TOSHIBA

FPGA-based Simulated Bifurcation Machine

Kosuke Tatsumura, Alexander R. Dixon, and Hayato Goto

FPL2019 @ Barcelona
2019.09.09

Contents

An ultra-fast solver for combinatorial optimization problems

Combinatorial optimization problems

Economically valuable but computationally hard

Enhancing productivity

Finance

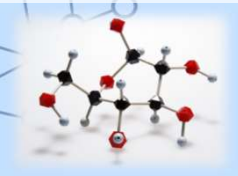
Manufacture

Logistics

Management

Medicine

Material



Combinatorial optimization problem

Routing

Allocation

Placement

Scheduling

Nondeterministic polynomial time (NP)-hard

Standard approach: Simulated annealing (SA)

New approach: Ising machine

1. Map to an Ising problem,
2. Special-purpose machine solves it quickly

Problem
NP-hard

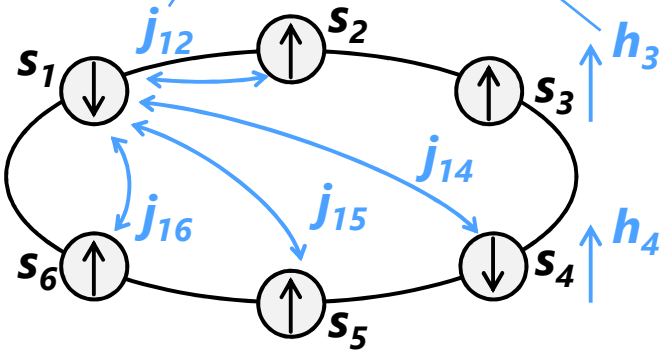
Combinatorial optimization problem

mapping

Ising problem
NP-hard & NP-complete

$$J=\begin{bmatrix} 0 & j_{12} & j_{13} \\ j_{21} & 0 & j_{23} \\ j_{31} & j_{32} & 0 \end{bmatrix} \quad h=\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}$$

coupling *bias*



spin: binary variable

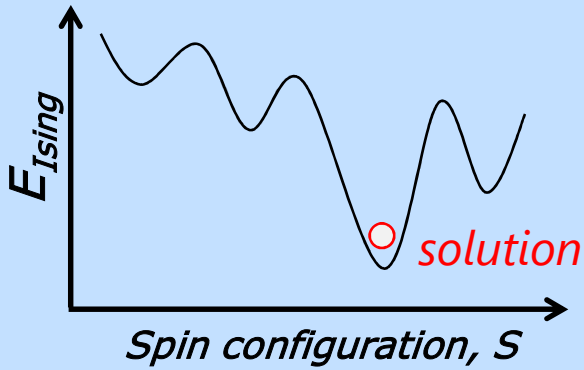
input

Ising machine
Special-purpose

search for ground-state ***s*** minimizing ***E***

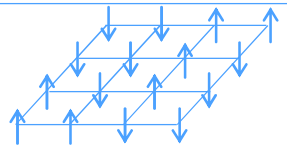
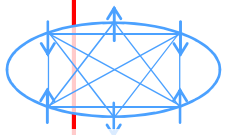
Ising Energy

$$E = -\sum j_{ij} s_i s_j + \sum h_i s_i$$



Ising machines

Very competitive, CIM deserves attention

	D-Wave Sys.[*1]	HITACHI [*2]	FUJITSU [*3]	NTT/Stanford/U-Tokyo [*4]
	2011-	2015-	2016-	2016-
	Quantum Annealer	**CMOS annealing machine**	**Digital annealer**	**Coherent Ising machine (CIM)**
				
Conne-ctivity	 Locally connected (sparse)		 Fully connected (dense)	
Principle	Quantum annealing	Simulated annealing (SA)		CIM (optical bifurcation)
				state-of-the-art [R. Hamerly, *Sci. Adv.* eaau0823, '19]

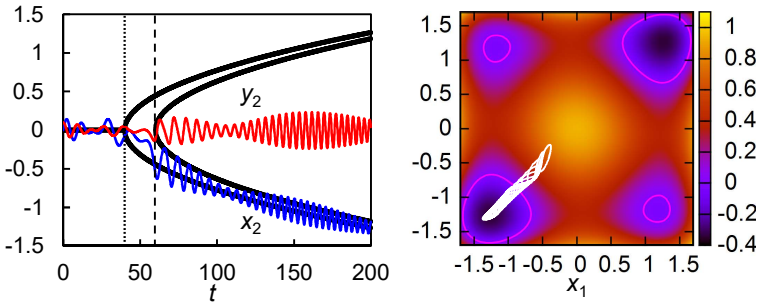
*1 https://www.dwavesys.com/d-wave-two-system
*2 https://www.hitachi.co.jp/New/cnews/month/2019/02/0219.html
*3 https://www.fujitsu.com/global/about/resources/news/press-releases/2018/0515-01.html
*4 https://www.ntt.co.jp/news2017/1711e/171120a.html

A recently proposed, quantum-inspired algorithm

Simulated Bifurcation (SB)

[H. Goto, K. Tatsumura, A. R. Dixon, *Sci. Adv.* ***5***, eaau0823, '19]

Simulated Bifurcation (SB)

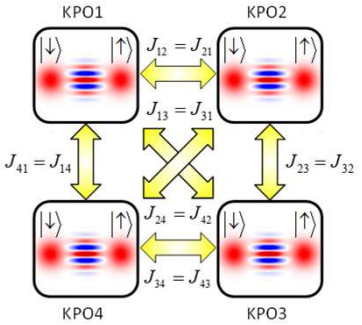


Derived as the classical counterpart

Quantum Bifurcation (QB) machine

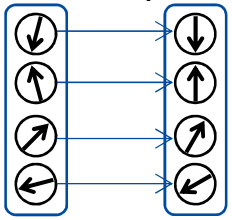
a quantum adiabatic optimization method

[H. Goto, *Sci. Rep* ***6***, 21686, '16]



Plentiful parallelism

Parallel updating

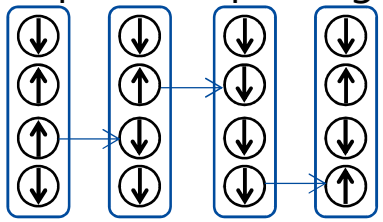


→Substantial speedup by massively parallel processing

vs.

Simulated Annealing (SA)

Sequential updating



Objective of this work

Designing & evaluating massively-parallel custom accelerator for SB

Motivation:

Be the world's fastest Ising machine

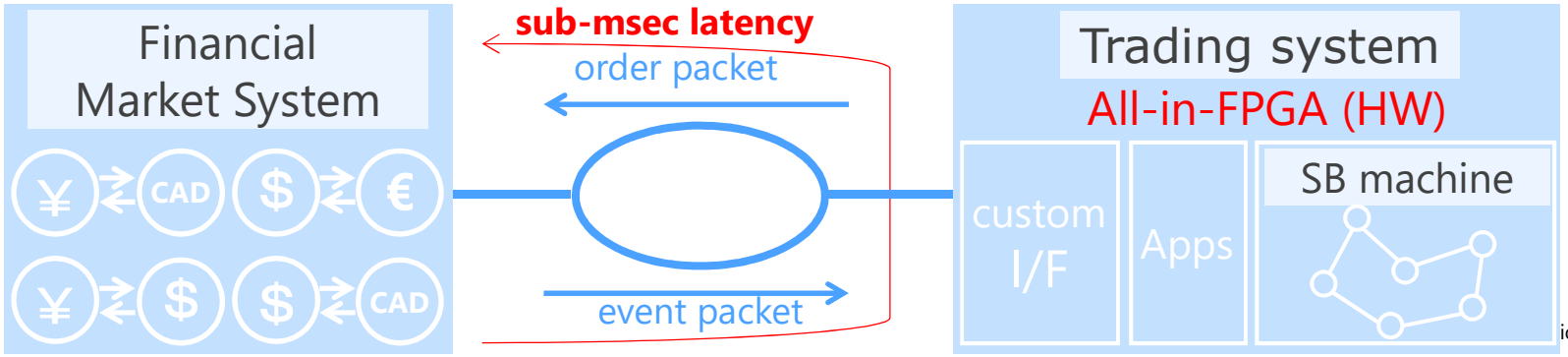
→ Pursue a custom circuit that can fully exploit the parallelism in SB

Meet the diversity of problems (size, bit precision, I/F)

→ Describe the design in an HLS to make it flexible & scalable

Toward "Intelligent real-time response systems"

→ Be a component of ALL-in-HW system enabling sub-msec latency



Contents

An ultra-fast solver for combinatorial optimization problems

How it works: Simulated Bifurcation (SB)

N-body system dynamically searches for a good solution

Movement of the system in *N*-dimensional space

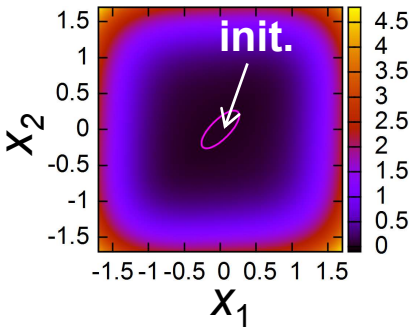
Example: ***N*=2**

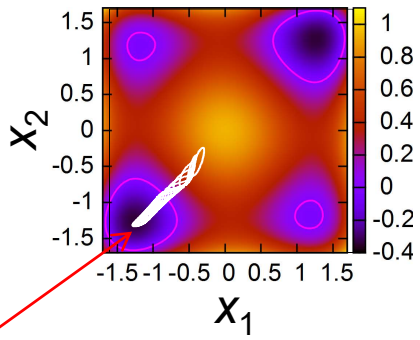
a single local minimum

Bifurcation
(adiabatic process)

multiple local minima
(target cost function)

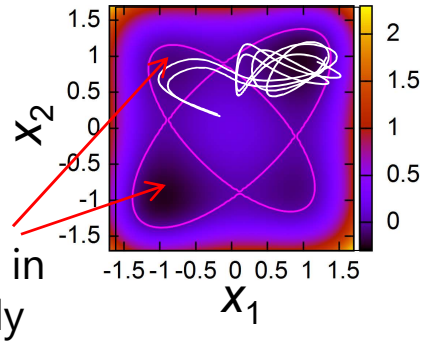
best solution
(-1,-1)

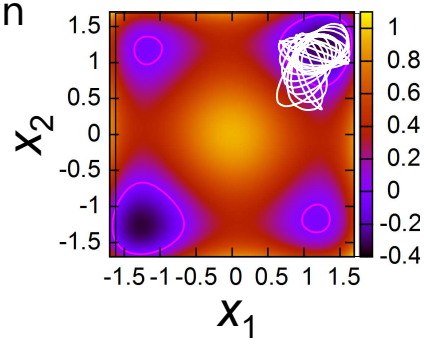




Adiabatic Search
chase one of the minima

Multiple minima in the energetically allowable region





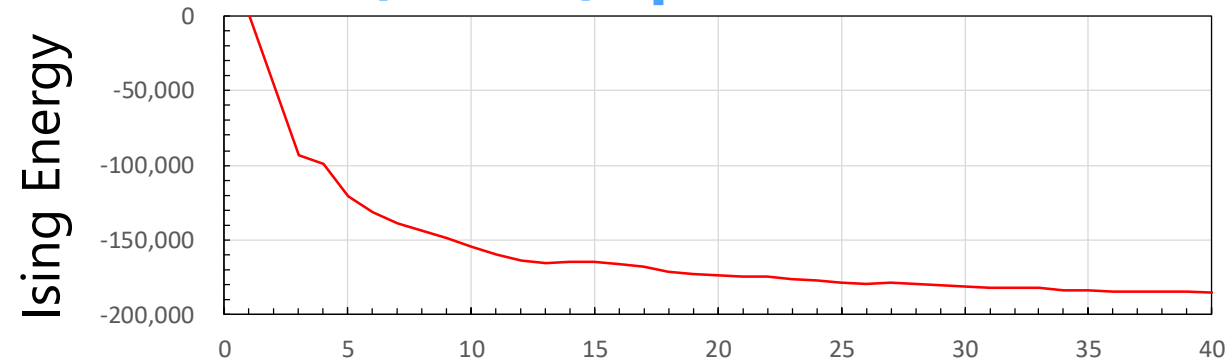
Ergodic Search
find better one with higher probability

If *N* is large,
imagine the process of finding the brightest "star" among 2^N stars in the dark

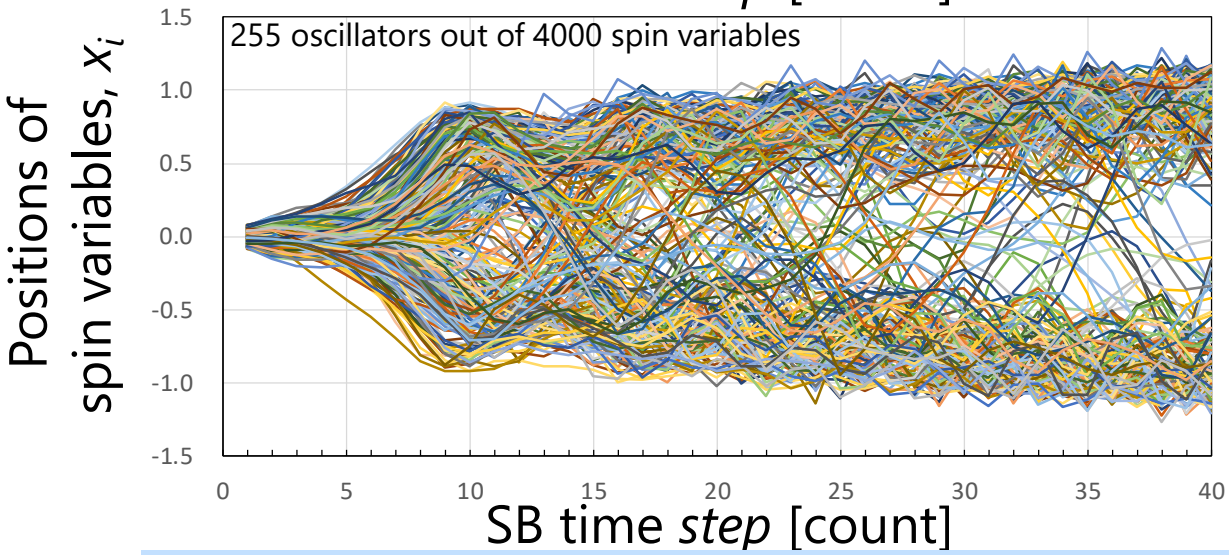
How it works: Simulated Bifurcation (SB)

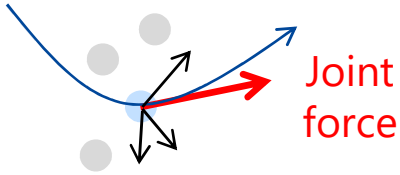
Time evolution of *N*-body system

Movements of *N*(=4000) spin-variables as a function of time



better solution



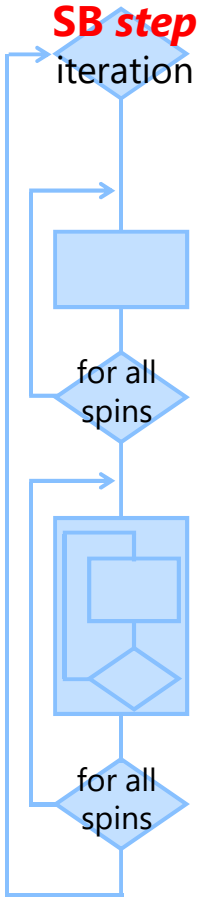


Many-body interactions depending on all other spins

This is what FPGA-based SB machine computes

Algorithm of SB and it's parallelism

SB *step*: spin state at t_{n+1} ← the previous state at t_n



Matrix-Vector Multiplication (**MM**)

$$\begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \square \\ \square \\ \square \\ \Delta p_N \end{bmatrix} = \begin{bmatrix} j_{11} & j_{12} & \square & \square & \square & \square \\ j_{21} & j_{22} & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & j_{NN} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ \square \\ \square \\ \square \\ x'_N \end{bmatrix}$$

ΔP | J | X'

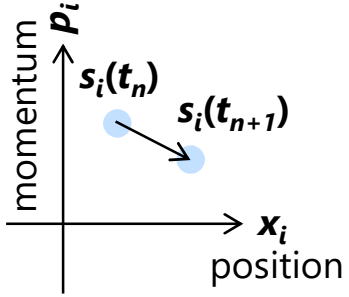
Many-body interaction

$$\Delta p_i = \sum j_{ij} x'_i$$

Horizontal partition (Block parallelism)

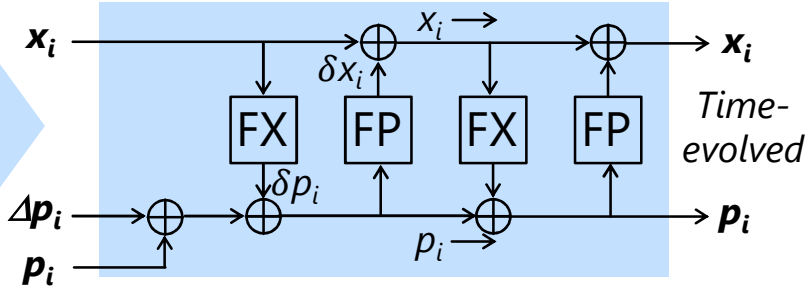
Time evolution (TE)

$$(x_i, p_i)_{@tn+1} = \mathrm{TE}(x_i, p_i, \Delta p_i)_{@tn}$$



Sequential but independent for each spin

TE pipeline

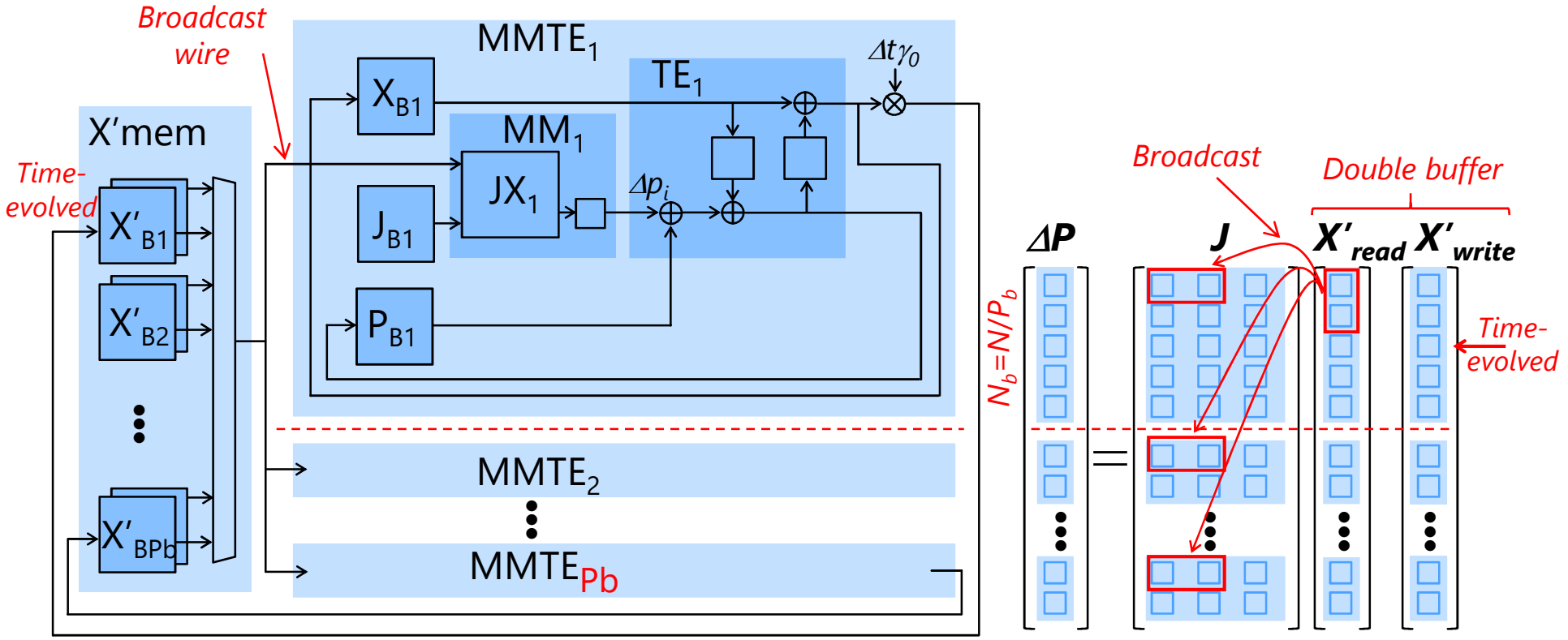


Top-level parallelism: **Simultaneous update of *N* spins is possible**

Design: Top-level circuit architecture

End-to-end HW implementation of SB

1. SB *step* iteration → Circulative structure
2. Block parallelism (P_b) → P_b number of MMTEs (responsible for $N_b=N/P_b$ spins)
3. Data dependency for ***x*** → *Double buffer for X'mem*



Design: the most computationally intensive part

Matrix-vector multiplication module (MM)

1. Column/Row parallelism (P_c, P_r)

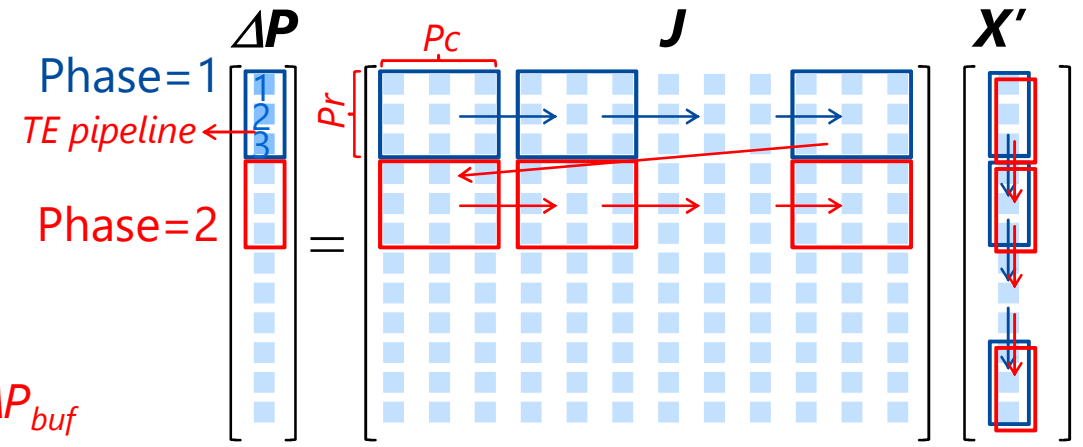
   ↓

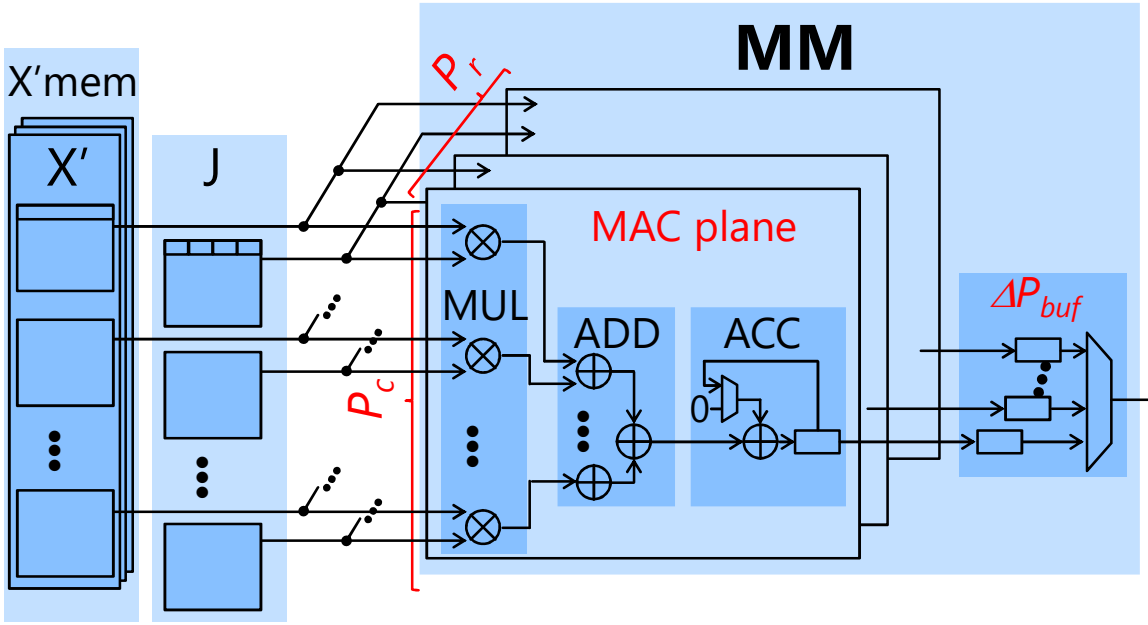
   P_r number of P_c-input MAC planes

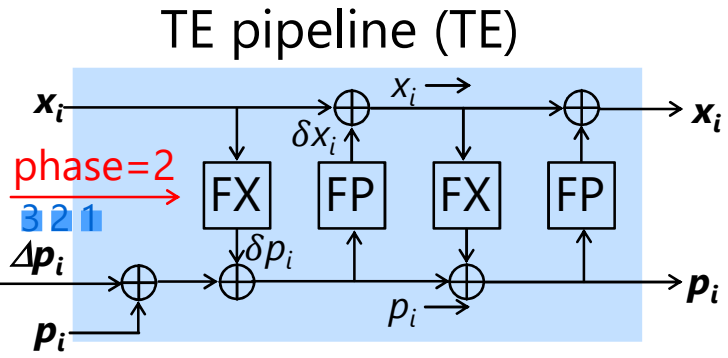
2. Overlapping TE with MM

   ↓

   Parallel-input/Sequential-output ΔP_{buf}

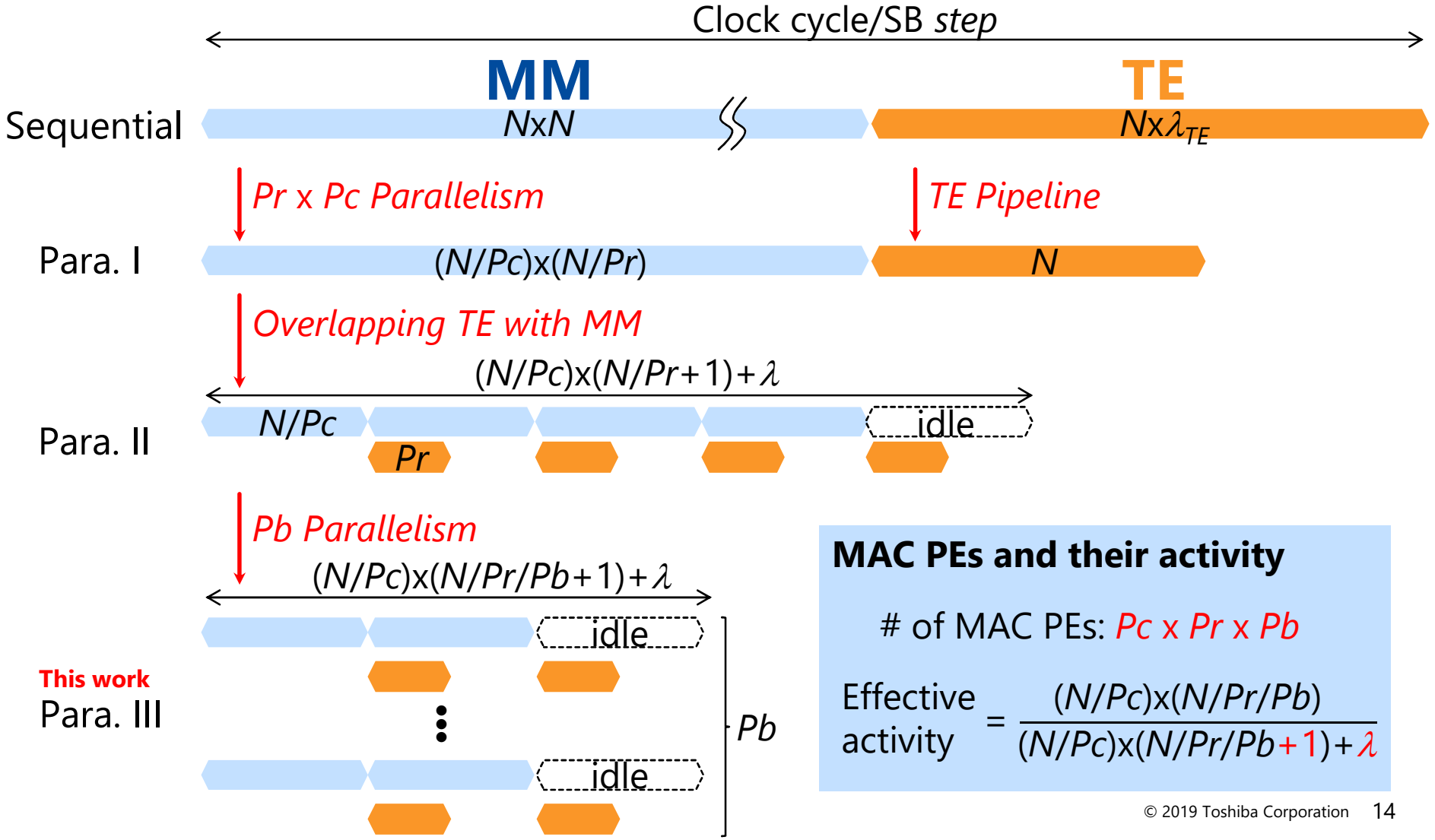






Timing Design: spatial and temporal parallelization

Pc x *Pr* x *Pb* Speedup for MM & hiding TE part



MAC PEs and their activity

# of MAC PEs: *Pc* x *Pr* x *Pb*

$$\text{Effective activity} = \frac{(N/Pc)\text{x}(N/Pr/Pb)}{(N/Pc)\text{x}(N/Pr/Pb+1)+\lambda}$$

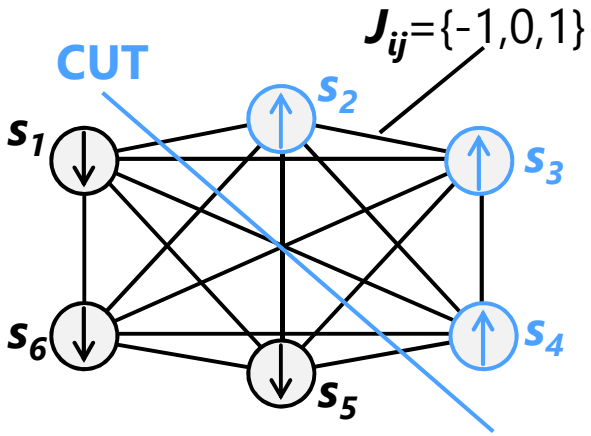
Contents

An ultra-fast solver for combinatorial optimization problems

Implementation for MAX-CUT benchmark problem

4,096-spin SB machine on Arria10 FPGA
Massively Parallel, High Utilization, 4X larger vs CIM

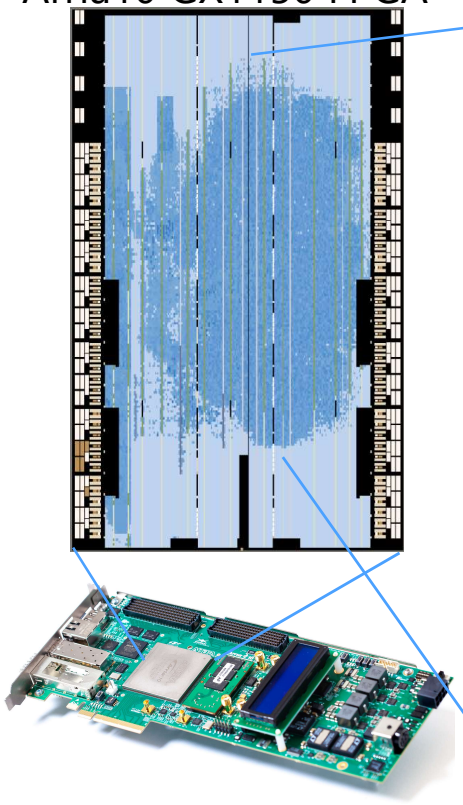
MAX-CUT problem

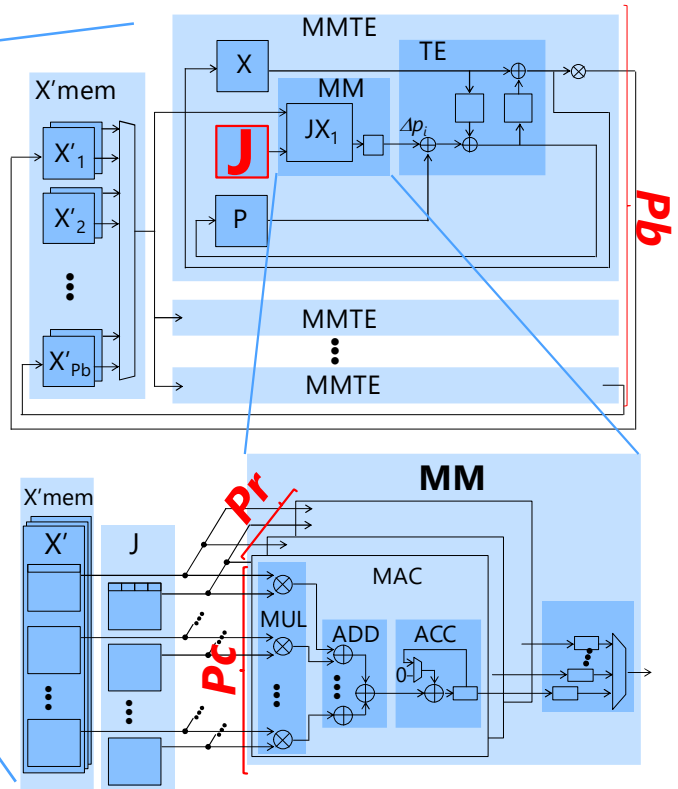


Implementation for MAX-CUT benchmark problem

4,096-spin SB machine on Arria10 FPGA
Massively Parallel, High Utilization, 4X larger vs CIM

Arria10 GX1150 FPGA





	4,096-size
Architecture	
Pr/Pc/Pb	32/32/8
# of MAC PEs	**8,192**
Effective activity	**92%**
Resource	
ALM	40%
BRAM	**56%**
DSP	7%
System Clock	[MHz]
Fsys	**269**

Spin-Size: limited **by BRAM (for J matrix)**

MAX-CUT: J_{ij}={-1,0,1}, $J_{4096x4096}$→16Mbit (1,024 BRAMs)

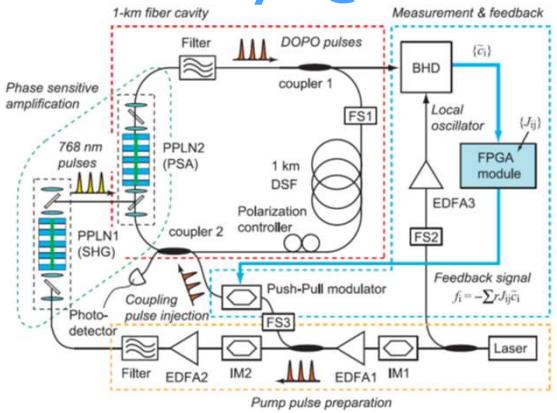
Parallelism: limited **by routing congestion**

Evaluation: FPGA-SB vs. CIM

14X faster, 288X more energy efficient than CIM

Coherent Ising Machine

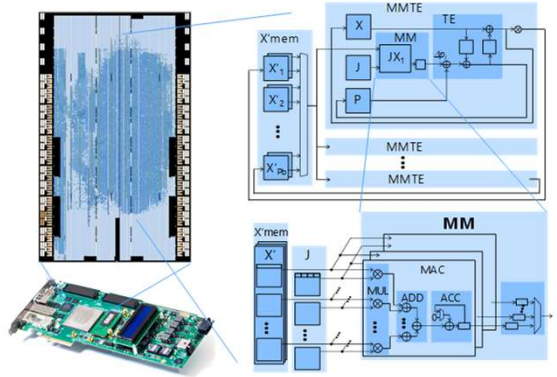
800 GMAC/s @ 1000 W

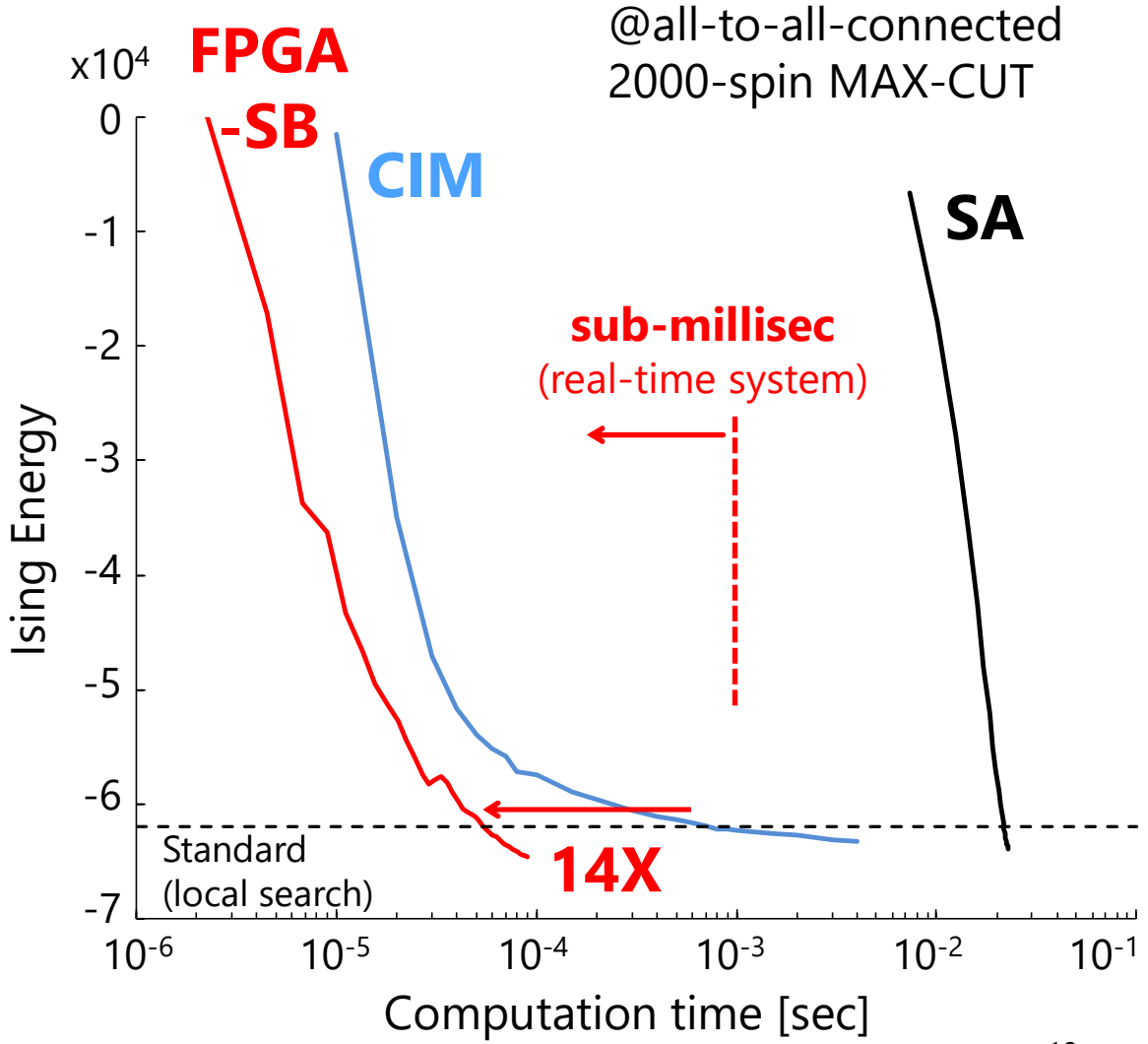


[T. Inagaki, Science 354, 603, '16]

FPGA-SB

1,873 GMAC/s @ 49 W

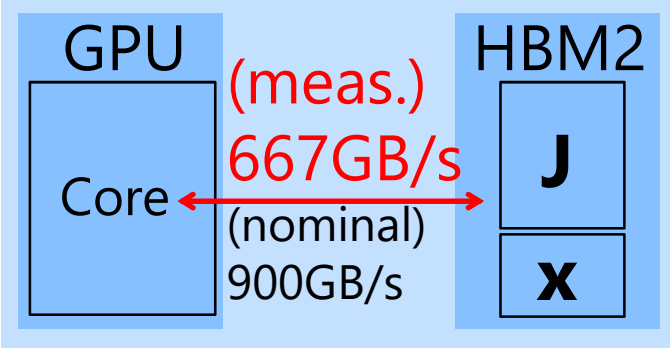




Evaluation: FPGA-SB vs. GPU-SB

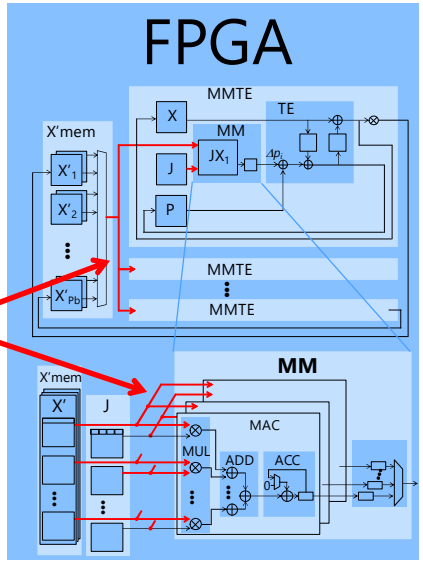
FPGA is computation-bound, GPU memory-bound
-11X faster, 26X more energy efficient than GPU-SB

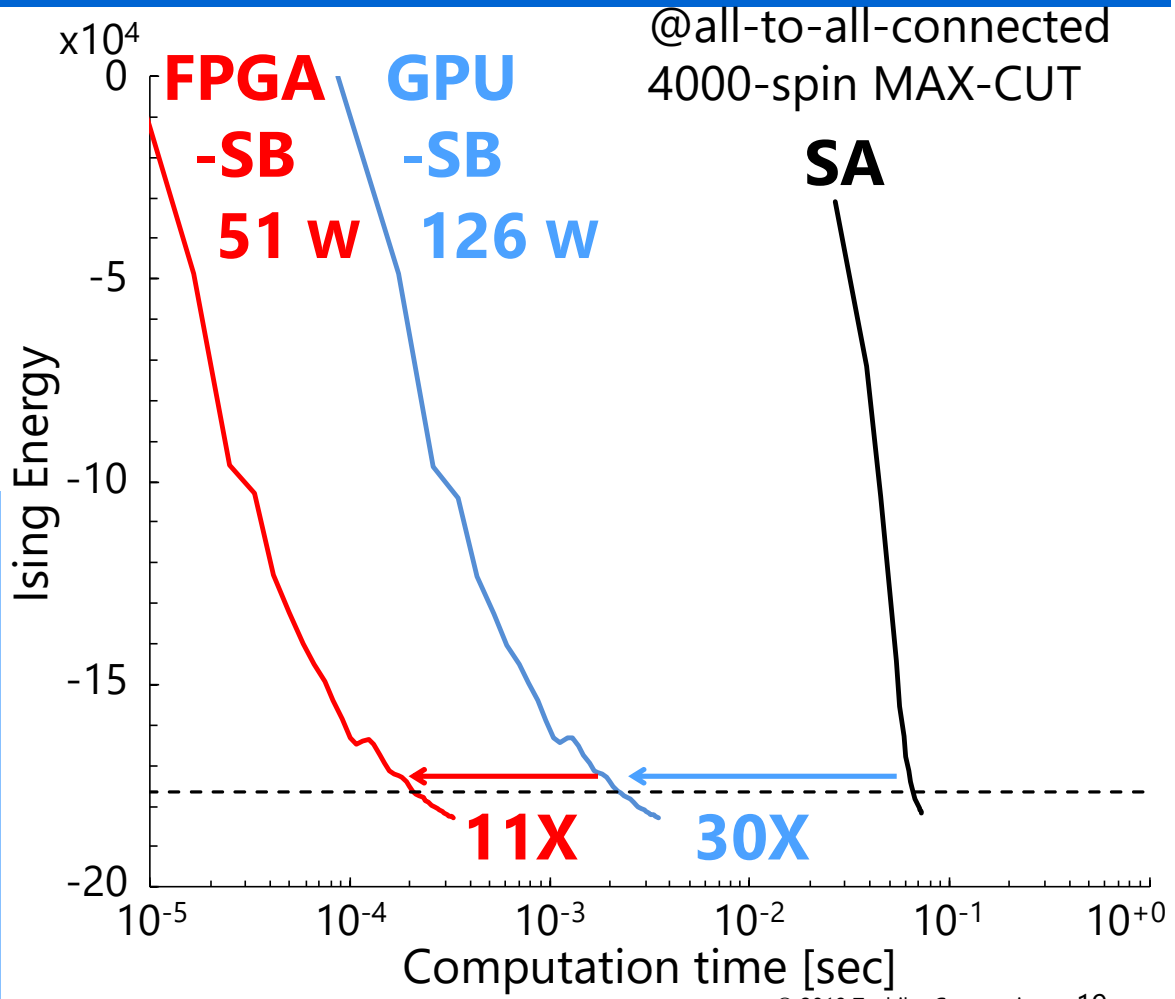
GPU-SB (Tesla V100)



FPGA-SB

peak rate
4,678GB/s





Contents

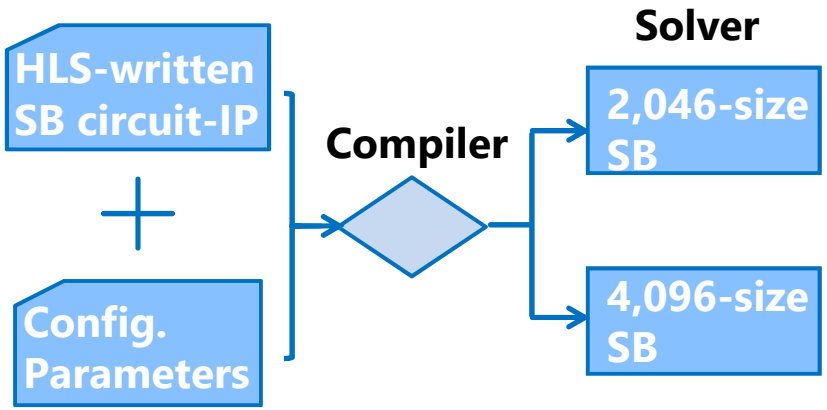
An ultra-fast solver for combinatorial optimization problems

Hit record-high speed, meet diversity of problems

Write fully(bit-by-bit/clock-by-clock) customized HWs in an HLS language (OpenCL)

HLS: high-level-synthesis

Good size-scalability of our parameterized design



*all-to-all-connected MAX-CUT

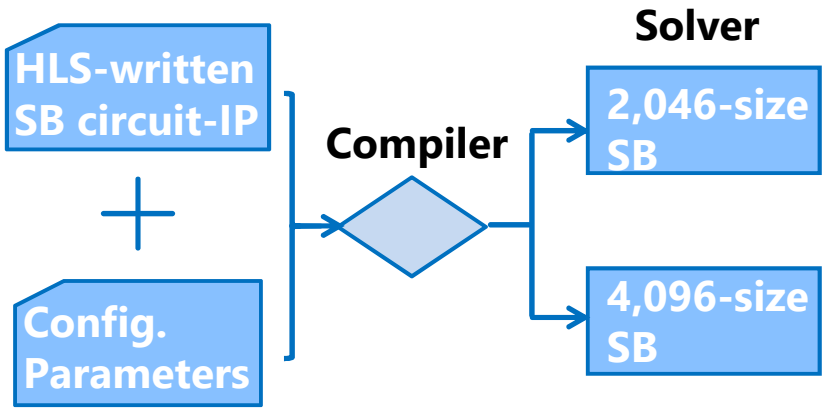
Problem	K2000*	K4000*	Ratio
Complexity per SB step	4M	16M	4X
Solver	**2K-size SB**	**4K-size SB**	**Ratio**
Time-to-solution [ms]	0.055	0.211	3.9X
Energy-for-solution [mJ]	2.7	10.8	4.0X

Hit record-high speed, meet diversity of problems

Write fully(bit-by-bit/clock-by-clock) customized HWs in an HLS language (OpenCL)

HLS: high-level-synthesis

Good size-scalability of our parameterized design



*all-to-all-connected MAX-CUT

Problem	K2000*	K4000*	Ratio
Complexity per SB step	4M	16M	**4X**
Solver	**2K-size SB**	**4K-size SB**	**Ratio**
Time-to-solution [ms]	0.055	0.211	**3.9X**
Energy-for-solution [mJ]	2.7	10.8	**4.0X**

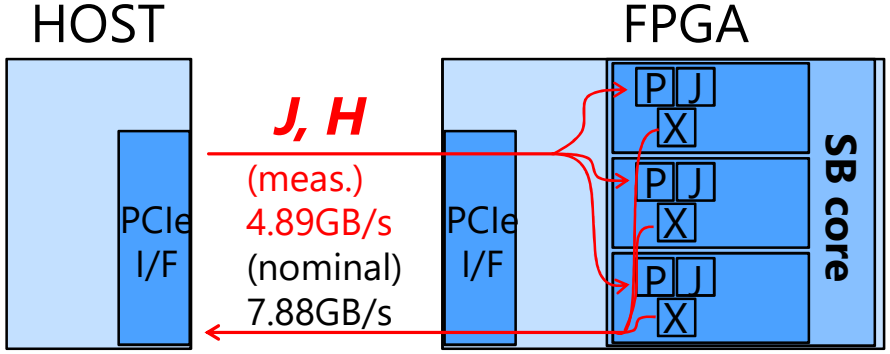
FPGA+HLS is a good choice for practical problems

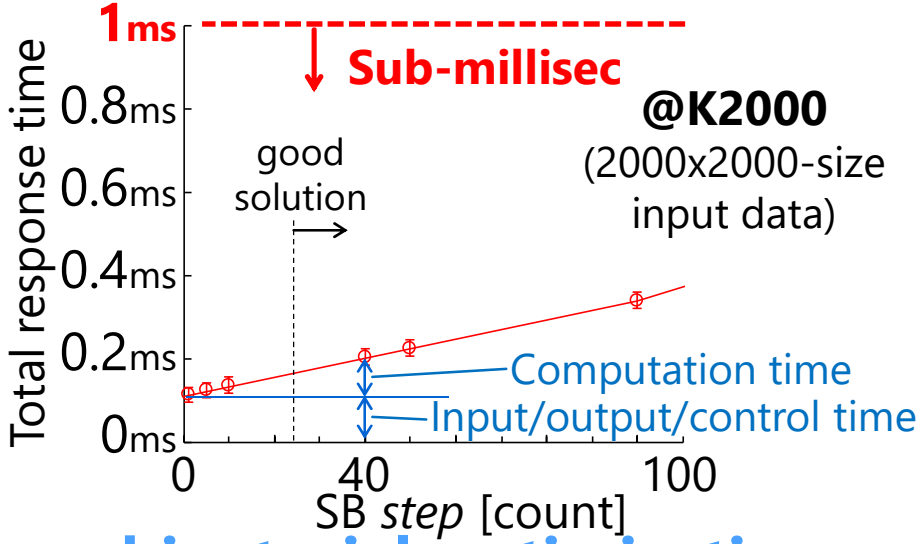
Ising problems	J matrix	h vector	size
MAX-CUT	Ternary, {-1,0,1}	none	various
Arbitrage (path search)	Ternary, {-1,0,1}	FP32	various
Portfolio optimization	FP32	none	various
●●●	**various**	**various**	**various**

Toward "Intelligent real-time response systems"

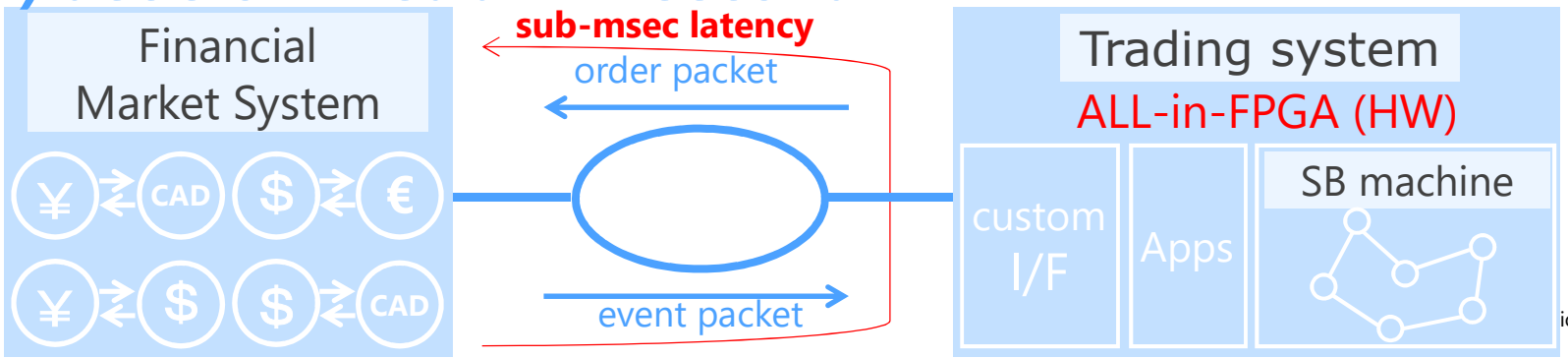
Interface (I/F) is important as well; Be flexible, wide, directly connected to external I/F

Total response time including input/output/control





New concept: Making intelligent (combinatorial-optimization-based) decision in sub-millisecond



Summary

FPGA-based Simulated Bifurcation Machine

An ultra-fast solver for combinatorial optimization problems

Massively-parallel, fully-customized SB accelerator

-the world's fastest, most energy-efficient
(14X faster, 288X more energy efficient than CIM)

Very practical

-No refrigerator, no laser, but just an FPGA
-written in an HLS language, flexible & scalable to meet the diversity of problems

Be a key component for intelligent real-time response systems

-End-to-End HW implementation that can be directly connected to external systems, enabling *optimal* responses in sub-msec