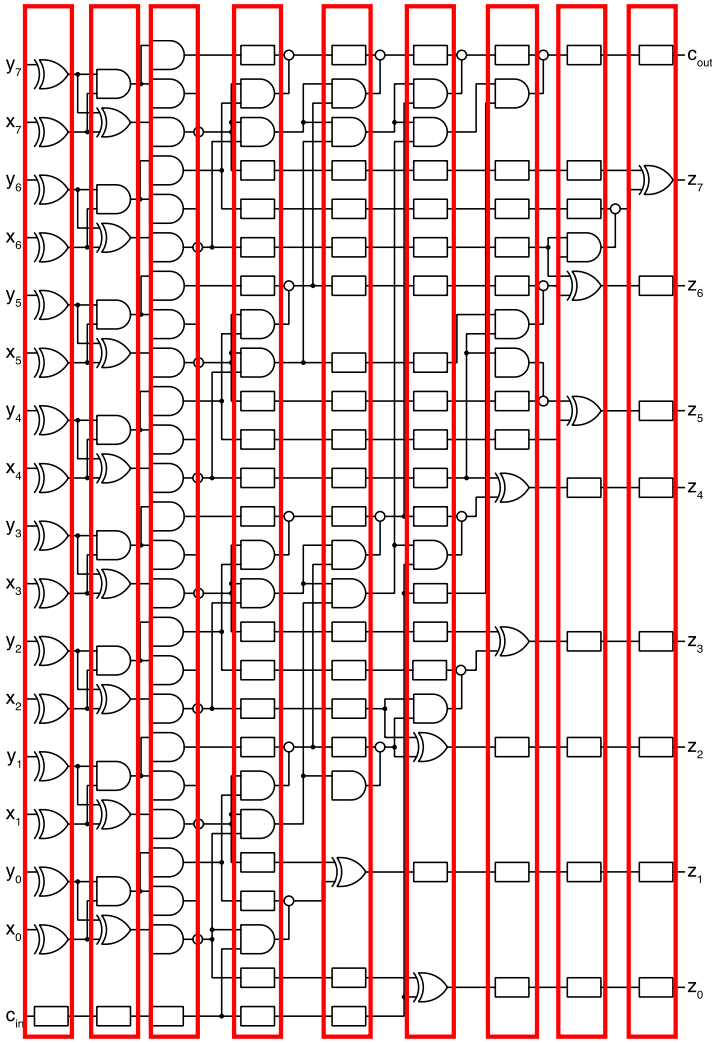


32GHz 6.2mW Processor
VLSI Symposium'20
Selected as a featured paper



50GHz AI Accelerator
MICRO'20
(simulation)

8-bit Bit-Parallel ALU Design: ISLPED 2017



Number of pipeline stages: 9

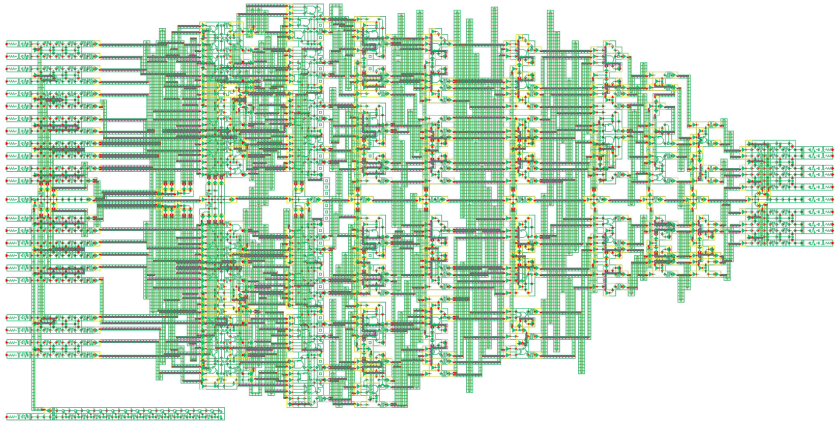
- ✓ **Target frequency: 50 GHz**
- ✓ **Gate-level pipelining**
- ✓ Functions: ADD, SUB, AND OR, XOR, NOR, etc.
- ✓ Data length: 8 bits

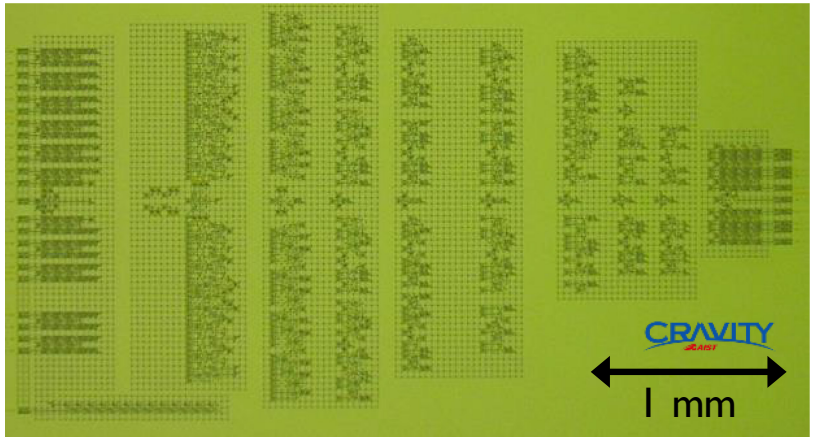
Based on Brent-Kung adder

- Minimum number of logic gates (w/o D flip-flops)
- Sparse wiring tracks
- Small fanouts (Max. 3)
- Maximum logic depth

R. Brent and H. Kung, *IEEE Trans. Comput.* **G31** (1982) 260

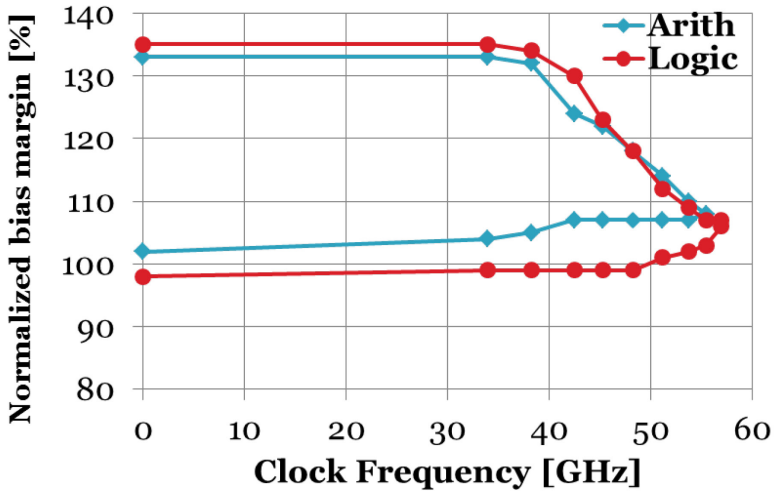
It Works!





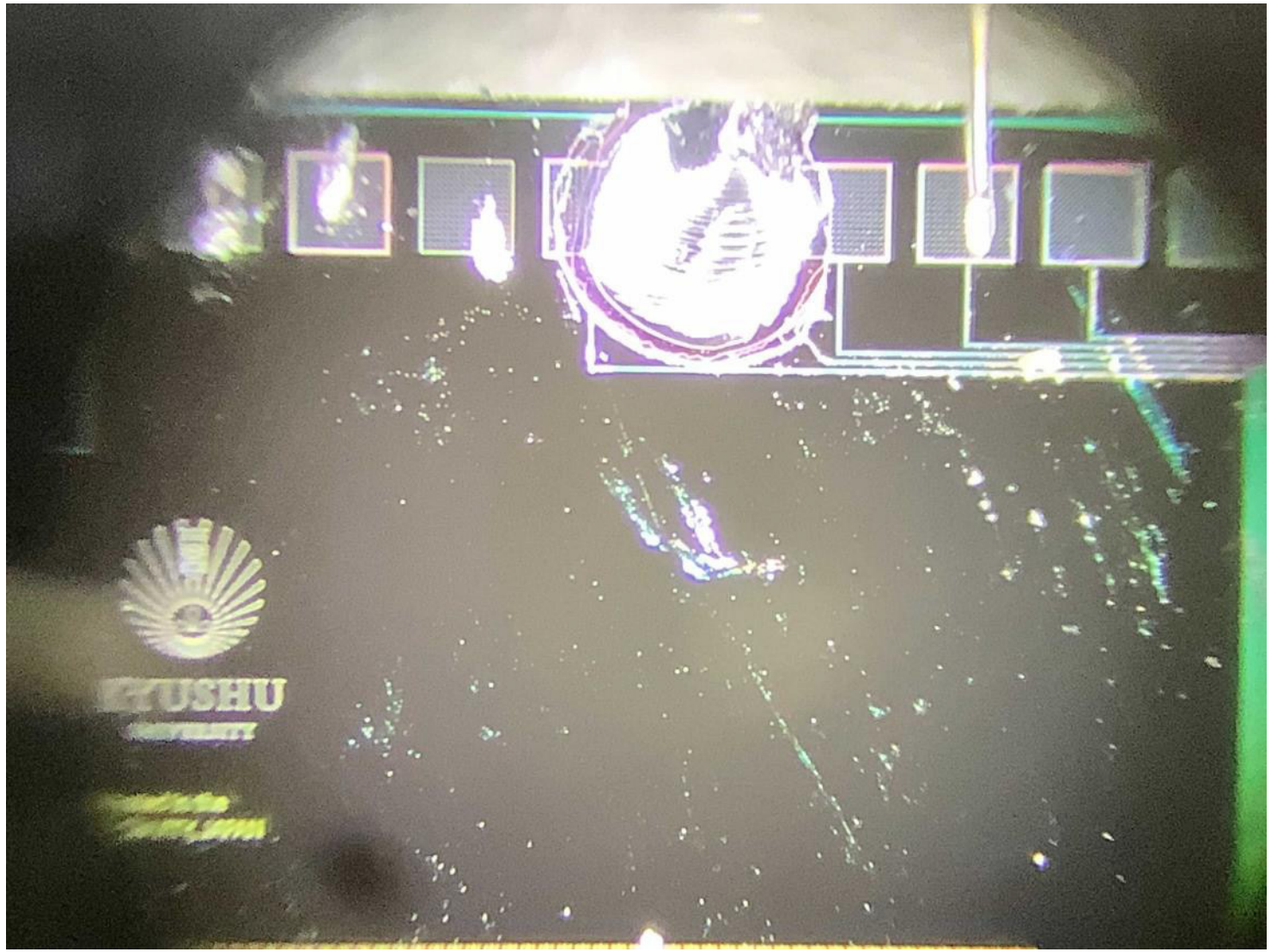
1.6 mW, 56 GHz 8-bit ALU
~35 TOPS/W

→Next design achieved
112 TOPS/W

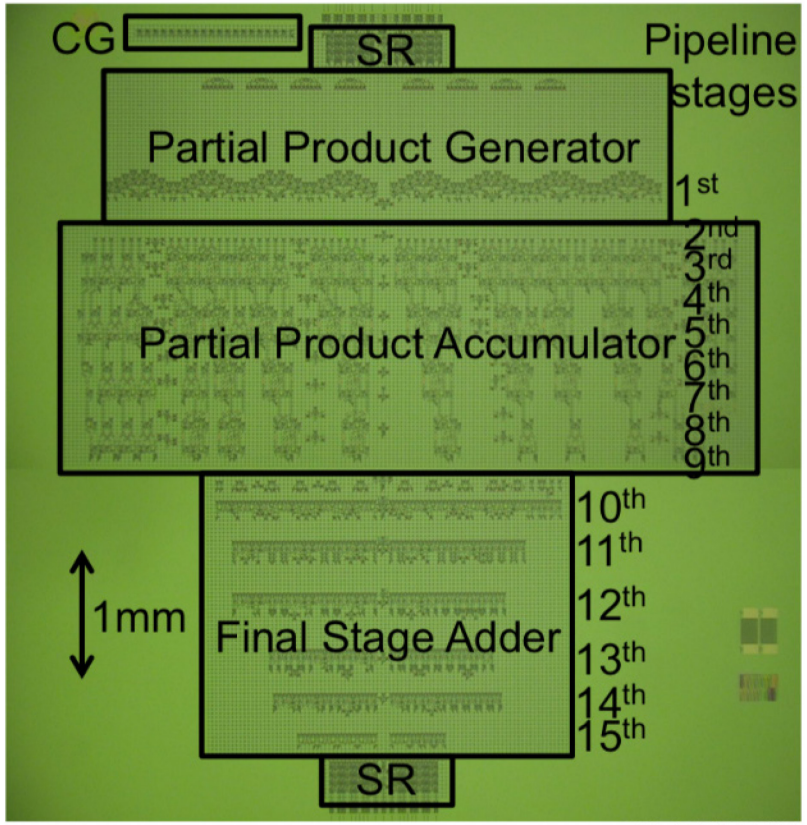


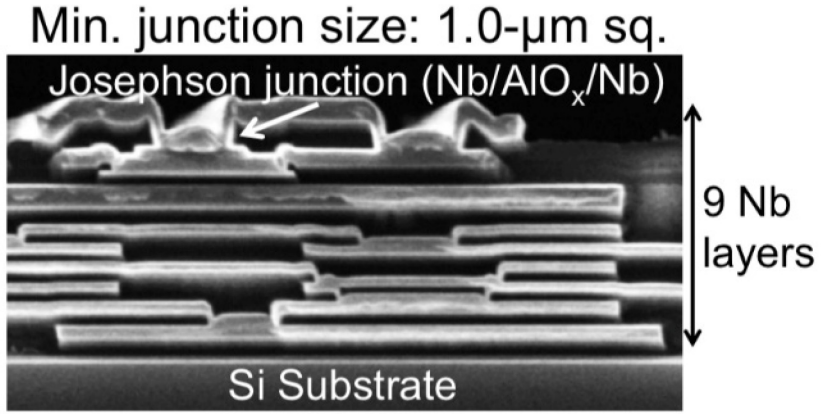
Youtube Movie
https://www.youtube.com/watch?v=jZP7sXWHyZs

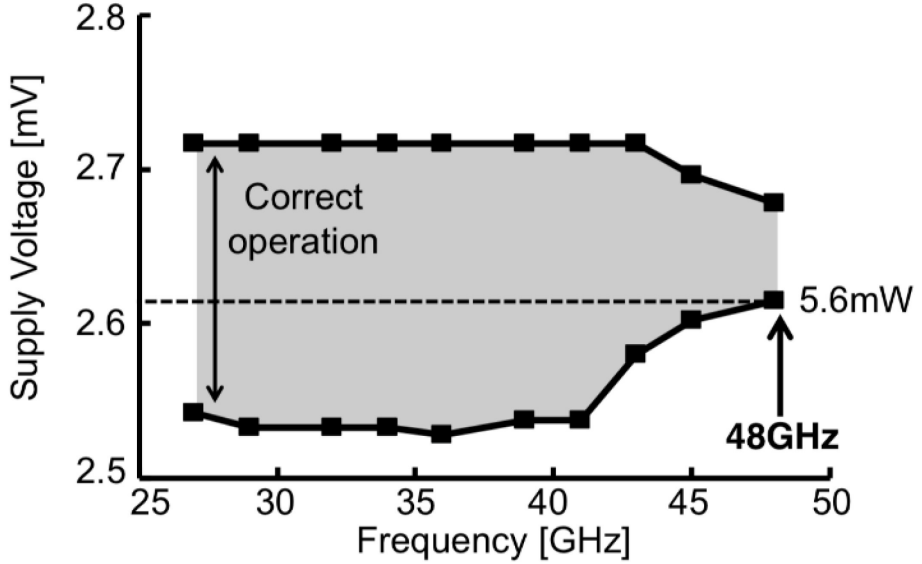
Sometimes…



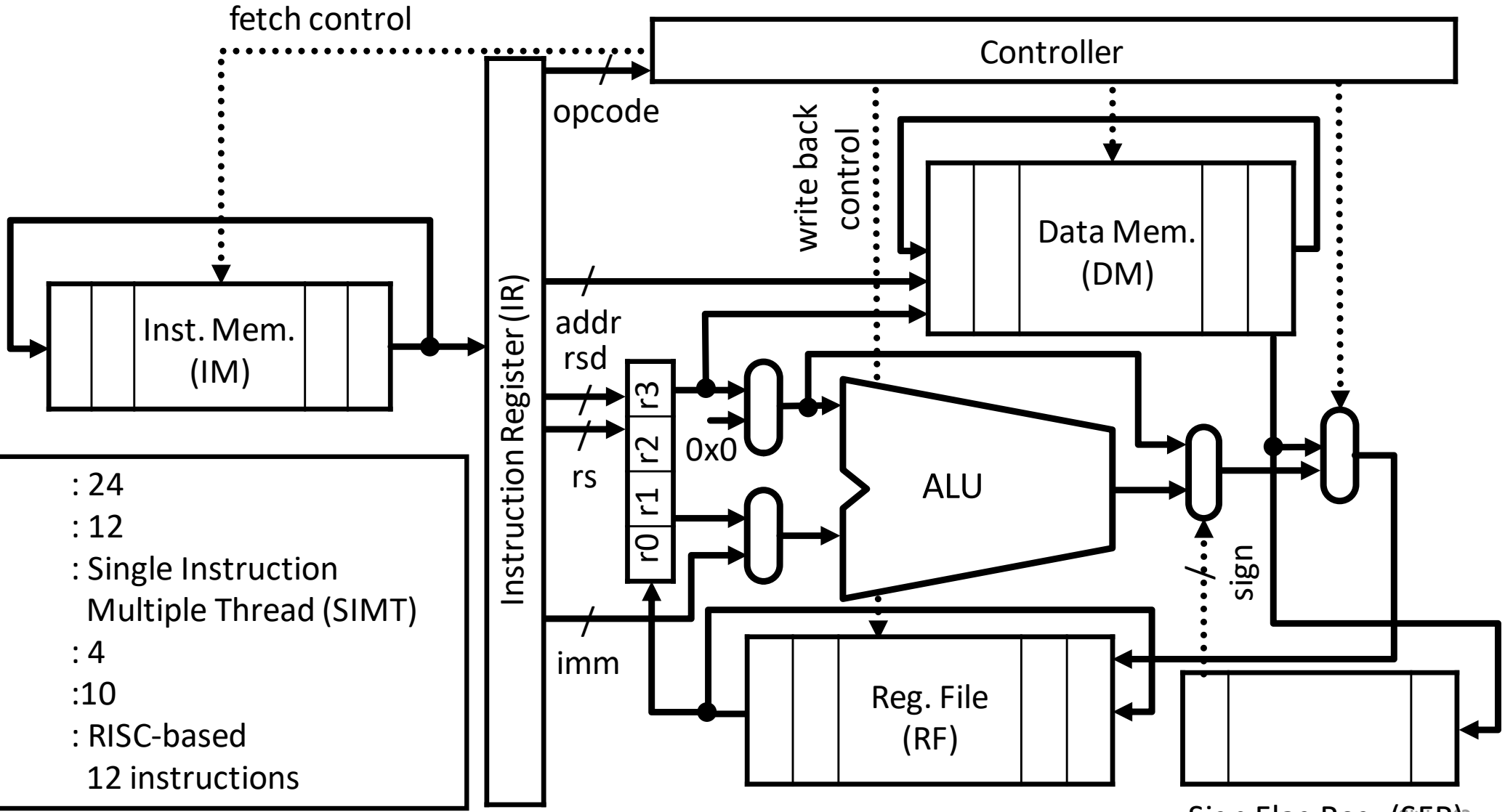
48 GHz 5.6mW Multiplier: ISSCC 2019





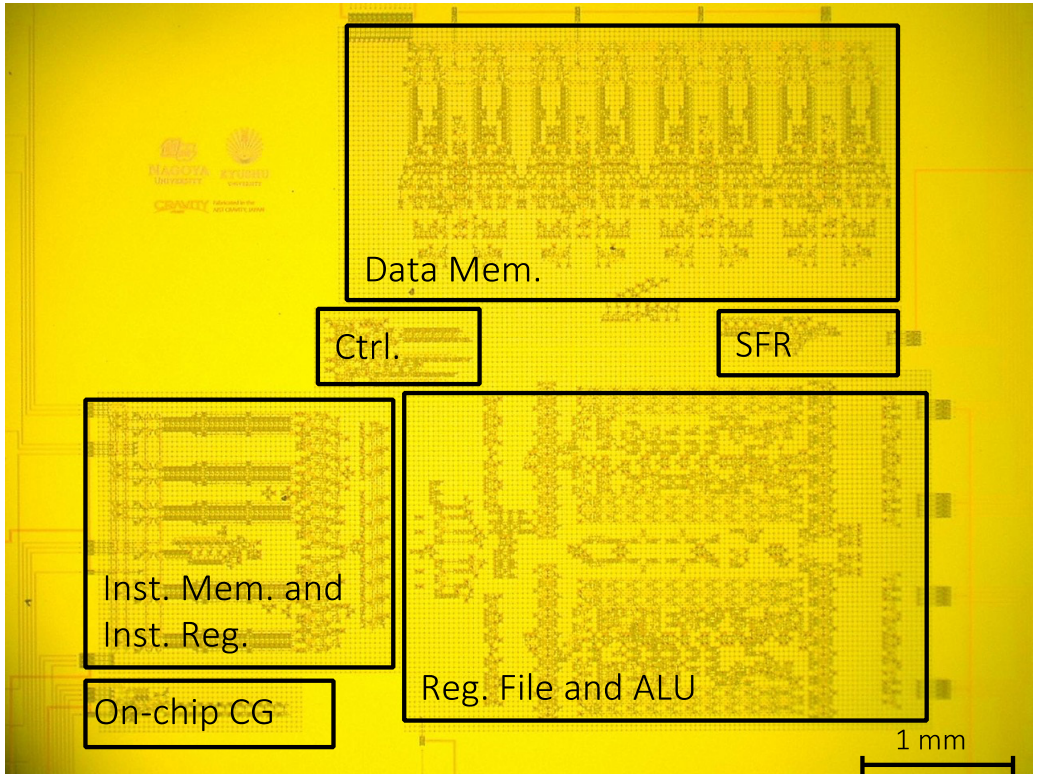


4-bit Microprocessor:VLSI Symposium 2022



- ✓ #Stages : 24
- ✓ #Threads : 12
- ✓ Execution : Single Instruction Multiple Thread (SIMT)
- ✓ Bit width : 4
- ✓ Inst width :10
- ✓ ISA : RISC-based 12 instructions

4-bit Microprocessor: VLSI Symposium 2022

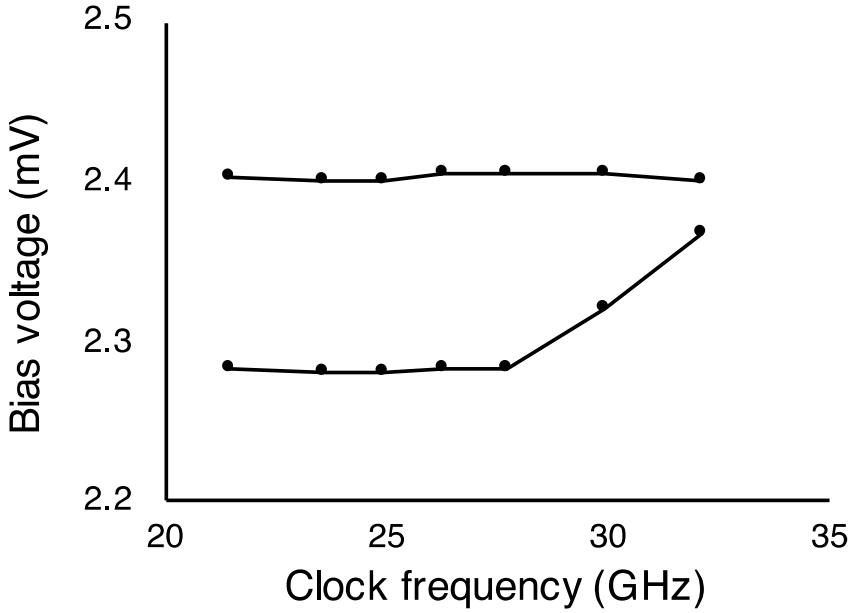


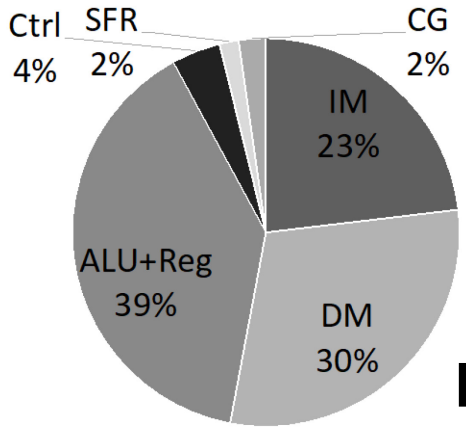
1.0-μm, 9-layer Nb process

Area : 4.08 mm x 5.31 mm

JJs : 23,713 JJs

Operating margin

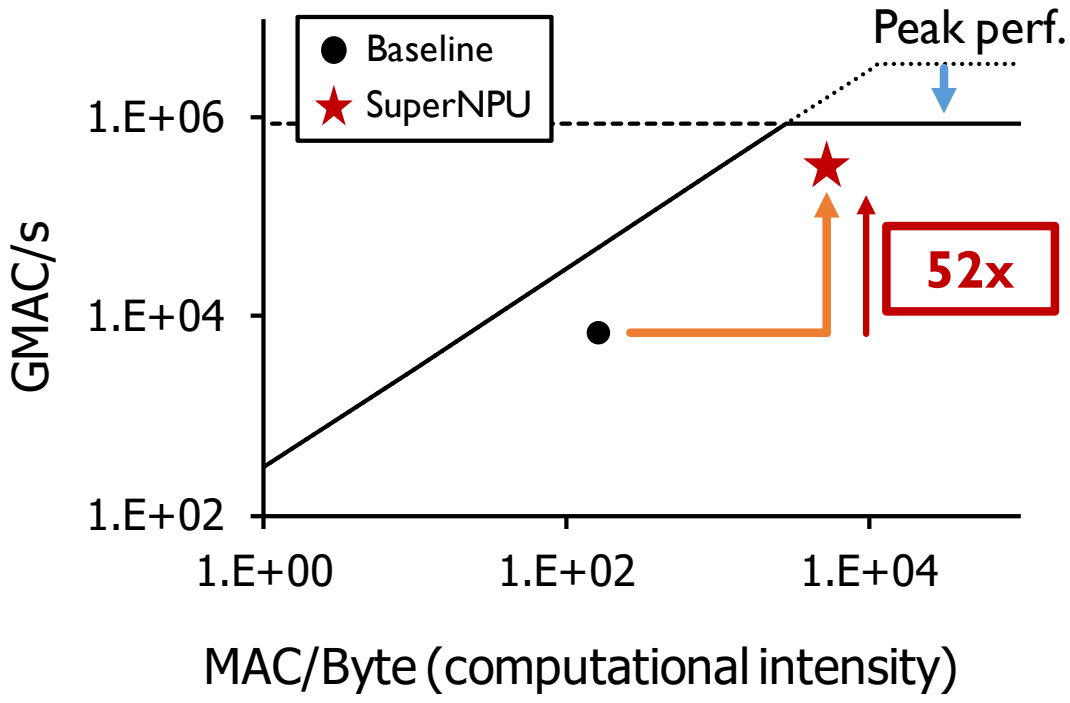


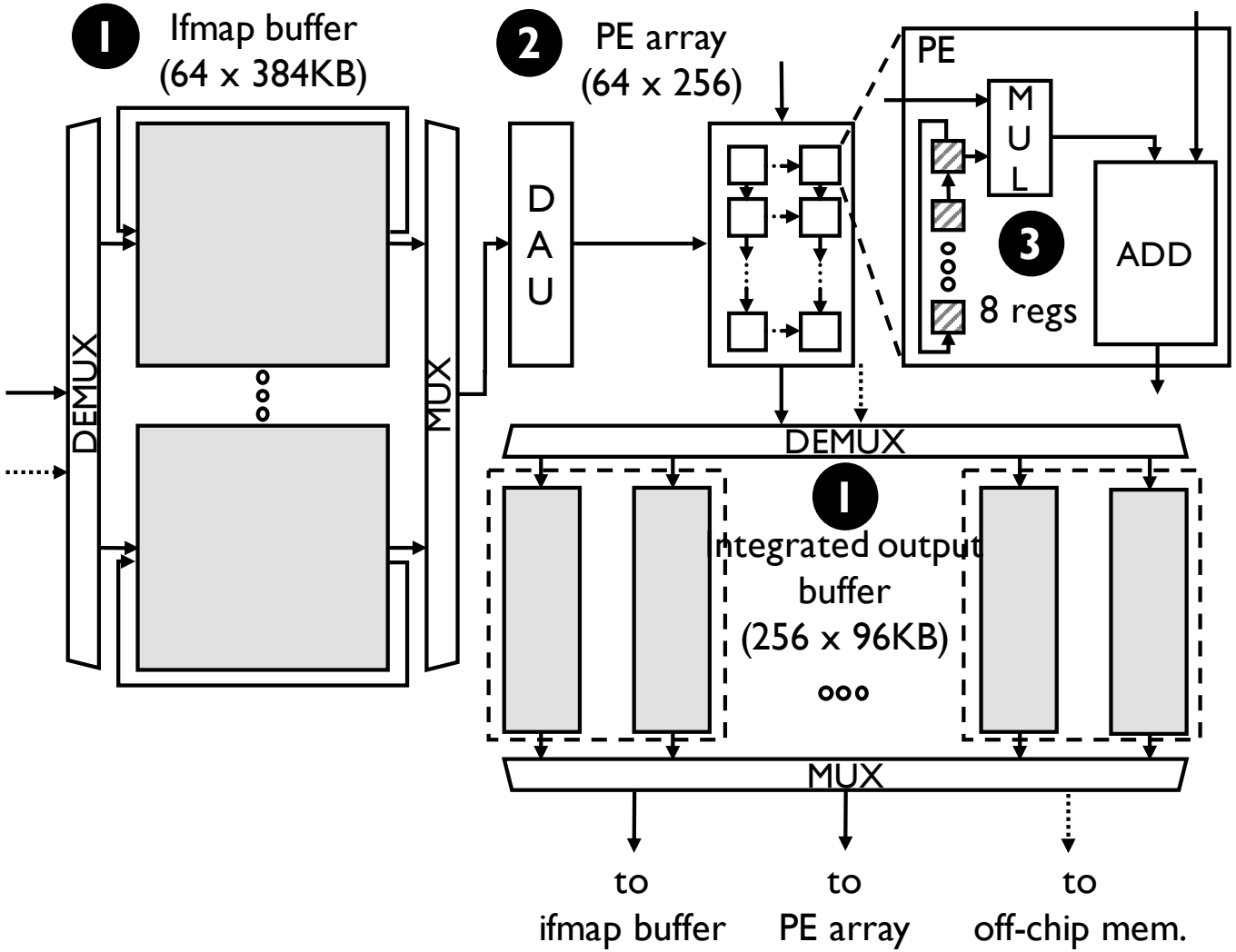


Power breakdown

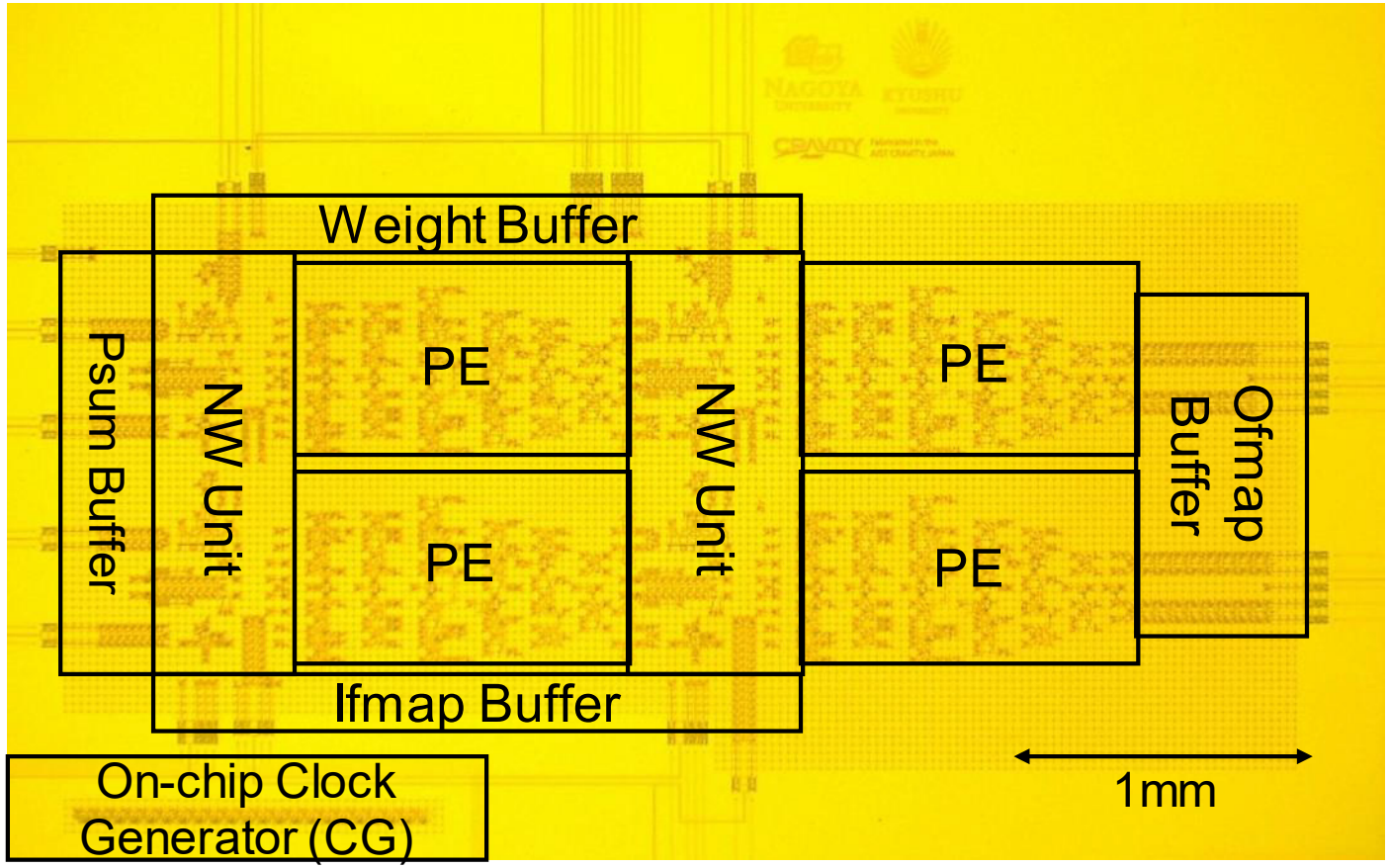
SuperNPU: MICRO 2022

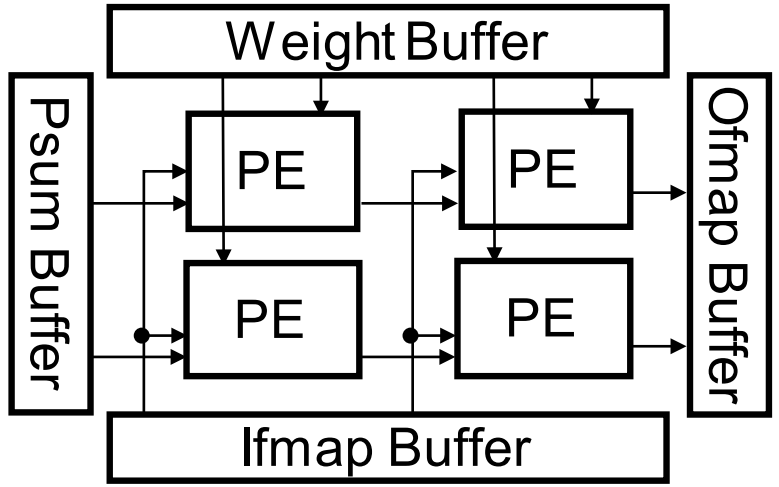
1. **Buffer division**
2. **Recourse balancing**
3. **Increase #registers in PE**

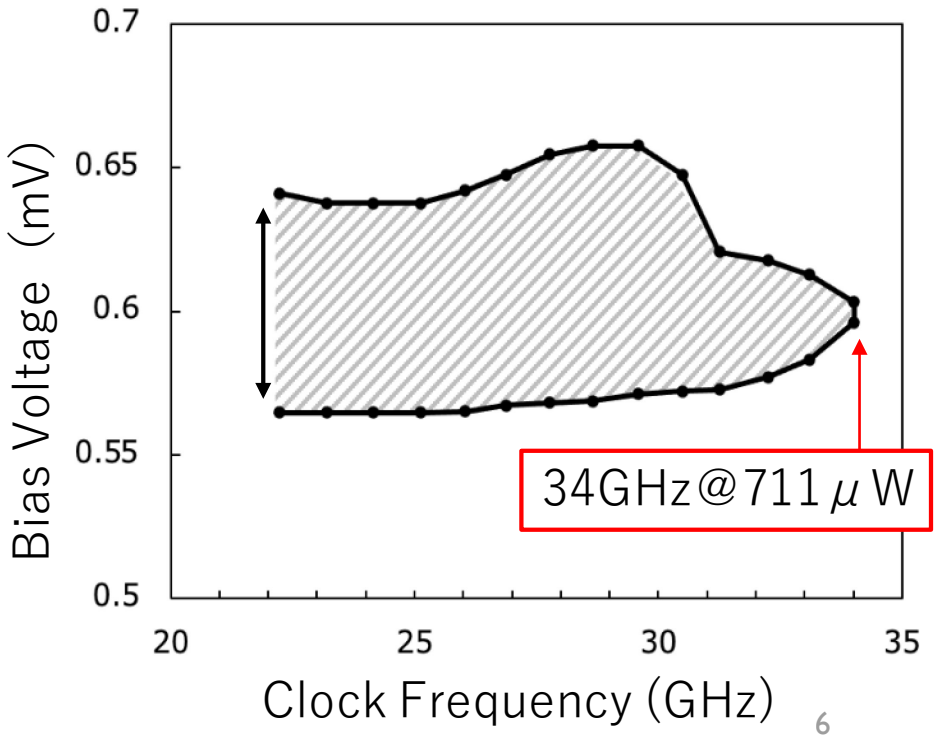




2x2 PE Array





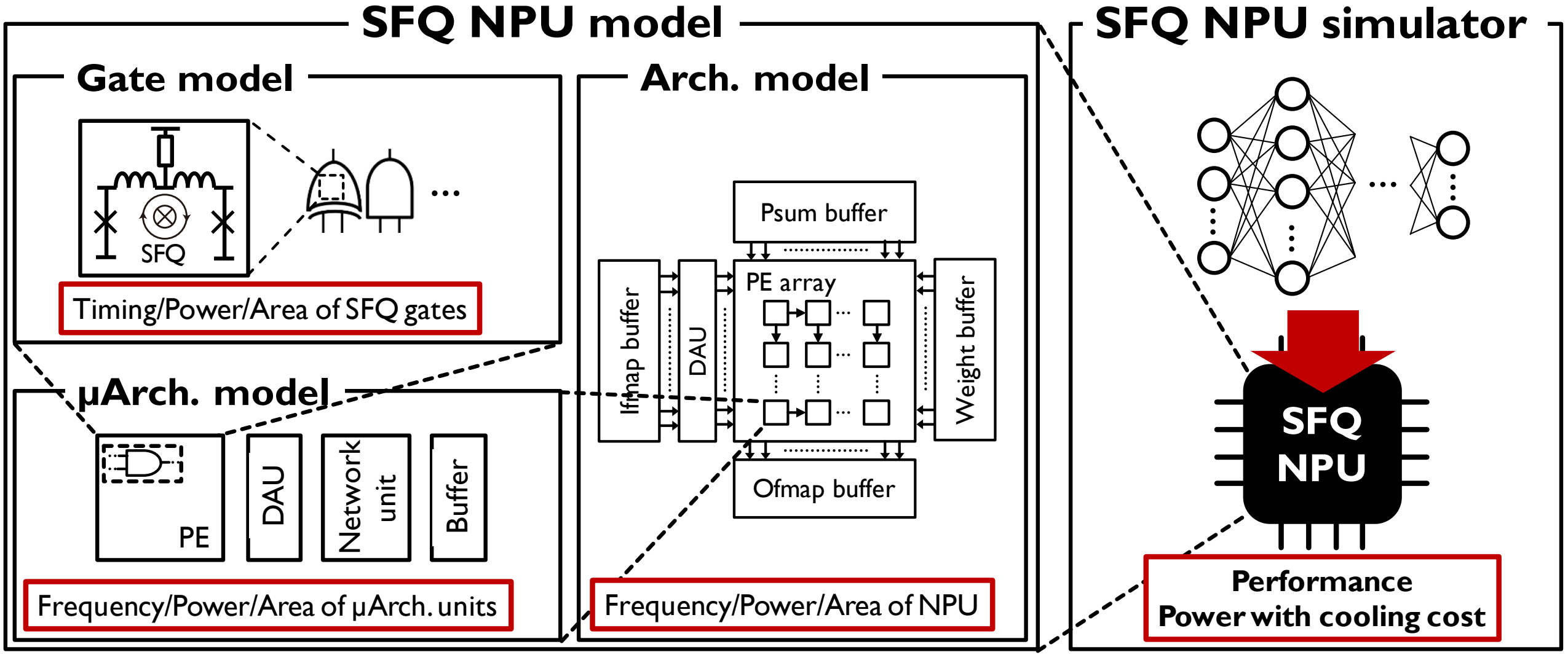


1.0-μm, 9-layer Nb process

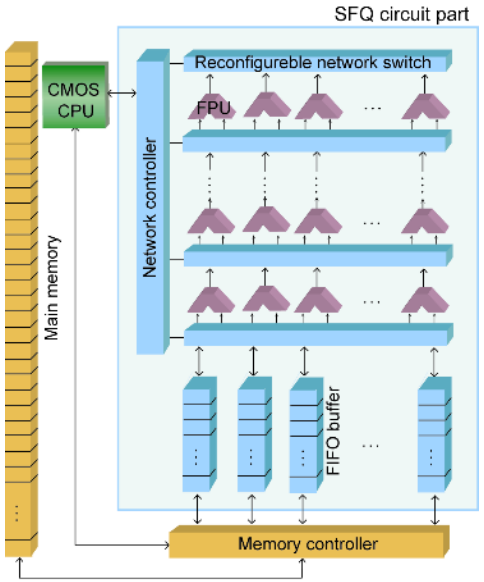
Area : 2.34 mm x 4.59 mm

JJs : 9,293 JJs

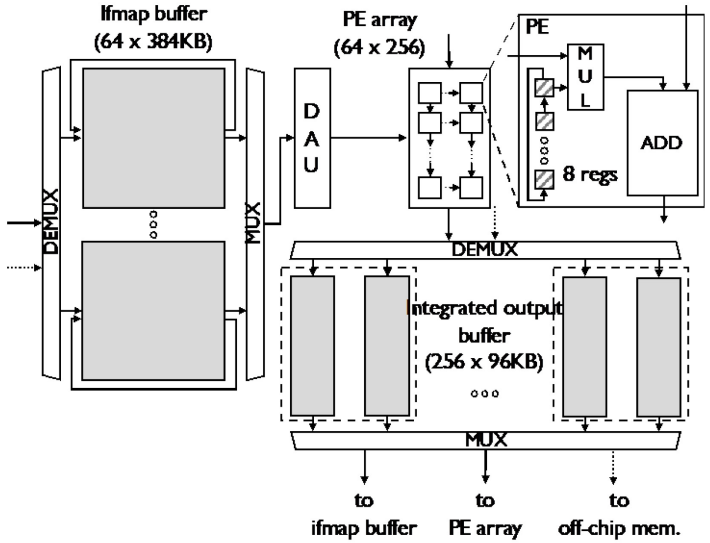
Simulation framework overview

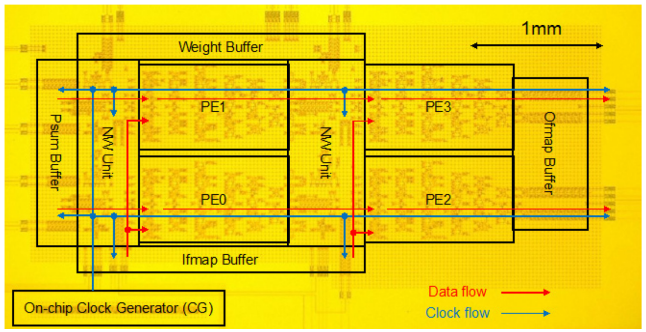


RSFQ RDP vs. SuperNPU



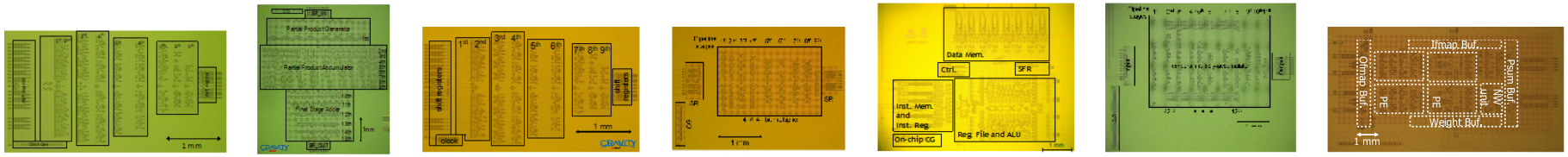
	SFQ-RDP	SuperNPU
Application	HPC (MO)	AI Inference
Data Reuse (on-chip)	Low	High
Datapath	Bit-Serial FP w/ course-grained pipelining	Bit-Parallel Int w/ gate-level pipelining
On-chip network	Complex & Frexible	Simple & Fixed
On-chip memory	Simple input-output buf	Simple & integrated buf
Optimization mainly focused	DFG mapping and routing	Microarchitecture





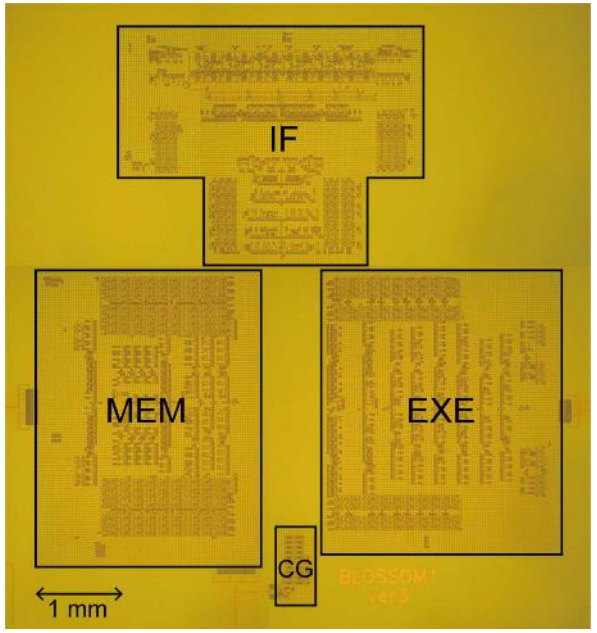
Series of our RSFQ design and micrographs of fabricated chips

Fabricated Chip	Purpose	Frequency [GHz]	Power [mW]	Efficiency [TOPS/W]	#of JJs	Year
1: 8-bit ALU	First demo. of gate-level pipeline	56	1.6	40	4,846	2017
2: 8-bit array-type multiplier	large-scale circuit design	48	5.6	8.5	20,251	2018
3: low voltage 8-bit ALU	0.5mV low-voltage operation	30	0.276	109	7,451	2019
4: low-voltage 4-bit multiplier	large-scale low-voltage operation	51	0.134	381	4,498	2019
5: 4-bit microprocessor	large-scale datapath	32	6.5	2.5	25,403	2019
6: low-voltage 4-bit MAC	basic function for AI acceleration	38	0.366	104	9,739	2020
7: 2x2 systolic PE array	prototype of *SuperNPU*	34	0.711	382	9,263	2021



1 2 3 4 5 6 7

State-of-the-art Designs

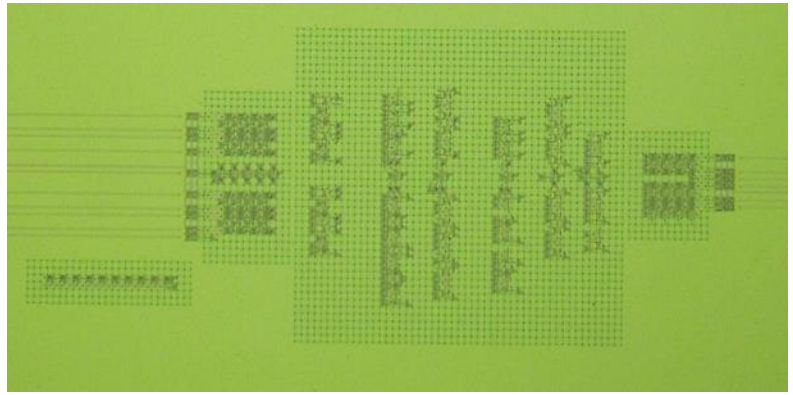


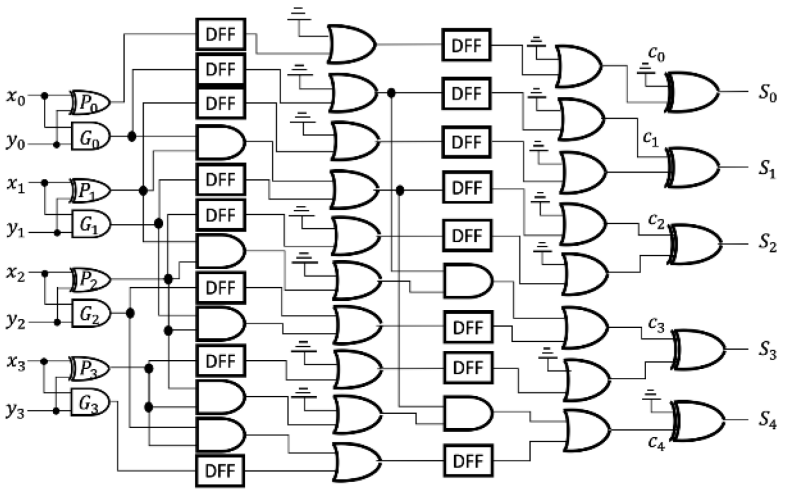
32,712 JJs, 5.8 x 6.0 mm^2

Types	Instruction	Tested components	Frequency (GHz)
Arithmetic instruction	ADD	IF, EXE	64.8
	SUB	IF, EXE	62.9
	INC	IF, EXE	64.8
	DEC	IF, EXE	64.8
Conditional-branch instruction	SKNE	IF, EXE	57.2
	SKLT	IF, EXE	57.2
	JMP	IF	60.8
Memory-access instruction	LW	IF, EXE, MEM	78.2
	SW	IF, EXE, MEM	71.9

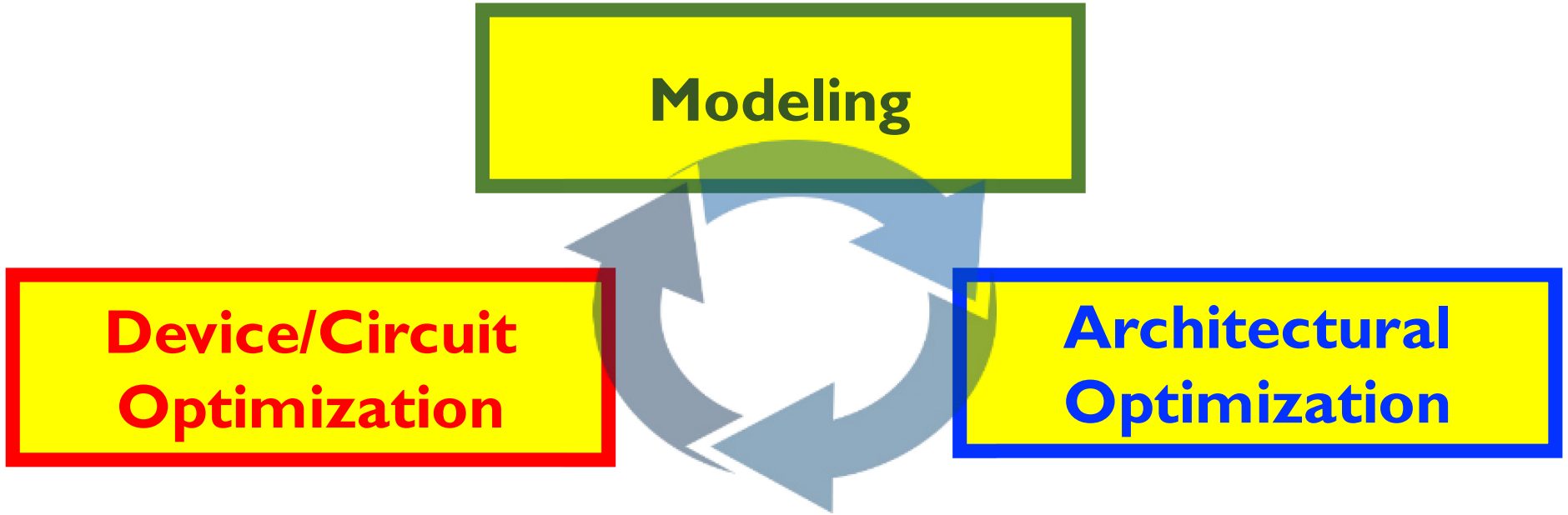
57.2GHz 11.2mW 8-bit General Purpose Superconductor Microprocessor with Dual-Clocking Scheme (ASSCC 2022)

>100 GHz Bit-parallel Adder





Conclusions
~ SFQ-based computing is still at an early stage! ~



Significant potential, but a lot of issues that computer architects would solve!