

インテル® FPGA PACで実現する疑似量子計算機 シミュレーテッド分岐マシン™

TOSHIBA

著者

株式会社東芝
研究開発センター
情報通信プラットフォーム研究所
コンピューター&
ネットワークシステムラボラトリー

濱川 洋平

辰村 光介

はじめに

金融取引の最適化や、産業用ロボット動作の最適化、移動経路の最適化など、社会・産業システム上の課題の多くは、膨大な組合せパターンから最良のものを選び出す組合せ最適化問題に帰着します。これは、解の候補数が問題規模の指数関数で増加する、いわゆる組合せ爆発のために、解くことが大変難しい問題として知られています。そのため、組合せ最適化専用計算機の研究開発が国内外で活発に行われています。これらの多くはイジングマシンと呼ばれるもので、磁性体の簡易モデルであるイジングモデルの基底状態を求める問題を解く専用計算機です。多くの組合せ最適化問題がイジング問題に定式化できることから、イジングマシンに高い期待が持たれています。

東芝グループは、従来のデジタル計算機技術を生かしたイジングマシンとして、シミュレーテッド分岐マシン™(Simulated Bifurcation Machine™: SBM) を2019年に発表³しました。これは、量子コンピューターである量子分岐マシン^{1,2}の理論から生まれた、シミュレーテッド分岐(SB)アルゴリズムという並列計算に適した独自のアルゴリズムを、FPGAやGPUを用いて超並列実装したものです。2021年には、約10倍の高速化と、精度・規模の大幅向上を実現した第2世代のSBアルゴリズムを発表しています。⁴

ここで、FPGA実装⁵においては、SBアルゴリズムの並列性を活かした空間並列化に加え、逐次処理部分における時間並列化(パイプライン化)など、SBに特化したカスタマイズの超高密度・超高速実装を行っています。また、完全なハードワイヤー化により、処理時間がクロックサイクル・レベルで決定論的に決まる(予測可能であること)ことも特長の1つです。

目次

はじめに	1
シミュレーテッド分岐マシン™	2
組合せ最適化問題と イジングマシン	2
量子原理から発見された SBアルゴリズム	2
計算機科学からみたSBの特長	3
SBアクセラレーターの設計	4
オンプレミス版サービス	5
サービス概要	5
システム構成	5
回路IP(FPGA構成データ群)	6
ソフトウェア技術者が利用できるAPI	7
リファレンス・デザイン	8
FPGAソリューションのポイント	9
FPGAを利用するメリット	9
OpenCLによる開発	10
まとめ	10
東芝について	10
参考資料	11

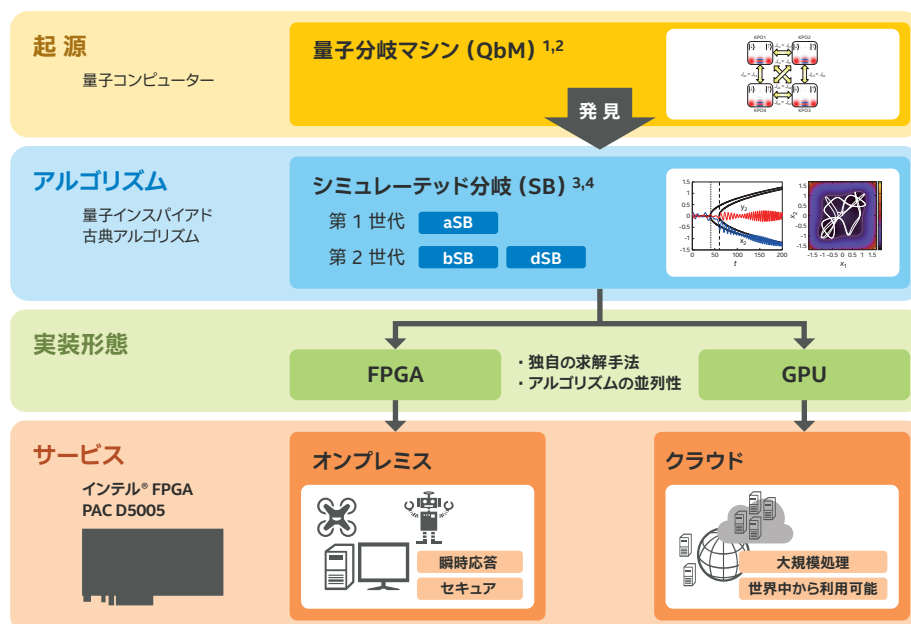


図1. SBサービスの全体構成

これまで、金融や通信、ロボットなどの分野で見られるリアルタイム・システムでは、複雑な組合せ最適化を行うことへの潜在的な要求があっても、特定時間内の応答を要するという時間的制約から、それを実現できませんでした。FPGA 実装 SBM の登場は、「リアルタイム・システムで高度な組合せ最適化を利用できる = 入力に対し瞬時に最適な応答が可能なシステム (瞬時最適応答システム)」を実現できる可能性をもたらしたといえます。実際、リアルタイム・システムの一例として、SB および周辺ロジックや入出力を FPGA に実装し、マイクロ秒級で速度で利益率の高い取引経路を検出可能な、多数通貨間の裁定取引マシンを実機実証しています。⁶

現在、東芝デジタルソリューションズ株式会社から、2 種類の SBM 実装形態 (FPGA、GPU クラスタ) を、それぞれオンプレミス版、クラウド版⁷としてサービス提供しています (図 1)。このうち、オンプレミス版では、インテル® FPGA PAC D5005 で動作する回路データと、C/C++ または Python API を提供します。これにより、一般のソフトウェア技術者が前述の裁定取引マシンのような革新的なシステムを開発可能になります。また、使用する SB アクセラレーターの種類を用途に応じて API 経由で切り替えるなど、FPGA の特長を活かした機能も提案しています。

このホワイトペーパーでは、FPGA 実装 SBM によるオンプレミス版サービスを対象とし、アルゴリズム、回路アーキテクチャ、サービス内容について紹介します。また、複数のバリエーションを持つ回路群をそれぞれ高い実装密度で設計するため、今回 OpenCL 言語を用いた High Level Synthesis (HLS) 設計を採用しました。これらの設計手法も含めた FPGA ソリューションのポイントについても紹介します。

シミュレーテッド分岐マシン™

組合せ最適化問題とイジングマシン

多くの組合せ最適化問題は、イジングモデルのエネルギー最小化問題 (イジング問題) に変換して解くことができます。イジングモデルとは、磁性体の簡易モデルであり、複数のスピン (各スピンは ± 1 のいずれかの値をとる) と、スピン間の相互作用、各スピンに独立に作用する外部磁場から構成されます。イジングモデルのエネルギー (イジングエネルギー $E(s)$) は次の式で定義されます。

$$E(s) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij} s_i s_j + \sum_{i=1}^N h_i s_i$$

ここで、 s_i は i 番目のスピン、 N はスピンの総数、 J_{ij} はスピン間の相互作用の強さ、 h_i は s_i にかかる外部磁場の強さを表します。イジング問題を解くとは、与えられた問題固有の J 、 h に対し、 $E(s)$ を最小にする $s(s_1, s_2, \dots, s_N)$ を求めることに相当します。

イジング問題を効率的に解く専用計算機をイジングマシンと呼びます。イジングマシンを使用して組合せ最適化問題を解く流れを図 2 に示します。

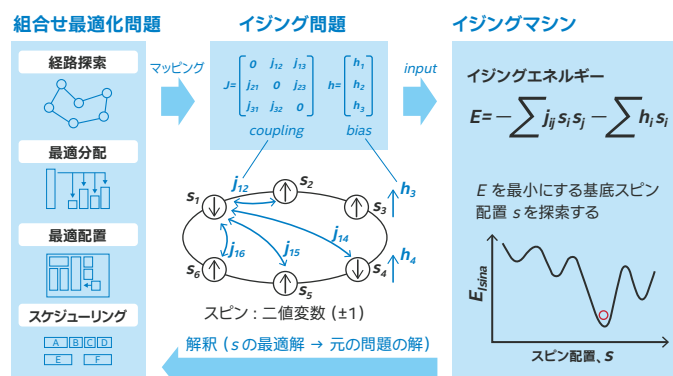


図 2. イジングマシンを使って組合せ最適化問題を解く流れ

量子原理から発見された SB アルゴリズム

シミュレーテッド分岐 (SB) アルゴリズム³は、量子分岐マシン (Quantum bifurcation Machine: QbM)^{1,2}と呼ばれる量子断熱最適化手法に対する古典対応物から導かれた、量子インスパイア最適化計算アルゴリズムです。現在、デジタル計算機 (電子回路) 実装によるイジングマシンでは、求解アルゴリズムとしてシミュレーテッド・アニーリング (SA) をベースにしたものが多く見られます。これに対し、SB は分岐現象を示す古典的な非線形振動子ネットワークの断熱時間発展を数値的に模擬すること (つまり、古典計算機による力学系のシミュレーション) をベースとしたアルゴリズムであることが特徴的な点です。以下、図 3、図 4 を参照しながら、アルゴリズムのポイントを紹介します。

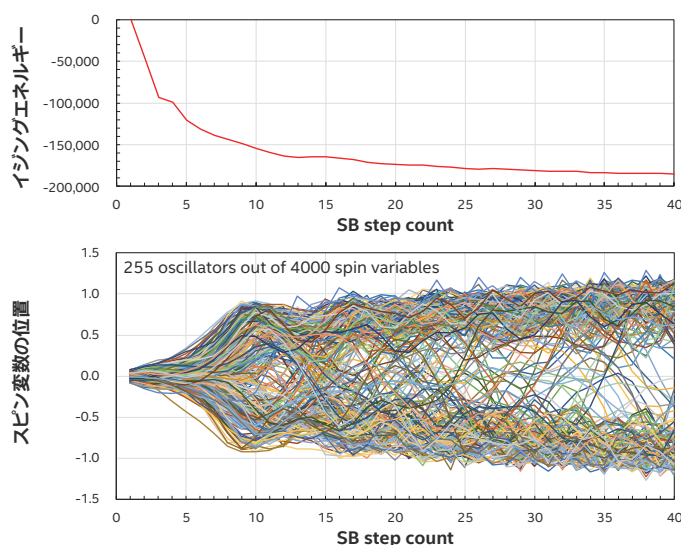
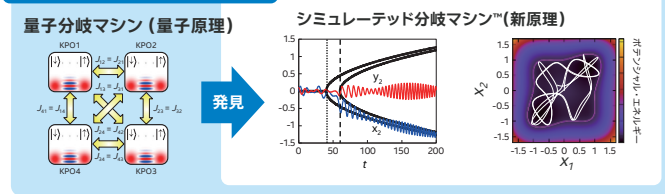
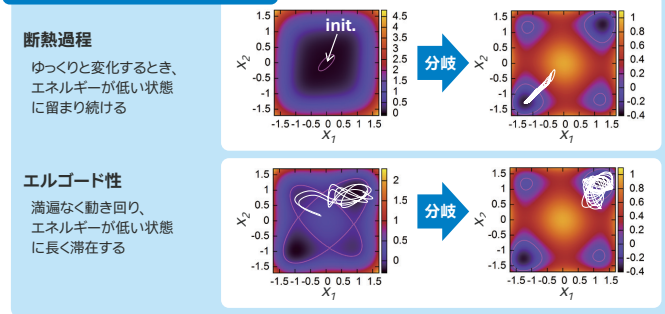


図 3. イジングエネルギー (上) と振動子位置の時間発展 (下)⁵

SB アルゴリズムの発見



解探索のメカニズム



第 2 世代アルゴリズム

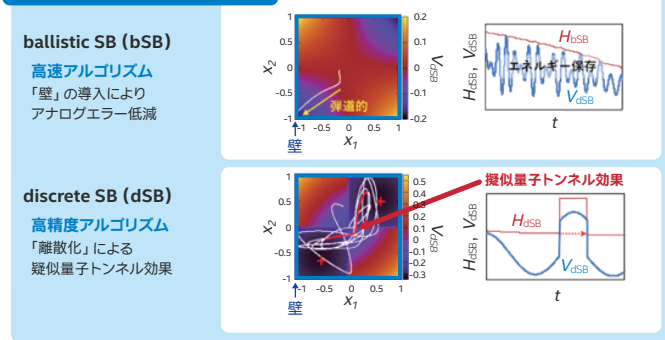


図 4. SB アルゴリズム 1, 2, 3, 4

- 分岐現象：**SB では、1つの振動子 (例えば i 番目の振動子) を実数である位置変数 x_i と運動量変数 y_i によって表現します。N 個の相互作用する振動子は、時間発展ダイナミクスにより ± 1 のいずれかの方向に徐々に分岐しますが、所定の時間発展後の最終的な位置変数 x_i を二値化したものが、イジングモデルにおける i 番目のイジングスピン (s_i) に相当します。ここで、SB のステップ処理 (以降、SB ステップと呼びます) とは、すべての変数 ($\mathbf{x}, \mathbf{y}$) を離散時刻 t_n から次の時刻 t_{n+1} へ更新する手続きのことを言います。図 3 は、ある問題を対象に、SB ステップが進む (時間発展する) につれ振動子が分岐し、この分岐とともにイジングエネルギーが減少する (系全体が基底状態に近い状態に近づく) 様子を示しています。
- 解探索のメカニズム：**SB は、「分岐現象」、「断熱過程」および「エルゴード性」といった古典力学の興味深い現象を、解探索のメカニズムとして利用します。これにより、最適解または近似解 (最適解に近く実用上十分良い解) を動的に探索します。

- 第 2 世代アルゴリズム⁴：**2021 年に、速度と精度を向上させる 2 つのアルゴリズムを提案しました。bSB (ballistic SB) は、従来の 4 次ポテンシャルの代わりに非弾性衝突壁を導入することで、精度低下の原因であったアナログエラーを軽減し、高速化・高精度化を実現します。discrete SB (dSB) は、bSB をベースに、運動方程式中の一部の項で位置 x を二値 (± 1) に離散化したものです。この場合、量子トンネルと似た現象 (疑似量子トンネル現象) が起こることが確認されており、この効果により更なる高精度化を実現しています。

計算機科学からみた SB の特長

SB アルゴリズムは高い計算並列性を持ちます。この高い並列性を生かすことで、SB は並列計算機により容易に高速化できます。ここでは、SB をデジタル計算機に実装する上での特長について列挙します。

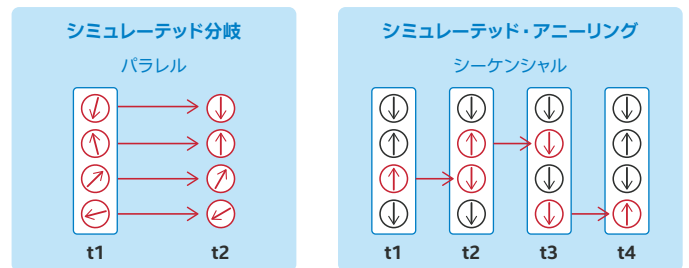


図 5. スピン状態の更新

- 超並列アルゴリズム：**SB アルゴリズムの最上位の並列性は、N 個の仮想粒子の状態を同時に更新可能である点 (同時更新性) にあります。図 5 に、SB における仮想粒子 (スピン) 状態の更新イメージを示します。t1 における N 個の仮想粒子間相互作用を元に現時刻 t2 における y_i が更新され、 y_i によって x_i を更新します。同時刻における x_i と y_i はそれぞれ他の仮想粒子の状態に依存しないため、独立かつ並列に更新可能です。一方、(特別な工夫がないナイーブな) SA では、1つのスピンを反転した場合、次のスピンはその反転後の状態に基づいて決めなければなりません。これは、SB アルゴリズムが並列計算により高速実装できることを意味します。
- 演算の単純性：**SB に必要な演算は和と積のみであり、特殊な関数の演算が不要です。これは、FPGA などによるアルゴリズムのハードワイヤー化にとって望ましい特性です。また、SA アルゴリズムで必要とする「演算過程における疑似乱数の生成」が不要である点も、SB のハードウェア実装に対するメリットの 1 つです ($\mathbf{x}$ と $\mathbf{y}$ の初期値の設定時のみ乱数を使用します)。

SB アクセラレーターの設計

前記の特長を活かし、SBを高速処理するSBアクセラレーターを開発⁵し、FPGAに実装しました。ここでは、SBアクセラレーターの回路アーキテクチャーについて、疑似コード (図6) と回路アーキテクチャー (図7) を参照しながら説明します。

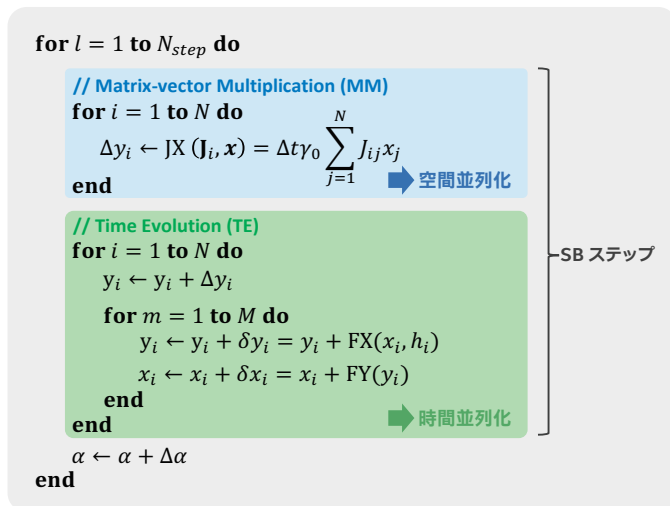


図6. SB疑似コード (文献⁵より一部抜粋し編集)

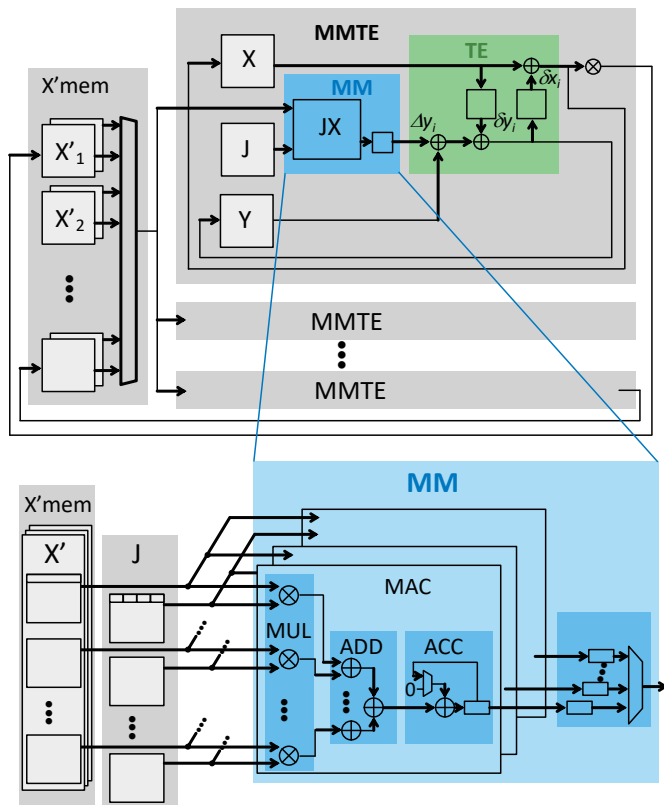


図7. SBアクセラレーターの回路アーキテクチャー⁵

• **構成要素**: 図6の疑似コードに示すとおり、SB全体は、SBステップの繰り返しにより構成されます (繰り返し数はハイパーパラメーターであり、ここでは N_{step} と記載します)。ここで、SBステップは、行列ベクトル積部 (Matrix-vector Multiplication: MM) と時間発展部 (Time Evolution: TE) の2つの部分で構成されます。

• **MM部の高速化 (空間並列化)**: MM部は、各振動子について他の振動子との多体相互作用に基づく運動量の修正量 (Δy_i) を計算します (計算複雑度 $O(N^2)$)。この処理の大部分は積和演算 (MAC) によるものであるため、MACを行う処理エレメント (PE) 回路を並列に複数配置することでMM部の処理を高速化します (空間並列化)。

• **TE部の高速化 (時間並列化)**: TE部は、各振動子について、はじめ運動量を Δy_i だけ更新し、続いて自己の運動量と1体ポテンシャルから計算される修正量 δx_i と δy_i だけ位置と運動量を更新します (計算複雑度 $O(N)$)。これらは逐次的な処理であるため、パイプライン化 (時間並列化) により高速化します。さらに、各仮想粒子間の独立性を利用し、このパイプラインを複数備えることで高速化を加速します。

• **回路アーキテクチャー**: SBアクセラレーターの回路アーキテクチャー⁵を、図7に示します。トップ階層では、N個の仮想粒子を複数のサブグループに分割し、各サブグループに属する仮想粒子の状態更新を担当するモジュール (MMTE: MM + TE) を複数備えています。回路全体としては、前記の時間発展処理を実現するため、MMTEを循環接続する構造をとっています。前述のMM部の空間並列化 (MAC PEの並列配置) は、各PEに十分なデータ供給が可能な特殊なメモリー・サブシステムを構築することにより実現しています。また、MM部とTE部の独立的な高速化に加え、MM部出力段バッファとダブルバッファ構造であるX'memモジュールを利用し、MM部の処理とTE部の処理を時間軸においてオーバーラップさせることでTE部の処理時間を実質的に隠蔽し、さらに全体の処理時間を短縮しています。

SBアルゴリズムは空間並列化と時間並列化それぞれにより高速化できる部分を持ちますが、各高速化手法を同時に具現化できるのはFPGA特有のメリットと言えます。また、このアーキテクチャーをベースにFPGA実装された第2世代SBアルゴリズムであるbSBとdSBは、種々のベンチマーク問題で極めて高い性能を示しています。⁴

オンプレミス版サービス

サービス概要

オンプレミス版のSBM®は、オンプレミス型のユーザーシステム（サーバーやデスクトップPC、エッジ機器）で動作するシミュレーテッド分岐マシン™です。市販の汎用FPGAボードであるインテル® FPGA PAC D5005で動作するSB専用高速処理回路と周辺ソフトウェアを提供します。クラウドサービスとの大きな違いは下記のとおりです。

- SBアルゴリズムを、ユーザーシステムに組み込み可能（システム単独動作が可能）
- SBアルゴリズムを、FPGA（インテル® FPGA PAC D5005）で動作する専用高速処理回路（SBアクセラレーター）で実装
- システム内のSBアクセラレーターに、ネットワーク接続なしでダイレクトにアクセス可能

これらのSBアクセラレーターとオンプレミス型システムの組合せにより、超高速処理・瞬時応答（低遅延）を実現できることがオンプレミス版の大きな特長です。このような特長から、オンプレミス版は図8に示すような用途に適しています。

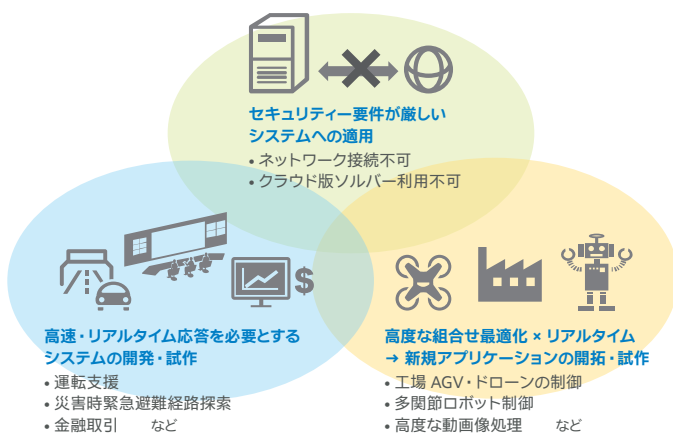


図8. オンプレミス版のSBMの用途例

システム構成

オンプレミス版のSBMによる応用システムの構成例を、図9に示します。ユーザーが所有するハードウェアと、東芝グループから提供するソフトウェア一式によりシステムが構成されます。

- ハードウェア（ユーザーが用意）
 - ホストサーバー（サーバーまたはデスクトップPC）
 - FPGA ボード（インテル® FPGA PAC D5005）
- ソフトウェア（東芝グループから提供）
 - 回路IP（Intellectual Property）
 - ソフトウェア・ライブラリー（APIを提供）
 - リファレンス・デザイン

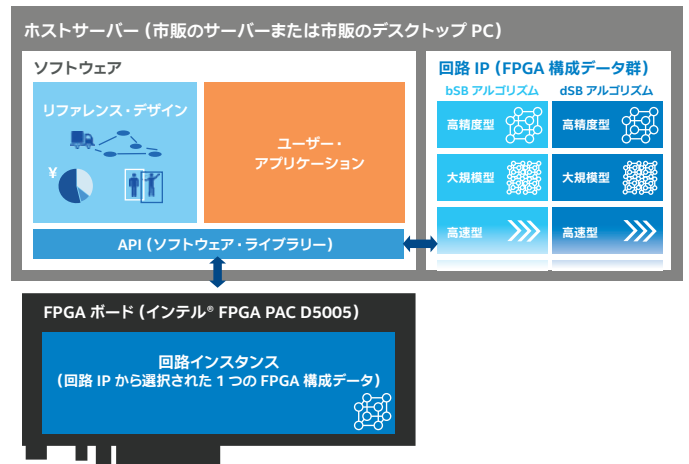


図9. オンプレミス版のシステム構成

SBアクセラレーターは、FPGA 構成データ（インテル® FPGA PAC D5005 向けバイナリーデータ）として提供します。ここで「回路インスタンス」は1つのFPGA 構成データを、「回路IP」は複数個の回路インスタンスからなるデータセット全体を指します。ユーザーが用意するハードウェアは、サーバーまたはデスクトップPC（パソコン）とFPGA ボード（インテル® FPGA PAC D5005）だけであり、ほかの特殊な機材は必要ありません。ソフトウェア・ライブラリーは、FPGA に対する一連で複雑な処理を隠蔽し、C/C++ およびPythonで利用できるAPIを提供します。これにより、一般のソフトウェア・エンジニアが、ハードウェア構成を意識することなく簡単にイジングマシンを扱うことができます。

回路 IP (FPGA 構成データ群)

バリエーション

第2世代のSBアルゴリズムにはbSBとdSBの2つがあり、それぞれ特長が異なります。例えば、bSBは少ないステップ数で良解（高精度な近似解）を見つける高速アルゴリズム、dSBはより高精度な解を見つける高精度アルゴリズムであることが特徴的な点です。FPGAのリソースを最大限に利用しながら、用途に応じてアルゴリズムを使い分けられるよう、アルゴリズムごとに最適化した回路インスタンスを用意しました。また、FPGAの上限リソースは決まっているため、図7の並列数（主にMAC PE数を指し、高速化の度合いを示す）やスピン数、スピン間結合係数のビット幅の大きさは、トレードオフの関係にあります。そこで、ユーザーがその用途によりこれらの特性を選択できるよう、スピン数（規模）やスピン間結合係数精度（精度）の異なる複数の構成（回路インスタンス）を用意しています。本ホワイトペーパー執筆時点で用意している回路インスタンスは表1のとおりです。ここで T_{step} は、SBMステップ当たりの処理時間で、小さいほど高速であることを意味します。

表 1. 回路インスタンス一覧

id	アルゴリズム	スピン数	スピン間結合係数精度 (bit)	T_{step} (usec)
1	bSB	2048	32 (FP)	4.301
2	bSB	2048	1	0.683
3	dSB	2048	32 (FP)	4.349
4	dSB	2048	2	0.359
5	dSB	8192	2	3.645

速度性能

2021年2月に米国オンライン科学雑誌「Science Advances」に掲載された論文⁴では、種々のベンチマーク問題において、第2世代のSBアルゴリズムが世界最先端のイジングマシンと比べても高い競争力を持つことが示されています。オンプレミス版で提供する回路は、市販の汎用FPGAカード（インテル® FPGA PAC D5005）で動作し、実用的・汎用的機能実現のためユーティリティ回路を備えたものでありながら、同論文の実証データと同等の速度性能を有しています（図10）。

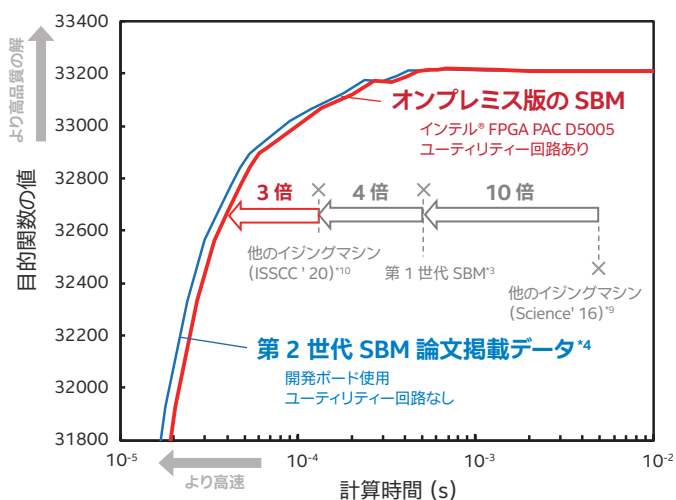


図 10. 速度性能(2000 スピン全結合問題ベンチマーク)^{3,4,9,10}

ソフトウェア技術者が利用できる API

前節のとおり、オンプレミス版のSBMでは複数の回路インスタンスを提供します。ハードウェアに近い層では、これらはそれぞれ異なる手順で制御する必要があります。今回、これらの個別かつ複雑な手順を隠蔽・抽象化し、イジングマシンとして統一的なインターフェイスを提供するためにAPIを用意しました。このAPIは、統一的・簡易的であることを前提に、ユーザーのさまざまな設計レベルに対応できるように設計しており、以下に示すような3つの特長を持っています。

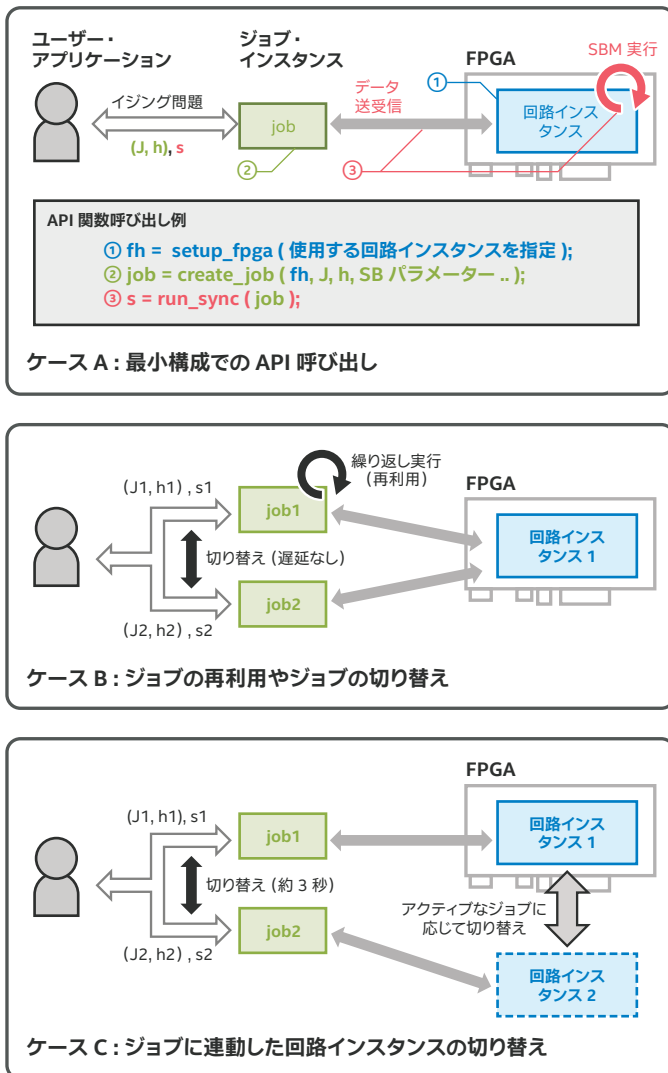
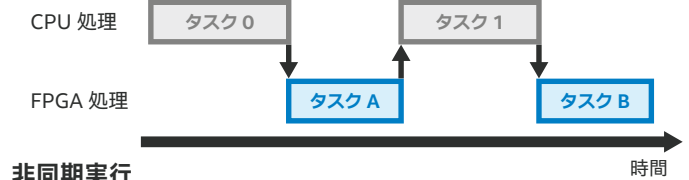


図 11. APIおよびソフトウェア・ライブラリーの動作フロー

同期実行



非同期実行

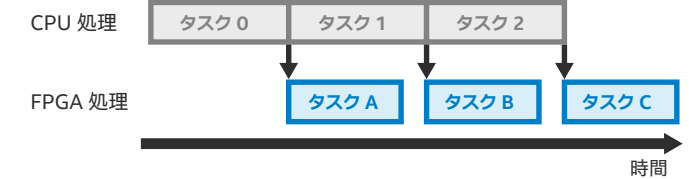


図 12. CPU 処理と FPGA 処理の同期実行および非同期実行

- 簡索性 (Simplicity):** 単体のイジング問題を解くために必要な API 関数は、わずか3つであり、簡単にイジングマシンを使用することができます。図 11 のケース A に、関数呼び出し例とその動作イメージを示します。初めに① `setup_fpga` 関数を呼び出すことで、回路インスタンスを選択して FPGA に書き込みます。次にユーザーが解きたいイジング問題を指定するために② `create_job` 関数を呼び出します。最後に、FPGA 上の SBM でイジング問題を解くために③ `run_sync` 関数を実行します。
- 柔軟性 (Flexibility):** 問題設定を抽象化したデータ概念(ジョブ)を導入しています。図 11 のケース B のように、一度作成したジョブの再利用や複数の異なるジョブをスイッチングさせるなど、ジョブ構成の工夫により時間効率の良いプログラム実装が可能です。また、図 12 に示す非同期実行をサポートし、CPU 処理と FPGA 処理のオーバーラップにより FPGA 処理時間を隠蔽することも可能です。
- 独自性 (Uniqueness):** ここでの独自性とは、他のソリューションにはない FPGA 特有の機能である動的パースナル・リコンフィギュレーションを活用する API を備えていることを指します。API 関数や図 11 のケース C に示すジョブ連動切り替えの仕組みにより、アプリケーション内で回路インスタンスを切り替えることができます。この切り替えは、システムをリブートすることなく稼働状態を維持しながら行うことができます。切り替えに要する時間は約 3 秒であり、用途により十分実用的であると考えられます。この仕組みにより、外部状況の変化に応じて最適な回路インスタンスに動的に切り替えることができます。

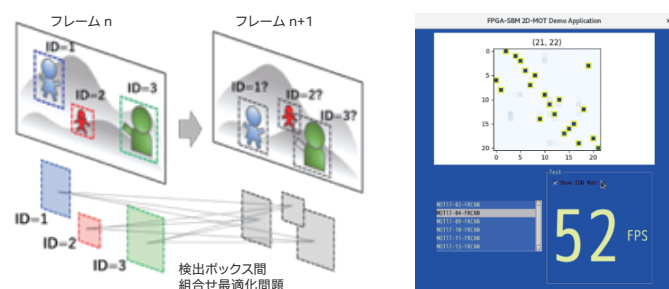
リファレンス・デザイン

3種類のリファレンス・デザインを用意しました。ユーザーは、リファレンス・デザインのソースコードをベースにシステム開発をスムーズに進めることができます。また、このリファレンス・デザイン自体が、オンプレミス版のSBMのリアルタイム動作可能性の実証例となっています。以下で、リファレンス・デザインの内容を説明します。

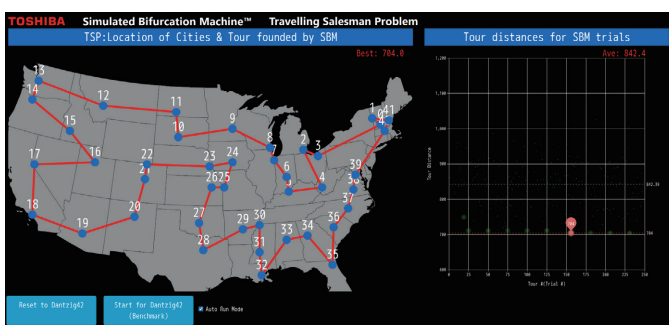
- **複数物体追跡¹²**：既存の複数物体追跡手法¹¹をベースに、動画像中の複数物体追跡を実装したアプリケーション例です（図13(a)）。追跡処理の一部は、フレーム間の物体の対応関係が最ももともらしい組合せを求める2部グラフ最大マッチング問題に帰着します。この最適化問題をイジング問題に定式化し、1フレームごとに解を求めます。システムトータルで30フレーム/秒以上の処理速度を実現しており、リアルシステムへの組込み可能性を最も端的に示す例と言えます。

- **インタラクティブ最短巡回経路探索¹³**：「巡回セールスマン問題」は、都市の集合においてすべての都市を一度だけ訪れる巡回路のうち総距離が最短である経路を求める、有名な組合せ最適化問題です。図13(b)に示すアプリケーションは、任意の位置に配置される42都市の巡回セールスマン問題を対象としています。Graphical User Interface (GUI) からユーザーが都市位置をマウスのドラッグ・アンド・ドロップで指定後、即座に経路探索を開始し、1回当たり約7msの高速探索を複数回実行しながら更新された最短経路を順次画面表示します。位置が変化し続けるロボット群の移動パターンや通信経路の最適化などへの応用が考えられます。

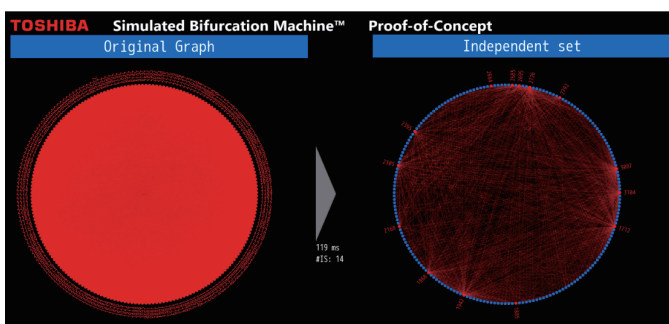
- **ストリームデータ処理型最大独立集合検出¹⁴**：株式データから、銘柄間の値動きの相関を表すグラフ（市場グラフ）を生成し、最大独立集合問題を解くことで多様化ポートフォリオの構成候補となる銘柄セットを見つけるアプリケーション例です（図13(c)）。最大独立集合問題とは、ある集合から互いに相関がない（最大の）部分集合を求める、Non-Deterministic Polynomial (NP) 困難に属する組合せ最適化問題です。得られる銘柄セットは、互いに値動きの相関が低いため、個別の銘柄の値動きに左右されない安定した株式ポートフォリオを構築できます。開発したアプリケーションでは銘柄数（上限は2,048）によらず、約120msで良解を返すことができ、刻一刻と変化するストリームデータ処理への適用可能性を示す実例となっています。



(a) 複数物体追跡



(b) インタラクティブ最短巡回経路探索



(c) ストリームデータ処理型最大独立集合検出

図 13. リファレンス・デザインとして開発したアプリケーション

FPGA ソリューションのポイント

これまでFPGAに実装したSBMのアーキテクチャーやオンプレミス・サービスについて紹介してきました。ここでは、今回のFPGAによるソリューション事例を振り返り、そのポイントを議論します。

FPGA を利用するメリット

設計自由度の高さ(対CPU、GPU)

FPGAでは、ターゲットとなる処理の特性に応じ、論理回路レベルの粒度でハードウェアを最適化することができます。例えば図7のアーキテクチャーは、SBの特性を活かした並列処理実装(空間並列化)が特長の1つです。このような並列処理実装を十分生かすため、全MACに毎サイクル新しいデータを供給(広範囲にブロードキャスト)可能な特殊なメモリー・サブシステムをカスタム設計しています。さらに、SBの一部に存在する(空間並列化が不可能な)逐次処理部については、時間並列化(パイプライン化)により、極限まで高速化を突き詰めた実装を行っています。このような設計自由度の高さが、CPUやGPUと比較したときのFPGAのメリットの1つであり、ターゲット(タスク)の特性によっては大幅に性能を向上させることが可能です。

動的再構成可能なハードウェア

SBには特長が異なる複数のアルゴリズム(第2世代の場合、bSBとdSB)が存在すること、用途によって規模(スピン数)・速度・精度に対し重視する特性が異なることから、イジングマシンの具体的構成は複数種類考えられます。今回は、ユーザー自身が使用する構成を選択できるよう、それぞれのマシン構成ごとに最適化した回路インスタンスを用意しました。これらの回路インスタンスは、パーシャル・リコンフィグ(部分再構成)できるよう設計しており、システム稼働を維持しながら(APIを通じて)数秒で回路切替が可能です。これは、FPGA特有の動的再構成機能を顕著に活かしたシステム仕様と言えます。もちろん、今後のSBアルゴリズムの進化や新しいイジングマシン構成(ビット幅バリエーションの追加など)の登場に応じ、同一ハードウェア(インテル® FPGA PAC D5005)上で動く回路インスタンスを適宜リリースできる点も大きなメリットです。

汎用高速インターフェイスを持つインテル® FPGA PAC

インテル® FPGA PAC D5005は、ハイエンドのFPGAデバイス(インテル® Stratix® 10 SX FPGA)を搭載し、PCI Express(PCIe) Gen3x16のインターフェイスを持つFPGAボードです。SBMの演算に必要なJ行列データ(イジングモデルのスピン間結合係数を表す2次元行列)は、演算を開始する前にホストからFPGAにコピー(転送)されます。このJ行列は全結合イジングモデルでは巨大なサイズになるため、オフロード時のレイテンシーを小さくするためには高速なバス転送が必要です。今回のソリューションの場合、PCIe Gen3x16(片側15.75GB/s)の帯域をフルに活用して高速転送を行うことができます。このインターフェイスを使うのに、独自のハードウェアやソフトウェアを設計する必要はなく、インテル® FPGA PACの共通フレームワークの中で提供されるI/O回路やドライバー・ライブラリーを利用できます。この共通フレームワークの存在は、「一般的なサーバーやPCに差すだけ」で使えるなど、ユーザーにとっても大きなメリットがあります。

OpenCLによる開発

今回のソリューションでは、高位設計開発環境(インテル® FPGA SDK for OpenCL ver 19.2)を用い、High Level Synthesis (HLS)言語であるOpenCLを使用しました。

高い設計生産性

Verilog-HDLやVHDLといったハードウェア記述言語によるRegister Transfer Level(RTL)設計と比べ、C/C++をベースとするOpenCL言語では高抽象度の設計が可能です。このため、レジスターレベルでの設計が不要で、コード量が少ない、などのメリットがあります。図7のアーキテクチャーの場合、ベースのアーキテクチャー検討から、並列処理部の回路構成パラメーターの検討など多数の試行回数をともなう開発を短期間で行うことができました。さらに、今後リリースを検討するものも含めたバリエーション展開(アルゴリズム改善や、J行列結合係数のビット幅、スピンの数などの新しい組合せ)の設計を短期間で行うことも可能です。

エミュレーターによる高速なテスト

OpenCL開発環境では、デバイス(FPGA)とそれを制御するホスト(CPU)を統一的に(同じ言語で)設計します。コード設計のみならず、テスト～実機動作まですべての開発工程で共通の開発環境を利用できるため、開発環境の整備・管理の容易さに加え、テスト工程の準備・実行が簡単に行えることも大きなメリットです。特に、インテル® FPGA SDK for OpenCLではエミュレーション・モード(エミュレーター向けにOpenCLデバイスコードを仮合成し、エミュレーションを行う)が用意されています。エミュレーターを用いることで、(数時間かかるような)ビットストリームまでの合成を行うことなく、回路動作確認レベルのテストを短TATで繰り返し実行することが可能です。

高密度・高速な回路を実現

一般的に、RTL設計と比べた場合、HLS設計は、(その高い設計生産性の代償として)十分に回路性能が引き出せないケースがあると言われています。今回、我々は通常のHLS設計フローの中ではケアしにくい配置配線の効率向上を行う独自のノウハウを確立し、適用しました。その結果、HLS設計でありながら、RTL設計の場合と同等以上の回路性能(表2)を実現しました。

表2では、条件の異なる2例の合成結果を示しています。ここで、「# of MAC PEs」は、図7におけるMAC PEの合計数を指します。MAC PEは行列ベクトル積計算で使いますが、ここはアルゴリズム全体の計算量に占める割合が大きい部分です(計算複雑度 $O(N^2)$)。そのため、空間並列数を示すMAC PEの数が、並列実装による高速化度合いを測る主な指標になります。J:32bit FPの例(左側)ではBRAMとDSPの使用量が大きく、J:1bitの例(右側)では「# of MAC PEs」とLogicの使用量が高いのが特徴的です。いずれもデバイス上限に近いリソースを使用しながら、高い F_{max} を実現しています。また、実際の演算性能としてもJ:32bit FPで $2.42 TFLOPS_{peak}$ 、J:1bitで $6,141 GMACS^{-1}$ と、いずれも極めて高い性能を示しています。

表2. FPGA 合成結果

インテル® Stratix® 10 GX2800 FPGA	bSB J:32bit FP	bSB J:1bit
Architecture		
Spin size	2,048	2,048
# of MAC PEs	4,096	32,768
Resource (usage)		
Logic utilization	359,061 (38%)	724,657 (72%)
BRAM	9,395 (80%)	2,299 (20%)
DSP	4,994 (87%)	258 (4%)
Performance		
F_{max} [MHz]	259	292
$TFLOPS_{peak}$	2.42	-
$GMACS^{-1}$	975	6,141

まとめ

リアルタイム・システムにおいて大規模組合せ最適化を可能とする、FPGA実装によるSBアクセラレーターを開発しました。また、インテル® FPGA PAC D5005で動作する複数のSBアクセラレーター(回路IP)とAPIを提供するソフトウェア・ライブラリーをパッケージングし、オンプレミス版サービスとして研究用途向けにリリースを開始しています。今回のソリューションにおいては、HLS(OpenCL)設計手法の活用と独自のノウハウにより、高い性能を有する複数バリエーションの回路を短期間で開発を行うことができました。

今後も、スケールアウト技術¹⁵の導入など、継続的な機能・性能向上を行いながら、幅広い分野での実応用開拓や、新たな価値の共創に取り組んでいきます。

東芝について

東芝グループは、「人と、地球の、明日のために。」を経営理念の主文に掲げ、インフラ事業や電子デバイス事業など幅広い領域で事業を展開するインフラサービスカンパニーです。創業以来永年にわたり培ってきた製造業としてのフィジカル技術と、情報処理やデジタル・AI技術といったサイバー技術を融合したサイバー・フィジカル・システム(CPS)テクノロジーを駆使して、気候変動などSDGsに代表される社会課題やお客様が直面する課題解決に貢献します。そして、企業として持続的に発展するためESGの取り組み強化に努め、脈々と受け継がれるベンチャースピリットを胸に新しい未来を始動させることが、私たちの存在意義です。

参考資料

1. H. Goto, Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network, Scientific Reports 6, 1-8, 2016.
<https://doi.org/10.1038/srep21686> (英語)
2. H. Goto, Quantum Computation Based on Quantum Adiabatic Bifurcations of Kerr-Nonlinear Parametric Oscillators, Journal of the Physical Society of Japan, vol.88, 061015, 2019.
<https://doi.org/10.7566/JPSJ.88.061015> (英語)
3. H. Goto et al., Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems, Science Advances 5, eaav2372, 2019.
<https://doi.org/10.1126/sciadv.aav2372> (英語)
4. H. Goto et al., High-performance combinatorial optimization based on classical mechanics, Science Advances 7, eaab7953, 2021.
<https://doi.org/10.1126/sciadv.aab7953> (英語)
5. K. Tatsumura et al., FPGA-Based Simulated Bifurcation Machine, IEEE International Conference on Field Programmable Logic and Applications (FPL), 59-66, 2019.
<https://doi.org/10.1109/FPL.2019.00019> (英語)
6. K. Tatsumura et al., A Currency Arbitrage Machine based on the Simulated Bifurcation Algorithm for Ultrafast Detection of Optimal Opportunity, IEEE International Symposium on Circuits and Systems (ISCAS), 1-5, 2020.
<https://doi.org/10.1109/ISCAS45731.2020.9181114> (英語)
7. 東芝デジタルソリューションズ「組合せ最適化ソルバー Simulated Bifurcation Machine」
<https://www.global.toshiba/jp/products-solutions/ai-iot/sbm.html>
8. 濱川 洋平、日高 亮、辰村 光介「オンプレミス版シミュレーテッド分岐マシン〜リアルタイムシステムに組み込める疑似量子計算機〜」東芝レビュー、76、5、p49-52、2021。
<https://www.global.toshiba/content/dam/toshiba/jp/technology/corporate/review/2021/05/f01.pdf>
9. T. Inagaki et al., A coherent Ising machine for 2000-node optimization problems, Science 354, 603-606, 2016.
<https://doi.org/10.1126/science.aah4243> (英語)
10. K. Yamamoto, et al., "STATICA: A 512-spin 0.25 M-weight full-digital annealing processor with a near-memory all-spin-updates-at-once architecture for combinatorial optimization with complete spin-spin interactions," 2020 IEEE Int. Solid-State Circuits Conf.-(ISSCC), IEEE, pp.138-140, 2020.
11. Bewley, A. et al., "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP 2016). Phoenix, AZ, 2016-09, IEEE. 2016, p.3464-3468.
12. Toshiba Corporation. 2021.
On-premises simulated bifurcation machine™: multiobject tracking by solving the maximum matching problem.
<https://www.youtube.com/watch?v=bWMZjwHtm5g>
13. Toshiba Corporation. 2021.
On-premises simulated bifurcation machine™: a user-interactive interface for solving the travelling salesman problem.
<https://www.youtube.com/watch?v=NSh3nQTpOW8>
14. Toshiba Corporation. 2021.
On-premises simulated bifurcation machine™: stream data processing of market graph for finding the diversified portfolio through solving the maximum-independent-set problem.
<https://www.youtube.com/watch?v=WIDoO-DJClo>
15. K. Tatsumura et al., Scaling out Ising machines using a multi-chip architecture for simulated bifurcation, Nature Electronics 4, 208-217, 2021.
<https://doi.org/10.1038/s41928-021-00546-4> (英語)
16. パフォーマンス測定に使用したシステム構成を示します。サーバー: HPE ProLiant DL380 Gen10、CPU (インテル® Xeon® Silver 4215R プロセッサ)、MEM(256GB)、SSD(960GB)、BIOS(U30 v2.42)、PCIe(Gen3 x16) [<https://www.hpe.com/jp/ja/product-catalog/servers/proliant-servers/pip.specifications.hpe-proliant-dl380-gen10-server.1010026818.html>]、OS: CentOS Linux release 7.9.2009 (Core)。



性能は、使用状況、構成、その他の要因によって異なります。詳細については、<http://www.intel.com/PerformanceIndex/>(英語) を参照してください。

性能の測定結果は、システム構成に記載された日付時点のテストに基づいています。また、現在公開中のすべての更新プログラムが適用されているとは限りません。構成の詳細については、補足資料を参照してください。絶対的なセキュリティを提供できる製品またはコンポーネントはありません。インテルは、サードパーティーのデータについて管理や監査を行っていません。ほかの情報も参考にしてデータの正確さを評価してください。

実際のコストや結果は異なる場合があります。

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

Intel、インテル、Intel ロゴ、Stratix、Xeon は、Intel Corporation またはその子会社の商標です。

その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

©2021 Intel Corporation. 無断での引用、転載を禁じます。

349407-001JA