

資料6-1

AIチップ開発加速のためのイノベーション推進事業

研究開発項目②

「AIチップ開発を加速する共通基盤技術の開発」全体概要

研究開発責任者 内山邦男（産総研）

1. 概要

- ・ 研究開発の背景と狙い
- ・ AIチップ設計拠点の体制・運営・利用形態
- ・ 実施項目・体制・スケジュール

2. 成果

- ・ 事業目標と実績
- ・ 拠点の構築と運営
- ・ 外部発表、人材育成

3. 2023年度からの拠点運営

- ・ 運営方針
- ・ 利用方法・利用例・実績

1. 概要：研究開発の背景と狙い

- ✓ 我が国では、ベンチャー企業等を中心に、AIチップを基にした新たなビジネスを創出させる種が多数存在。
- ✓ 一方、AIチップ設計には、高額なEDAツールやIP、検証装置(エミュレータ等)が必要であり、これらがビジネス化に向けた高いハードルとなっている。
- ✓ AIチップ設計に必要な**共通基盤技術開発**と**拠点整備**により、イノベーション実現のためのAIチップ開発を加速する。

革新的AIチップ のアイデア

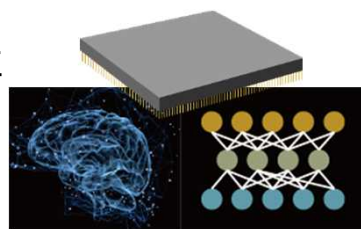


国内中小企業
ベンチャー企業

高額なEDAツール、
IP、検証装置が必要

高いハードル

AIチップ プロトタイプ試作



学習、推論、認識を
低電力かつ高速に

超スマート社会 (Society5.0) の実現

- ・次世代モビリティ
自動運転, 無人配送, ...
- ・次世代ヘルスケア
AI診断, 自動モニタリング, ...
- ・次世代サプライチェーン
スマート保安, 無人工場, ...
- ・農林水産業スマート化
無人農業車両, 水中ロボット, ...
- ・FinTech
:

AIチップ設計拠点

本事業において、

共通基盤技術の開発し、拠点を構築

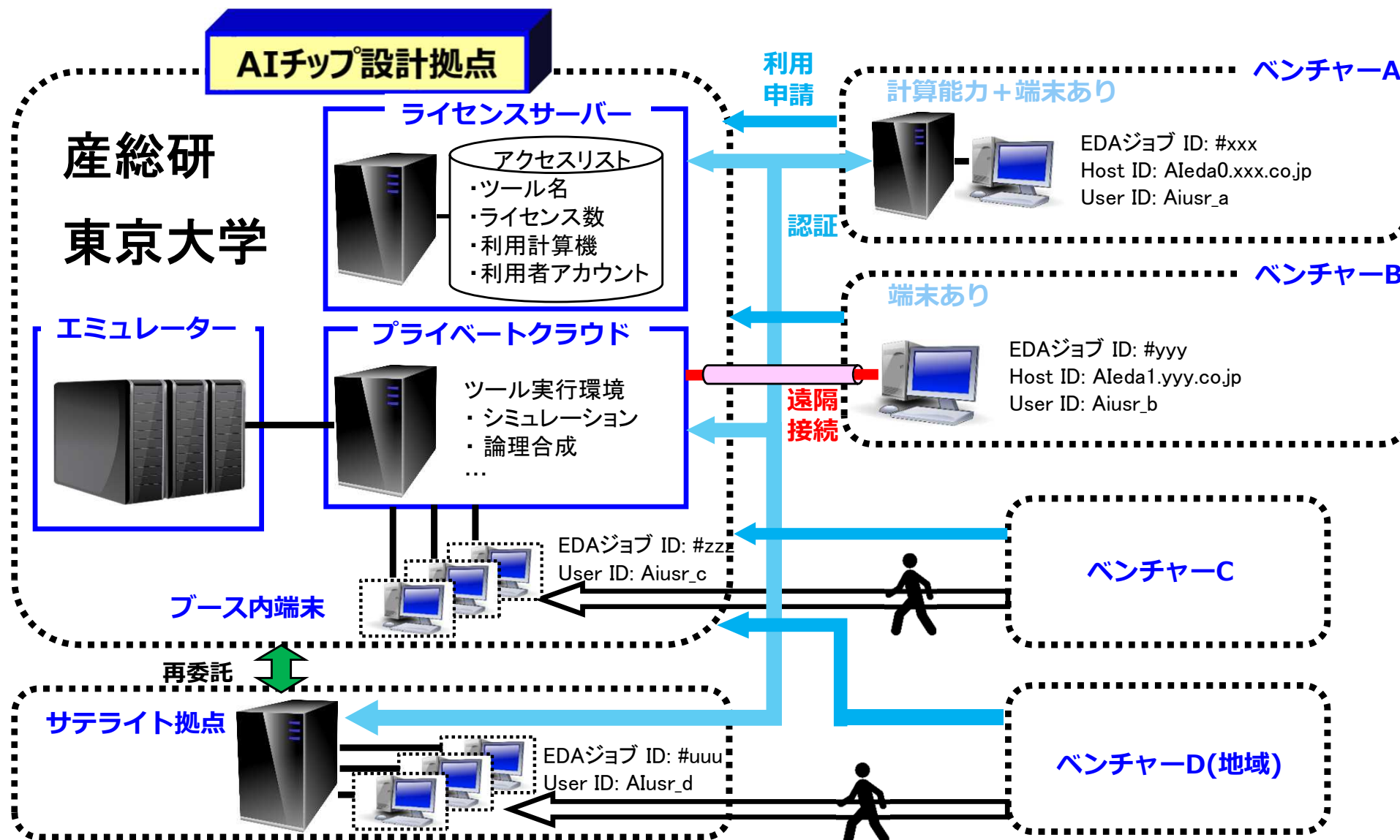
1. 概要： AIチップ設計拠点の体制と運営

赤字: AIチップ設計拠点 運営組織



1. 概要： AIチップ設計拠点の利用形態

- ✓ 企業毎の設計環境に応じた拠点利用形態を整備し、中小・ベンチャー企業等が使い易い拠点をめざす



1. 概要：研究開発の実施項目

実施項目1：AIチップ開発に必要な共通基盤技術の研究開発

1-1 AIチップ向け設計
ツールの研究開発

1-2 ハードウェア開発
垂直立ち上げ実現の
ための研究開発

1-4 センサ機能を含むチップ
のための新規デバイスモデル
の研究開発

1-5 国内外FABの活用
と最適化ライブラリ
の研究開発

検証装置とシミュレータの協
調設計フロー

ソフトウェアからハードウェア
設計を一気通貫に実現でき
るツールチェーン

センサーなどの機能素子
モデル化技術

国内外FABとの連携
AI-One/Twoプラットフォーム

1-3 AIチップ設計に向けた
リファレンス設計の研究開発

AIチップ
リファレンス設計



アイデア

第1の壁

ハード化
(半導体化)の壁

小規模RTL

大規模化の壁

第2の壁

大規模RTL

第3の壁

物理設計
特殊機能
の具現化の壁

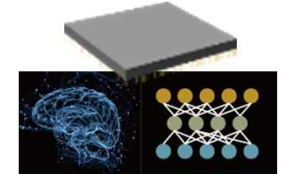
2-2 人材育成と拠点機能の整備

1-1～1-5の共通基盤技術を活用した教育カリキュラムの
整備および、AIチップ設計開発のための教育を行う

2-1 AIチップの研究開発に必要なEDAツールの整備

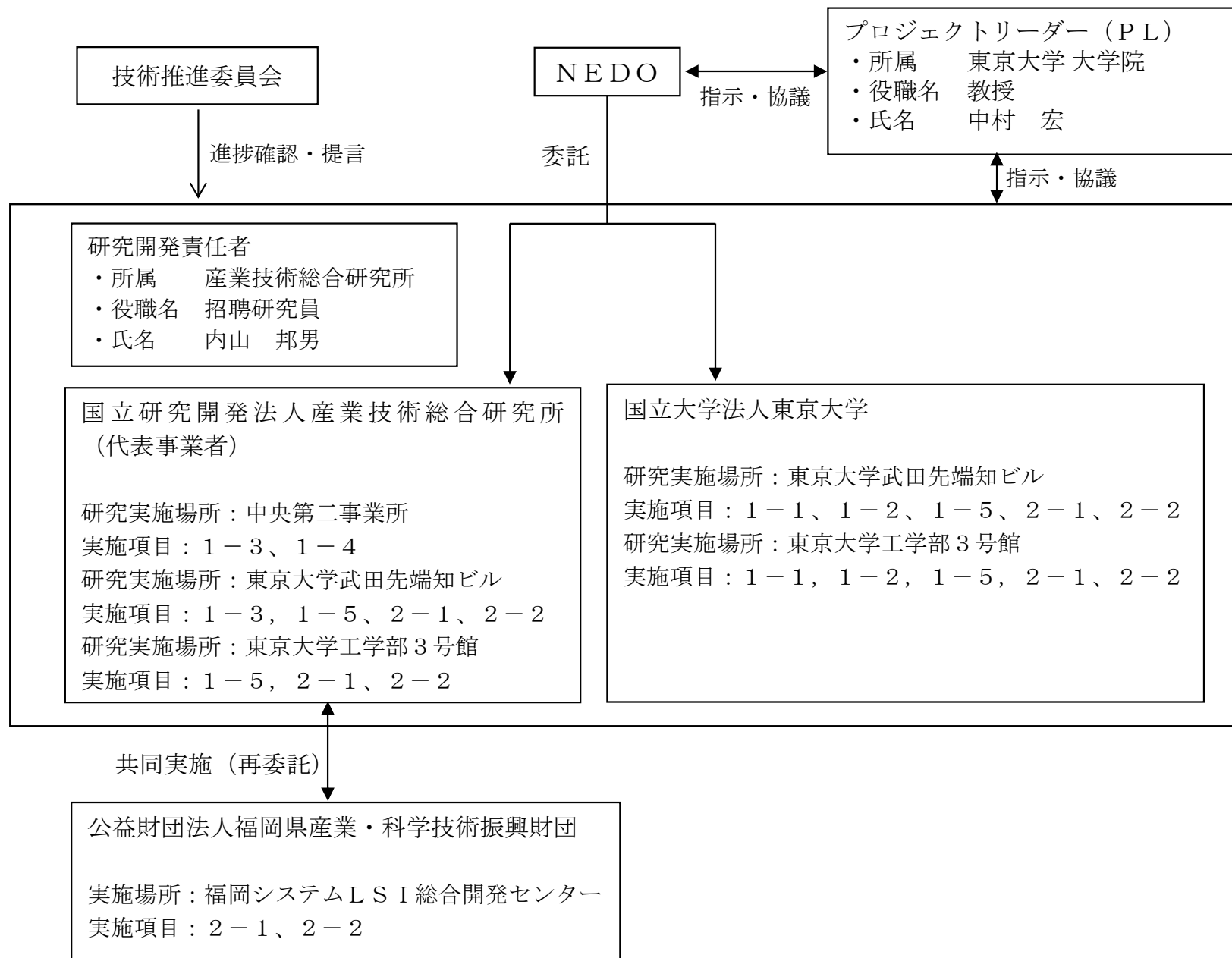
高額な回路設計ツール群を整備し、中小・ベンチャー企業等からの拠点
活用事例を通じ、広く活用されるような環境として提供する

AIチップ

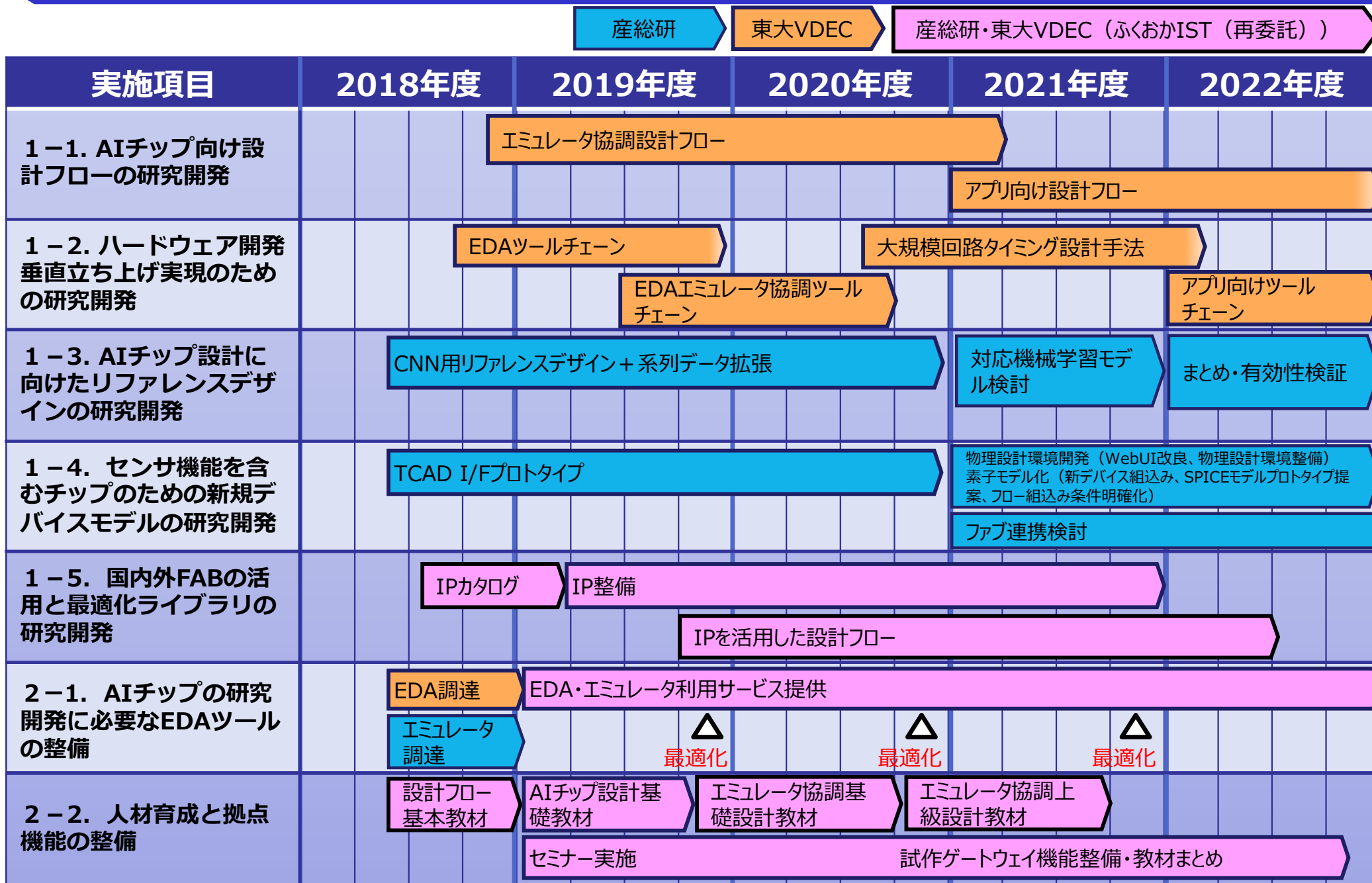


実施項目2：AIチップ開発拠点の整備

1. 概要：研究開発の実施体制



1. 概要：研究開発のスケジュールと進捗



[中間目標](令和2年度:2020年度)

本事業を通じて開発、整備したAIチップ設計のための共通基盤技術、学習環境、設計環境の活用件数**10件以上**を目標とする。

⇒ 2021年3月時点で**35件**の実績

[最終目標](令和4年度:2022年度)

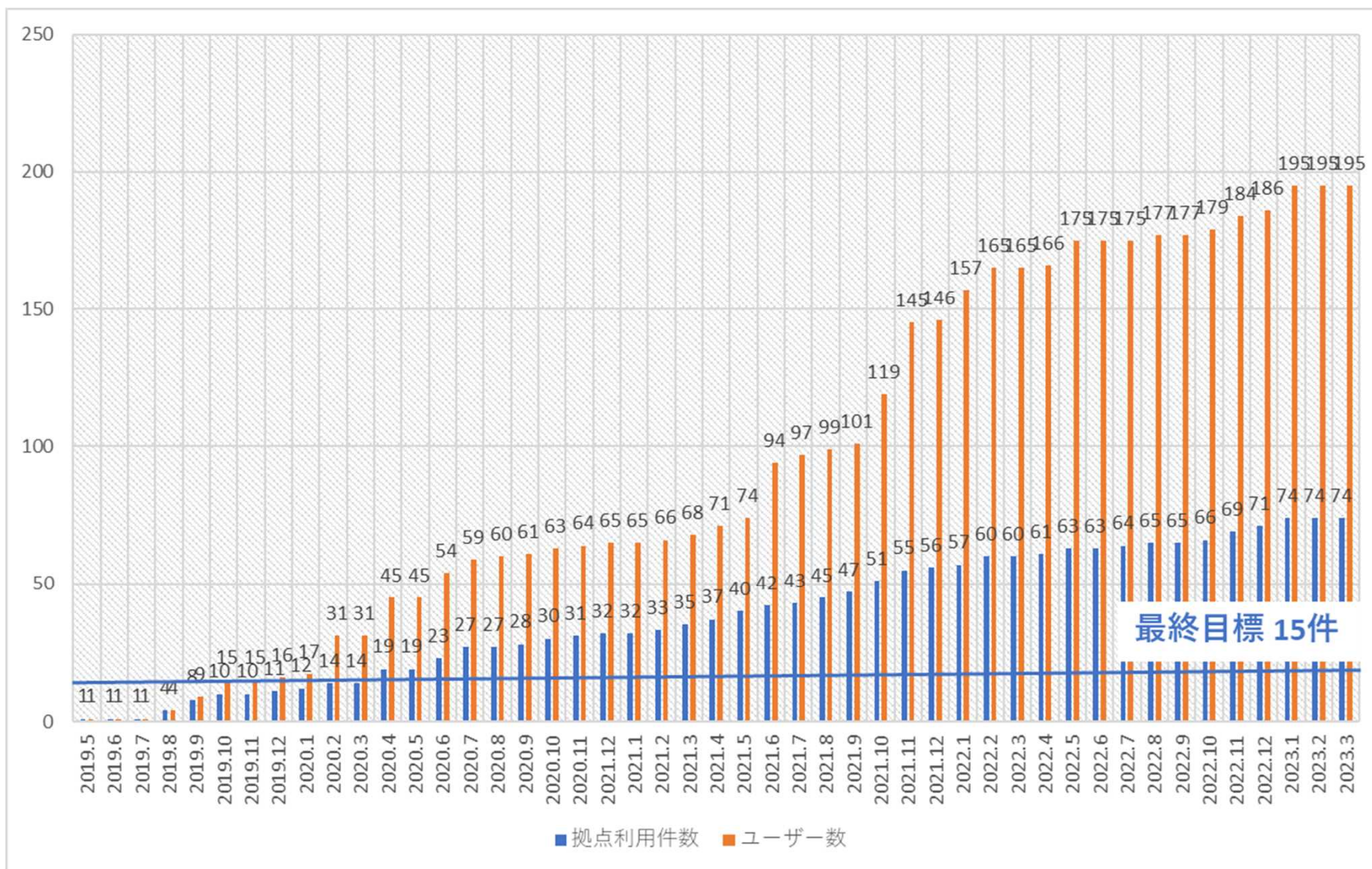
本事業を通じて開発、整備したAIチップ設計のための共通基盤技術、学習環境、設計環境の活用件数**15件以上**を目標とする。

⇒ 2023年3月時点で**74件**の実績

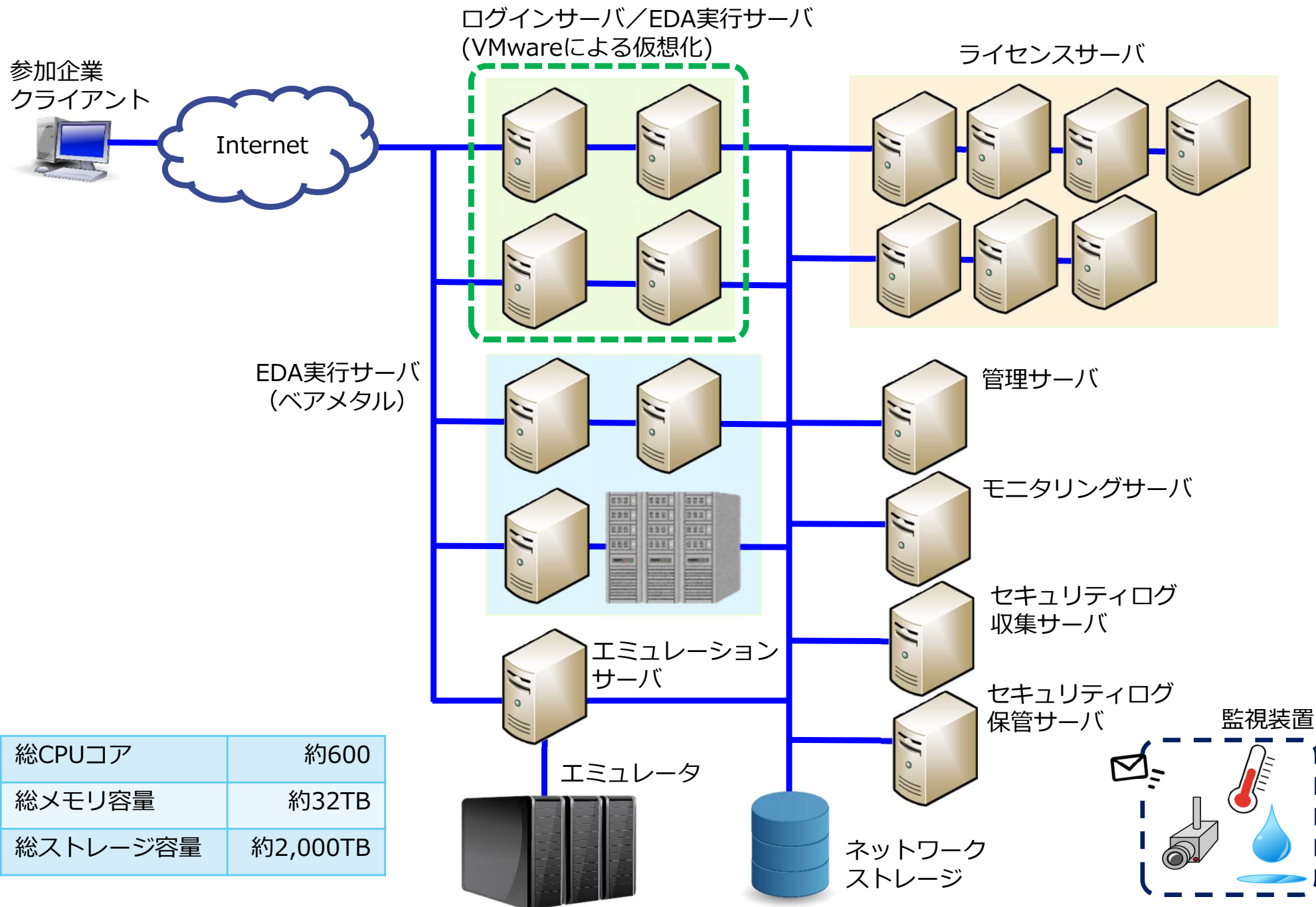
2. 成果：拠点活用件数/ユーザ数の推移

2023年3月時点の拠点活用件数(累積) 74件

内訳 企業：55件、大学：12件、国研：6件、その他：1件（一社）



2. 成果：拠点クラウドの構築



2. 成果：拠点の構築

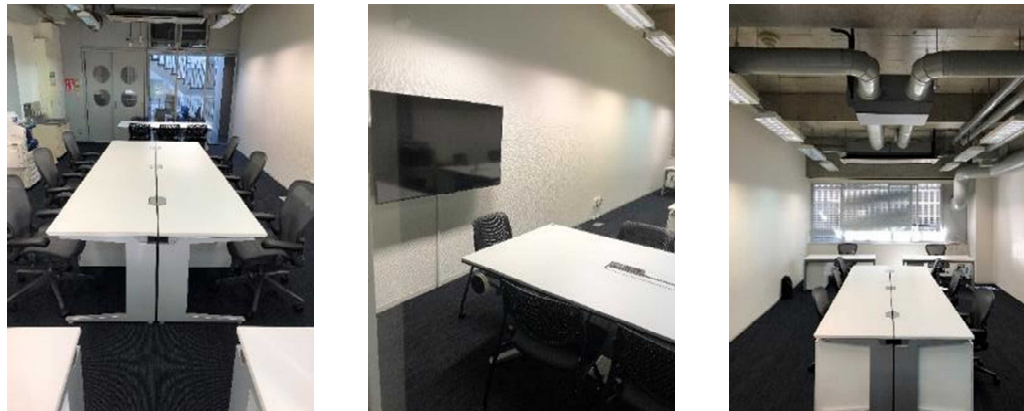
工学部3号館107号室(ブース)



サーバ室(クラウド環境、エミュレータ)



武田先端知ビル203号室(執務室)



サテライト拠点(福岡IST)



2. 成果：EDAツールの整備

Cadence, Synopsys, Siemensのツールを整備

アーキテクチャ検証、高位合成、論理設計/検証、回路設計、物理設計/検証、論理工ミュレータ、FPGAプロトタイピング、ボード設計、etc.

#Vendor	Feature	大項目	用途・カテゴリ	
cadence	Stratus_HLS_-.XL	アーキテクチャ検証 高位合成システムツール群	高位合成 (High-level Synthesis)	
cadence	Stratus_Floating_Point			
mentor	Catapult-Prime_Ap_SW	アーキテクチャ検証 高位合成システムツール群	高位合成	
mentor	Oasys-RTL_Designer_Ap_SW			
mentor	Oasys-RTL_Floorplanning_AddOn_SW			
mentor	Oasys-RTL_Architect_AddOn_SW			
mentor	Oasys-RTL_Low_Power_AddOn_SW			
mentor	Catapult_Coverage_Ap_SW			
mentor	SLEC-HLS_Ap_SW		高位検証	
mentor	PowerPro-Optimizer_Ap_SW		消費電力の推定・解析	
synopsys	DC_Explorer		論理合成ツール (LogicSynthesis)	
synopsys	DC_Ultra			
synopsys	HDL_Compiler_Verilog			
synopsys	VHDL_Compiler			
synopsys	Power_Compiler			
synopsys	DFT_Compiler			
synopsys	Design_Vision	論理合成結果の可視化・最適化ツール (GUI & Advanced Optimization)		
synopsys	Design_Compiler_Graphical			
synopsys	Library_Compiler	論理合成向け基本関数ライブラリ (Datapath library for Logic Synthesis)		
synopsys	DesignWare_Library			
synopsys	DW_minPower_Components	配置配線ツール (P&R)		
synopsys	IC_Compiler_II_AG_8-core			
synopsys	Custom_Compiler_ADV			
synopsys	IC-Compiler-II-AF-Add-on			
synopsys	StarRC_Ultra	寄生成分抽出ツール (Extraction)		
synopsys	Custom_Compiler_ADV			
synopsys	IC_Validator_NXT			
synopsys	IC_WorkBench_Edit/View_Plus			
synopsys	PrimeTime_ADV	タイミング検証・電力解析ツール (Timing & Power Analysis)		
synopsys	Prime_Power			
synopsys	Formality		等価性検証ツール (Equivalence checking)	
synopsys	PowerReplay			
synopsys	Verdi-3	論理デバッグツール (Debug)		
synopsys	Verdi_-.Power_Aware_Debug			
synopsys	nAnalyzer			
synopsys	nECO			
synopsys	Verdi_HW_SW_Debug_Add-on			
synopsys	Verdi_Advanced_AMS_Debug_Add-on			
synopsys	Verdi_Performance_Analyzer			
synopsys	SpyGlass_Lint			
synopsys	SpyGlass_Lint_Turbo_Add-on			
synopsys	SpyGlass_CDC_Advanced_Add-on			
synopsys	SpyGlass_RDC_Add-on	RTL解析ツール (RTL analysis)		
synopsys	SpyGlass_DFT_ADV_Add-on			
synopsys	SpyGlass_SVA_Generation_Add-on			
synopsys	VC_LP			
synopsys	SpyGlass_Power_Explorer_Add-on			
synopsys	VC_Formal		形式的機能検証 (Formal verification)	
synopsys	VC_Formal_FSV_Add-on			
synopsys	VC_Formal_FTA_Add-on			
synopsys	Certitude_Base_w_Unlimited_Sim		検証品質確認ツール (Functional Qualification)	
synopsys	DFTMAX			
synopsys	DFTMAX_Ultra_Add-on			
synopsys	DC_Ultra	テスト設計検証ツール (Design for test)		
synopsys	Design_Vision			
synopsys	HDL_Compiler_Verilog			
synopsys	TetraMAX_II_ADV_8-core_ATPG			
synopsys	HSPICE	回路レベルシミュレーションツール (Circuit simulation)		
synopsys	Custom_WaveView_ADV			
synopsys	CustomSim_with_FineSim2			
synopsys	FastSPICE_Analysis_Suite			
synopsys	Custom_Compiler_ADV			
synopsys	VCS_MX			
synopsys	TestMax_CustomFault			
synopsys	X-Prop_Add-on			
synopsys	VCS_MX		論理シミュレーションツール (Logic simulation)	
synopsys	NLP_Add-On			

cadence	Genus_Synthesis_Solution	論理設計検証ツール群	論理合成 (Logic Synthesis)	
cadence	Genus_Low_Power_Option			
cadence	Genus_Physical_Option			
cadence	Isosys_RTL_Power_Solution		電力解析	
cadence	Conformal_Low_Power_-.XL		寄性検証	
cadence	Innovus_Implementation_System		配置配線 (Place&Route)	
cadence	Innovus_20/15/14nm_Option			
cadence	Innovus_Hierarchical_Design_Option			
cadence	Tempus_Timing_Signoff_Solution		タイミング検証 (Timing Analysis)	
cadence	Tempus_Timing_Signoff_Solution_ECO			
cadence	Xcelium_Single_Core	論理検証 (Logic Verification)		
cadence	Xcelium_Digital_Mixed_Signal_Option			
cadence	vManager_Integration_Server			
cadence	vManager_Multi-Project_Application			
cadence	vManager_Linux_Client			
cadence	JasperGold_Formal_Verification_Platform		フォーマル検証	
cadence	Virtuoso_Schematic_Editor_L		アナログ・ミックスドシグナル設計	回路図入力 (Schematic Editor)
cadence	Virtuoso_Schematic_Editor_XL			
cadence	Virtuoso_ADE_Assembler			レイアウト入力 (Layout Editor)
cadence	Virtuoso_Layout_Suite_L			
cadence	Virtuoso_Layout_Suite_XL	アナログ・ミックスドシグナル設計	シミュレーション	
cadence	spectre_Multi-Mode_Simulation_with_AHS		回路図入力 (Schematic Editor)	
synopsys	Custom_Compiler_ADV		レイアウト入力 (Layout Editor)	
synopsys	Galaxy_Custom_Router			
synopsys	HSPICE			
synopsys	Custom_WaveView_ADV		アナログシミュレーション環境 (Analog Simulation)	
synopsys	CustomSim_with_FineSim2			
synopsys	FastSPICE_Analysis_Suite			
synopsys	TestMax_CustomFault			
synopsys	Custom_Compiler_ADV		設計機能検査・寄生成分抽出ツール (DRC & Extraction)	
synopsys	StarRC_Ultra	電力解析ツール (Power Analysis)		
synopsys	IC_Validator_NXT			
synopsys	Custom_Compiler_ADV			
synopsys	FastSPICE_Analysis_Suite			
synopsys	CustomSim_with_FineSim2	物理検証ツール群	物理検証 DRC LVS (Physical Verification)	
synopsys	Custom_Compiler_ADV			
mentor	Calibre_nmDRC_Ap_SW		寄生成分抽出	
mentor	Calibre_nmDRC-H_Op_SW			
mentor	Calibre_nmlVS-Ap_SW			
mentor	Calibre_nmlVS-H_Op_SW			
mentor	Calibre_xACT_3D_Ap_SW		対話型実行環境 結果表示	
mentor	Calibre_RVE/QQB-H_Ap_SW			
mentor	Calibre_Interactive_Ap_SW			
mentor	Calibre_DeSigner_Ap_SW			
cadence	Physical_Verification_System_DRC	物理検証ツール群	物理検証 DRC LVS (Physical Verification)	
cadence	Physical_Verification_System_Advanced_Analysis_Option_for_PVS_DRC			
cadence	Physical_Verification_System_LVS			
cadence	Physical_Verification_System_Advanced_Device_Option			
cadence	Quantus_QRC_Extraction		寄生成分抽出 (Parasitic Extraction)	
cadence	Quantus_QRC_Advanced_Analysis			
cadence	Quantus_QRC_Advanced_Modeling			
cadence	Quantus_QRC_Advanced_Node_Modeling			
cadence	Voltaus_IC_Power_Integrity_Solution		電力解析 (Power Analysis)	
cadence	Voltaus_IC_Power_Integrity_Solution_Advanced_Analysis_GXL_Option			
cadence	Voltaus_IC_Custom_Power_Integrity_Solution	エミュレータ		
cadence	Palladium_Z1_GXL_576_always-on-domains			
cadence	Palladium_Z1_56Gbps_36_ports_switch			
cadence	Palladium_Z1_2-port_host_card			
cadence	Palladium_Dynamic_Power_Analysis_2.0			
cadence	Virtual_Debug_Interface_to_Software_Debuggers_4-pack			
cadence	Palladium_Series_Memory_model_Multiple_Subscription			
cadence	Palladium_PcE_4.0_VirtualBridge_Kit			
cadence	Palladium_VirtualBridge_8-port_pack			
cadence	Incise-Enterprise-Simulator-XXL			
cadence	Palladium_Accelerated_VIP_for_AMBA_AXI	エミュレータ用検証モデル		
cadence	Palladium_Accelerated_VIP_for_AMBA_AHB			
cadence	Palladium_Accelerated_VIP_for_PcE_4.0			
cadence	Accelerated_VIP_for_AMBA_USB_3.1			
cadence	Accelerated_VIP_for_HDMI_2.0			
cadence	Accelerated_VIP_for_MIP1_CSI_2		エミュレータ FPGAプロトタイプ	
cadence	Accelerated_VIP_for_MIP1_DSI_2			
cadence	Accelerated_VIP_for_CSI1			
cadence	Protium-SI-Implementation-and-Debug-Software			
cadence	Protium_SI_FPGA-based_prototyping_system_B_UltraScale_VU140_FPGAs		FPGAプロトタイプ	
cadence	Protium_SI_accessory_2_chassis_cable_kit			
cadence	Protium_SI_Accessory_High_Performance_XDRAM_Card_16GB			
cadence	PCIE_4.0_Emulation_Dev_Kit_for_Win10_on_Intel_platform			
cadence	PCIE_4.0_Emulation_Dev_Kit_for_Win10_on_Intel_platform			
cadence	Protium_SI_Accessory_QCS_Adapter_for_PcE_4.0_SpeedBridge			
cadence	Protium_SI_Accessory_DDR1_direct_connect_8GB			
cadence	Protium / Protium_SI_Accessory_14-HB_56AH_300MH			
cadence	Protium_SI_Accessory_204-pin_SODIMM_Adapter			
cadence	Protium_7_Protium_SI_Accessory_PcE_Kit			
cadence	Allegro_SB_for_HD_serial_interfaces_RPIS			
cadence	Protium_SI_Accessory_VHDCI_Adapter_for_VcHo_SpeedBridge			
cadence	Allegro PCB Designer	BoardDesign	PCB board Design	

Synopsysの40nm, 28nm, 12nm 標準IP群 (CPU, DSP, DMAC, DDR, PCIe, I2C, etc.) を整備 (物理系はTSMC向け)

	40nm	28nm	
Catgry	Edge	Graph	IP packageName
MCU, CPU	MCU for RToS	CPU for Linux	ARC EM9D Core
	DSP		
	Debug	Debug	
	Timer	Timer	
	WDT	WDT	
	INTC	INTC	
	XYIF		
	MemIF		
	ROMIF		
	Sybsystem RTL		Data Fusion IP Subsystem
	FPU		ARC FPU Option for EM
	DMAC		ARC uDMA Option for EM
		SRAMIF	ARC HS38x2 Core
		ROMIF	
		DMAC	AXI DMAC
Bus IF	AHB/I2C	AHB/I2C	APB Advanced Peripherals
	APB/I2C	APB/I2C	
	APB/UART	APB/UART	
	SPI	SPI	SSI Core
	AHB Master/Slave	AHB Master/Slave	AHB Fabric
	AHB I2C IF	AHB I2C IF	
	AHB SDMI40 IF	AHB SDMI28 IF	
	ADC IF		
	DAC IF		
	APB/GPIO	APB/GPIO	APB Peripherals
	SDMI40	SDMI28	SDHC
			ARC EM ESP Option
Security	TEE		ARC Crypto Pack Option
	AES		
	ECC		
	SHA		
	AES_SW		Cryptography SW Library
	ECC-SW		
	SHA-SW		
	OTP		NVM OTP TS40LP

AIチップ設計拠点 Confidential Document

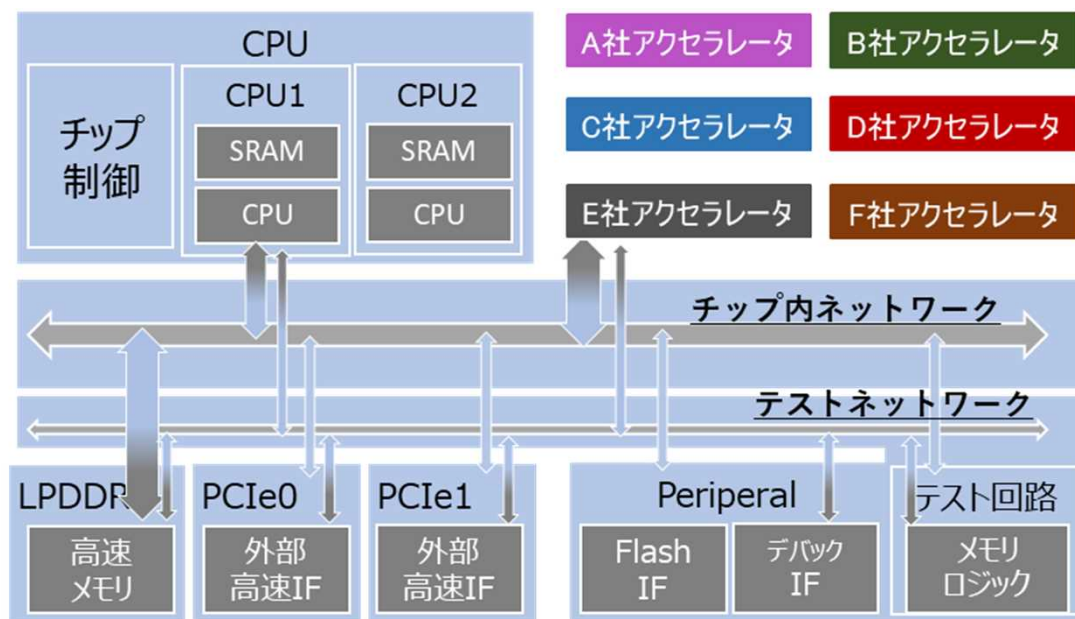
Catgry	Edge	Graph	IP packageName
DDR		DDR-cntl	Universal DDR MCTL2 MP - v3.50a
		DDR-opt	Adv Feature Pkg for uMCTL2
		DDR-addOnLP	LPDDR4 Add-on to uMCTL2
		DDR-Phy	LPDDR4 multiPHY V2 28HPC+
PCIe		PCIeG2 Std	PCIe 2.0 Standard AMBA
		PCIeG2-Phy	PCIe V2.x Consumer 6Gx4, 28HPC+18
USB	USB2.0 40-OTG	USB2.0 28-OTG	usb_2_0_hs_otg_v4
	USB2.0 40-Phy		USB2 picoPHY 40LP25
		USB2.0 40-Phy	USB2 femtoPHY 28HPC+18
MIPI	MIPI 40-Csi	MIPI 28-Csi	MIPI CSI2 Host Controller
	MIPI 40-Dsi	MIPI 28-Dsi	MIPI DSI Host Controller
	MIPI 40-Tx		mipi_dphy_bd4_t40lp25
	MIPI 40-Rx		mipi_dphy_rx4_t40lp25
		MIPI 28-Tx	mipi_d_t4_tsmc28hpcp18ns
		MIPI 28-Rx	mipi_d_r4_tsmc28hpcp18ns
ADC	ADC		ADC-IQIF 12b250M 40LP
		ADC	ADCIQ12b320M 28HPCPNS
	DAC		DAC-IQ, 12b160M 40LP
		DAC	DACIQ12b640M TSMC28HPCPNS
StdCell, RAM, Bist, Yield	Std40		Duet 40LP
	SRAM40 1Port		
	ROM40		
	SRAM40 2Port		Duet TSMC 40LP 2PUHD Add-on
	Std40-HPC		Duet TSMC 40LP HPC Design Kit
		Std28	Duet 28HPC+
		SRAM28 1Port	
		ROM28	
		SRAM28 2Port	Duet 28HPC+ 2PUHD Add-on
		Std28 HPCt	Duet 28HPC+ HPC Design Kit
	Bist, Test, Repare	Bist, Test, Repare	SMS v5.X Test and Repair
	Bist, Yield	Bist, Yield	SMS Yield Accelerator
	TAP compiler	TAP compiler	SHS IP Compiler
TAP Yield	TAP Yield	SHS Yield Accelerator	
Board, Dev Tools	Evaluation Board		ARC DWC IoT Development Kit
		Evaluation Board	ARC HS Development Kit
	Debug tools	Debug tools	ARC MetaWare Devel Toolkit
	Debugger, SW	Debugger, SW	
	RToS	Linux	The embARC Community
	Drivers	Drivers	
	Sample Apps	Sample Apps	



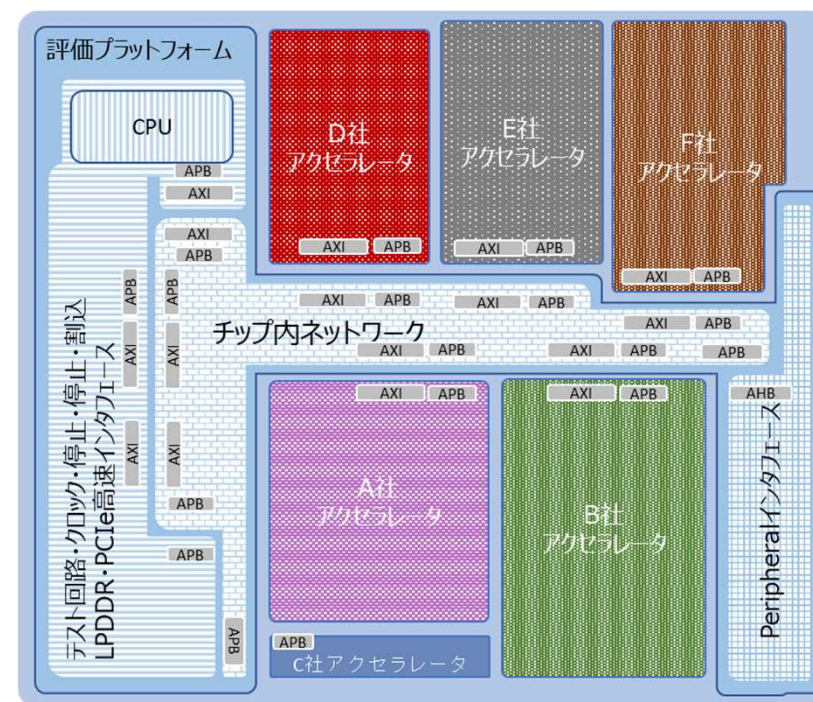
2. 成果：AIアクセラレータのための評価プラットフォーム(AI-One)構築

- ・ 拠点導入IP(28nm)を活用して、拠点が設計プラットフォームを準備
- ・ 乗合チップ参加企業は各社のAIアクセラレータをプラットフォームに接続
- ・ 拠点がまとめてチップ実装を行いファブに試作依頼、各社にチップ(+ボード)を配布
- ・ 各社は試作チップ (ボード)を用いて、実証実験を進める

AIアクセラレータ開発会社
Axell, DMP, Logic Research,
LeapMind, Privatech, AIST



チップ内部構成



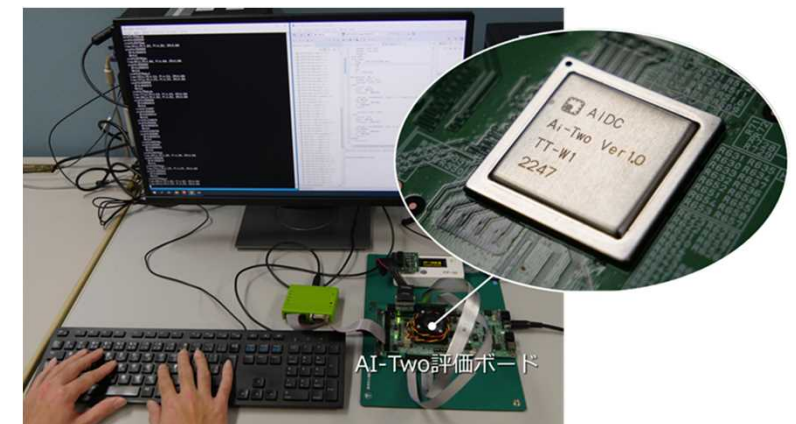
チップ実装イメージ

AI-One

- ・ファブ：TSMC 28nm
- ・標準IP：Synopsys
- ・AIアクセラレータ：6種
- ・参加企業：5社, 産総研
- ・2021/6月 チップ完, 11月 PKG完,
2022/1月 ボード完

AI-Two

- ・ファブ：TSMC 12nm
- ・標準IP：Synopsys
- ・AIアクセラレータ：3種
- ・参加企業：2社, 産総研
- ・2022/11月 チップ完,
2023/1月 ボード完



AI-Two
評価ボード

Palladium Z1 (Cadence社製, 2018年度補正予算で導入)

- 容量：23億ゲート, 4.6Tバイト（ユザメモリ）, 4.6Tバイト（デバッグメモリ）
- シミュレーション速度：最大4MHz, コパ°ル速度：140Mゲート/時
- ソフトウェア・シミュレーションに対して3,700倍～14,500倍の高速化(拠点での実績)



AIチップ設計拠点 サーバ室

2. 成果：拠点HP (https://www.ai-chip-design-center.org/)



プロジェクトID申請には、既にプロジェクトIDをお持ちで追加機能申請の場合は、本フォームにプロジェクトに参加される拠点利用者全員のEmailアドレスを記載願います。プロジェクトID入手後、拠点利用者全員に拠点ID申請をお願いいたします(プロジェクトID申請は、管理責任者以外は必要ありません)

プロジェクト

プロジェクトID(1)

プロジェクト名(1) **必須**

プロジェクト概要(1) **必須**

プロジェクトID(2)

AIチップ設計拠点 設計ツール Ai-One 拠点利用方法 Information

イベント

拠点フォーラム
e-講座・講演

イベント > e-講座 >

6:40 AIチップ設計拠点フォーラム (第26回)

2021/08/17 09:00 夏季休暇とシステム稼働不安定の報告(事後連絡)

Webinarと教材Download

4. 教材
Emulatorと論理Simulator, 高位設計、論理設計と検証

- エミュレータフローの一般論 (初級編)
- デジタル設計フローの一般論 (初級編)
- 高位合成を使ったデジタル設計 (基礎編)
- エミュレータ論理検証の基礎 (初級編)
- エミュレータ論理検証の基礎 (応用編)
- A Design Verification Management Platform

論理設計検証技術

- エミュレータ活用 速習1.5時間コース
- エミュレータ活用 関連セミナー(1)
- エミュレータ活用 関連セミナー(2)

5. 演習
Emulatorと論理Simulator, 高位設計、論理設計と検証

Spyness演習1

Public / お問い合わせ

お名前 **必須**

姓(例: 山田) 名(例: 太郎)

お名前(カタカナフリガナ) **必須**

姓(例: ヤマダ) 名(例: タロウ)

会社名(法人の方) **必須**

所属機関(例: 株式会社〇〇〇〇、〇〇大学)派遣元や複数所属している機関も全て記載をお願いします。

メールアドレス **必須**

例: yourname@sample.com

お問い合わせ内容 **必須**

お問い合わせ内容をご記載ください。入力は、日本語と英語の文字をお願いします。プログラムコードで 사용되는特殊文字を入れると問い合わせできないことがあります。

設計拠点のプライバシーポリシーに 同意する **必須**

記載内容確認。まだ送信されません。

(1) 外部講演

- ・NV-FPGA研究会, エナジーハーベストコンソーシアム総会, トリオンノード研究会, 学振154委員会, DAシンポジウム, DSF2020, RISC-V day, LSIとシステムのWS, ITmedia EXPO, HIYA研究会, SNUG, IEEE CMPT, 光・電波フォーラム など17件

(2) 展示会

- ・CEATEC2019(2019/10/15-18), NEDOフェスタin関西2019(2019/12/17-18), DAC2022(2022/7/10-14), SNUG Japan(2022/9/14-26), CEATEC2022(2022/10/18-21), EdgeTech+2022(2022/11/16-18) DSF2022(2022/11/25) の7件

(3) ニュースリリース

- ・2019/10/7「AIチップ開発加速のための「AIチップ設計拠点」が稼働開始」
—設計・評価ツールの提供により、中小・ベンチャーのチップ開発加速を目指す—
- ・2021/5/10「複数のAI アクセラレータを搭載した評価チップの設計を完了、試作を開始」
—短期間で低コストのAI チップ設計・評価手法の確立へ—
- ・2022/3/22「複数のAIアクセラレータを搭載した実証チップ「AI-One」の動作を確認」
—従来比45%以下の短期間で低コストのAIチップ設計・評価が可能に—
- ・2023/3/17「AIチップ設計拠点の本格運用を開始」
—設計環境の提供により、中小・ベンチャー企業などのAIチップ開発加速を目指す—

AIチップ設計拠点フォーラム

- ・AIチップ、コンピューティング、LSI設計などの技術情報を共有し、議論の場を提供
- ・月1回のペースで開催(2019/5～)、2023年3月度が第45回
- ・毎回百数十名の参加者

第45回 AIチップ設計拠点フォーラム (2023/3/31)

13:30-13:35 AIチップ設計拠点フォーラムについて
(産総研／内山邦男)

13:35-14:35 脱炭素とデジタル化を支える半導体ソリューション
(インフィニオン テクノロジーズ ジャパン／川崎郁也氏(代表取締役社長))

14:35-15:35 The introduction to ONNC – Anchors while both hardware and software are changing in AI semiconductors.
(Skymizer Taiwan, Inc.／Luba Tang氏(CEO))

15:40-16:40 ISSCC2023におけるAIチップ研究動向
(産総研／更田裕司氏)

教材

1. デジタル設計の基礎
2. 高位合成を使ったデジタル設計 (基礎編)
3. 高位合成を使ったデジタル設計 (続基礎編)
4. 高位合成を使ったデジタル設計 (実践編)
5. 高位合成を使ったデジタル設計 (実践編 2)
6. 高位合成を使ったデジタル設計 (実践編 3)
7. 高位合成を使ったデジタル設計 (実践編 4)
8. デジタルチップ・物理設計 (応用編)
9. デジタルチップ・物理設計 (低消費電力編)
10. ハードウェア・エミュレータでの論理検証の基礎 (初級編)
11. ハードウェア・エミュレータでの論理検証の基礎 (応用編)
12. Accelerating the crypto design – labs
13. Accelerating Crypto Design on Palladium
14. Crypto Pairing Design Bring Up on EmuForge

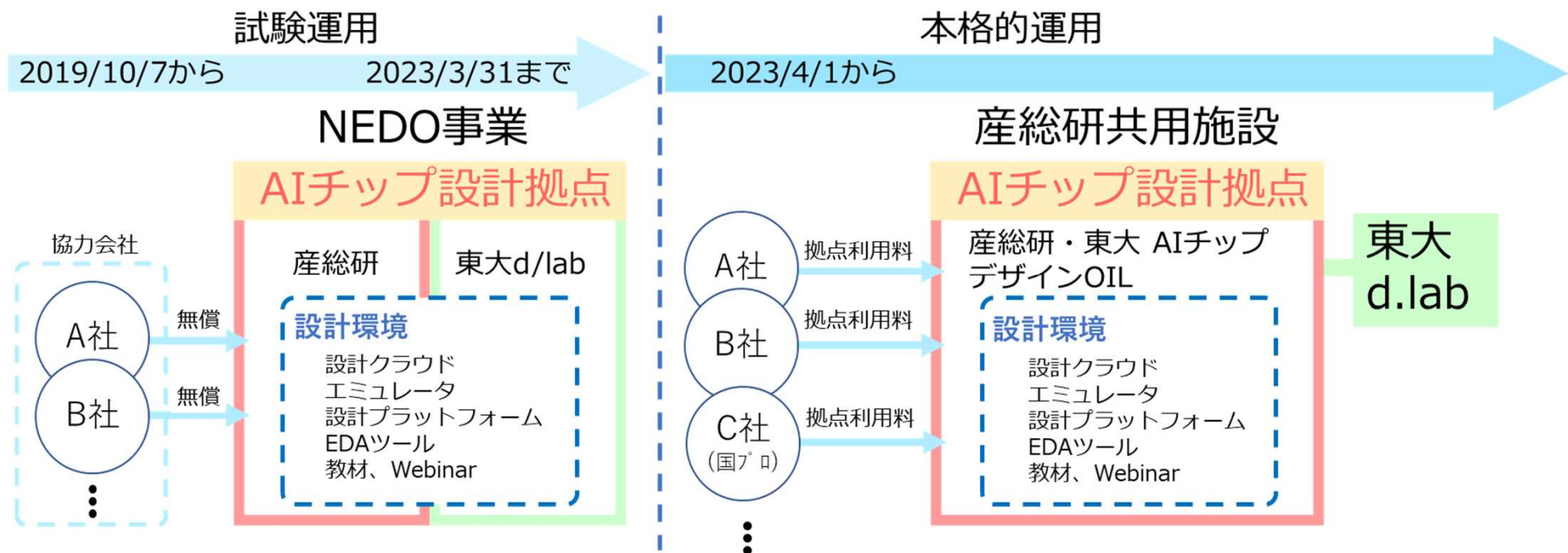
ハンズオンセミナー

1. 2021年6月30日 13:00 – 17:00 【エミュレータを利用した検証の高速化に向けたテストベンチ作成方法】
2. 2021年7月1日 13:00 – 16:00 【シミュレータ・エミュレータによる協調検証環境を用いた検証効率向上】
3. 2021年11月30日 13:00 – 17:30 【エミュレータトレーニング by Cadence】
4. 2022年3月10日 13:30 – 17:00 【エミュレータ・ワークショップ with Cadence】
5. 2022年4月28日 13:30 – 15:00 【エミュレータ・セミナー with Cadence】
6. 2022年6月7日 13:30 – 17:30 【エミュレータトレーニング by Cadence】
7. 2022年9月13日 13:30 – 17:00 【エミュレータ・ワークショップ with Cadence】
8. 2022年9月28日 13:30 – 17:30 【RISC-VベースSoCのシミュレーション・エミュレーション協調検証トレーニング】
9. 2022年12月23日 13:30 – 17:00 【拠点フォーラム：検証特集】
10. 2023年2月14日 13:30 – 17:30 【エミュレータトレーニング by Cadence】
11. 2023年3月14日 13:30 – 17:00 【エミュレータ・ワークショップ with Cadence】
12. 2023年3月20日 10:00 – 17:00 【RISC-VベースSoCのシミュレーション・エミュレーション協調検証トレーニング】

Webinar 「RTLガイドラインとiocheck実行」, 「形式検証の導入、実行、デバック」, 「CDC検証」, etc.

3. 2023年度からの拠点運営：運営方針

- 産総研内にAIDLを設置して、AIチップ設計拠点を本格的に運営
AIDL：産総研・東大 AIチップデザインOIL (オープンイノベーションラボラトリ) (産総研の研究組織)
- 本NEDO事業において開発、構築した成果は、AIチップ設計拠点に実装
- 利用機関は、約款により定めた拠点利用料を支払って拠点の設計環境を利用



3. 2023年度からの拠点運営：利用方法

- ✓ プランとオプションを組み合わせることで、利用者が設計環境を構築可能
- ✓ 月ごとにプランとオプションを設定可能 (一部に複数月のみの設定あり)

1. おすすめプラン

- ① スタンダード
- ② PDK/IP持ち込み
- ③ エミュレータ



1.1 追加オプション

- ① 追加ストレージ
- ② 追加CPUcore
- ③ 追加Memory
- ④ 追加ユーザ
- ⑤ 追加ドメイン(エミュレータ)
- ⑥ 追加設計環境利用時間
- ⑦ 追加設計ツール



1.2 PDK・IP持ち込み専用強化オプション

- ① 5.1GHz CPU実行サーバ
- ② 768GB実行サーバ
- ③ 1500GB実行サーバ

2. ベアメタルサーバプラン

- ① 5.1GHz CPU Loginサーバ
- ② 5.1GHz CPU実行サーバ
- ③ 768GB実行サーバ
- ④ 1500GB実行サーバ



2.1 追加オプション

- ① 追加ストレージ
- ② 追加ユーザ
- ③ 追加設計環境利用時間
- ④ 追加設計ツール

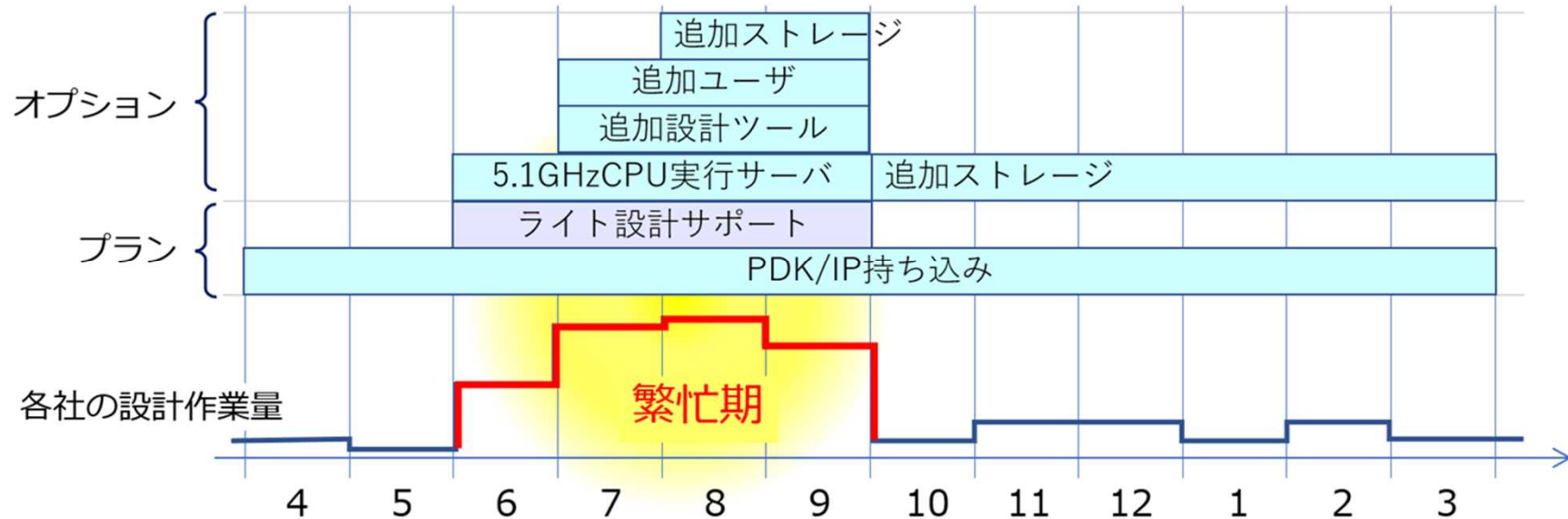
3. サポートプラン

- ① RTLガイドライン参照
- ② 実装確認依頼
- ③ SoCプラットフォーム参照
- ④ SoCプラットフォーム開発
- ⑤ ライト設計サポート
- ⑥ 設計サポート

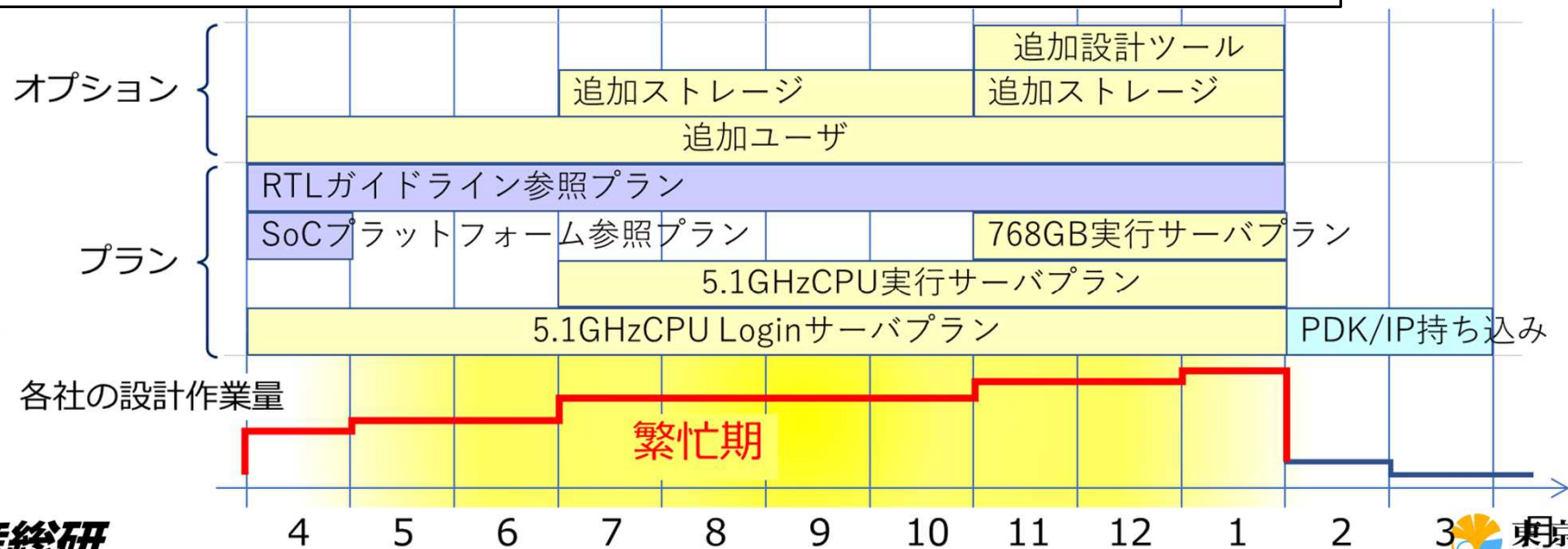
4. ブース利用プラン

3. 2023年度からの拠点運営：利用例

①標準的な設計環境を用いて小規模モジュール・チップを設計する場合

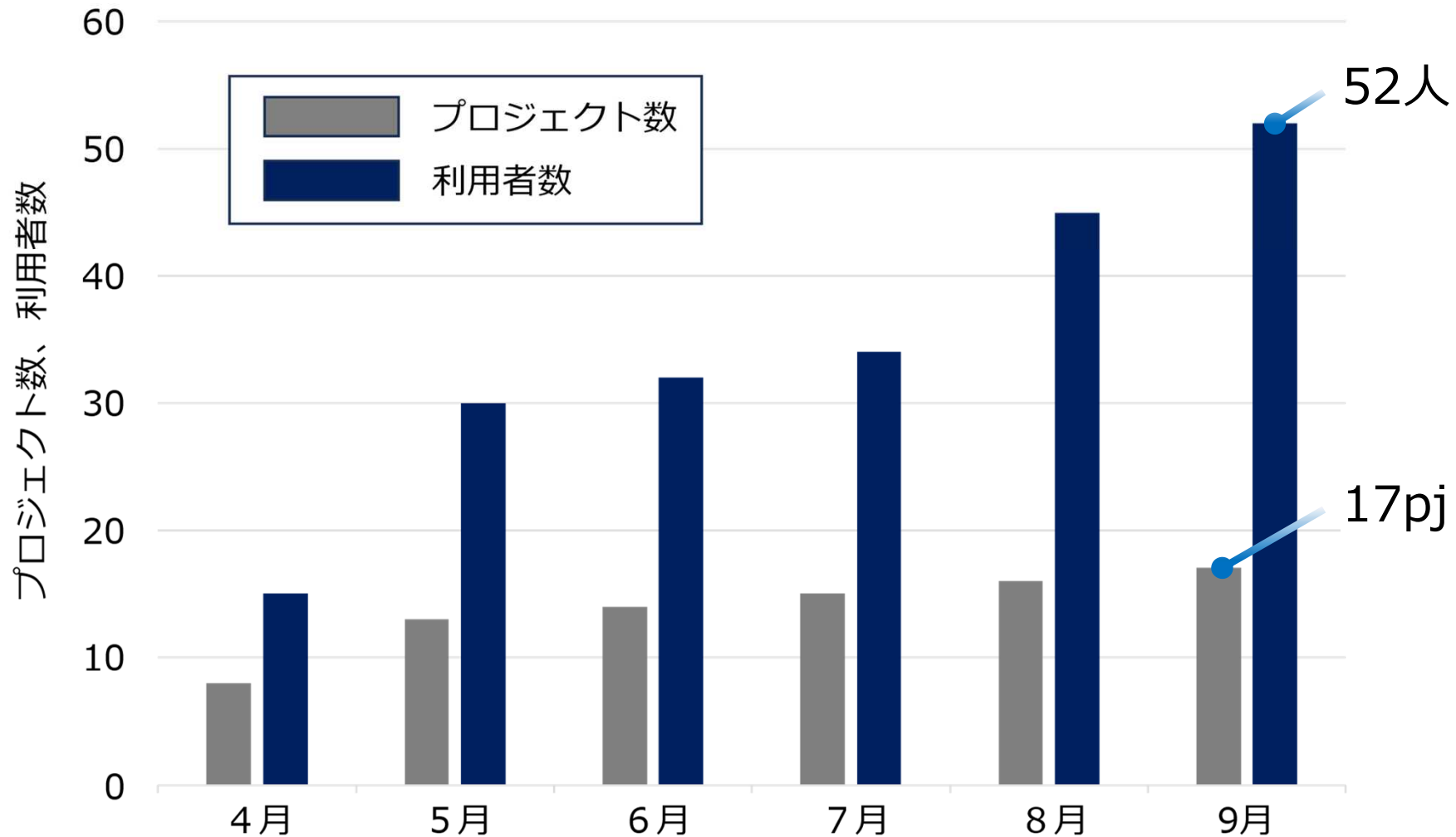


②圧倒的な処理能力をもつ設計環境を用いてチップを開発する場合



3. 2023年度からの拠点運営：利用実績

2023年9月時点の活用件数 **17件** (企業：12、大学：2、国研：3)



毎月の利用状況

AIチップ開発加速のためのイノベーション推進事業

- 実施項目 1 – 1 「AIチップ向け設計フローの研究開発」
実施項目 1 – 2 「ハードウェア開発垂直立ち上げ実現
のための研究開発」

担当機関 東京大学
発表者 池田誠（東京大学）

1. 概要

- 背景・目的・課題
- 取り組み内容・技術の特徴
- 目標・計画

2. 成果

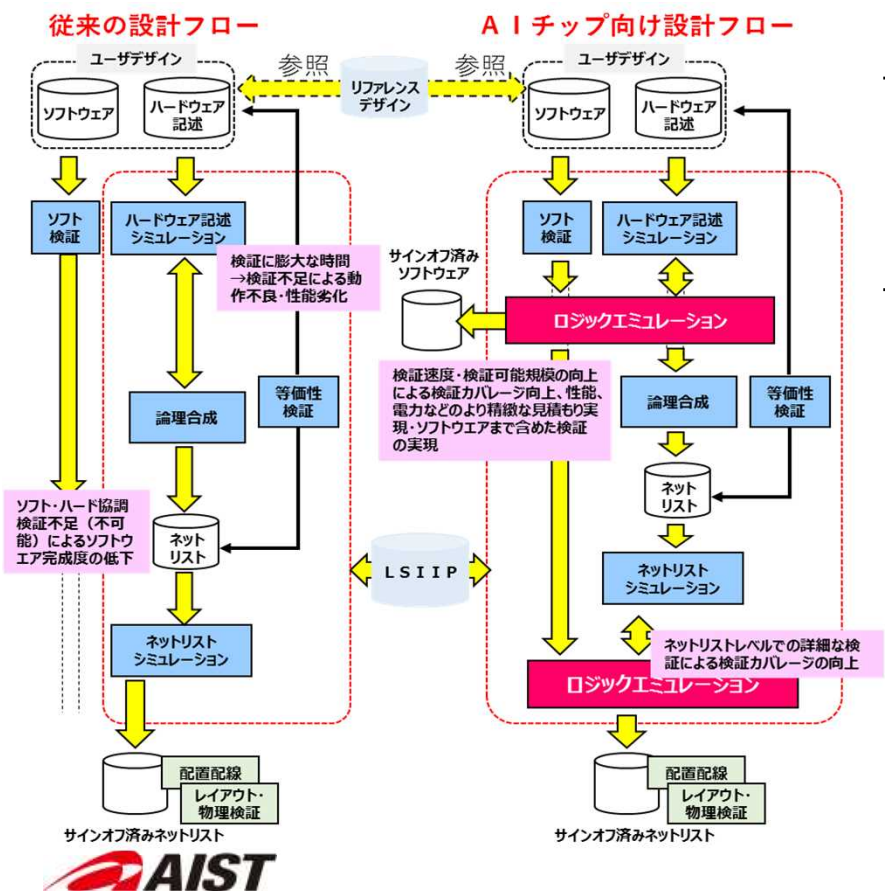
- 成果・意義
- 目標達成度 など

背景・課題

- AIチップ開発での、設計と実チップでの動作や性能に乖離が生じる
- 特に、高速、大規模、超低消費電力、高電力効率を謳う場合、設計段階での実チップの精緻な見積もりが不可欠

目的

- 失敗なく動作するチップ試作が可能となるためのノウハウを有した、AIチップ開発向けの論理設計以降の設計フロー(手順)を構築



取り組み内容

- AIチップ設計に特化したエミュレータを組み込んだ設計・検証のフローを構築する

技術の特徴

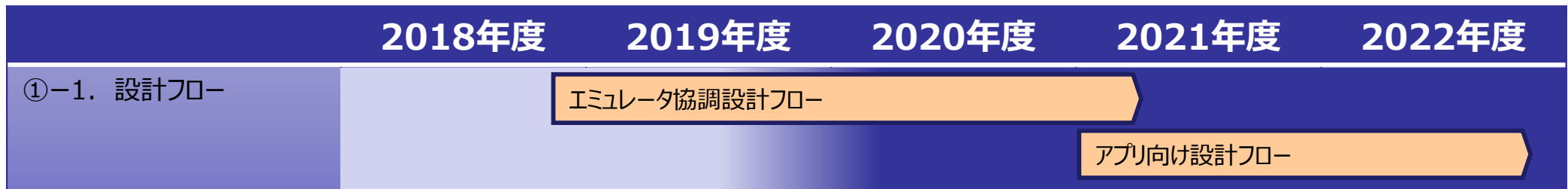
- エミュレータを用いた検証を行うことで大規模な設計に対して十分な検証ステップでの検証・精緻な性能見積もりを可能とする
- スムーズにAIチップの実現、十分な検証によるファーストシリコンからの稼働、ファーストシリコンと同時にソフトウェアが稼働するシステムの実現を可能とする設計検証環境の実現

中間(2020年度末時点)目標

- RTL記述を用いたエミュレータ・シミュレータ協調設計検証フローで設計検証を実施
- ネットリストレベルの記述を用いたエミュレータ・シミュレータ協調設計検証フローで設計検証実施
- 協調設計検証フローによる設計検証効率の向上可視化

最終(2022年度末時点)の目標

- より大規模(>1BG)な回路に対し、エミュレータ・シミュレータ協調設計検証フローで設計検証を実施
- 高速チップ・超低消費電力チップ・高電力効率チップ等のアプリケーション向けのエミュレータ・シミュレータ協調設計検証フローで、設計検証を実施



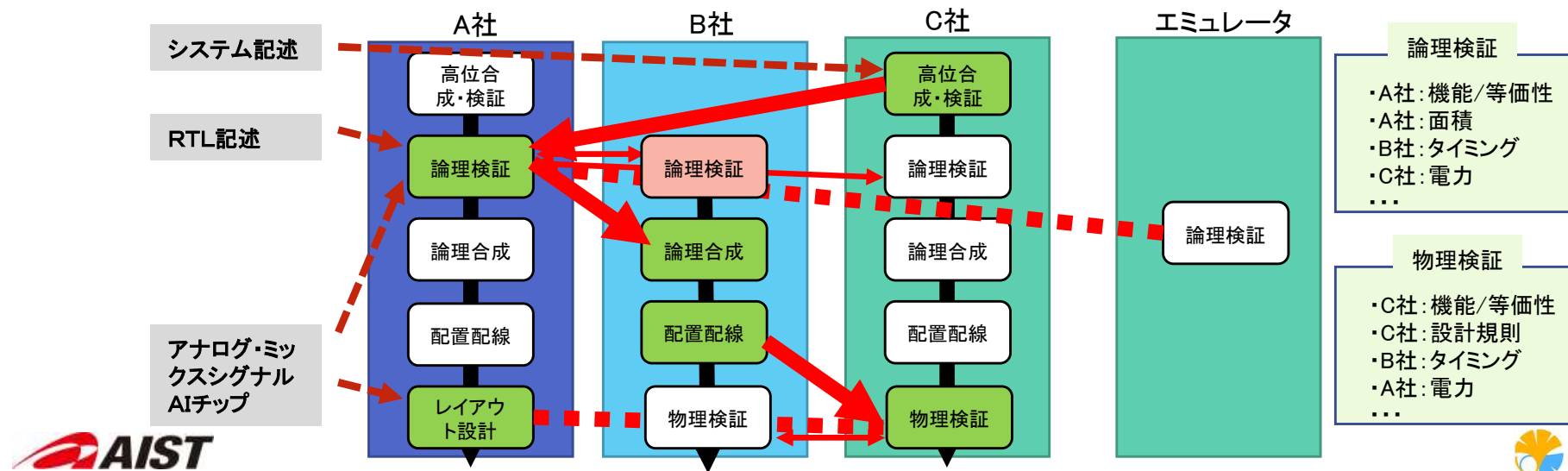
1. 1-2概要(背景・目的・課題)

背景・課題

- AIにかかわるソフトウェアやアルゴリズムの開発を行ってきた会社のAIハードウェアの開発への新規参入の障壁
 - 様々な設計ツール及び検証作業を要する
 - 当該ツール群に特有な個々のデータベースやファイル形式を取り扱う必要がある
- ハードウェア開発への新規参入は非常に敷居が高く、それらがAIに係るハードウェア開発の阻害要因となっている
- これを如何に軽減するかがAIハードウェアの研究開発の大きな課題

目的

- 多種多様な設計ツールを取捨選択し、ソフトウェアからハードウェア設計のサインオフまでを一気通貫に実現できるツールチェーンを策定する



取り組み内容

- ハードウェア開発を生業としないためシステムレベル記述言語の知見しかないベンチャー企業等の研究開発者でも、多種多様な設計ツールを適切に取捨選択でき、システムレベル記述言語からハードウェア設計のサインオフまでを一気通貫に実現できるツールチェーンを策定する。アナログ設計も対象とする。

技術の特徴

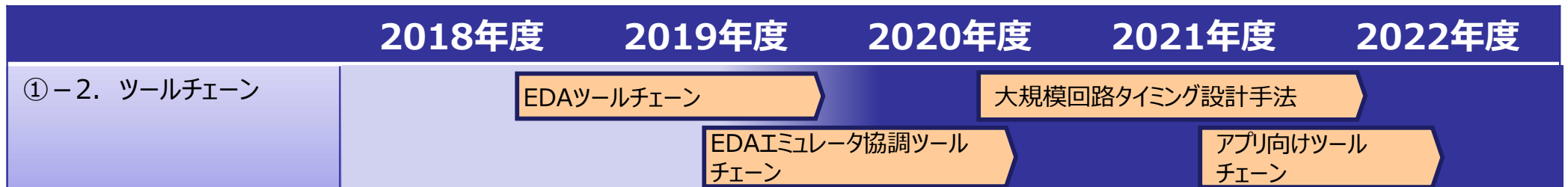
- 複数のEDAベンダー企業が取りそろえる多種多様な設計ツールの中から最適なツールを取捨選択
- 大手のIDMではノウハウの積み上げにより構築された設計フロー(手順)と同等なものを拠点でハードウェア開発を生業としないベンチャー企業等でも活用できるものとする

中間(2020年度末時点)目標

- モデルベースのエミュレータ・EDAツールの協調設計ツールチェーンを開発し、小・中規模の実設計において実証する。3件以上がこのモデル設計フローを活用した設計を行う

最終(2022年度末時点)の目標

- アプリケーション向けのツールチェーンを策定し、拠点利用者が事業期間全体で5件程度、より大規模(>1BG)な設計を実施する



成果

- エミュレータ利用ガイドライン及びエミュレータ利用マニュアル(エミュレータ利用モデルを含む)を電子的に作成し拠点のホームページ(以下HPと略)にて公開した
- エミュレータ・シミュレータ協調検証関連の教材を作成しHPに公開した
- RTL記述を用いたエミュレータ・シミュレータ協調設計検証フロー(VeriForge)環境を構築した
- 各種ハンズオントレーニングを実施した
- エミュレータ利用促進のための講習会・セミナー・ワークショップを実施した
- エミュレータによる検証速度向上を定量的に検証しHPに公開した

意義

- エミュレータの利用を加速することで、従来乖離が多かった検証結果と実機との差を埋めることが可能となるとともに、大規模な検証での検証漏れをなくすことが可能となる
- 実際のサンプルRTLを用いた教材整備により、難解で使いこなすことが困難と思われがちなエミュレータ利用の障壁を大幅に低減できたことで、利用者の増加が期待できる
- さらに、教材を活用したセミナーを実施することで、エミュレータに対する理解を深め更なる利用者の獲得につながる

エミュレータの利用を前提とした設計検証手法教材

デジタル設計教材
デジタル設計の基礎

AIチップ
東京大学システム

【目次】

【デジタル設計の基礎編】

1章 論理設計概要

1.1. LSI設計概要

1.2. 論理設計フロー

1.3. 論理回路の基本

1.4. HDL/SystemVerilog概要

2章 検証概要

2.1. 検証について

2.2. 検証手順

2.3. 検証方法

2.4. 検証ツール

3章 仕様説明

3.1. サンプルモジュールの概要

3.2. 基本モジュール MADD

3.3. 並列化モジュール MADD_FF

4章 基本モジュール(MADD)検証フロー

4.1. デレトリ構築

4.2. MADDの検証環境

4.3. MADDの検証実行フロー

4.4. MADDの結果確認

5章 並列化モジュール(MADD_FF)検証フロー

5.1. デレトリ構築

5.2. MADD_FFの検証環境

5.3. MADD_FFの検証実行フロー

5.4. MADD_FFの結果確認

A章 Appendix

本書は【デジタル設計の基礎編】になります。

ハードウェア・エミュレータでの論理検証の基礎
【初級編】

AIチップ
東京大学システム

【目次】

【デジタル設計の基礎編】

1章 論理設計概要

1.1. LSI設計概要

1.2. 論理設計フロー

1.3. 論理回路の基本

1.4. HDL/SystemVerilog概要

2章 検証概要

2.1. 検証について

2.2. 検証手順

2.3. 検証方法

2.4. 検証ツール

3章 仕様説明

3.1. サンプルモジュールの概要

3.2. 基本モジュール MADD

3.3. 並列化モジュール MADD_FF

4章 基本モジュール(MADD)検証フロー

4.1. デレトリ構築

4.2. MADDの検証環境

4.3. MADDの検証実行フロー

4.4. MADDの結果確認

5章 並列化モジュール(MADD_FF)検証フロー

5.1. デレトリ構築

5.2. MADD_FFの検証環境

5.3. MADD_FFの検証実行フロー

5.4. MADD_FFの結果確認

A章 Appendix

本書は【ハードウェア・エミュレータでの論理設計

ハードウェア・エミュレータでの論理検証の基礎
【応用編】

AIチップ設計拠点
東京大学システムデザイン研究センター

【目次】

【デジタル設計の基礎編】

1章 論理設計概要

1.1. LSI設計概要

1.2. 論理設計フロー

1.3. 論理回路の基本

1.4. HDL/SystemVerilog概要

2章 検証概要

2.1. 検証について

2.2. 検証手順

2.3. 検証方法

2.4. 検証ツール

3章 仕様説明

3.1. サンプルモジュールの概要

3.2. 基本モジュール MADD

3.3. 並列化モジュール MADD_FF

4章 基本モジュール(MADD)検証フロー

4.1. デレトリ構築

4.2. MADDの検証環境

4.3. MADDの検証実行フロー

4.4. MADDの結果確認

5章 並列化モジュール(MADD_FF)検証フロー

5.1. デレトリ構築

5.2. MADD_FFの検証環境

5.3. MADD_FFの検証実行フロー

5.4. MADD_FFの結果確認

A章 Appendix

本書は【ハードウェア・エミュレータでの論理設計の基礎(応用)編】になります。

【ハードウェア・エミュレータでの論理設計の基礎(初級)編】

6章 ハードウェア・エミュレータでの論理検証

6.1. エミュレータ概要

6.2. エミュレータでの検証環境

7章 ハードウェア・エミュレータでの検証フロー

7.1. デレトリ構築

7.2. MADD_FFの検証環境

7.3. MADD_FFの検証実行フロー

7.4. MADD_FFの結果確認

A章 Appendix

【ハードウェア・エミュレータでの論理設計の基礎(応用)編】

8章 ニューラルネットワークの回路仕様

8.1. 半書き数字記法のニューラルネットワーク

8.2. サンプルモジュール回路の仕様

9章 ニューラルネットワークの検証フロー

9.1. デレトリ構築

9.2. 検証環境

9.3. 検証実行フロー

9.4. 検証結果確認

A章 Appendix

Accelerating the crypto design - labs

Revision 1.0.0
Mar 2021

Accelerating Crypto Design on Palladium

Agenda

- Understanding the DUT (crypto de
 - Clock analysis
 - Data analysis
 - Functionality (pairing)
- Simulating the design
- Moving the design to Palladium
 - Signal Based Acceleration
- Accelerating the design on Palladium
 - Transaction Based Acceleration
 - Stage 1 - one transaction per WellFF2
 - Stage 2 - one transaction per iteration
- Further Optimizations

Crypto Pairing Design Bring up on EmuForge

Belmont Computing, Inc
July 2021

Agenda

- Overview
 - What is EmuForge
 - EmuForge Overview
 - Features
- Design on boarding
 - Introduction
 - Describing design source file dependencies
 - Core file
 - Writing core file
- Tool on boarding
 - Introduction
 - File list generation
 - Build-run flow
 - Tool specific hpcn file
- Configuration to run design on palladium
 - Emulation config hpcn
 - Running on EmuForge
- Configuration to run design on xcelium
 - Running on Xcelium
 - Tool switchability

デジタル設計教材 デジタル設計の基礎

AIチップ設計拠点
東京大学システムデザイン研究センター

1章 論理設計概要

1.1. LSI設計概要

設計部門が担当する設計工程は、更に

- 【仕様検討】から【機能検証】までを“フロントエンド設計”
- 【実装（レイアウト）】、【実装検証】を“バックエンド設計”
- 両者をインタフェースし、データの加工（【論理合成】）や実装後の検証（【等価/タイミング検証】）等を行う工程を“ミドルエンド設計”

と分ける場合もあります。

半導体の微細化技術が進歩して一つのLSIに搭載できるトランジスタの数が飛躍的に増大したため各工程毎に専門性を高める必要性が強まった為です。

論理回路部を設計・検証する工程を一般的に論理設計と称します。論理設計については、次の1.2章で説明します。



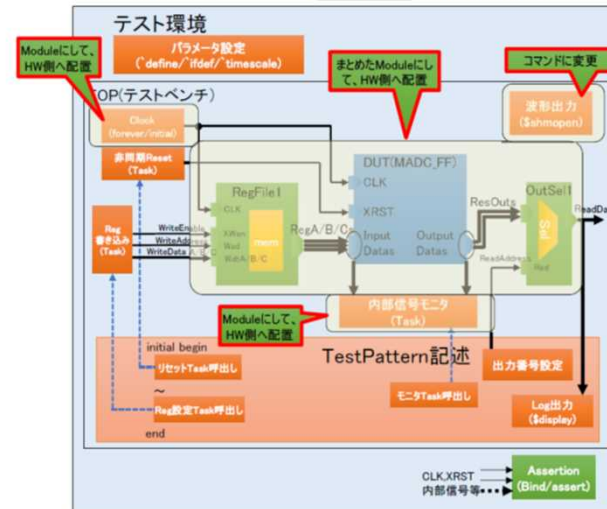
ハードウェア・エミュレータでの論理検証の基礎 【初級編】

AIチップ設計拠点
東京大学システムデザイン研究センター

7章 ハードウェア・エミュレータでの検証フロー

7.2. MADC_FFの検証環境

7.2.1 テストベンチ 【シミュレータでの検証環境イメージ図】

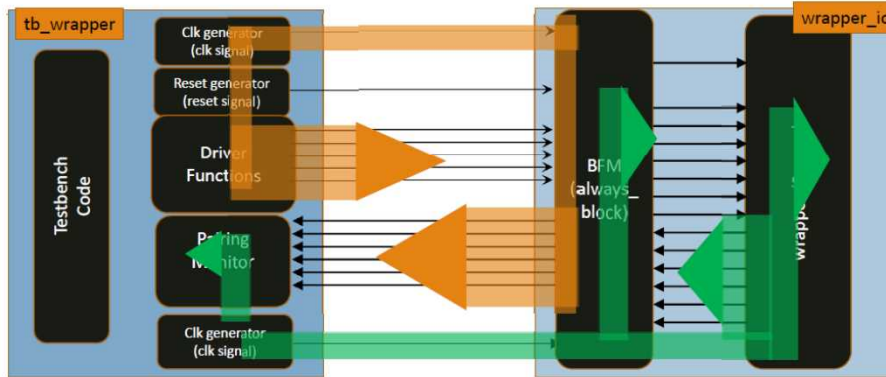


左図は「デジタル設計の基礎編」で説明したシミュレータでのMADC_FF検証環境のイメージに対し、エミュレータでの対応する項目を示しています。詳細は次ページ以降を参照ください。

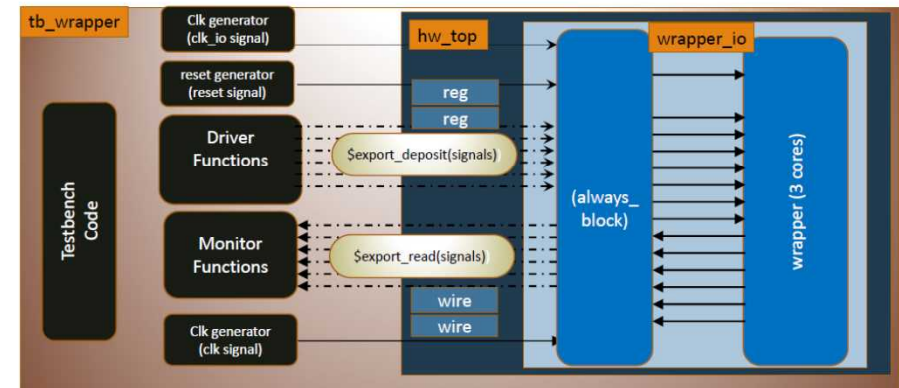
<p style="text-align: center;">デジタル設計 デジタル設計(</p> <p style="text-align: center;">AIチップ設計拠点 東京大学システムデザイン</p> <p style="text-align: center;">--</p>	<p style="text-align: center;">ハードウェア・エミュレータでの論理検証の基礎 【応用編】</p> <p style="text-align: center;">AIチップ設計拠点 東京大学システムデザイン研究センター</p> <p style="text-align: center;">--</p>	<p style="text-align: center;">ハードウェア・エミュレータでの論理検証の基礎 【初級編】</p> <p style="text-align: center;">AIチップ設計拠点 東京大学システムデザイン研究センター</p> <p style="text-align: center;">--</p>
<p>1章 論理設計概要 1.1. LSI設計概要</p> <p>設計部門が担当する設計工程は、更に</p> <ul style="list-style-type: none"> ➢ 【仕様検討】から【機能検証】までを“フロントエンド設計” ➢ 【実装（レイアウト）】、【実装検証】を“バックエンド設計” ➢ 両者をインタフェースし、データの加工（【論理合成】）や後の検証（【等価/タイミング検証】）等を行う工程を“ミッドエンド設計” <p>と分ける場合もあります。 半導体の微細化技術が進歩して一つのLSIに搭載できるトランジスタの数が飛躍的に増大したため各工程毎に専門性を必要とすることが強まった為です。</p> <p>論理回路部を設計・検証する工程を一般的に論理設計と称します。論理設計については、次の1.2章で説明します。</p> <p style="text-align: center;">--</p>	<p>8章 ニューラルネットワークの回路仕様 8.1 手書き数字認識のニューラルネットワーク</p> <p>(1)全結合ニューラルネットワークの構成 28×28(784)画素の濃淡画素データを入力し、出力層で0~9の認識結果を出力します。</p> <p style="text-align: center;">--</p>	<p>ハードウェア・エミュレータでの検証フロー</p> <p>ハードウェア・エミュレータでの検証環境イメージ図</p> <p>左図は「デジタル設計の基礎編」で説明したシミュレータでのMADC_FF検証環境のイメージに対し、エミュレータでの対応する項目を示しています。 詳細は次ページ以降を参照ください。</p> <p style="text-align: center;">--</p>

2. 成果: 「エミュレータを利用した検証の高速化に向けたテストベンチ作成方法」教材の抜粋

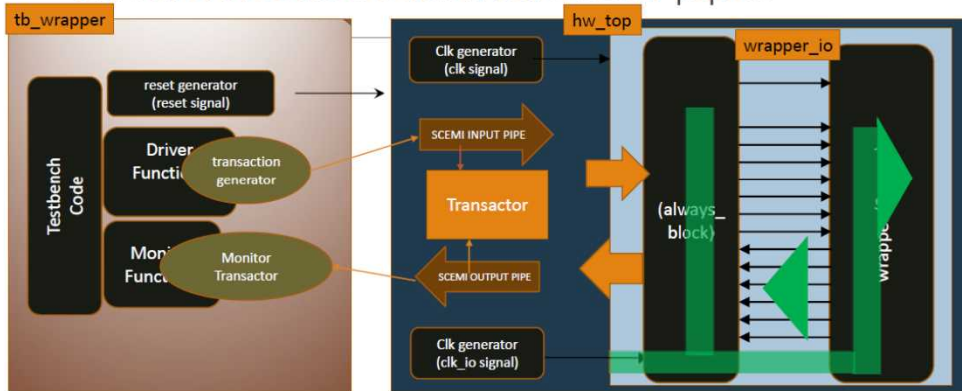
Clock usage



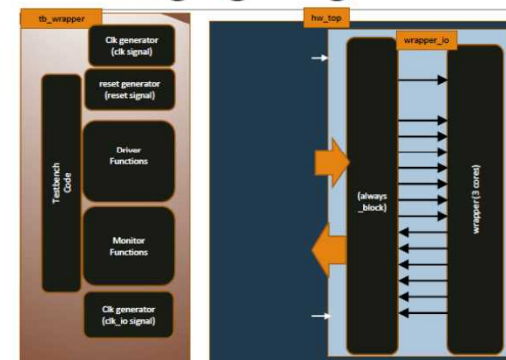
Overall SBA architecture



TBA architecture based on SCEMI pipes



Managing clk generation



- Create a two top architecture
 - Hw_top which is a wrapper on top of the wrapper_io
 - tb_wrapper which is the entire testbench
 - Two independent tops
 - Testbench does not instantiate the DUT
- Move the clock generations into the Palladium
- Make testbench independent of clock signals

2. 成果:エミュレータの利用加速に向けた取り組み

本プロジェクトにおける人材育成活動の一環として、高度な論理検証技術を有する人材育成を目指した取り組み

- ロジックエミュレータ Palladium 利用方法の最新情報共有
- ロジックエミュレータ Palladium 利用事例の共有と、相互学習の場の提供
- ベンダからの最新情報提供、問題解決に向けたFAQ共有の場の構築

【エミュレータを利用した検証の高速化に向けたテストベンチ作成方法】

日時 2021年6月30日 (水) 13:00 - 17:00

【シミュレータ・エミュレータによる協調検証環境 (EmuForge) を用いた検証効率向上】

日時 2021年7月1日 (木) 13:00 - 16:00

【エミュレータトレーニング】

日時 2021年11月30日 (木) 13:00 - 17:30

【エミュレータ・ワークショップ】

日時 2022年3月10日 (木) 13:30 - 17:00

【エミュレータ・セミナー】

日時 2022年4月28日 (木) 13:30 - 15:00

【エミュレータトレーニング】

日時 2022年6月7日 (木) 13:30 - 17:30

【エミュレータ・ワークショップ】

日時 2022年9月13日 (火) 13:30 - 17:00

【RISC-VベースSoCのシミュレーション・エミュレーション協調検証トレーニング】

日時 2022年9月28日 (水) 13:30 - 17:30

【拠点フォーラム:検証特集】

日時 2022年12月23日 (金) 13:30 - 17:00

【エミュレータトレーニング】

日時 2023年2月14日 (火) 13:30 - 17:30

【エミュレータ・ワークショップ】

日時 2023年3月14日 (火) 13:30 - 17:00

【RISC-VベースSoCのシミュレーション・エミュレーション協調検証トレーニング(再)】

日時 2023年3月20日 (月) 10:00 - 17:00

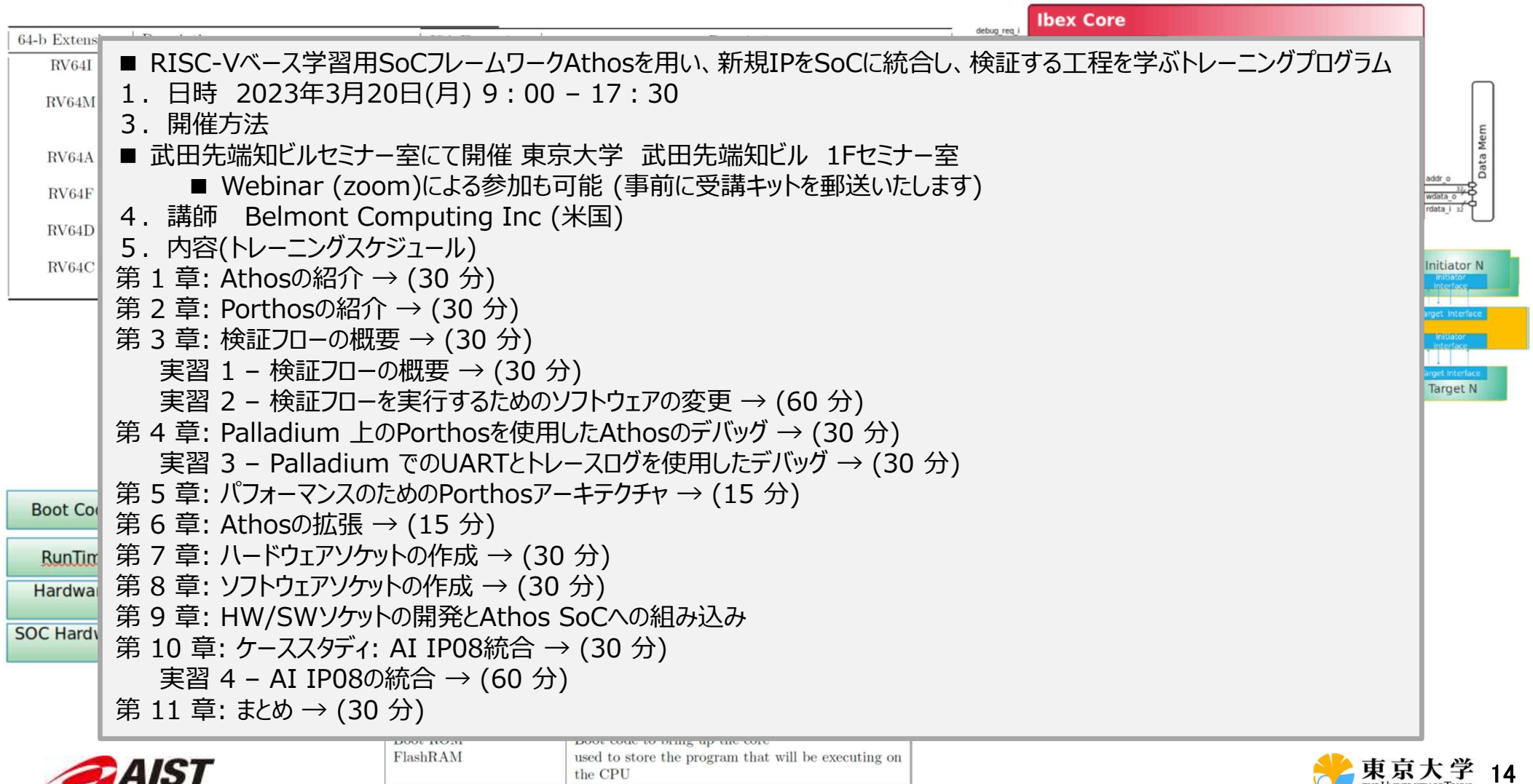
Open Sourceな環境でのSoCのハード・ソフト協調検証環境の実現を目指して

1. フリーなRISC-V, NoC, IOを用いたSoCプラットフォームの構築
2. プラットフォーム上でのソフトウェアの実行環境構築と検証環境の実現
3. セミナーの実施

2022/1/26 「RISC-V SoC検証法」セミナー開催・・・少々調整不足でハンズオンはできず

2022/9/28 「RISC-VベースSoCのハード/ソフト・シミュレータ/エミュレータ協調検証」トレーニング

2023/3/20 「RISC-VベースSoCのハード/ソフト・シミュレータ/エミュレータ協調検証」トレーニング(再)



Ibex Core

64-b Extens... debug req |

RV64I
RV64M
RV64A
RV64F
RV64D
RV64C

Boot Co...
RunTir...
Hardwa...
SOC Hardv

addr_o
wdata_o
rddata_i

Data Mem

Initiator N
initiator interface
target interface
initiator interface
target interface
Target N

FlashRAM
used to store the program that will be executing on the CPU

AIST

東京大学 THE UNIVERSITY OF TOKYO 14

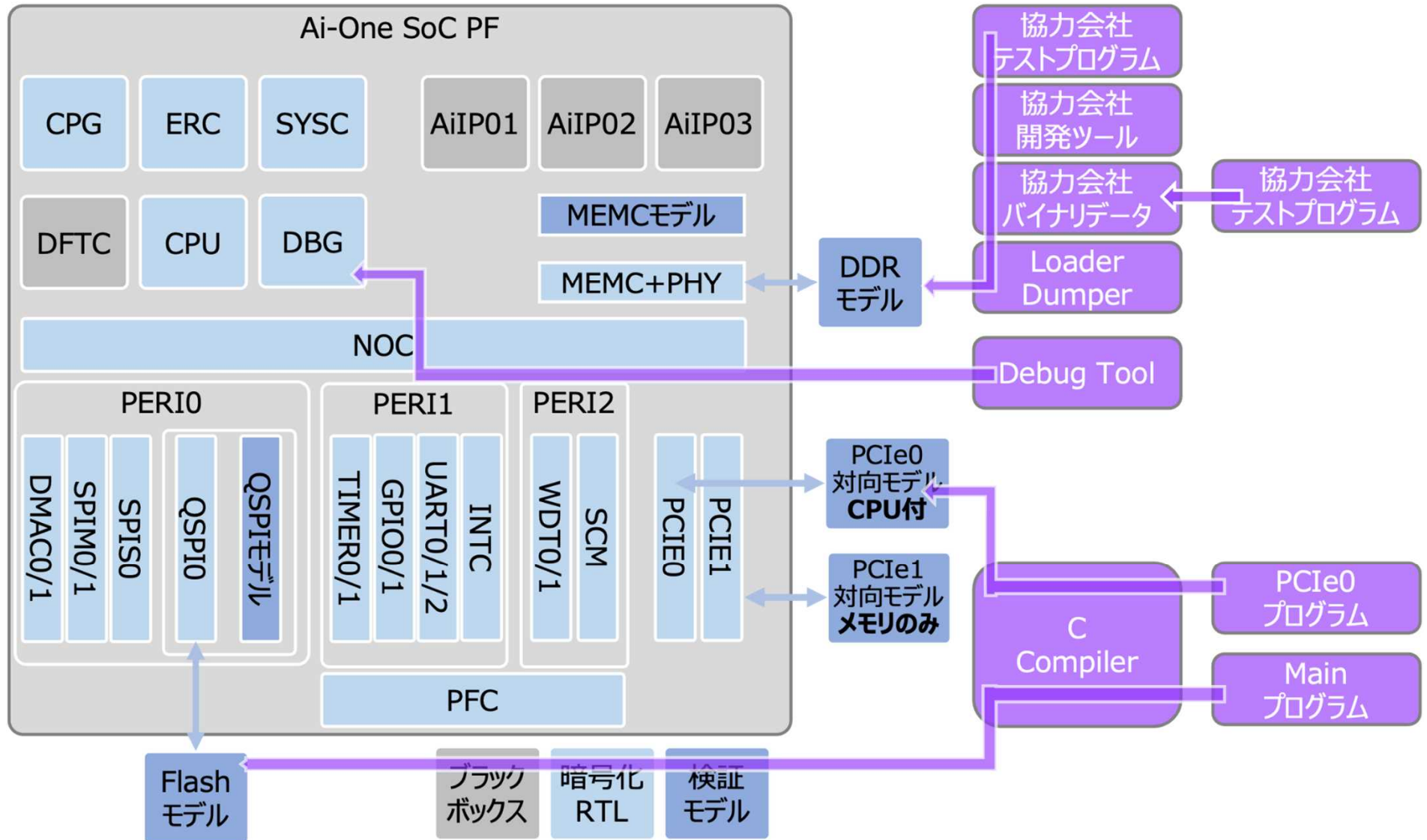
■ RISC-Vベース学習用SoCフレームワークAthosを用い、新規IPをSoCに統合し、検証する工程を学ぶトレーニングプログラム

1. 日時 2023年3月20日(月) 9:00 - 17:30
3. 開催方法
 - 武田先端知ビルセミナー室にて開催 東京大学 武田先端知ビル 1Fセミナー室
 - Webinar (zoom)による参加も可能 (事前に受講キットを郵送いたします)
4. 講師 Belmont Computing Inc (米国)
5. 内容(トレーニングスケジュール)
 - 第1章: Athosの紹介 → (30分)
 - 第2章: Porthosの紹介 → (30分)
 - 第3章: 検証フローの概要 → (30分)
 - 実習1 - 検証フローの概要 → (30分)
 - 実習2 - 検証フローを実行するためのソフトウェアの変更 → (60分)
 - 第4章: Palladium 上のPorthosを使用したAthosのデバッグ → (30分)
 - 実習3 - Palladium でのUARTとトレースログを使用したデバッグ → (30分)
 - 第5章: パフォーマンスのためのPorthosアーキテクチャ → (15分)
 - 第6章: Athosの拡張 → (15分)
 - 第7章: ハードウェアソケットの作成 → (30分)
 - 第8章: ソフトウェアソケットの作成 → (30分)
 - 第9章: HW/SWソケットの開発とAthos SoCへの組み込み
 - 第10章: ケーススタディ: AI IP08統合 → (30分)
 - 実習4 - AI IP08の統合 → (60分)
 - 第11章: まとめ → (30分)

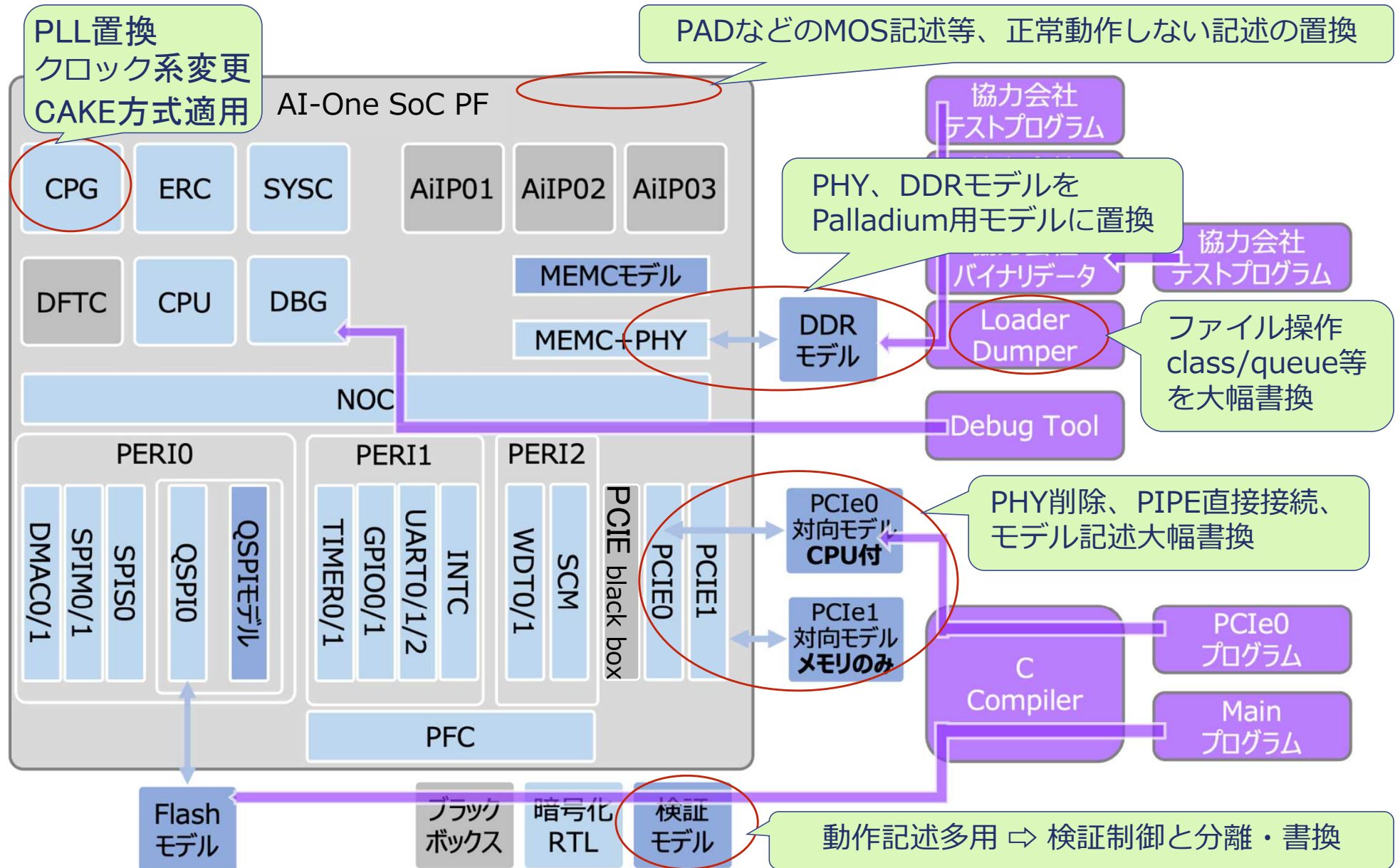
より実用的なSoCでの検証環境の実現を目指して

AI-One SoCプラットフォームの検証環境を: RTLシミュレーション、エミュレータ上に構築

AI-Oneに搭載のCPU, NoC, ペリフェラルIPを実装し、SoCプラットフォームの検証、搭載AI-IPの検証を実現



・ エミュレータポーティング向けのRTL変更



IPコア単体の検証の場合の速度向上例

Simulation performance data

No. of Pairings	Xcelium Run time	
	Wall clock (mm:ss)	Simulation time (ns)
1	00:17	425,400
2	00:24	546,630
5	00:53	989,430
10	01:35	1,674,750
50	07:23	7,315,800
100	14:34	14,366,190
250	36:29	35,518,080
500	1:13:23	70,772,580
700	1:39:54	98,974,950

Average performance
8.8 secs per pair

Palladium runtimes (v2_1_sba)

No. of Pairings	Palladium Run time		
	Wall clock (mm:ss)	Simulation time (ns)	Sp
1	00:26	425,400	0.65
2	00:27	546,630	0.88
5	00:28	989,430	1.89
10	00:30	1,674,750	3.16
50	00:50	7,315,800	8.86
100	01:15	14,366,190	11.65
250	02:29	35,518,080	14.5
500	04:34	70,772,580	16
700	06:13	98,974,950	16

16x speedup over simulation

Palladium run times (writefp2 level)

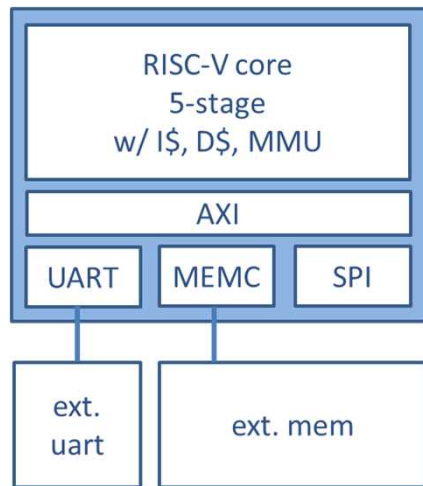
No. of Pairings	v2_1 Xcelium Run time (Compile+elab = 11s)		v2_1_scemi_static Palladium emulation	
	Wall clock (mm:ss)	Simulation time (ns)	Wall clock (mm:ss)	Speed up wrt v2_1
1	00:17	425,400	00:25	0.68
2	00:24	546,630	00:25	0.96
5	00:53	989,430	00:26	2
10	01:35	1,674,750	00:25	3.8
50	07:23	7,315,800	00:28	15
100	14:34	14,366,190	00:30 (tbsyncs = 4130)	29.1
250	36:29	35,518,080	00:58	59
500	1:13:23	70,772,580	01:01	91
700	1:39:54	98,974,950	01:03	103

91x speedup over simulation
Average performance
0.1 sec per pairing

その他の設計例を用いた速度向上検証結果

対象デザイン: 10MG程度の数値演算回路		シミュレータ		エミュレータ			
		コンパイル時間[s]	実行時間 (loop 50)[s]	コンパイル時間[s]	実行時間 (loop 50)[s]	実行時間 (loop 500)[s]	
RTL	何も対処しない	NA	410.98	391.82	57.79	239.95	x17
	データの入出力にメモリを使用	--	--	388.75	45.93	188.10	x22
	制御入力をメモリ使用			442.18	27.40	41.97	x98
	SCEMI Pipeを使用			468.20	27.40	42.24	x20,000
ネットリスト	何も対処しない	NA	419,806.74	3,978.91	60.59	208.94	x21,000
	データの入出力にメモリを使用			4,633.13	78.09	198.74	x100,000
	制御入力をメモリ使用			3,910.68	33.42	42.00	x100,000
	SCEMI Pipeを使用			3,898.73	37.27	41.57	x100,000

2. 成果: エミュレータによる検証速度の向上: RISC-Vの例



ブロック図

- 5段パイプライン、命令・データキャッシュ、MMUあり
- 東工大一色研の高位システム設計検証環境で開発
 - C/C++からVerilog RTL、テストベンチを生成
- システム構成
 - RISC-Vコア、AXIバス、外部メモリ/F、UART I/F、SPI I/F
- 外部UARTモジュール
 - UART出力される文字コードを\$write, \$displayで表示
- テストベンチ
 - Verilogのinitial文で動作記述、クロック・リセットも記述
 - 毎サイクル、テスト対象の入出力ログを読み出し
 - 検証のために外部端子以外も出力し32ビットに圧縮してチェック
 - 入力ログはテスト対象に入力
 - ログはバークレーブートローダ (bbl) によるLinuxのブート

	parameters			boot cycles		Linux Boot			1M cycles		
	ext. mem.	log check	UART			RTL	Gate	G/R	RTL	Gate	G/R
1	log	Yes	1/2	116M	Emu.	13'38"	15'06"	1.11	0'46"	2'01"	2.63
					Sim.	262:46'20"	561:47'40"	2.14	2:15'55"	4:50'35"	2.14
					S/E	1,156	2,232		177	144	
7	mem.	No	1/234	132M	Emu.	2'20"	2'31"	1.08	0'22"	0'53"	2.41
					Sim.	145:09'48"	609:19'36"	4.20	1:05'59"	4:36'58"	4.20
					S/E	3,733	14,527		180	314	

2. 成果: エミュレータによる検証速度の向上: AI-Oneの例

- 論理合成可能な**DUT部分は比較的容易**にポーティング可能
- **動作記述を多用**している部分は、大幅な書き換え**も**必要
- **書き換えやデバッグにはエミュレータに対する深い理解が必要**
- エミュレータ用の記述は、エミュレータ専用ライブラリ以外はシミュレータでも動作するので、最初から**双方での運用を前提**に環境構築不可欠
- エミュレータの速度は、**700kHz~800kHz**でシミュレータの**100~2,500倍**
- 本来最大で2MHz程度が期待、今回は、短いテストサンプルのため**初期化オーバーヘッド**の割合が大きく、速度向上率が劣化

Pattern	シミュレータ (kHz)	CAKEなし (kHz)	CAKE 1x (kHz)	CAKEなし 速度向上	CAKE 1x 速度向上
base	7.62	499.32	817.39	66	105
base_pcie_000	0.69	374.49	769.58	543	1,115
base_pcie_001	0.33	404.15	799.37	1,225	2,422
base_pcie_002	0.31	543.21	790.48	1,752	2,550
base_pcie_004	0.59	549.47	782.13	931	1,326
base_pcie_005	0.65	405.72	784.58	624	1,207
base_memc	3.89	457.75	703.52	118	181

成果

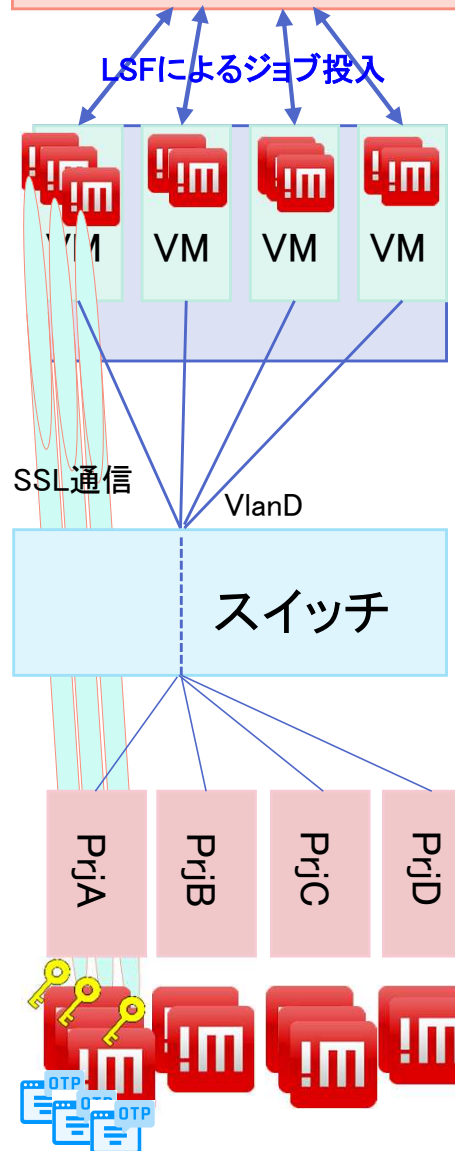
- 2-1で整備した拠点におけるEDA利用環境に即したツール利用設定を整備しWEBにて公開
- 高位合成を使ったデジタル設計【基礎編】【続基礎編】【応用編】【実践編】を作成しHPにて公開した
- デジタル物理設計教材【基礎編】【応用編】【低電力編】を作成しHPにて公開した

意義

- マニュアルに加え、導入ツールに即した教材の整備、ツールごとの環境変数の一括した整備により、EDAツール利用初心者の利用開始の障壁が劇的に低減し、利用者の増加が期待される
- 各種教材の整備によりより実践的な設計にまで容易に取り組める環境を整備した

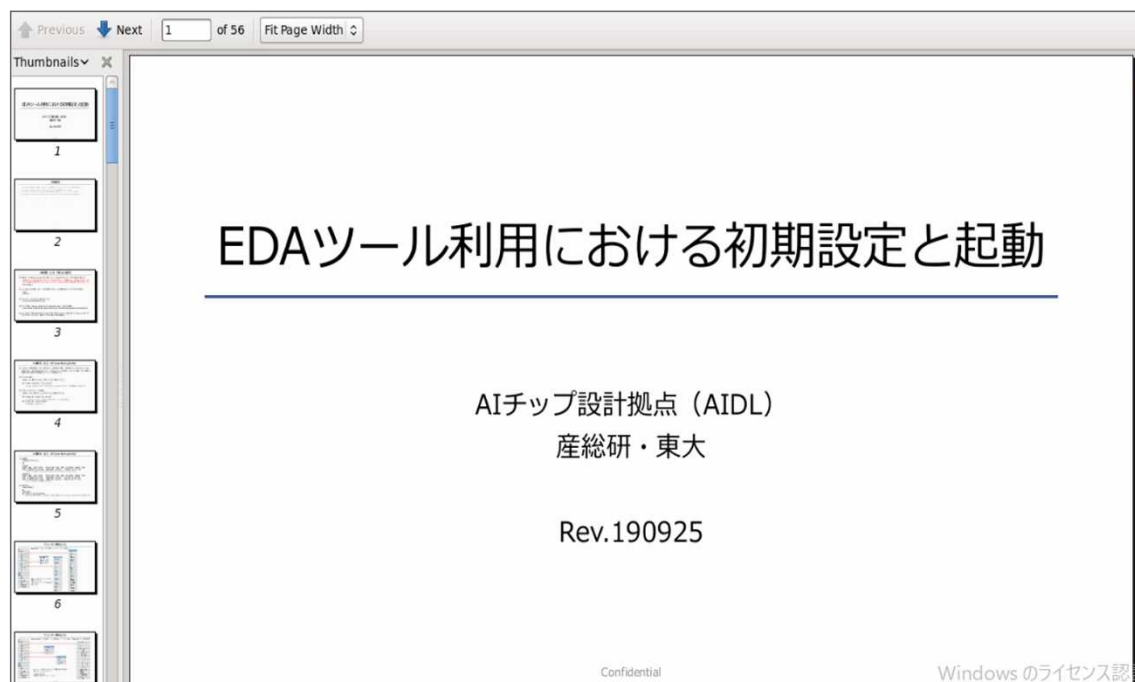
2. 成果(成果・意義)

計算サーバ



左:2-1で整備した拠点のEDA利用環境
下:拠点の環境に即したツール利用のための環境設定ファイルをツール毎に整備、利用法の作成と公開

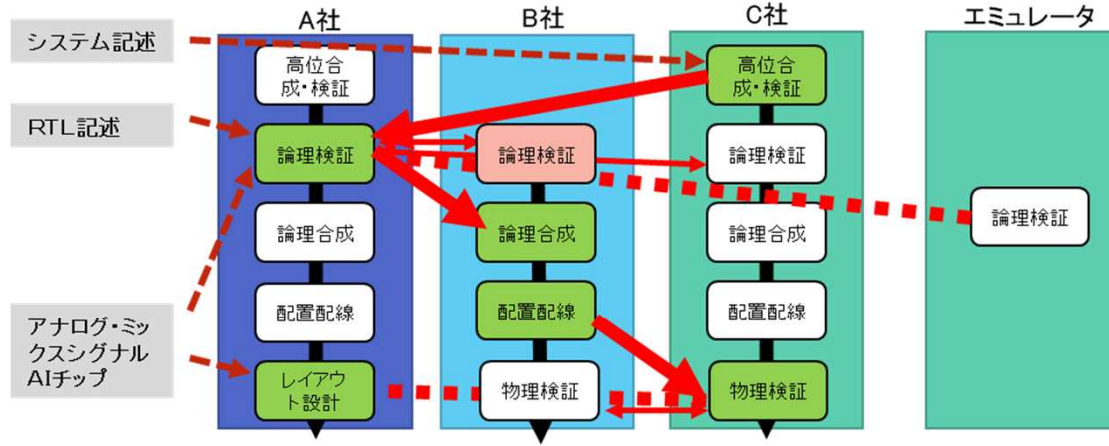
ベンダー	設定ファイル
Cadence	cshrc.innovus, cshrc.joules, cshrc.mdv, cshrc.qrc, cshrc.stratus, cshrc.virtuoso, cshrc.vxe, cshrc.genus, cshrc.jasper, cshrc.lec, cshrc.pvs, cshrc.spectre.batch, cshrc.tempus, cshrc.voltus, cshrc.xcelium
Mentor	cshrc.calibre, cshrc.catapult, cshrc.powerpro, cshrc.slec
Synopsys	cshrc.ct, cshrc.hspice, cshrc.lc, cshrc.pt, cshrc.spyglass, cshrc.vcs, cshrc.verdi, cshrc.cc, cshrc.dc, cshrc.icc, cshrc.prime, cshrc.spyglass-lint, cshrc.starrc, cshrc.vcs-mx, cshrc.waveview



2. 成果: EDAツールの協調設計に向けた教材整備



高位合成→シリコンまでのツールチェーンの完成を目指す



- 論理検証**
- ・A社: 機能
 - ・A社: 面積
 - ・B社: タイミ
 - ・C社: 電力
 - ...
- 物理検証**
- ・C社: 機能
 - ・C社: 設計
 - ・B社: タイミ
 - ・A社: 電力
 - ...

デジタル設計教材 高位合成を使ったデジタル設計【基礎編】	デジタル設計教材 デジタル物理設計【基礎編(ブロック設計)】
AIチップ設計拠点 東京大学システムデザイン研究センター	AIチップ設計拠点 東京大学システムデザイン研究センター
【目次】 高位合成を使ったデジタル設計【基礎編】 1章 高位合成の概要 2章 高位合成の設計フロー 3章 高位合成の検証 4章 高位合成の物理設計 5章 高位合成のレイアウト 6章 高位合成の物理検証 7章 高位合成の物理設計 8章 高位合成の物理検証 9章 高位合成の物理設計 10章 高位合成の物理検証 11章 高位合成の物理設計 12章 高位合成の物理検証 13章 高位合成の物理設計 14章 高位合成の物理検証 15章 高位合成の物理設計 16章 高位合成の物理検証 17章 高位合成の物理設計 18章 高位合成の物理検証 19章 高位合成の物理設計 20章 高位合成の物理検証	【目次】 デジタル物理設計【基礎編(ブロック設計)】 1章 デジタル物理設計の概要 2章 デジタル物理設計の設計フロー 3章 デジタル物理設計の検証 4章 デジタル物理設計の物理設計 5章 デジタル物理設計のレイアウト 6章 デジタル物理設計の物理検証 7章 デジタル物理設計の物理設計 8章 デジタル物理設計の物理検証 9章 デジタル物理設計の物理設計 10章 デジタル物理設計の物理検証 11章 デジタル物理設計の物理設計 12章 デジタル物理設計の物理検証 13章 デジタル物理設計の物理設計 14章 デジタル物理設計の物理検証 15章 デジタル物理設計の物理設計 16章 デジタル物理設計の物理検証 17章 デジタル物理設計の物理設計 18章 デジタル物理設計の物理検証 19章 デジタル物理設計の物理設計 20章 デジタル物理設計の物理検証

デジタル設計教材
高位合成を使ったデジタル設計【基礎編】

AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル設計教材
高位合成を使ったデジタル設計【実践編 2】

AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル設計教材
高位合成を使ったデジタル設計【実践編 3】

AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル設計教材
高位合成を使ったデジタル設計【実践編 4】

AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル設計教材
高位合成を使ったデジタル設計【実践編 5】

AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル設計教材テキスト
デジタルチップ・物理設計フロー
【高位合成～物理設計編】

2023年2月24日
AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル設計教材
物理設計【低消費電力編】

AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル設計教材
物理設計【応用編】

AIチップ設計拠点
東京大学システムデザイン研究センター



2. 成果: 「高位合成を使ったデジタル設計: 基礎編・続基礎編」教材の抜粋

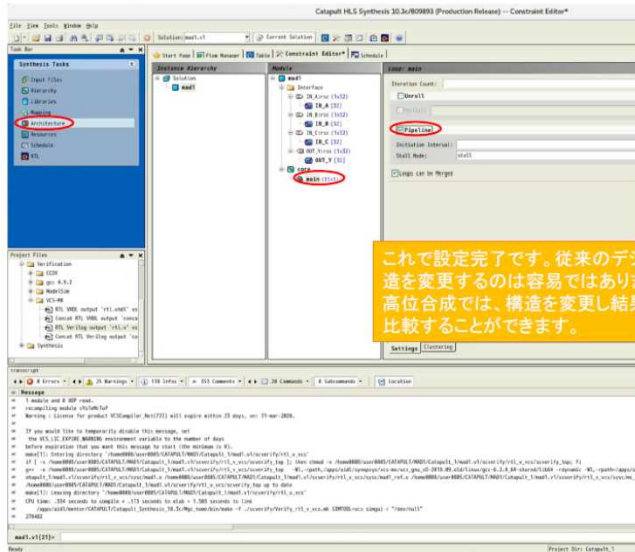
デジタル設計教材 高位合成を使ったデジタル設計

AIチップ設計拠点
東京大学システムデザイン研究セン

14章 パイプライン化

14.1. Architectureの変更

Catapultに戻り、Task Barの「Architecture」をクリックします。次にModule>mainを選択し、最後に右下の「Apply」をクリックします。



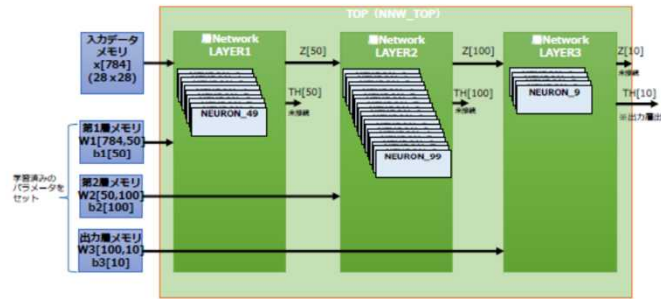
デジタル設計教材

高位合成を使ったデジタル設計【続基礎編】

AIチップ設計拠点
東京大学システムデザイン研究センター

5章 機能仕様書

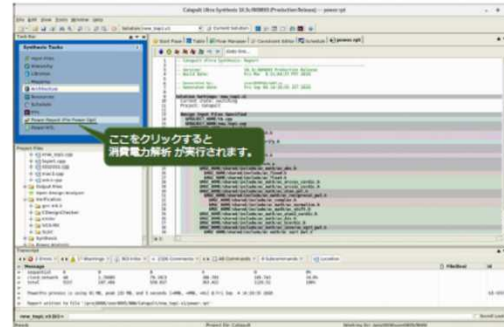
(4) 構成 (全体イメージ図)



8章 消費電力最適化フロー

8.1. 消費電力解析

Catapultを使用して、デザインで消費される電力を解析します。このフローでは、CatapultがSCVerify内のデストベンチを使用してスイッチング情報を収集し、PowerProを使用して、デザイン内のレジスター、メモリ、および組み合わせたロジックの動的電力とリーク電力を分析します。Synthesis Tasksメニューの「Power Report (Pre Power Opt)」をクリックします。



```

W1, W2, W3 = network["W1"], network["W2"], network["W3"]
b1, b2, b3 = network["b1"], network["b2"], network["b3"]

a1d = fdot1x2(x, W1)
a1 = fplus1x1(a1d,b1)
z1 = fsigmoid1(a1)

a2d = fdot1x2(z1, W2)
a2 = fplus1x1(a2d,b2)
z2 = fsigmoid1(a2,fxmax)

a3d = fdot1x2(z2, W3)
a3 = fplus1x1(a3d,b3)
y = fsigmoid1(a3)
    
```

行列の積計算
行列の加算計算
行列のSigmoid計算

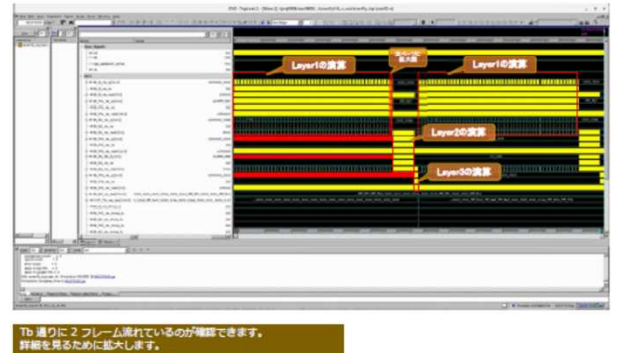
7章 RTL生成と等価性検証フロー

レポートをチェックします。まずは、満たしていない箇所を探していきます。

Questa Statement Coverage Report

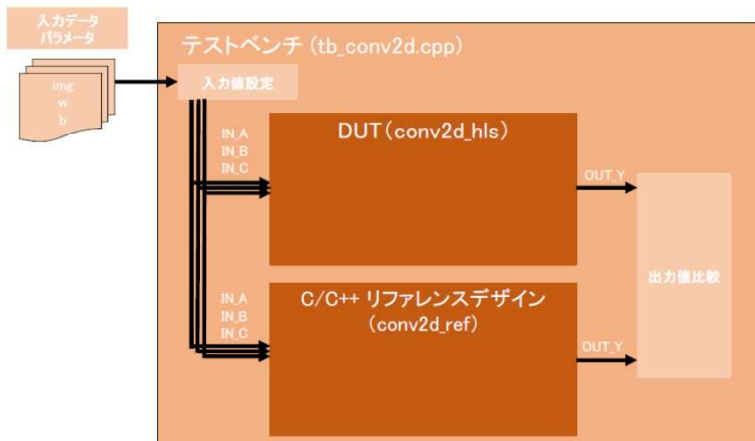


9章 シミュレーション



2. 成果: 「高位合成を使ったデジタル設計: 実践編」教材の抜粋

実践編2では CONV2D のリファインを検討しやすくするために、テストベンチを下図のように C/C++ と H/W 用の記述をそれぞれ置いて、同じ入力を与えて結果を比較する構成としました。本編でも同様なテストベンチを用いて残りのレイヤのリファインを行っていきます。

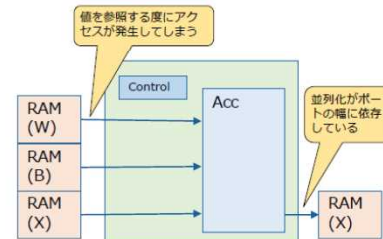


- 16 -

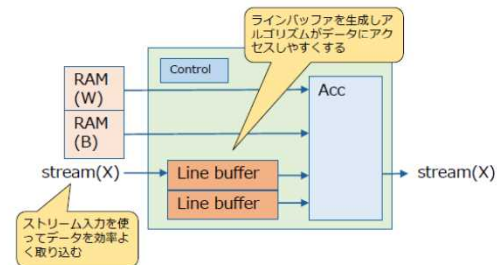
ここからは回路の最適化を考えていきます。「実践編2」で最適化した「CONV2D」以外の以下のレイヤに関して最適化していきます。

- 3.1.1. 「MAXPOOLING2D」
- 3.1.2. 「FLATTEN」
- 3.1.3. 「DENSE2D」

以前の実装



本編で目指す実装



「実践編2」で「CONV2D」に対して行ったのと同様に、画像データをストリームで扱うよう変更しブロック内のラインバッファに保存してアルゴリズムが効率よく演算できるようにします。

- 17 -

3章 高位合成設計フローと回路の最適化

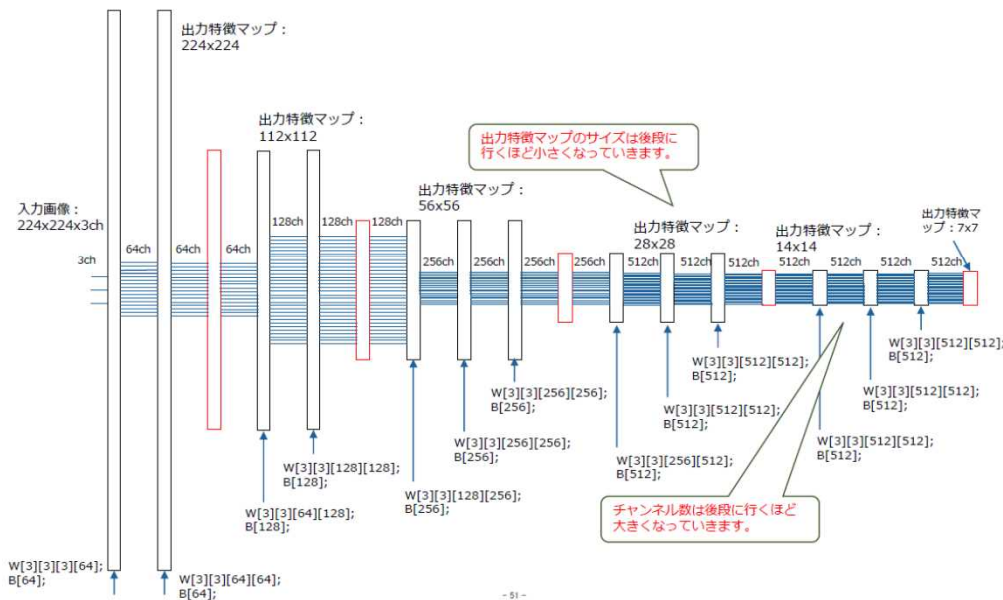
Layer (type)	Output Shape	Param #	
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792	Block1 のパラメータ数
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928	1792 + 36928 = 38720
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0	
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856	Block2 のパラメータ数
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584	73856 + 147584 = 221440
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0	
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168	Block3 のパラメータ数
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080	295168 + 590080 + 590080 = 1475328
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080	
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0	
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160	Block4 のパラメータ数
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808	1180160 + 2359808 + 2359808 = 5899776
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808	
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0	
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808	Block5 のパラメータ数
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808	2359808 + 2359808 + 2359808 = 7079424
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808	
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0	
flatten (Flatten)	(None, 25088)	0	
fc1 (Dense)	(None, 4096)	102764544	Full Connection以降のパラメータ数
fc2 (Dense)	(None, 4096)	16781312	102764544 + 16781312 + 4097000 = 123642856
predictions (Dense)	(None, 1000)	4097000	
Total params: 136,357,544			
Trainable params: 136,357,544			
Non-trainable params: 0			

左図は vgg16 モデルのレイヤ構成を示しています。赤枠で囲った部分が各レイヤが必要となるパラメータ数です。
参考として 32bit / 1パラメータで換算すると、
Block1: 155KB
Block2: 886KB
Block3: 5.9MB
Block4: 23.6MB
Block5: 28.4MB
それ以降: 495MB
となります。

パラメータ数は後段に行くほど大きくなっていきます。

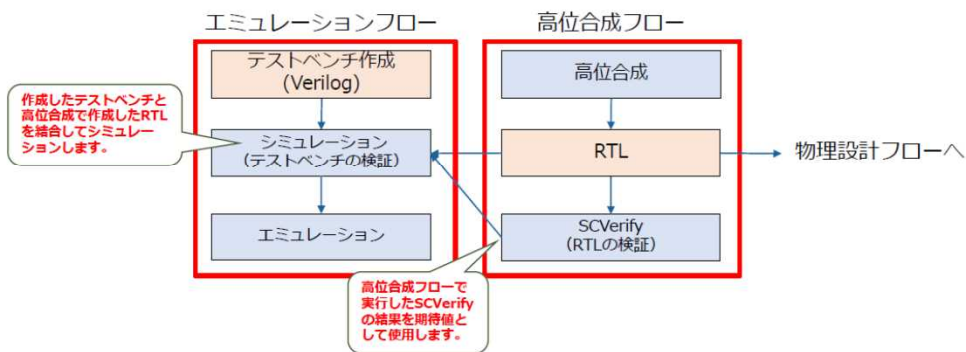
- 50 -

3章 高位合成設計フローと回路の最適化



- 51 -

2. 成果: 「高位合成を使ったデジタル設計:実践編」教材の抜粋



今回使用した Catapult 2021.1 は、エミュレータに対応していないため、C++で記載したテストベンチを Verilog で書き換え高位合成で作成した RTL と結合してシミュレーションを実行します。その際の期待値として、高位合成の SCVerify の結果を使います。また、この環境を使ってエミュレータを実行します。

高位合成【実践編】では、実用的なニューラルネットワークとして「VGG16」を高位合成ツールを使って設計しました。高位合成ツールで設計するためには、C/C++ 言語での実装が必要のため、Python モデルを参考にしながら C/C++ 言語を使って設計し、高位合成ツールを使って RTL を生成しました。ところが、C/C++ 言語でアルゴリズムを実装しただけでは期待した通りの H/W の生成には至りませんでした。そこで、高位合成【実践編 2】では「VGG16」中のレイヤである「CONV2D」に対して期待している H/W を生成するためのコードの修正「リファイン」を実施しました。さらに高位合成【実践編 3】では、高位合成【実践編 2】と同様の「リファイン」を「VGG16」の残りのレイヤに対しても行い、「VGG16」全体として期待する H/W を生成するコードを作成しました。その後、高位合成【実践編 4】ではチップ化を考慮し、物理設計へ渡す RTL を生成しました。



【実践編 4】にてパラメータ保持ブロックと推論実行ブロックそれぞれを高位合成でRTLを作成しました。

1章 はじめに

1.3. 高位合成【実践編 5】の概略

高位合成【実践編 5】では、高位合成【実践編 4】で作成したチップ化を考慮した「VGG16」のサンプル回路を検証する方法について説明します。具体的には下記の作業を行います。

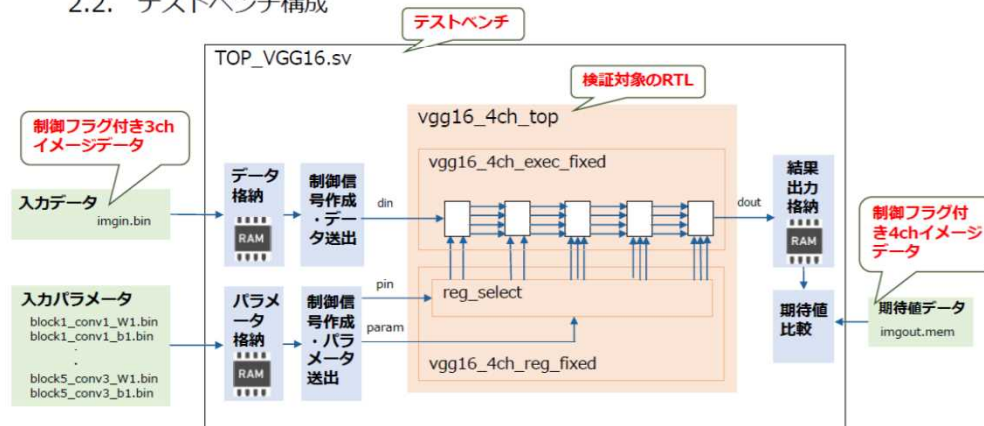
- ・シミュレータ「Cadence Xcerium」による検証
- ・エミュレータ「Cadence Palladium」による検証

シミュレータに対してエミュレータがどの程度加速できるのかを体験します。



2章 エミュレーションのためのテストベンチ作成

2.2. テストベンチ構成



次にテストベンチ構成を考えます。入力データと入力パラメータは C++ のシミュレーション時に実際に入力したデータをダンプして作成します。データ格納部・パラメータ格納部にて一旦RAMにすべてのデータを取り込みます。制御信号作成・データ送出部にて RTL に対するストリーム制御信号を生成し、データを送出します。パラメータも同様です。結果出力格納部にて出力データをすべてRAMに格納し、シミュレーション終了時に期待値データと比較判定を行います。

2. 成果: 「デジタル物理設計: 基礎編」教材の抜粋

デジタル設計教材

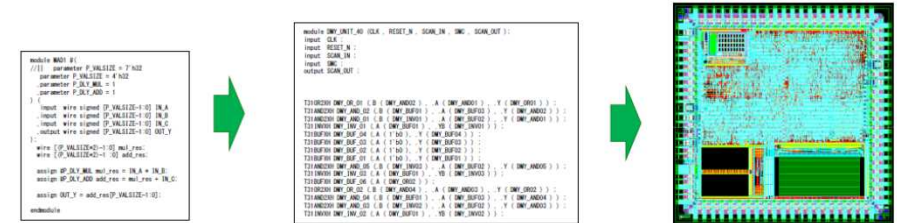
デジタル物理設計【基礎編(ブロック設計)】

AIチップ設計拠点
東京大学システムデザイン研究センター

デジタル物理設計とは、論理設計で記述した抽象的な論理回路に対して、実際にどの素子を使うか、それらをどう配置し、どう配線するかを決める工程です。

一般的に論理回路の設計は最終的に配置実装する素子レベルではなく、もっと抽象的な記述で行います。そのままではLSIとして製造できないため、素子単位での具体的な回路に変換する作業を行います。まず素子による回路構成に変換する論理合成、その後配置配線を行う自動レイアウトで工場での製造に必要な最終データを作成します。

デジタル物理設計では、論理設計で確定した機能を変更しないように、クロック周波数、入出力仕様、ピン配置、消費電力、面積、形状などの目標を達成できるように設計を行います。



CONFIDENTIAL

- 1 -

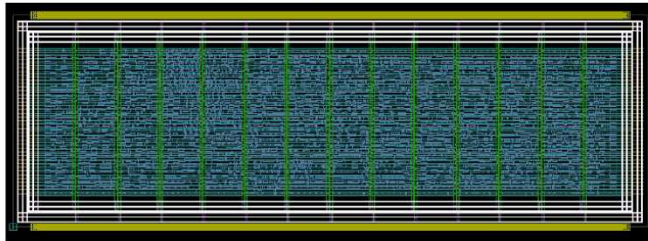
CONFIDENTIAL

- 4 -

5章 P&Rレイアウト工程概要

5.2.3 コマンドによるプラン作成

【create_powerの結果確認】【RING】



```
icc2_shell> change_selection [get_shapes {PATH_35* PATH_36*}]
```

pns_example.tclで設定した通りRINGが形成されているか確認

```
/proj0025/Work_PnR/dp/TOP/rm_icc2_dp_scripts/pns_example.tcl
```

```
# PG RING CREATION
create_pg_ring_pattern ring_pattern -horizontal_layer M5 ¥
-horizontal_width {4.5} -horizontal_spacing {0.5} ¥
-vertical_layer M6 -vertical_width {4.5} -vertical_spacing {0.5}
set_pg_strategy core_ring ¥
-pattern {{name: ring_pattern} ¥
{nets: {VDD VSS}} {offset: {3 3}} -core
```

RING除去コマンドはこちら

```
/proj0025/Work_PnR/dp/TOP/rm_icc2_dp_scripts/compile_pg_example.tcl
compile_pg -strategies core_ring
```

```
remove_routes -ring
```

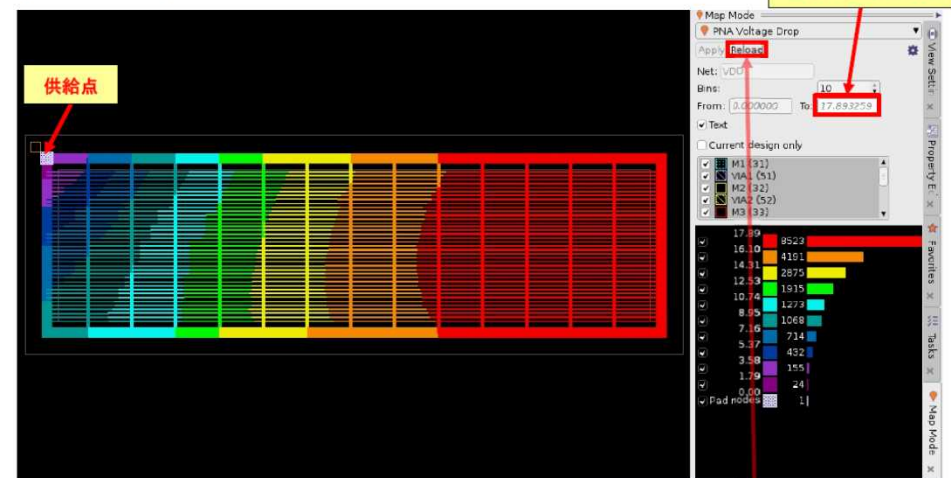
CONFIDENTIAL

- 10 -

5章 P&Rレイアウト工程概要

5.7.4 IR-Drop と EM について

【IR-Drop(VDD)】



ReloadでVDD/VSSのどちらかに選択可能

CONFIDENTIAL

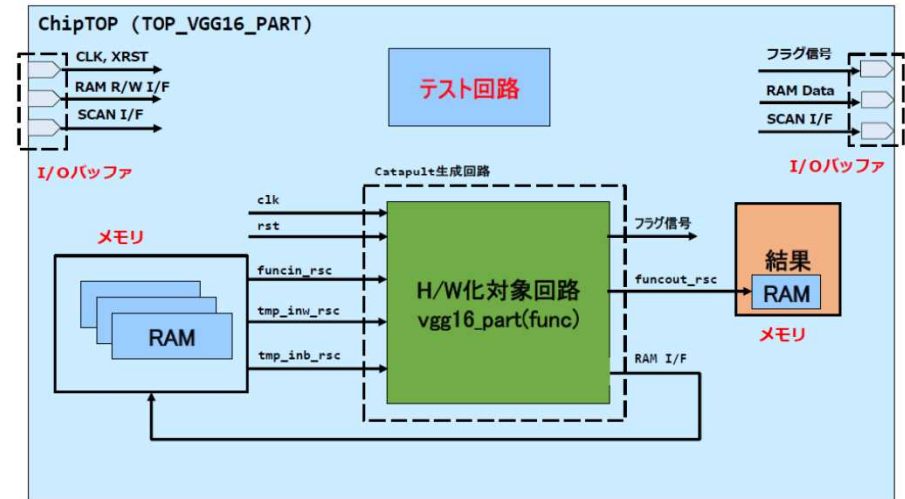
- 344 -

2. 成果: 「デジタル物理設計: 応用編」教材の抜粋

ブロック設計ではシステムやメモリに接続するための整合性が取れたI/Fの実装ができればブロックとしての設計を完結することができました。チップ設計ではクロックやリセット、メモリ、テスト回路、I/Oバッファといったシステム全体で必要とされる要素を全て実装する必要があります。

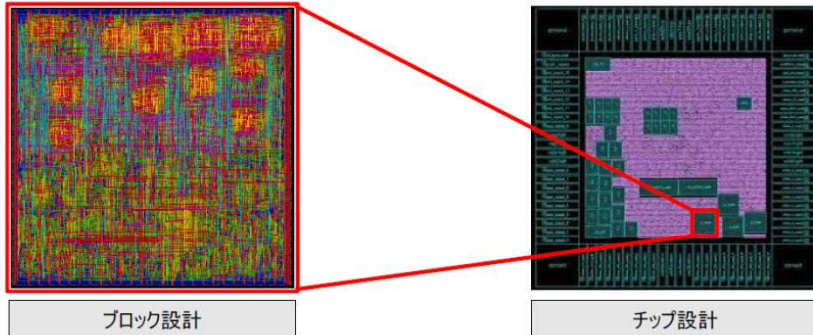
デジタル設計教材

デジタルチップ・物理設計【応用編】



1章 デジタルチップ設計概要

1.2. ブロック設計とチップ設計の違い



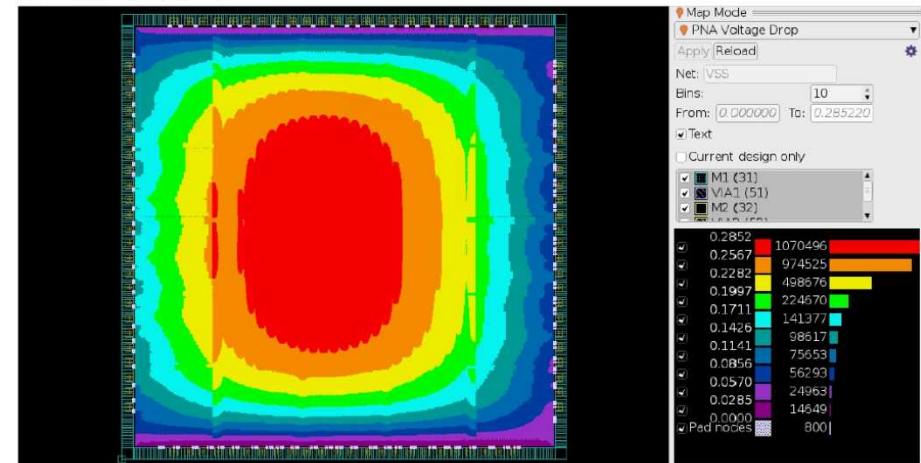
ブロック設計とチップ設計の違いは以下の通りです。

- IO配置、PAD配置
- ハードマクロ配置
- チップレベルの配置・CTS・配線

8章 チップのPower検証

8.4.1. IR-Drop結果の確認

【scan】【IR-Drop(VSS)】



2. 成果: 「デジタル物理設計: 低消費電力編」教材の抜粋

デジタル設計教材

デジタルチップ・物理設計【低消費電力編】

2022年 3月7日
 凸版印刷株式会社
 エレクトロニクス事業本部
 株式会社トッパン・テクニカル・デザインセンター

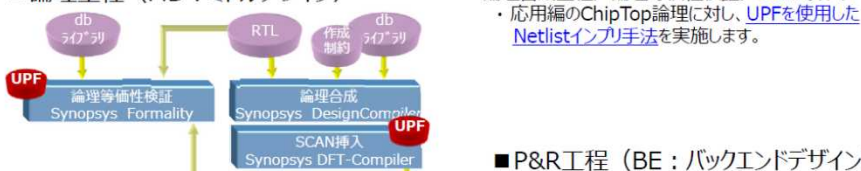
CONFIDENTIAL

- 1 -

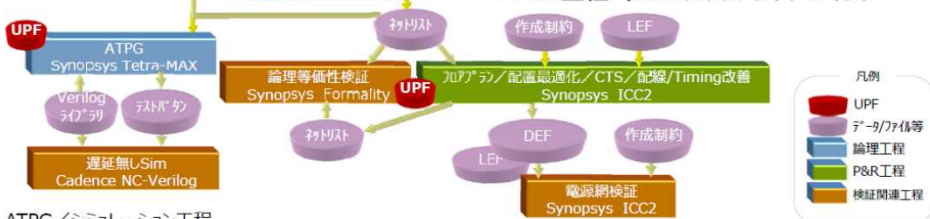
1章 電源遮断について

1.2. 電源遮断設計のフロー

■ 論理工程 (MD : ミドルデザイン)



■ P&R工程 (BE : バックエンドデザイン)



ATPG/シミュレーション工程
 1・全電源ドメインがON状態での動作確認を行います。
 2・一部電源ドメインがOFF状態になった場合、期待値が不一致になることを確認します。

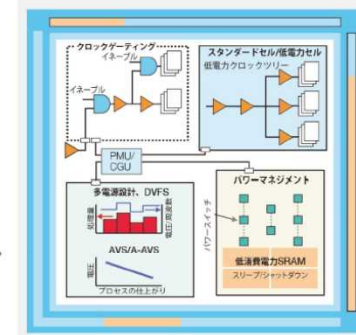
P&Rレイアウト工程:
 低消費電力編では、UPFを使用してアイソレーションセルによる電源遮断を主体に教材と環境を構築します。電源遮断に必要な要件について、フロアプラン~P&Rレイアウトに至るまで対応いたします。

CONFIDENTIAL

- 13 -

【多電源設計/パワーマネジメント】

- 多電源設計
 LSI内に異なる電圧を供給する設計のことです。高速動作の回路ブロックには高電圧、低速動作の回路ブロックには低電圧を供給し、動作時の消費電力を抑えます。
- パワーマネジメント
 パワーゲーティングとSRAMのスリープ、シャットダウンモードを統合的に制御する手法です。電源オン/オフ切り替え時に発生する電源ノイズを抑制してLSIの誤動作を回避します。



出典: 「低消費電力化技術」株式会社ソシオネクスト
<https://www.socionext.com/jp/products/customsoc/design/low-power.html>

本編では、UPFを採用することによって、電源遮断回路を含めた、物理設計・検証が一括してできるようになります。UPFの詳細は【1.3. UPFについて】を確認してください。

CONFIDENTIAL

- 6 -

1章 電源遮断について

1.3. UPFについて

【パワードメインの定義】

電源の有効範囲を定義する「パワードメイン」を定義、使用する電源/グラウンドを指定します。トップ階層のパワードメイン (PD_TOP) を定義します。供給電源はSS_VDDH、その他の電源はSS_VDDLを定義します。

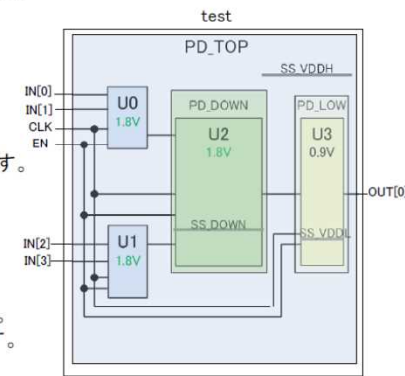
```
create_power_domain PD_TOP ¥
-supply {primary SS_VDDH} ¥
-supply {extra_supplies_0 SS_VDDL} ¥
-include_scope
```

電源遮断領域のパワードメイン (PD_DOWN) を定義します。供給電源はSS_DOWN、その他の電源はSS_VDDLを定義します。

```
create_power_domain PD_DOWN ¥
-supply {primary SS_DOWN} ¥
-supply {extra_supplies_0 SS_VDDH} ¥
-elements {U2}
```

電源電圧が異なるパワードメイン (PD_LOW) を定義します。供給電源はSS_LOW、その他の電源はSS_VDDHを定義します。

```
create_power_domain PD_LOW ¥
-supply {primary SS_VDDL} ¥
-supply {extra_supplies_0 SS_VDDH} ¥
-elements {U3}
```



赤字: ユーザ指定
 青字: コマンド特有の定義済み設定

CONFIDENTIAL

- 32 -

2. 1-1成果(目標達成度):2020年度終了時

項目	成果	達成度	
		水準	内容
拠点利用者が、小・中規模の回路に対し、拠点を構築したRTL記述を用いたエミュレータ・シミュレータ協調設計検証フローで設計検証を実施(5件以上)	・RTLのエミュレータを用いた検証を実施した	○	小・中規模の回路に対して前年度累計で5件、RTL記述を用いたエミュレータ・シミュレータ協調設計検証フロー(手法)で実証 より大規模な回路設計に対応するためエミュレータ・シミュレータ協調設計検証のテストベンチ作成の半自動化を試行
拠点利用者が、小・中規模の回路に対し、拠点を構築したネットリストレベルの記述を用いたエミュレータ・シミュレータ協調設計検証フローで設計検証を実施(3件程度)	・ネットリストのエミュレータを用いた検証を実施した	○	小・中規模の回路RTLを用いたエミュレータ・シミュレータ協調設計検証フローの利用実績5件のうち3件に関してネットリストレベルでの検証を行った
エミュレータ・シミュレータ協調設計検証フローを用いることによる設計検証効率の向上を、エミュレータのみを用いた設計検証フローと比較することで(検証時間・検証カバレッジ等を用いて)可視化するとともに結果を電子的に公開する	・検証時間等に関して検証フローの効果を可視化した	○	ネットリストレベル記述を用いたエミュレータ・シミュレータ協調設計検証フローを構築・確立し、電子的にマニュアルを作成、公開 エミュレータ・シミュレータ協調設計検証フローを用いることによる設計検証効率の向上をエミュレータのみを用いた設計検証フローと比較することで効果の可視化

◎ 大きく上回って達成、○達成、△達成見込み(中間)、×未達

2. 1-1成果(目標達成度):2022年度終了時

項目	成果	達成度	
		水準	内容
拠点利用者が、より大規模(>1BG)な回路に対し、エミュレータ・シミュレータ協調設計検証フローで設計検証を実施(3件程度)	・大規模設計データに対してエミュレータ・シミュレータ協調設計検証フローを用いた検証を実施した	○	AI-One/AI-Twoおよび協力会社の3件の1BG程度・超の設計のエミュレーション検証環境を整備し検証の実行
拠点で構築した高速チップ・超低消費電力チップ・高電力効率チップ等のアプリケーション向けのエミュレータ・シミュレータ協調設計検証フローで、拠点利用者が設計検証を実施(事業期間全体で10件程度)	・種々のアプリケーション向けの検証フローでの検証を実施	○	中間までの検証実績(RTL:5件、ネットリスト:3件)に加え、AI-One/Twoおよび協力会社の検証、RISC-Vプラットフォームの検証:2件(1件はハンズオンセミナーにも使用)を加え、13件の検証を実行

◎ 大きく上回って達成、○達成、△達成見込み(中間)、×未達

2. 1-2成果・進捗状況(中間時点:2020年度の成果)

◎大幅達成、○達成、△一部未達、X未達

項目	成果	達成度	
		水準	内容
モデルベースのエミュレータ・EDAツールの協調設計ツールチェーンを開発し、小・中規模の実設計において実証する。3件以上がこのモデル設計フローを活用した設計を行う	ツールチェーンを開発 小・中規模の実設計において実証	○	モデルベースのエミュレータ・EDA協調設計ツールチェーンを開発し、実証、公開 実施項目①-1において実現されたエミュレータ・シミュレータ協調設計フロー及び、本実施項目の成果によるツール間の接続を活用し、RTL-エミュレータ論理設計から物理設計にいたる最適設計フローを実現 拠点利用者の設計を例題とし、フローの有効性を実証 モデルベース設計ツールチェーンを活用した設計を2020年度累計で3件実施

2. 1-2成果・進捗状況(2022年度終了時)

◎大幅達成、○達成、△一部未達、×未達

項目	成果	達成度	
		水準	内容
アプリケーション向けのツールチェーンを策定し、拠点利用者が事業期間全体で5件程度、より大規模(>1BG)な設計を実施する	AI-One: 6 AI-IPおよびSoC全体、AI-Two; 3 AI-IPおよびSoC全体の設計を実施し、実チップが設計通り機能・性能を出すことを確認した	◎	全体として9件のAI-IPに対してツールチェーンを活用した設計を実施した。また、AI-One/AI-TwoのSoC全体の設計(おおむね1BG程度の規模)に対して設計フローを適用して実チップにて動作を確認した

AIチップ開発加速のためのイノベーション推進事業

実施項目 2 - 1 「AIチップの研究開発に必要なEDAツールの整備」

実施項目 2 - 2 「人材育成と拠点機能の整備」

担当機関 東京大学
産業技術総合研究所
発表者 池田誠（東京大学）

1. 概要

- 背景・目的・課題
- 取り組み内容・技術の特徴
- 目標・計画

2. 成果

- 成果・意義
- 目標達成度 など

背景・課題

- AIチップ開発において、その性能を高精度に予測し設計するための設計ツール群は必要不可欠
 - 設計ツール群は非常に高価 → AIチップの開発には莫大な費用が必要で、大きな障壁
- 標準的な高速インターフェースやメモリアンターフェース等のIPが不可欠
- 大規模なAIチップの設計のためにはエミュレータの適切な活用が不可欠
 - 管理運用及び利活用の方法が大きな課題
 - 利用者である中小・ベンチャー企業によっては、手元で設計ツールの管理や設定が可能な企業と、そうではない企業がありうる

目的

- アナログ系、デジタル系論理設計・物理設計のための設計ツール群、高位合成のためのツール群及びサインオフ検証向けツール群を整備
- 拠点を利用する中小・ベンチャー企業に広く活用されるような環境を整備
- 国内外のFABの代表的なプロセステクノロジー向けの設計環境を整備
 - AIチップ開発に新たに取り組むことになる中小・ベンチャー企業の設計環境整備を支援

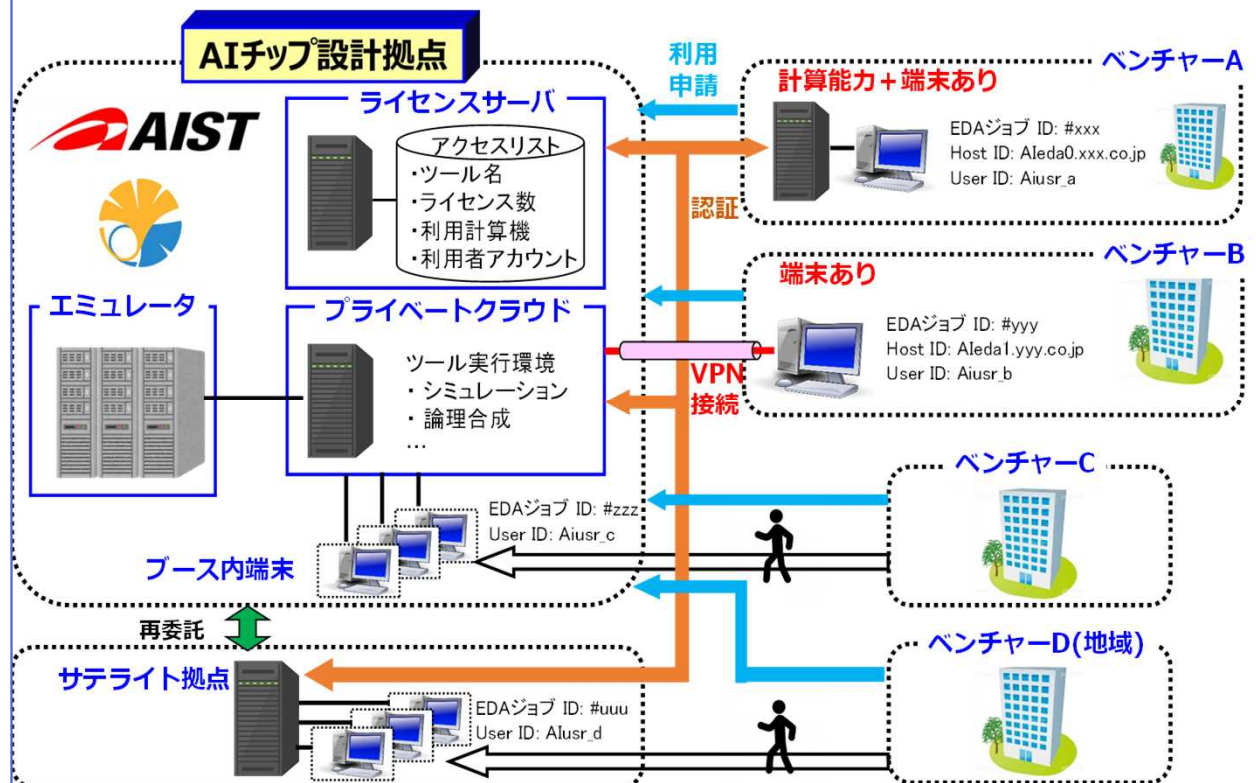
取り組み内容

- アナログ系、デジタル系論理設計・物理設計のための設計ツール群、高位合成のためのツール群及びサインオフ検証向けツール群を整備
- 拠点を利用する中小・ベンチャー企業に広く活用されるような環境を整備

技術の特徴

- ベンチャーの環境に応じ複数の利用形態を整備する

- A)ベンチャーで十分な計算能力、ツールの整備能力のある場合にはライセンスのリモート認証による利用
- B)ベンチャーの端末から拠点プライベートクラウドに接続しツールを利用
- C)拠点のブースでのツールの利用
- D)福岡ISTのブースでのツールの利用

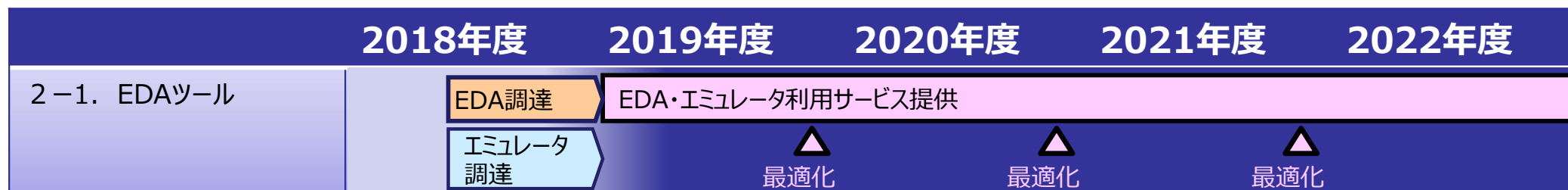


中間(2020年度末時点)目標

- 利用動向の監視と利用状況の可視化
- ツール利用環境の改良整備、HPの機能追加、拠点運用管理体制の強化
- セキュリティを確保した上で、拠点の規模及び能力の拡充
- 四半期ごとのEDAツールライセンス数を最適化
- ツール群安定最新バージョンへのアップデート及びアップデート後の動作確認

最終(2022年度末時点)の目標

- 利用状況の可視化により四半期ごとのEDAツールライセンス数を最適化
- 拠点運用管理体制の強化
- セキュリティ対策の向上、拠点の規模及び能力の拡充
- ツール群安定最新バージョンへのアップデート及びアップデート後の動作確認



背景・課題

- AIチップ開発拠点の機能として組み込む共通基盤技術＝ツールや設計フロー、ツールチェーンやリファレンスデザイン等の準備だけではエミュレータやEDAツール群を使いこなしてAIチップの開発ができるわけではない。
- AIチップ開発初心者にとってAIチップ製造のFABの選定、守秘等の契約、設計のためのデザインマニュアル・PDK・ライブラリの導入は非常に時間がかかり容易ではない。
- これらの機能が最大限活用されるようにするためには、それら利用方法の教育等の人材育成が不可欠 & 教育する側の人材育成も重要。

目的

- 拠点環境の整備人材育成・セミナーなど教育活動を通して
 - 拠点利用者への利便性向上
 - 拠点利用者を増やすための設計人材、AIチップ設計人材増加を目指す。

取り組み内容

- 拠点に整備する機能の利用方法についてマニュアルの整備
- AIチップ設計・検証ガイドラインの策定
- AIチップ設計の教育カリキュラムを構築
- 中小・ベンチャー企業のAIチップ開発に係る人材育成を実施
- 代表的なFABに対する窓口を整備し、試作に向けた支援を実施
- 勉強会を開催し、中小・ベンチャー企業等の意見を集約し、マーケティングやコンソーシアム活動を通して拠点の自立運営を検討

技術の特徴

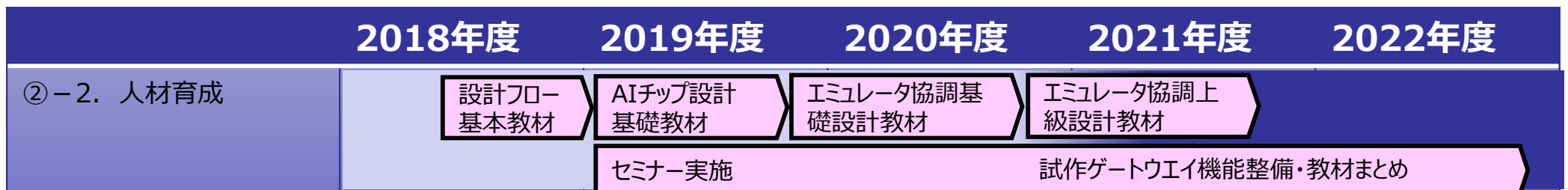
- AIチップ実現のためのワンストップ窓口となること

中間(2020年度末時点)目標

- エミュレータ・シミュレータ協調設計フローに即したカリキュラム(初級編)を構築、教材を電子的に整備し、HPにて公開
- 年間1回以上の教育コースを拠点利用者等に対して実施
- 1-1「ネットリスト検証フロー利用目標」のうち1件程度が本設計試作ゲートウェイ機能を利用
- 拠点運営について運営形態等検討会を立ち上げる

最終(2022年度末時点)の目標

- 1-1「協調設計検証フロー利用目標」のうち3件程度が本設計試作ゲートウェイ機能を利用
- 教育カリキュラムに則ったセミナー等を拠点利用者等に対して実施
- 設計ノウハウのFAQを更新し、拠点利用者に公開しノウハウの共有を図る
- 拠点のユーザーとなり得る中小・ベンチャー企業等との勉強会を実施し、拠点運営に係る意見を集約



背景・課題

- エッジ側におけるAI機能とセンシング機能の融合が進行
- AI処理において画像処理応用範囲が広く、イメージセンサとの融合が期待
- イメージセンサ技術は産業的に日本が強い分野であり、そのセンサーを活用した設計に向けた、物理モデルに基づいたデバイス設計環境の必要性
- イメージセンサを組み込んだAIチップを設計する際、TCAD等のデバイス設計ツールに精通していない設計者でも設計できる環境整備が不可欠
- 現状機能デバイスとAIチップの協調シミュレーション環境・モデルが存在しない

目的

- Impulse TCADをベースとしてAIチップに搭載する際に利用可能な独自の機能素子の物理設計環境実現に向けた課題を抽出しと改良仕様を策定
- 機能デバイス設計ツールのクラウド上での利用環境の構築
- 特定の用途向けユースケースに対して有効性を検証
- 機能デバイス設計環境の拠点利用者等の利用を通じた有効性の実証とAIチップ設計フローへ組み込むために必要な条件等の明確化

中間(2020年度末時点)目標

- Impulse TCADをベースとしてAIチップに搭載する際に利用可能な独自の機能素子の物理設計環境のプロトタイプを、例えばフォトセンサーといった特定の用途を想定して開発・評価し、課題を抽出し、改良仕様を策定する

最終(2022年度末時点)の目標

- 機能デバイス設計ツールを改良し、クラウド上で利用可能な環境を構築、特定の用途を想定したケースに対して有効性を検証する。その一環として機能デバイス設計ツールの拠点利用者等の利用を通してその有効性を確認する。また、検討した新規デバイスのモデル化技術等については、AIチップ設計フローへ組み込みとそれに必要な条件等の明確化を図る。

	2018	2019	2020	2021	2022
ユーザインタフェース開発	調査、仕様策定	プロトタイプ開発	評価・改良仕様	UI改良開発	物理設計環境整備
シミュレーションデータ開発		CISピクセル	CIS改良、圧力センサ	他デバイスへの展開	
モデル化技術				モデル化技術検討	設計フロー検討

成果

- 導入したEDAツールの利用マニュアルを公開
- EDAツール利用のための環境設定ファイルをツール毎に準備
- ライセンスの効率的な運用、設計データやIPの保護のために、VM(バーチャル マシン)にログインし、LSF(ライセンス・スケジューラ)を用いたツールの実行環境を整備
- ライセンスの利用実績、利用見込みから最適なライセンス数を導入し安定した運用を実現
- AI-One/AI-Twoといった大規模なSoCの設計にも対応可能なツール利用環境を実証した
- EDAツール利用が進み、利用件数は延べ74件となった。

意義

- ユーザー間の情報漏洩の防止、設計情報の漏洩の防止、拠点の情報の漏洩防止を図りつつ、EDAツール利用の利便性を最大限担保できるシステムの構築
- 企業ユーザーのにとってより安心して活用できる環境かつ、多くのユーザーの受け入れを実現

2. 成果: 拠点利用のための申請フロー

拠点利用のための申請フロー



ユーザID申請

全ての利用者は、申請します。
ユーザIDの共有は出来ません。

1.1 ユーザID申請フォーム記入

Emailとパスワード

1.2 ユーザID申請受付メール受領

2～5営業日 → **実効 1営業日未満**

1.3 ユーザID承認メール受領

1.4 申請したEmailとパスワードでWebサイトにログインします。

プロジェクトID申請

プロジェクトの代表者が、
プロジェクトID申請します。

2.1 プロジェクトID申請フォーム記入

装置の利用目的

プロジェクト参加者情報

2.2 プロジェクトID申請受付メール受領

2～5営業日 → **実効 1営業日未満**

2.2 EDA利用申請書をメール受領

利用するEDA機能名称を指定し返信

2～5営業日 → **実効 5～10営業日**

2.3 EDAベンダ契約書を受領。

EDAベンダ契約書に署名し返信

2～5営業日 → **実効 1～10営業日**

2.5 プロジェクトID発行メール受領

2.9 プロジェクトIDをプロジェクト参加者に通知し、参加者は装置ID申請を行う

装置ID申請

全てのEDA利用者は、申請します。
プロジェクトIDは、代表者から受領します。

3.1 装置ID申請フォーム記入

プロジェクト代表者より受領した
プロジェクトIDを記入

3.2 装置ID申請受付メール受領

2～5営業日 → **実効 1～2営業日**

3.3 装置ID発行メール受領

Username, Loginサーバ名, queue name

2～5営業日 → **実効 2営業日**

3.4 郵送レターパック受領

Private keyを格納したUSB、借用書

OTPを表示するセキュリティトークン

3.5 OTP借用書に署名+USBを返送

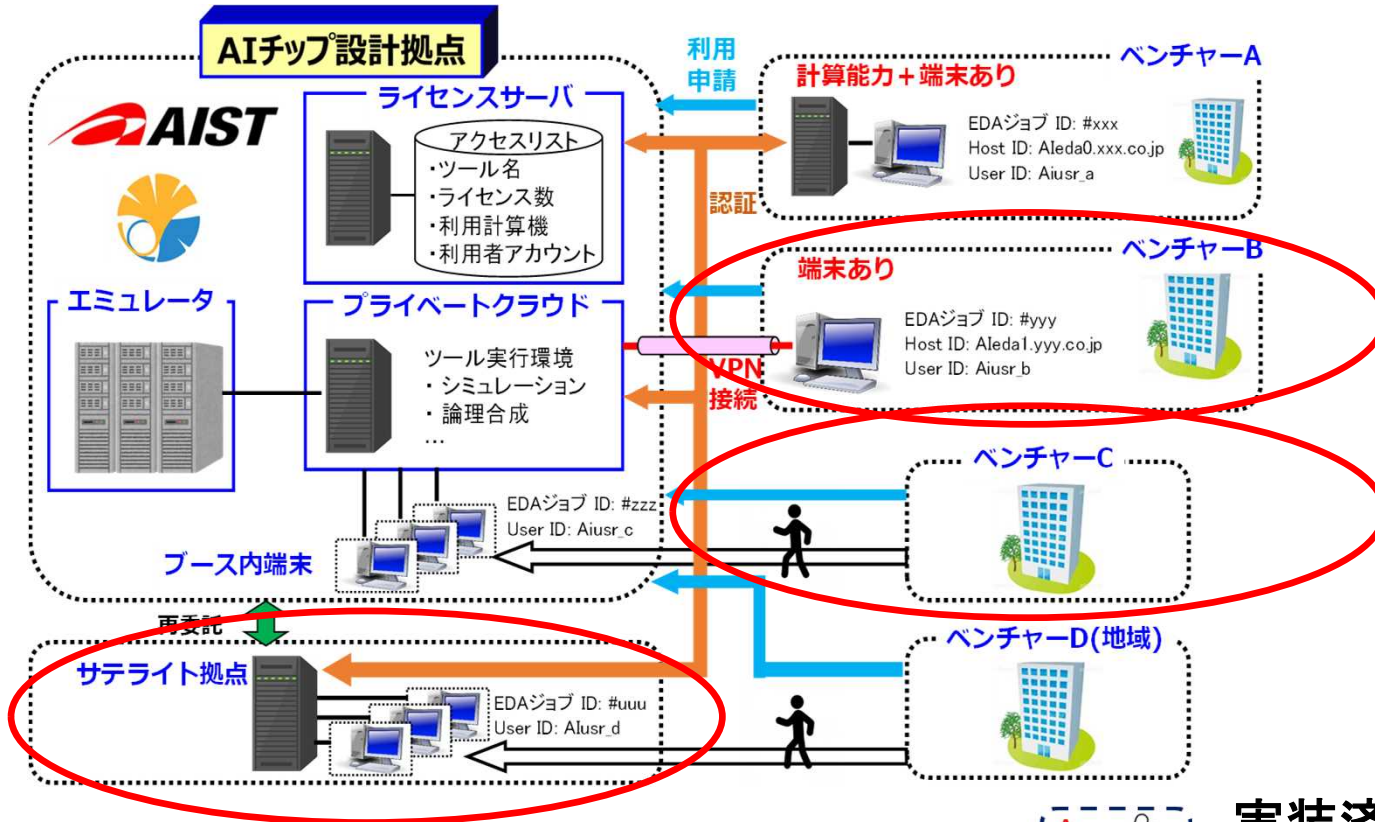
3.9 インストールと設定

ユーザID申請

プロジェクトID申請

装置ID申請

2. 成果:現状システム



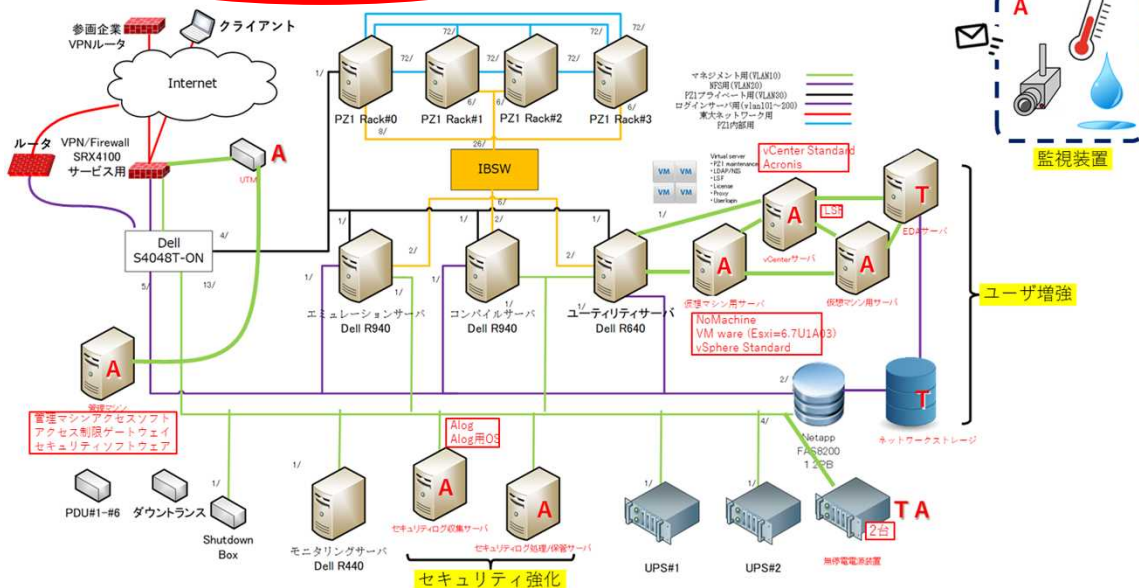
発足当初の計算リソース

- ・コア数: 96
- ・CPU合計性能: 9.2TFlops
- ・主記憶合計: 9.7TB



現時点の計算リソース

- ・コア数: >600
- ・CPU合計性能: > 70 TFlops
- ・主記憶合計: >25 TB



実装済みのシステム運用

- ・ 設計技術・契約に即した申請システム
- ・ NoMachineベースのセキュアなアクセス
- ・ Webインターフェースによるセキュアなファイル転送
- ・ LSFベースのEDAツール利用管理
- ・ 設計ロードの応じた、システムリソース管理

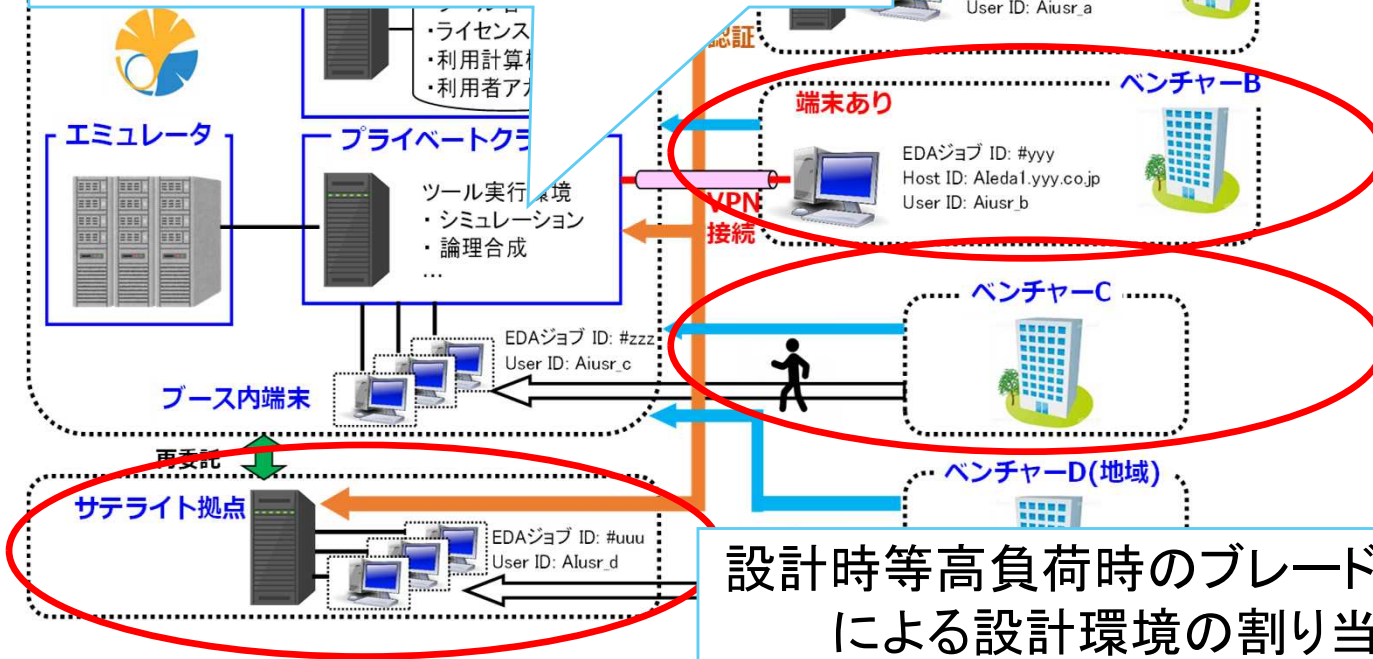
- NoMachineベースのセキュアなアクセス
 - EDAツールのバイナリ実体、IP等の設計ファイルの実体を参照することなく設計の実行が可能
 - 他ユーザとのデータの覗き見対策
- Webインターフェースによるセキュアなファイル転送
 - 利用者に自由にデータのUpload/Downloadをさせないことで、ウイルスの混入リスクの軽減、EDAバイナリやIP情報等のファイルの不正なダウンロード防止
 - ファイルをルールベースでの自動チェック、必要に応じて担当者の目視によるチェックによる不正防止
 - 問題ないファイルに関しては、半自動でのファイル転送許可
- LSFベースのEDAツール利用管理
- 設計ロードの応じた、システムリソース管理
 - AI-One/Two設計等負荷の高い設計においては、Bladeマシン等を直接利用することによる設計効率の向上と、他ユーザとの隔離を両立
- NASが2台 (NetApp/EMC)構成になったことで、IPやテクノロジ情報のNAS単位での分離によるさらなるセキュリティ向上
- 40nm/28nm/12nmのPDK、IPの設計環境を実現する、利便性と安全性を十分に評価実証したシステムとして運用
- IPをブラックボックス化してシミュレーション・エミュレーションし結果を返す検証環境の実現

2. 成果: 現状システム

EDAツール(毎)の実行権管理

AIチップ設計拠点

ライブラリ・PDKへのアクセス件管理



発足当初の計算リソース

- ・コア数: 96
- ・CPU合計性能: 9.2TFlops
- ・主記憶合計: 9.7TB

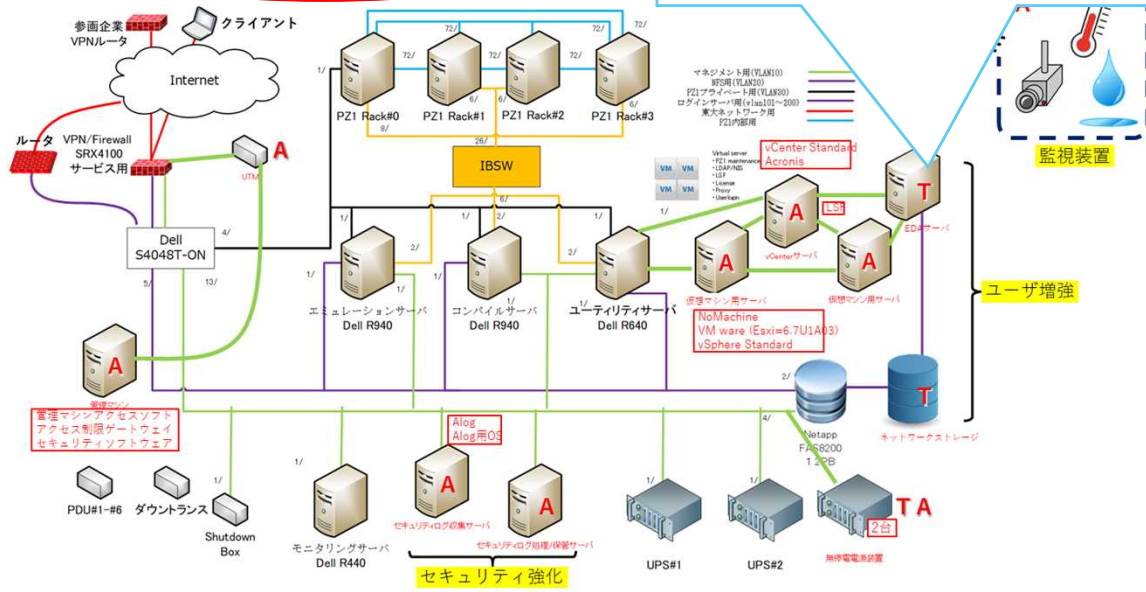


現時点の計算リソース

- ・コア数: >600
- ・CPU合計性能: > 70 TFlops
- ・主記憶合計: >25 TB

大規模なシステム運用

- ・設計技術・契約に即した申請システム
- ・NoMachineベースのセキュアなアクセス
- ・Webインターフェースによるセキュアなファイル転送
- ・LSFベースのEDAツール利用管理
- ・設計ロードの応じた、システムリソース管理



2. 成果: EADツールの整備

Cadence, Synopsys, Siemensのツールを整備

アーキテクチャ検証、高位合成、論理設計/検証、回路設計、物理設計/検証、論理工ミュレータ、FPGAプロトタイプング、ボード設計、etc.

#Vendor	Feature	大項目	用途・カテゴリ	
cadence	Stratus_HLS_-.XL	アーキテクチャ検証 高位合成システムツール群	高位合成 (High-level Synthesis)	
cadence	Stratus_Floating_Point			
mentor	Catapult-Prime_Ap_SW	アーキテクチャ検証 高位合成システムツール群	高位合成	
mentor	Oasys-RTL_Designer_Ap_SW			
mentor	Oasys-RTL_Floorplanning_AddOn_SW			
mentor	Oasys-RTL_Architect_AddOn_SW			
mentor	Oasys-RTL_Low_Power_AddOn_SW			
mentor	Catapult_Coverage_Ap_SW			
mentor	SLEC-HLS_Ap_SW		高位検証	
mentor	PowerPro-Optimizer_Ap_SW		消費電力の推定・解析	
synopsys	DC_Explorer		論理合成ツール (LogicSynthesis)	
synopsys	DC_Ultra			
synopsys	HDL_Compiler_Verilog			
synopsys	VHDL_Compiler			
synopsys	Power_Compiler			
synopsys	DFT_Compiler			
synopsys	Design_Vision	論理合成結果の可視化・最適化ツール (GUI & Advanced Optimization)		
synopsys	Design_Compiler_Graphical			
synopsys	Library_Compiler	論理合成向け基本関数ライブラリ (Datapath library for Logic Synthesis)		
synopsys	DesignWare_Library			
synopsys	DW_minPower_Components	配置配線ツール (P&R)		
synopsys	IC_Compiler_II_AG_8-core			
synopsys	Custom_Compiler_ADV			
synopsys	IC-Compiler-II-AF-Add-on			
synopsys	StarRC_Ultra			
synopsys	Custom_Compiler_ADV			
synopsys	IC_Validator_NXT		寄生成分抽出ツール (Extraction)	
synopsys	IC_WorkBench_Edit/View_Plus			
synopsys	PrimeTime_ADV		タイミング検証・電力解析ツール (Timing & Power Analysis)	
synopsys	Prime_Power			
synopsys	Formality	等価性検証ツール (Equivalence checking)		
synopsys	PowerReplay			
synopsys	Verdi-3			
synopsys	Verdi_-_Power_Aware_Debug			
synopsys	nAnalyzer		論理デバッグツール (Debug)	
synopsys	nECO			
synopsys	Verdi_HW_SW_Debug_Add-on			
synopsys	Verdi_Advanced_AMS_Debug_Add-on			
synopsys	Verdi_Performance_Analyzer			
synopsys	SpyGlass_Lint		RTL解析ツール (RTL analysis)	
synopsys	SpyGlass_Lint_Turbo_Add-on			
synopsys	SpyGlass_CDC_Advanced_Add-on			
synopsys	SpyGlass_RDC_Add-on			
synopsys	SpyGlass_DFT_ADV_Add-on			
synopsys	SpyGlass_SVA_Generation_Add-on			
synopsys	VC_LP			
synopsys	SpyGlass_Power_Explorer_Add-on			
synopsys	VC_Formal	形式的機能検証 (Formal verification)		
synopsys	VC_Formal_FSV_Add-on			
synopsys	VC_Formal_FTA_Add-on			
synopsys	Certitude_Base_w_Unlimited_Sim	検証品質確認ツール (Functional Qualification)		
synopsys	DFTMAX			
synopsys	DFTMAX_Ultra_Add-on			
synopsys	DC_Ultra			
synopsys	Design_Vision			
synopsys	HDL_Compiler_Verilog		テスト設計検証ツール (Design for test)	
synopsys	TetraMAX_II_ADV_8-core_ATPG			
synopsys	HSPICE		回路レベルシミュレーションツール (Circuit simulation)	
synopsys	Custom_WaveView_ADV			
synopsys	CustomSim_with_FineSim2			
synopsys	FastSPICE_Analysis_Suite			
synopsys	Custom_Compiler_ADV			
synopsys	VCS_MX			
synopsys	TestMax_CustomFault			
synopsys	X-Prop_Add-on			
synopsys	VCS_MX			
synopsys	NLP_Add-On	論理シミュレーションツール (Logic simulation)		

cadence	Genus_Synthesis_Solution	論理設計検証ツール群	論理合成 (Logic Synthesis)	
cadence	Genus_Low_Power_Option			
cadence	Genus_Physical_Option			
cadence	Isosys_RTL_Power_Solution		電力解析	
cadence	Conformal_Low_Power_-.XL		寄性検証	
cadence	Innovus_Implementation_System		配置配線 (Place&Route)	
cadence	Innovus_20/15/14nm_Option			
cadence	Innovus_Hierarchical_Design_Option			
cadence	Tempus_Timing_Signoff_Solution		タイミング検証 (Timing Analysis)	
cadence	Tempus_Timing_Signoff_Solution_ECO			
cadence	Xcelium_Single_Core	論理検証 (Logic Verification)		
cadence	Xcelium_Digital_Mixed_Signal_Option			
cadence	vManager_Integration_Server			
cadence	vManager_Multi-Project_Application			
cadence	vManager_Linux_Client			
cadence	JasperGold_Formal_Verification_Platform		フォーマル検証	
cadence	Virtuoso_Schematic_Editor_L		アナログ・ミックスドシグナル設計	
cadence	Virtuoso_Schematic_Editor_XL			回路図入力 (Schematic Editor)
cadence	Virtuoso_ADE_Assembler			
cadence	Virtuoso_Layout_Suite_L			レイアウト入力 (Layout Editor)
cadence	Virtuoso_Layout_Suite_XL			
cadence	spectre_Multi-Mode_Simulation_with_AHS	シミュレーション		
synopsys	Custom_Compiler_ADV	回路図入力 (Schematic Editor)		
synopsys	Custom_Compiler_ADV	レイアウト入力 (Layout Editor)		
synopsys	Galaxy_Custom_Router			
synopsys	HSPICE			
synopsys	Custom_WaveView_ADV	アナログ・ミックスドシグナル設計	アナログシミュレーション環境 (Analog Simulation)	
synopsys	CustomSim_with_FineSim2			
synopsys	FastSPICE_Analysis_Suite			
synopsys	TestMax_CustomFault			
synopsys	Custom_Compiler_ADV			
synopsys	StarRC_Ultra		設計機能検査・寄生成分抽出ツール (DRC & Extraction)	
synopsys	IC_Validator_NXT			
synopsys	Custom_Compiler_ADV			
synopsys	FastSPICE_Analysis_Suite			
synopsys	CustomSim_with_FineSim2		電力解析ツール (Power Analysis)	
mentor	Calibre_nmDRC_Ap_SW	物理検証ツール群	物理検証 DRC LVS (Physical Verification)	
mentor	Calibre_nmDRC-H_Op_SW			
mentor	Calibre_nmlVS_Ap_SW			
mentor	Calibre_nmlVS-H_Op_SW			
mentor	Calibre_xACT_3D_Ap_SW			寄生成分抽出
mentor	Calibre_RVE/QQB-H_Ap_SW			
mentor	Calibre_Interactive_Ap_SW			
mentor	Calibre_DeSigner_Ap_SW			対話型実行環境 結果表示
cadence	Physical_Verification_System_DRC			
cadence	Physical_Verification_System_Advanced_Analysis_Option_for_PVS_DRC			
cadence	Physical_Verification_System_LVS	物理検証ツール群	物理検証 DRC LVS (Physical Verification)	
cadence	Physical_Verification_System_Advanced_Device_Option			
cadence	Quantus_QRC_Extraction		寄生成分抽出 (Parasitic Extraction)	
cadence	Quantus_QRC_Advanced_Analysis			
cadence	Quantus_QRC_Advanced_Modeling			
cadence	Quantus_QRC_Advanced_Node_Modeling			
cadence	VoltaX_IC_Power_Integrity_Solution			
cadence	VoltaX_IC_Power_Integrity_Solution_Advanced_Analysis_GXL_Option			
cadence	VoltaX_IC_Custom_Power_Integrity_Solution			
cadence	Palladium_Z1_GXL_576_always-on-domains			
cadence	Palladium_Z1_56Gbps_36_ports_switch			
cadence	Palladium_Z1_2-port_host_card			
cadence	Palladium_Dynamic_Power_Analysis_2.0	エミュレータ		
cadence	Virtual_Debug_Interface_to_Software_Debuggers_4-pack			
cadence	Palladium_Series_Memory_model_Multiple_Subscription			
cadence	Palladium_PCE_4.0_VirtualBridge_Kit			
cadence	Palladium_VirtualBridge_8-port_pack			
cadence	Incise-Enterprise-Simulator-XXL			
cadence	Palladium_Accelerated_VIP_for_AMBA_AXI			
cadence	Palladium_Accelerated_VIP_for_AMBA_AHB			
cadence	Palladium_Accelerated_VIP_for_PCE_4.0			
cadence	Accelerated_VIP_for_AMBA_USB_3.1		エミュレータ用検証モデル	
cadence	Accelerated_VIP_for_HDMI_2.0	エミュレータ FPGAプロトタイプ		
cadence	Accelerated_VIP_for_MIP1_CSI_2			
cadence	Accelerated_VIP_for_MIP1_DSI_2			
cadence	Accelerated_VIP_for_CSI_X			
cadence	Protium-SI-Implementation-and-Debug-Software			
cadence	Protium_SI_FPGA-based_prototyping_system_8_UltraScale_VU140_FPGAs			
cadence	Protium_SI_accessory_2_chassis_cable_kit			
cadence	Protium_SI_Accessory_High_Performance_XDRAM_Card_16GB			
cadence	PCE_4.0_Emulation_Dev_Kit_for_Win10_on_Intel_platform			
cadence	PCE_4.0_Emulation_Dev_Kit_for_Win10_on_Intel_platform			
cadence	Protium_SI_Accessory_QCS_Adapter_for_PCE_4.0_SpeedBridge	FPGAプロトタイプ		
cadence	Protium_SI_Accessory_DDR1_direct_connect_8GB			
cadence	Protium / Protium_SI_Accessory_144MB_5RAM_300MHz			
cadence	Protium_SI_Accessory_204-pin_SODIMM_Adapter			
cadence	Protium_7_Protium_SI_Accessory_PCE_Kit			
cadence	Video_SB_for_HD_serial_interfaces_RHS			
cadence	Protium_SI_Accessory_VHDCI_Adapter_for_Video_SpeedBridge			
cadence	Allegro_PCB_Designer		BoardDesign	PCB board Design

2. 成果：EDAツールの整備・ライセンス数

フロー名	ツール名	オプション等	永久ライセンス		単価契約ライセンス												
			当初 1/31分	追加 4/16分	'19	'20				'21				'22			
						1	2	3	4	1	2	3	4	1	2	3	4
高位合成	Catapult	Catapult-Prime, Oasys-RTL, Catapult Coverage SLEC-HLS, PowerPro-Optimizer	0	0	1	4	3	3	1	2	2	2	2	1	1	1	1
論理合成	Design Compiler		2	4	4	2	4	2	0	1	1	0	0	0	0	0	0
配置配線	ICCompilerII		1	1	3	1	3	3	4	1	1	0	2	13	0	0	0
静的検証	PrimeTime/PrimePower		1	0	2	2	2	4	3	2	2	2	7	19	1	1	0
等価性検証	Formality		1	0	2	2	2	2	4	1	1	1	1	1	1	1	0
形式検証	DEBUG	Verdi SVTB, Verdi Power-Aware Debug, Verdi Coverage, Verdi Performance Analyzer, VC Formal, VC Verification IP Test suites, SpyGlass, PowerReplay, TetraMAX	0	0	1	7	5	2	1	3	3	2	2	3	3	1	2
レイアウト生成	Custom Compiler		2	0	2	1	1	0	0	1	1	0	0	0	0	0	0
寄生成分抽出	ICValidator/StarRC		1	0	4	2	2	3	3	2	2	1	1	2	2	2	0
回路シミュレーション	Analog Sim (VCS AMS, CustomSim, CustomSim Reliability, CustomFaultSim)		2	0	6	2	2	3	3	2	2	3	3	3	3	3	1
物理検証系	Calibre DRC, LVS, RVE, PEX, LFE		2	3	1	0	0	0	1	1	1	1	0	1	1	1	1

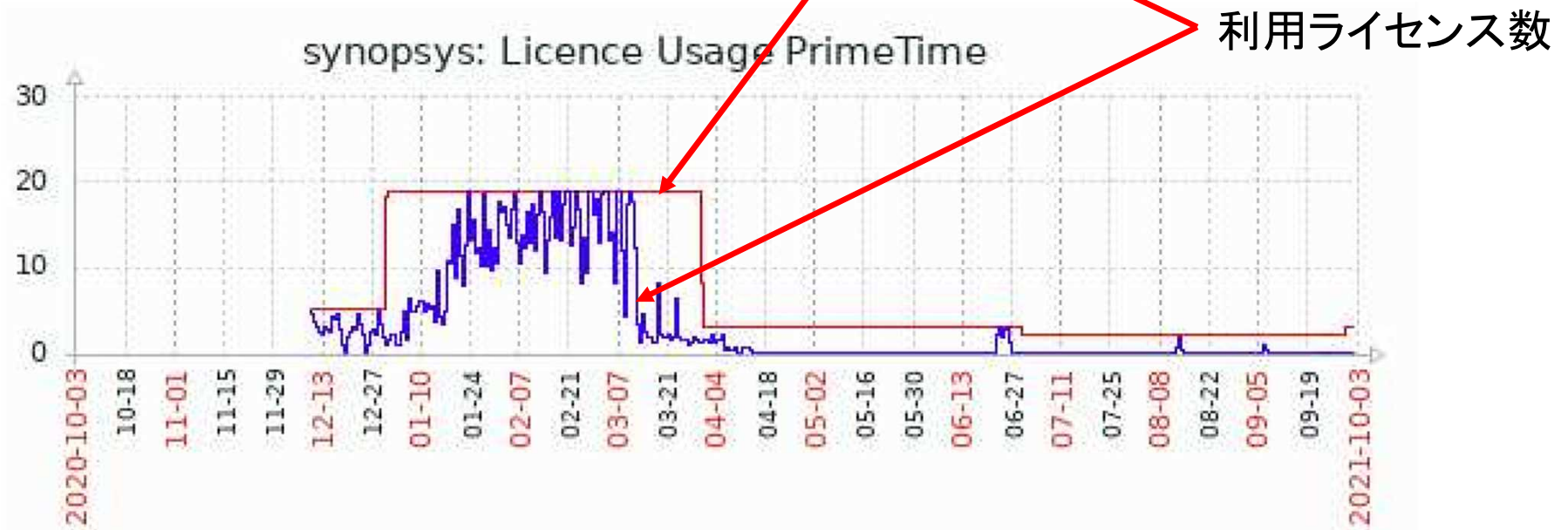
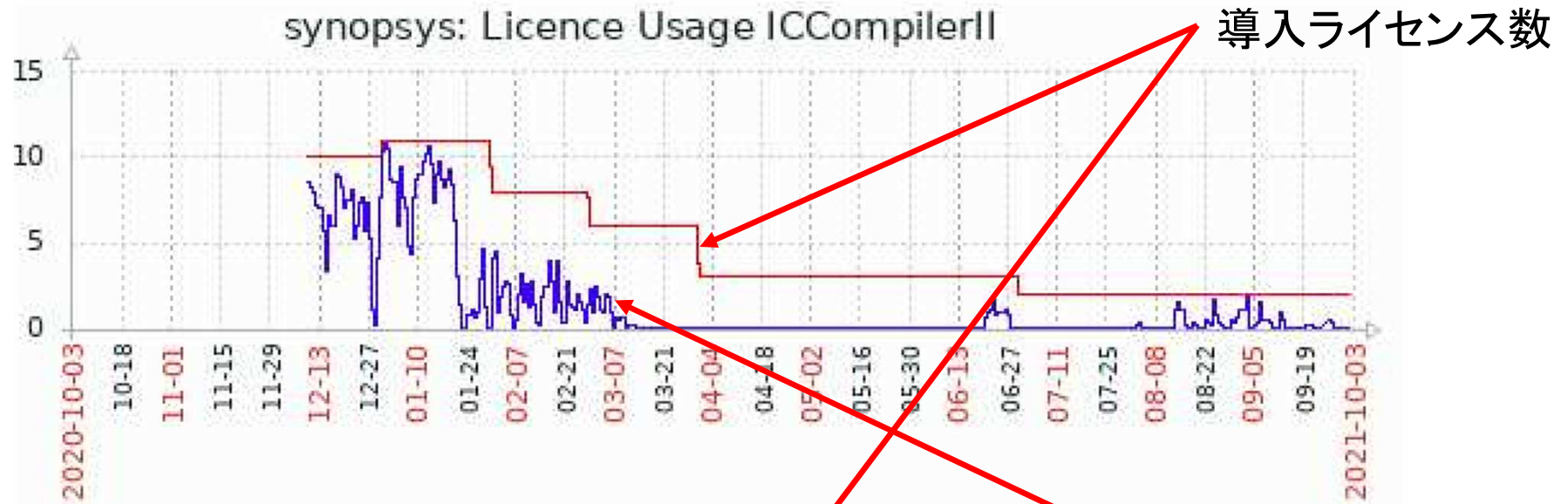
2020年度：当初単価契約以外のライセンス導入

Synopsys LogicBIST	2020/10/1-	1ライセンス
ICCompilerII	2020/12/1-12/31	5ライセンス
ICCompilerII	2020/12/1-12/31	5ライセンス
ICCompilerII	2020/1/1-1/31	5ライセンス
ICCompilerII	2020/2/1-2/28	2ライセンス
PrimeTime	2020/1/1-3/31	15ライセンス
Tweaker	2020/1/1-3/31	5ライセンス
CalibreDRC	2020/2/12-3/11	5ライセンス
Ansys Redhawk/PathFinder	2020/10/1-10/31	1ライセンス
Ansys Redhawk/PathFinder	2020/11/18-3/31	1ライセンス
Ansys Redhawk/PathFinder	2020/3/X-3/31	1ライセンス

2021年度：当初単価契約以外のライセンス導入

Synopsys LogicBIST	2021/4/1-	1ライセンス
Tweaker	2021/4/1-3/31	1ライセンス
Ansys Redhawk/PathFinder	2021/4/1-3/31	1ライセンス
Mentor CalibreMP	2022/1/1-3/31	2ライセンス
Mentor CalibrePERC	2022/1/1-3/31	1ライセンス
Mentor CalibreYE	2022/1/1-3/31	4ライセンス
Mentor CalibreADP	2022/1/1-3/31	1ライセンス
Synopsys ICCII	2022/2/1-2/28	3ライセンス
Synopsys ICCII	2022/3/1-3/31	5ライセンス
Synopsys PTADVP	2022/3/1-3/31	5ライセンス
Synopsys ICCII	2022/7/1-7/31	10ライセンス
Synopsys PTADVP	2022/6/1-6/30	10ライセンス
Synopsys PTADVP	2022/7/1-7/31	30ライセンス
+ Formality/Tweaker/LogicBIST/SHS/SMS	2022/4/1-	
Mentor CalibrePERC	2022/8/1-8/31	1ライセンス

2. 成果：EDAライセンス利用状況の可視化例



成果

- 拠点の整備を実現
- 人材育成の体系化に取り組んだ
- 一般向けに設計フォーラムを51回実施した(2023年9月時点)
- 「デジタル設計の基礎」、「ハードウェア・エミュレータでの論理検証の基礎【初級編】【応用編】」、「高位合成を使ったデジタル設計【基礎編】【続基礎編】【実践編】」、「デジタルブロック・物理設計編」を整備した
- 設計フローの詳細を扱うWebinarを整備しHPに公開した
- 教育セミナー・設計実践セミナーを14回実施した。
- 代表的なFABとして、TSMC、GFとNDAを締結しライブラリの供給を受け設計に供した

意義

- FABとのゲートウェイ機能の構築・強化することで、試作のハードルを軽減し、ハードウェア化の取り組みを推進。
- フォーラムやセミナーを介して様々な意見が伺え、人材育成強化の方向性変更や、より必要とされる人材育成の構築に反映、さらには、拠点アピールや利用促進につながる。

2. 成果: 拠点の構築

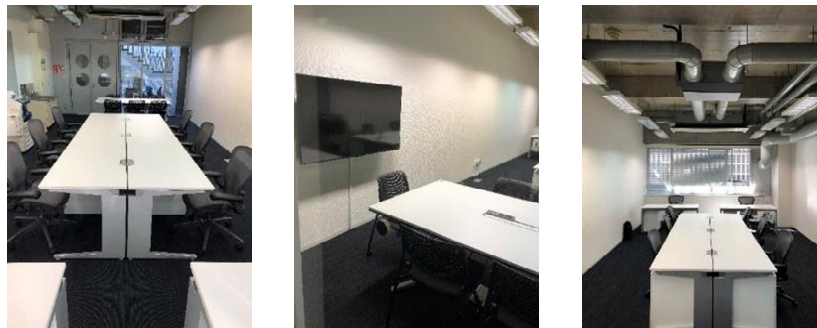
工学部3号館107号室(ブース)



サーバ室(クラウド環境、エミュレータ)



武田先端知ビル201 - 203号室



サテライト拠点(福岡IST)



2. 成果：成果設計人材育成の体系化

人材育成と拠点機能の整備・カリキュラム策定

- AIチップ開発に必要な座学は既存のカリキュラム(VDEC連携・福岡IST連携)を活用して網羅
- 重点1：未経験技術者（ソフト・アルゴリズム技術者）に向けた取り組み / フォーラム：毎月実施（～200名程度の参加）
- 重点2：実習レベルの教材準備（エミュレータ協調検証教材、デジタル設計教材、高位設計教材、設計フロー教材、AI-SoC設計教材）
- 重点3：Ai-One利用での実践トレーニング（システム設計者、FPGA経験者、高位記述設計者向け）

ハード開発経験知見	試作・製造	テスト	検証技術	EDAツール			AIアルゴリズム	AIチップ	システム設計	組込ソフト
			エミュレーション	デジタル設計	アナログ設計	設計フロー	ML/DL等			
あり	実習 手簿	AI拠点 重点2：教材となる題材（正しいデータ、間違っただデータなどを蓄積する） 重点3：SoCを教材とした実践トレーニング								
	座学	VDEC連携 座学 デジタル設計 手法	AI拠点 エミュレータ 協調検証	AI拠点 デジタル設計 トレーニング	AI拠点 アナログ設計 RF編	AI拠点 設計フロー	産総研 AIセンタ ー連携	AI拠点 AI SoC設計		あかIST 連携
なし	マッチング	AI拠点 重点1：フォーラム・セミナー（フランクな勉強会、マッチング）を月一回以上開催								
	成果アピール									

人材育成のグランドデザイン図

概論

AIとは(F)

LSI設計とは(F)

SoCとは(F)

最先端プロセス(F)

各論

高位合成(基礎・応用)

デジタル設計フロー
(基礎・応用)

論理検証・エミュレータ
を用いた設計検証

ロジックテストを用いた
SoCテスト手法

各ツールトレーニング

応用・実践編

拠点の利用方法

AiOne設計

IPコアのアクセスと設計

フルチップ検証手法

2. 成果：Forum開催によるアウトリーチ活動

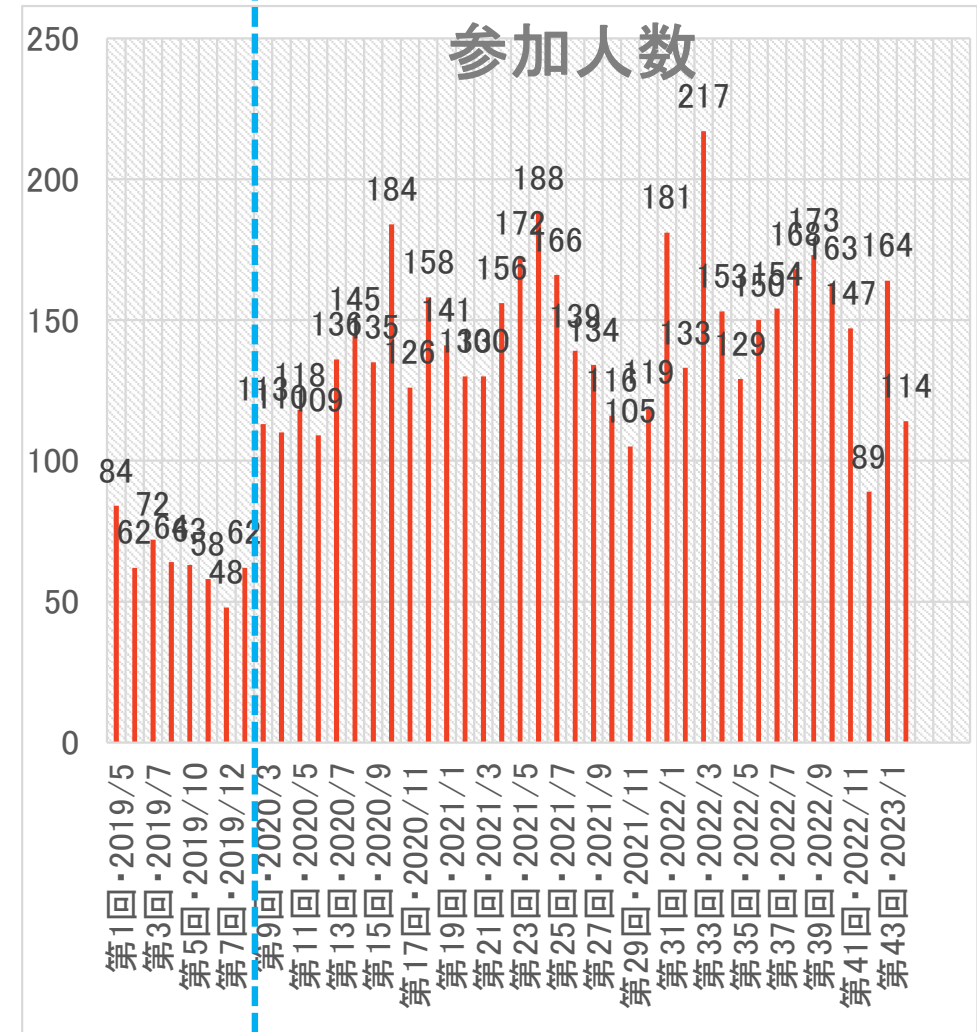
1. Webinarになった第9回から参加者が増加し、平均参加者は144名（第9～43回平均：それ以前64名）。

2. 第27回の中小企業の割合 = $\frac{\text{中小企業}}{\text{中小企業} + \text{大企業}} = 38\%$

第1回：5/17
 第2回：6/19
 第3回：7/26
 第4回：9/25
 第5回：10/30
 第6回：11/27
 第7回：12/23
 第8回：1/31
 第9回(W)：3/27
 第10回(W)：4/24
 第11回(W)：5/29
 第12回(W)：6/26
 第13回(W)：7/31(Fri)
 第14回(W)：8/28(Fri)

第15回(W)：9/29(Tue)
 第16回(W)：10/30(Fri)
 第17回(W)：11/27(Fri)
 第18回(W)：12/23(Wed)
 第19回(W)：1/29(Fri)
 第20回(W)：2/26(Fri)
 第21回(W)：3/26(Fri)
 第22回(W)：4/28(Wed)
 第23回(W)：5/28(Fri)
 第24回(W)：6/25(Fri)
 第25回(W)：7/21(Wed)
 第26回(W)：8/27(Fri)
 第27回(W)：9/24(Fri)
 ・
 ・
 第44回(W)：2023/2/24(Fri)

オンライン化



2. 成果: セミナーの実施

年月日	発表媒体	発表タイトル
2019年12月19日～20日	세미나	高位合成ツール「Catapult」セミナー
2020年1月15日～16日	세미나	エミュレータとエミュレータ・シミュレータ協調検証環境セミナー
2020年3月30日～31日	Webセミナー	論理検証に関するWebセミナー
2020年10月16日	Webセミナー	高位合成トレーニング
	Webセミナー	Cadenceパラジウム関係のデバッグ機構
2021年6月30日	Webセミナー	エミュレータを利用した検証の高速化に向けたテストベンチ作成方法
2021年7月1日	Webセミナー	シミュレータ・エミュレータによる協調検証環境（EmuForge）を用いた検証効率向上
2021/11/30, 2022/6/7, 2023/2/14	Webセミナー	エミュレータトレーニング
2022/3/10, 2022/9/10, 2023/3/14	Webセミナー	エミュレータ・ワークショップ
2022/9/28, 2023/3/20	Webセミナー	RISC-VベースSoCのシミュレーション・エミュレーション協調検証トレーニング

2. 成果:教材・Webinarの整備

問題1: メタスタビリティ

メタステーブル: 0や1に安定していない状態

• setup/holdウィンドウ内でデータが変化した場合に発生します
• 後段に伝搬すると誤動作を引き起こします
0:00 / 50:24

- 再生します。再生後、以下のアイコンが表示されます。
- Picture in Picture機能 (サムネール右下: をクリック)、新規Windowで視聴出来ます。
- Full screen機能 (サムネール右下: をクリック) で、解像度が上がります。
- 音量調節
音声が入切れます。対策版を入手中です。
本動画は、拠点ホームページでのみ閲覧ください。Downloadや配布等は禁止です。

基本操作

GUI-Debug

Tips

CDC検証

3. 論理合成と等価検証

基本操作と機能概要説明

論理合成グラフィカル

等価検証

4. 教材

Emulatorと論理Simulator, 高位設計、論理設計と検証

エミュレータフローの一般論 (初級編)

デジタル設計フローの一般論 (初級編)

高位合成を使ったデジタル設計 (基礎編)

形式検証編

Auto-Ungrouping 実行例

0:00 / 2:14:51

- 再生します。再生後、以下のアイコンが表示されます。
- Picture in Picture機能 (サムネール右下: をクリック)、新規Windowで視聴出来ます。
- Full screen機能 (サムネール右下: をクリック) で、解像度が上がります。
- 音量調節
本動画は、拠点ホームページでのみ閲覧ください。Downloadや配布等は禁止です。

基本操作

GUI-Debug

Tips

CDC検証

3. 論理合成と等価検証

基本操作と機能概要説明

論理合成グラフィカル

等価検証

4. 教材

Emulatorと論理Simulator, 高位設計、論理設計と検証

エミュレータフローの一般論 (初級編)

デジタル設計フローの一般論 (初級編)

高位合成を使ったデジタル設計 (基礎編)

12章 デザインフロー

12.8. Design Analyzer の起動

Project Files の「Open Design Analyzer」をダブルクリックすると、Design Analyzer が起動します。

Design Analyzer を開くと、このDPAウィンドウと併せて設計制約からエラーメッセージウィンドウが起動されます。

2. RTL形式検証

基本操作と機能概要説明

本講座の概要

基本操作

GUI-Debug

Tips

CDC検証

3. 論理合成と等価検証

基本操作と機能概要説明

論理合成グラフィカル

等価検証

12章 デザインフロー

12.9. Design Analyzer の画面構成

Instance View: デザインのInstanceを表示

RTL Schematic: デザインのRTL Schematicを表示

Object View: デザインのオブジェクトの情報を提供

Schedule View: デザインのスケジュールングをハードウェアのみに表示

Cross-linked G1: デザインのG1コードを表示。他のウィンドウと連携

Schedule View の重要 (redtext.html) をクリックすると、G1コード上の最適化が強調表示されます。同じくその他の最適化もクリックして、G1コードのどこが強調表示されるを確認して下さい。新しい回路のプロジェクトは、必ず、それらが、G1コードのどこによっていかに確認して下さい。

ウィンドウの右上のxをクリックし、Design Analyzer を終了します。

4. 教材

Emulatorと論理Simulator, 高位設計、論理設計と検証

エミュレータフローの一般論 (初級編)

デジタル設計フローの一般論 (初級編)

高位合成を使ったデジタル設計 (基礎編)

エミュレータ論理検証の基礎 (初級編)

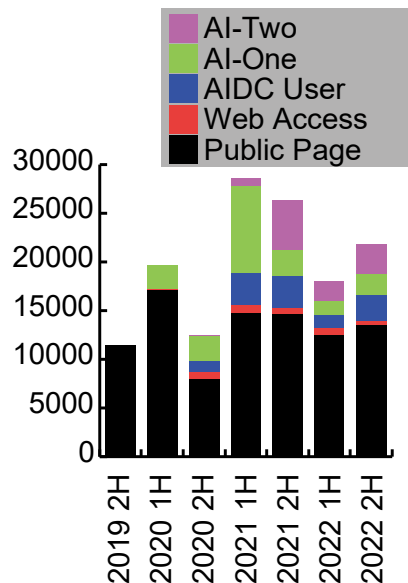
エミュレータ論理検証の基礎 (応用編)

Catapultを使用します
サンプル回路で演習できます

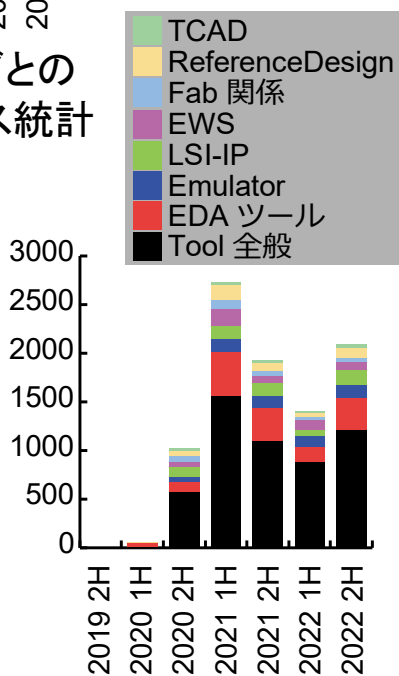
Download

高位合成編

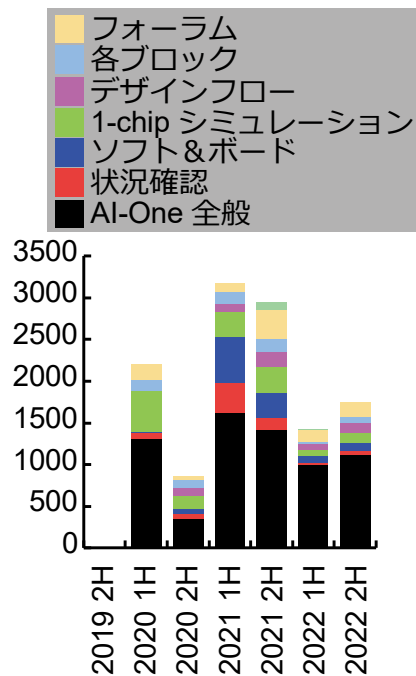
2. 成果: 拠点のアクセス状況



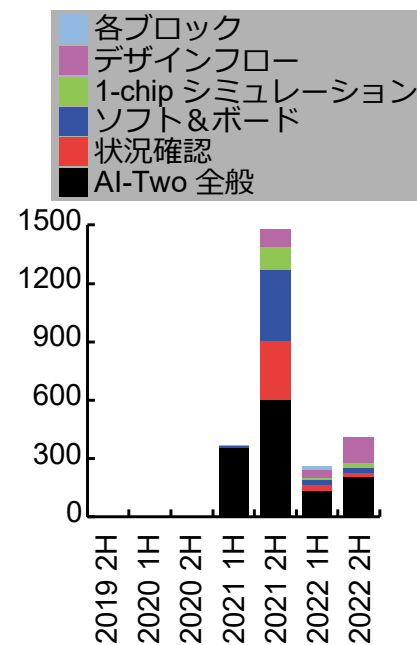
アカウント種別ごとのWebサイトアクセス統計



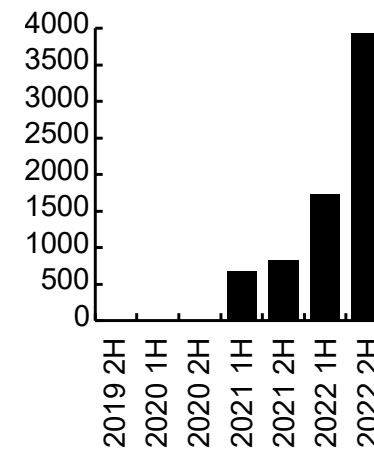
利用法関係のサイトアクセス状況



AI-One関係のサイトアクセス状況



AI-Two関係のサイトアクセス状況



Webinar視聴状況

2. 成果: 拠点利用者等のフィードバック状況

- フィードバック件数: 拠点利用者、一般(拠点利用検討者含む)、フォーラム参加者から合計530件
- 拠点利用方法の向上や技術情報に関するフィードバックに関しては、拠点ウェブサイトのFAQページやサーバ等に情報を集約し掲載。設計ツールに関しては、小規模ウェブセミナーを開催し、ノウハウを伝授している(下図、詳細非公開)。
- 上記フィードバックに対し、システム設計者、ソフト設計者を交え、技術打ち合わせを毎週～隔週実施

拠点FAQの構築状況 (2023/3/31 時点、詳細は非公開)

FAQ掲載場所と公開範囲		FAQ内容と件数			
		拠点接続	EDAツール	IP/SoC	EWS
拠点WEB	一般	10件	1件	5件	
	拠点利用者	30件	71件	213件	15件
拠点サーバ			181件		4件

2. 2-1 成果・進捗状況(2020年度までの成果)

◎大幅達成、○達成、△一部未達、×未達

項目	成果	達成度	
		水準	内容
ツール群の導入	永久ライセンス(ベース部分)および単価契約(変動部分)に分割してのEDAツールの導入を完了した	◎	「AIチップ開発加速のためのイノベーション推進事業／研究開発項目①: AIチップに関するアイデア実用化に向けた開発」の助成事業者等へのヒアリングを実施することで、EDAツール利用見込みを作成し、この見込みと2019年度の利用実績などに基づいて四半期ごとのEDAツールライセンス数を最適化した 利用状況可視化システムとしてOpenLMを導入し運用を開始
導入ツールの保守管理運用方針の策定と運用開始	利用規約の整備を行った	◎	ツール群安定最新バージョンへのアップデート及びアップデート後の動作を確認
EDA利用サーバの導入とツール群のインストールを行い、リモートからログインしてのツール群の利用環境を整備し拠点利用者へ公開	EDAライセンスサーバの導入とライセンスサーバのインストール起動、利用サーバへのツールのインストールと起動の確認を行った。	◎	2019年5月以降順次利用者を増やし、安定運用を実現している 2021年3月以降福岡サテライトでのライセンスの利用を実現 セキュリティ向上、拠点の規模、能力拡充を継続的に実施 ログ取得環境の整備と、ログのリアルタイム解析による監視強化を実現

2. 2-1 成果・進捗状況(2022年度までの成果)

◎大幅達成、○達成、△一部未達、×未達

項目	成果	達成度	
		水準	内容
EDAツールを安定的最新バージョンに更新、利用環境を整備し、拠点利用者へ公開	永久ライセンス(ベース部分)の保守の継続、利用状況に応じた単価契約(変動部分)でのEDAツール数の調整、加えて1か月単位でのスポットでのライセンス数の調整を実施、いずれのライセンスも適切な稼働と安定的な稼働を実現	◎	利用状況可視化システムとしてOpenLMの運用により動的にライセンスの利用状況を確認し、ライセンスの過不足を確認できる環境を実現 ツール群安定最新バージョンへのアップデート及びアップデート後の動作を確認、利用者の要望に応じたツールバージョンのメンテナンスを実施 EDAツールの利用実績、EDAツール利用見込みおよび設計の状況に応じたスポットでのライセンスの導入により、ライセンス数の最適化と過不足ない運用を実現
期間全体を通し15件程度の利用実績をあげる	合計でのべ74件の利用	◎	当初目標を大幅に上回る利用に対して大きな障害を生じることなく安定な運用を実現

2. 2-2成果・進捗状況(2020年度までの成果)

◎大幅達成、○達成、△一部未達、×未達

項目	成果	達成度	
		水準	内容
エミュレータ及びEDAツールを活用した設計フローの一般論(初級編)についての教材を電子的に整備し、拠点利用者へ公開する	<ul style="list-style-type: none"> 教材の整備の実施 セミナーの実施 	○	実施項目①-1、①-2において確立されたエミュレータ・シミュレータ協調設計フローに即した、エミュレータ・EDA協調設計検証カリキュラム(初級編)を構築した。また、その教材を電子的に整備し、HPにて公開した 整備した教材に基づき、1項目あたり年間1回以上の教育コースを拠点利用者等に対して実施した。必要に応じて内容の見直しを図った
国内外の代表的なFABと契約を締結し、設計環境を導入することで設計試作ゲートウェイ機能を整備する。これを拠点利用者へ公開することで、実施項目①-1の中間目標のネットリスト検証フロー利用目標3件のうち1件程度が本設計試作ゲートウェイ機能を利用して設計手順を実施	<ul style="list-style-type: none"> ユーザ企業への個別のヒアリングを実施 AI-Oneの設計を通じた整備を実施 	○	実施項目①-1中間目標のネットリスト検証フロー利用3件のうち1件が本設計試作ゲートウェイ機能を利用。そのフィードバックを通じて本設計試作ゲートウェイ機能の整備を行った
拠点運営について運営形態等の検討を行うコンソーシアム等を立ち上げる	<ul style="list-style-type: none"> 設計拠点自立運営検討会の立ち上げ 	○	拠点運営について運営形態等の検討を行うAIチップ設計拠点自立運営検討会を立ち上げた

2. 2-2成果・進捗状況(2022年度までの成果)

◎大幅達成、○達成、△一部未達、×未達

項目	成果	達成度	
		水準	内容
・エミュレータ・EDA協調設計検証論(上級編)を整備し、各種教育カリキュラムの教育コースを充実させる	<ul style="list-style-type: none"> 教材の整備の実施 セミナーの実施 	○	実施項目①-1、①-2において確立されたエミュレータ・シミュレータ協調設計フローに即した、エミュレータ・EDA協調設計検証カリキュラム(初級編)を構築した。また、その教材を電子的に整備し、HPにて公開した 整備した教材に基づき、1項目あたり年間1回以上の教育コースを拠点利用者等に対して実施した。必要に応じて内容の見直しを図った
実施項目①-1最終目標の協調設計検証フロー利用目標10件のうち3件程度が本設計試作ゲートウェイ機能を利用して設計手順を実施する	<ul style="list-style-type: none"> ユーザ企業への個別のヒアリングを実施 AI-One、AI-Twoの設計、試作を本設計試作ゲートウェイ機能を利用して設計手順を実施 	○	2020年終了時点のAI-Oneのテープアウト、試作に加え、3件のIPを搭載したAI-Twoのテープアウト、試作を行ったそのフィードバックを通じて本設計試作ゲートウェイ機能の整備を行った。

資料6-1

実施項目1-5 「国内外FABの活用と 最適化ライブラリの研究開発」

担当機関 東京大学
産業技術総合研究所
発表者 長谷川 淳（東大）

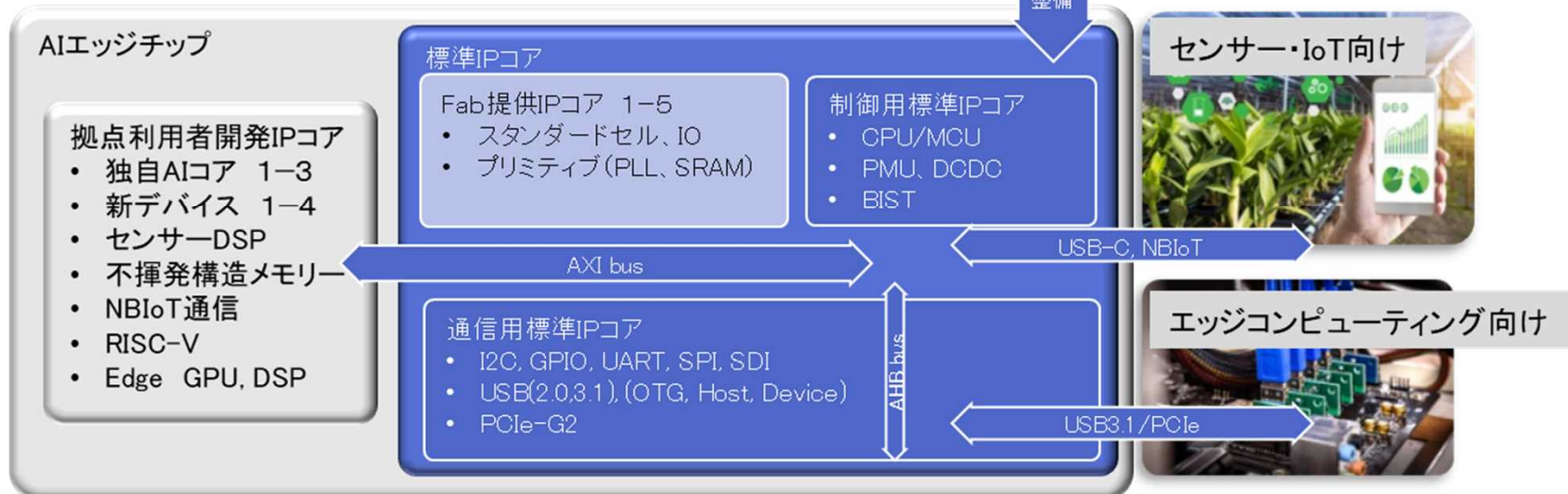
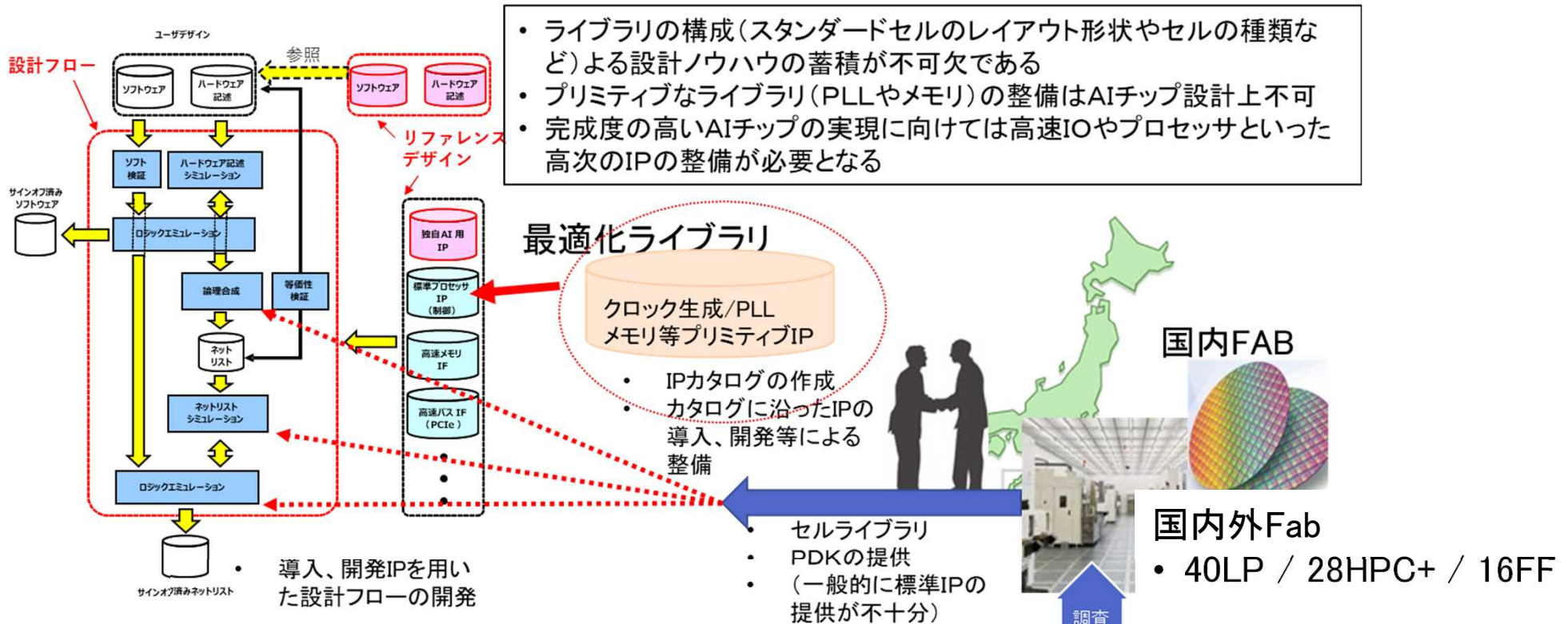
背景・課題

- 昨今の半導体チップは海外のFABで製造される場合が多い
- 半導体チップの偽造品や模倣品の市場への流通
- セキュリティを保證するアプリケーション向けの半導体の国内FABで製造への需要
- 国内のFABは自社以外の設計者の利用実績が極端に少なくノウハウの蓄積・公開が皆無
 - 中小ベンチャー企業を含む他社の技術者が国内FABの利用に向けた設計に容易に取り組める状況ではない
- 制御用標準IPコアや通信用標準IPコアは、AIチップを開発するベンチャー・中小企業が個々に準備する必要がある。これらを独自に組み上げた場合にはその組合せの動作検証が必要である、通信規格がある通信用標準IPコアを開発した場合、その規格検証やフィールドテストが必要になる。

目的

- 国内FABを中小ベンチャー企業等が活用できるようにIPを整備する
- 国外FABのIPの利用環境も充実させる
- 1-2で構築する設計フローを実際のチップに適用することで完成させる
- 設計事例により設計のノウハウを蓄積する
- 整備したIPを活用したAIチップ向け独自IPの設計評価手法を確立する

1. 概要(背景・目的・課題)



取り組み内容

- ベンチャー・中小企業に必要な標準IPコアの調査を行い、国内外のFabが標準で提供するIPコアに加え、制御・通信用標準IPコアを包含した標準IPコアを開発・整備する
- 設計フローやIPの準備、設計のノウハウの蓄積が十分ではない国内FABを活用できるようにIPの開発や整備を行う
- IPを活用した設計事例の実現、設計フロー、設計ノウハウの蓄積に向けたSoC (Ai-One)の設計を行う

技術の特徴

- 国内FABの利用が実質的に可能となるとともに、設計や試作経験のないAIベンチャー企業が国内FABを活用してAIチップ試作を実現できる。
- 個別に各拠点利用者が準備する標準IPコアを本施策で開発整備することで、拠点利用者は競争力のあるIPコアのみを開発して、開発期間の短縮を実現
- 国内外FABの活用と最適化ライブラリの研究開発で実施される設計フローの最適化可能となる
- 乗合チップ (Ai-One) は先端の通信IP、プロセッサIP、ネットワークIPを搭載し複数のAI-IPを搭載したヘテロ・コア型SoCとして非常にユニークな事例となり得る

中間(2020年度末時点)目標

- 整備すべきIPカタログの作成と整備順の策定
- カタログに沿ったIPを整備し、拠点利用者が利用できるようにWEBに掲載する
- AIチップ向け独自IPの評価手法を検討し、それに則った評価チップを設計する。

最終(2022年度末時点)の目標

- 整備したIPを用いたアプリケーション向けの設計フローを構築し、電子的にマニュアルを作成する。構築した設計フローは拠点利用者の利用を通してその有効性を実証する。
- AIチップ向け独自IPの評価手法を構築し、拠点利用者に提供できるように整備する。その利用法のマニュアル等を電子的に作成しHPにて公開する。

2018年度 2019年度 2020年度 2021年度 2022年度

1-5. FAB用ライブラリ

IPカタログ

IP整備

IPを活用した設計フロー

評価チップAi-One設計

Ai-One評価

Ai-Two設計

Ai-Two評価

成果

- 整備すべきIPカタログ、PRISMにより導入したIPリストを整備し、HPに掲載
- 拠点導入IPを最大限活用し、かつ拠点利用者設計の独自IPの実用化支援を目指すため、その第一弾として“**Ai-One SoCプロジェクト**” (TSMC28nmを活用)を立ち上げ、拠点利用者の独自IPを搭載した評価チップを設計、試作し、評価ボードを製作して評価を実施
- TSMC12nm FinFETプロセス向けIPを整備しHPに掲載
- 拠点利用者設計の独自IPの実用化支援の第2弾として**Ai-Two SoCプロジェクト**を立ち上げ、拠点利用者の独自IPを搭載した評価チップを設計、試作し、評価ボードを製作して評価を実施

意義

- AIチップ開発に必要なかつ、汎用的なインターフェースIP等が整備され利用可能になったことで、特殊機能の具現化が容易になる
- アイディアの実証に際して、必要となる評価手法、ノウハウなどの獲得につながる

2. 成果

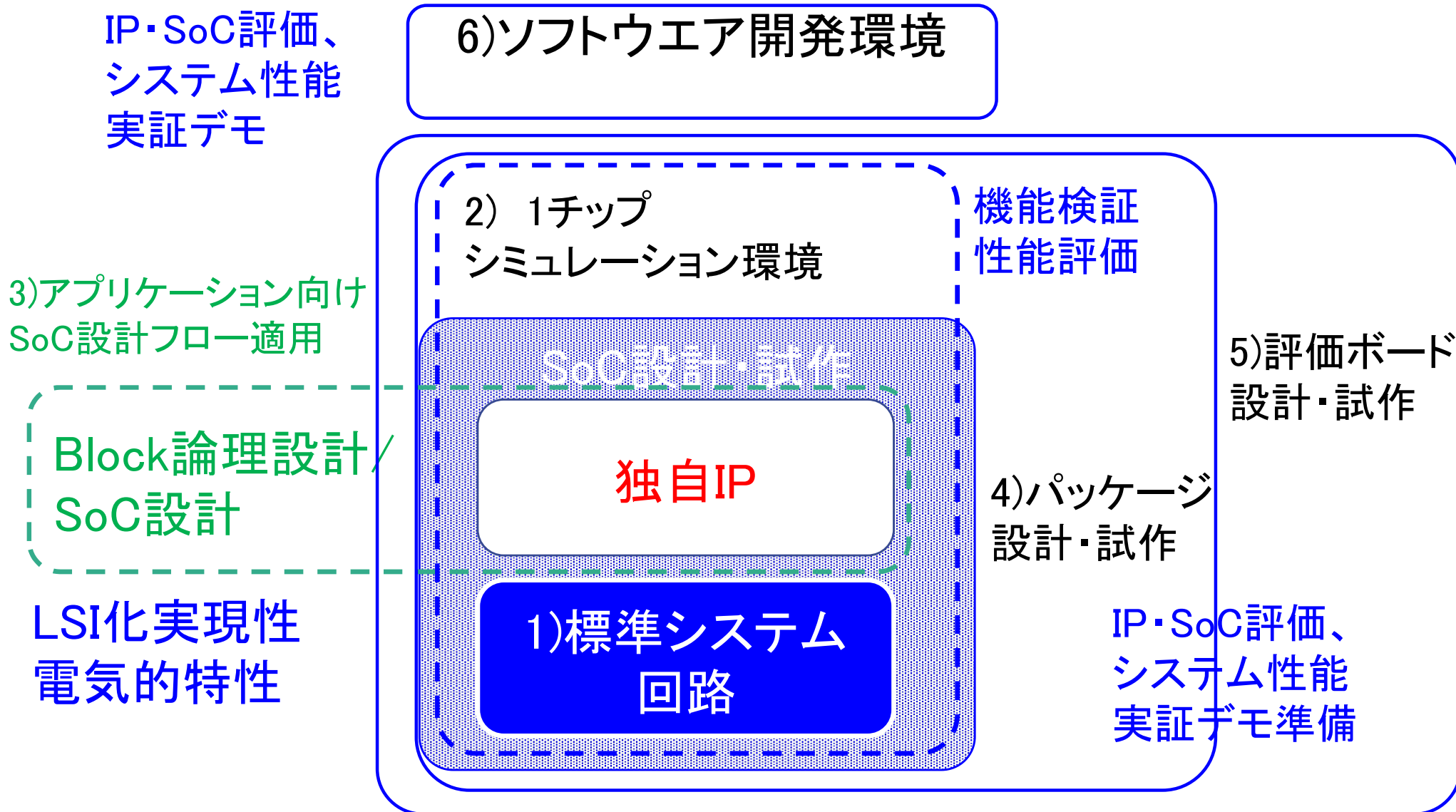
ライブラリおよびIPの整備状況

- 中間評価時点までに拠点協力社、拠点利用者からの要望・意見に基づき整備の優先順位を決定したライブラリ及びIPコアの整備を行った。整備したIPのリストをHP上に掲載
- 2020年3月、28nmSoCプラットフォーム向けにIPの追加整備を行った。
- 2021年度9月、さらに高度なAIチップのモデルケースとして、12nmFinFETプロセスSoCプラットフォームおよび評価チップAI-Twoの設計開発を開始。これに伴い12nmFinFETプロセス向けIPおよびライブラリの整備を行った。
- 追加整備したIPリストはHP上に掲載し拠点登録ユーザに提供している

Catgry	Edge	Graph	IP packageName
DDR		DDR-cntl	Universal DDR MCTL2P MP
		DDR-opt	Adv Feature Pkg for uMCTL2
		DDR-addOnLP	LPDDR4 Add-on to uMCTL2
		DDR-Phy	LPDDR4 multiPHY V2 TSMC 28HPC+
		DDR-Phy	LPDDR4 mPHY V2 TSMC 12FFC: Require TSMC NDA
PCIe		PCIeG3.1	PCIe 3.0 Standard AMBA II
		PCIeG2.1-PHY	PCIe 20 NS TSMC28HPCP X1
		PCIeG2.1-PHY	PCIe 20 NS TSMC28HPCP X2
		PCIeG3.1-PHY	PCIe 30 C8 NS TSMC28HPCP X4
		PCIeG3.1-PHY	PCIe 30 C10 NS TSMC12FFC X4: Require TSMC NDA
USB	USB2.0 40-OTG	USB2.0 28-OTG	usb_2_0_hs_otg_v4
	USB2.0 40-Phy		USB2 picoPHY 40LP25
		USB2.0 40-Phy	USB2 femtoPHY 28HPC+18
MIPI	MIPI 40-Csi	MIPI 28-Csi	MIPI CSI2 Host Controller
	MIPI 40-Dsi	MIPI 28-Dsi	MIPI DSI Host Controller
	MIPI 40-Tx		mipi_dphy_bd4_t40lp25
	MIPI 40-Rx		mipi_dphy_rx4_t40lp25
		MIPI 28-Tx	mipi_d_t4_tsmc28hpcp18ns
		MIPI 28-Rx	mipi_d_r4_tsmc28hpcp18ns
ADC	ADC		ADC-IQIF 12b250M 40LP
		ADC	ADCIQ12b320M 28HPCPNS
	DAC		DAC-IQ, 12b160M 40LP
		DAC	DACIQ12b640M TSMC28HPCPNS
StdCell, RAM, Bist, Yield	Std40		Duet 40LP
	SRAM40 1Port		
	ROM40		
	SRAM40 2Port		Duet TSMC 40LP 2PUHD Add-on
	Std40-HPC		Duet TSMC 40LP HPC Design Kit
		Std28	Duet 28HPC+
		SRAM28 1Port	
		ROM28	
		SRAM28 2Port	Duet 28HPC+ 2PUHD Add-on
		Std28 HPCt	Duet 28HPC+ HPC Design Kit
		Std12FFC	TSMC 12ffc 6t, 7d5t, 9t
		SRAM12ffc 1Port	TSMC 12ffc 1P SRAM, SP SRAM
	SRAM12ffc 2Port	TSMC 12ffc 2P SRAM, DP SRAM	

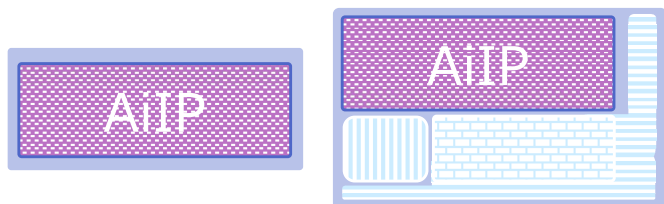
2. 成果

AIチップ向け独自IPコアの評価手法としてSoC設計評価プラットフォームを構築し
拠点利用者に提供できるように整備。その利用方法のマニュアルをHPに掲載



複数のヘテロジニアスIPコアを評価するためのSoCプラットフォーム

従来の単一IPテストチップ



単体TEGであったり、簡単なIPを有すのみ



再利用による
開発コスト低
減、期間短縮

プラットフォームベースのテストチップ

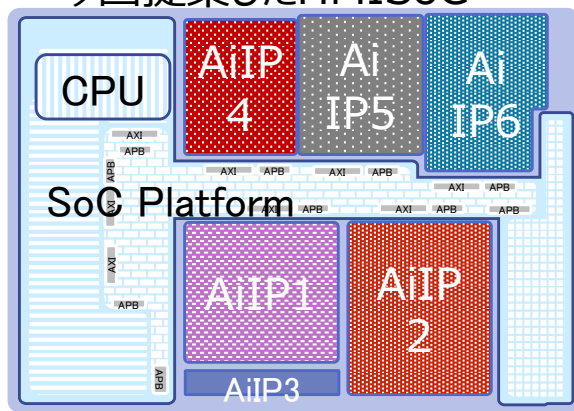


評価、デモ、POC開発に必要な共通IPを有す

複数のIPを単一の評価SoCに集積

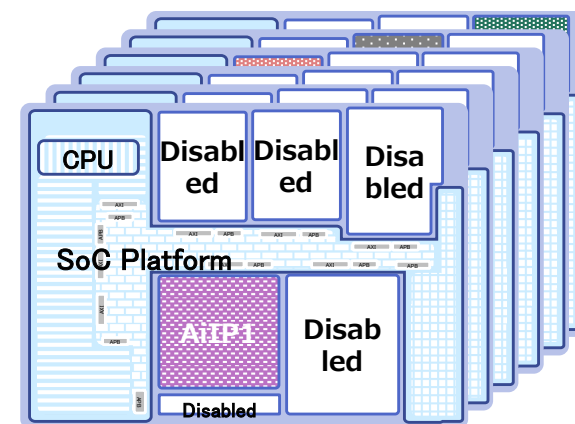
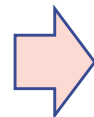


今回提案したHMISoC



開発費の大幅削減を実現する

各社に提供される
評価チップ



各社専用の評価チップを実現する

AIチップ向け独自IPの評価手法にのっとりた評価チップAI-One

- 協力会社5社のAI IPおよび設計拠点のリファレンスデザインを搭載した28nmプロセスSoCチップの設計、試作を行った
- 評価プラットフォームの有効性を確認できた

• 協力会社RTL提供	2020.05	
• テープアウトデータ完成	2021.02	約10か月
• チップ試作	2021.06	
• 評価ボード製造	2022.01	
• 標準システム回路動作確認	2022.02	約2か月
• 全IP動作確認・デモ	2022.03	約2か月

部品材料調達難により
チップ試作からボード製造完了まで想定外に時間がかかった

1チップシミュレーション環境を用い、約半数のソフトウェアテスト項目を事前デバッグ

ボードテスト機能で基板評価期間を短縮

AI-One SoC諸元

プロセス: TSMC 28nm HPC+ メタル10層

Dieサイズ: 9.0mm x 8.64mm

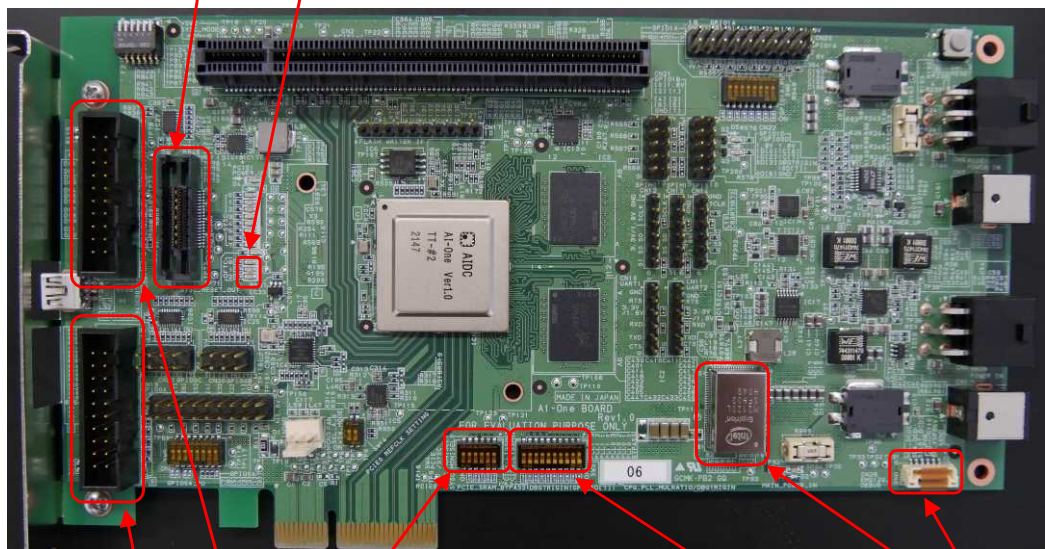
電源電圧 コア電源 0.9V, IO電源1.8V

パッケージサイズ: 27mm x 27mm 厚さ 3mm

最高動作周波数800MHz @all コーナーケース

⑥FLASHライターソケット

①MBIST (Memory Built-In Self Test)、LBIST (Logic Built-In Self Test)
テスト結果表示LED



- ソフトウェア開発主体の会社ではLSIテストの利用以前にLSIテストの使用準備のハードルが高い
- ボードのみである程度の評価が閉じればSoC開発のハードルが下がる
- 動作周波数、電源電圧の変更は評価ボードとホストPCで実施。
- 評価ボードによるテスト機能実現により製造した基板のテストが容易となり、また、試作時の評価を同時並列に実施することができる

⑥PCIeモード設定

④コア電源電圧設定及び消費電流計測

②JTAGデバッグインターフェイス ⑤PLL周波数設定切り替え

③JTAG2 DFTインターフェイス

12nmFinFET用IP、ライブラリを用いた評価チップAI-Two

- 28nm CMOSから12nmFinFETにプロセスを変えさらに高集積なAIアクセラレータの開発を可能とする環境を整備
- 3種のAI IPおよびAIDCのリファレンスデザインを搭載した12nmFinFETプロセスSoCチップAI-Twoの設計、試作、評価を行った
- 論理エミュレータを用いてソフトウェアのデバッグ・検証の短期化、効率化を実施
- 得られた知見をノウハウとして拠点に集積した



AI-TwoSoC諸元

プロセス: TSMC 12nm FFC メタル11層

Dieサイズ: 6.37mm x 6.82mm シュリンク後

電源電圧 コア電源 0.8V, IO電源 1.8V

パッケージサイズ: 23mm x 23 mm 厚さ 2.8mm

最高動作周波数 1GHz @ all コーナーケース

エミュレータを活用した1チップエミュレーション環境を用い、ソフトウェアについて、あらかじめLinuxのBootまで検証したことによる評価期間短縮効果大きい

2. 成果・進捗状況(中間時点:2020年度の成果)

◎大幅達成、○達成、△一部未達、×未達

項目	成果	達成度	
		水準	内容
整備すべきIPカタログの作成と整備順の策定	拠点利用者へのヒアリングに基づき整備すべきIPリストを作成した	○	ヒアリングに基づき優先順位を付けたIPリストを作成した
カタログに沿ったIPを整備し、拠点利用者が利用できるようにWEBに掲載する	IPリストに基づきPRISMにより導入したIPのリストをWEBに掲載した	○	PRISM事業での整備に加え、AI-Oneに向けたIPの整備を行った
AIチップ向け独自IPの評価手法を検討し、それに則った評価チップを設計する	拠点導入IPを最大限活用し、かつ拠点利用者の設計の実用化加速支援を目指したAi-One SoCプロジェクトを立ち上げSoCの設計を完了した	○	実用化支援を希望する6つの独自IPを搭載するAi-One SoCの設計をRTL提供から10か月で行い、チップ試作に向けたテープアウトを行った

2. 成果・進捗状況(2023年度終了時)

◎大幅達成、○達成、△一部未達、×未達

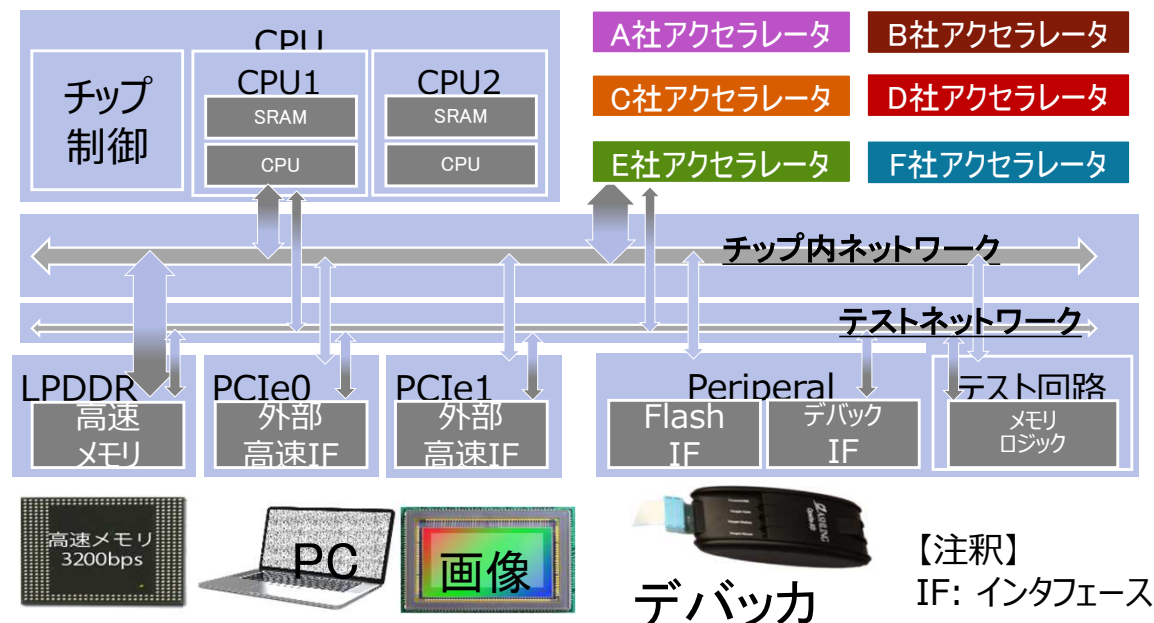
項目	成果	達成度	
		水準	内容
整備したIPを用いたアプリケーション向けの設計フローを構築し、電子的にマニュアルを作成する。構築した設計フローは拠点利用者の利用を通してその有効性を実証する。	AIチップ向け独自IPの設計・評価フローを構築し、作成したマニュアルをHPに掲載。評価チップAI-OneおよびAI-Twoの独自IP提供者による利用を通してその有効性を実証した。	◎	設計・評価フローのマニュアルをHPに掲載。 AI-Oneには6つの独自IPコアを搭載し、テープアウトから約1年で試作評価を行い、全IPの動作および目標性能達成を確認。 AI-Twoには4つの独自コアを搭載し、RTL提供から10か月でSoCの設計を完了。試作評価により全IPの動作および目標性能達成を確認。
AIチップ向け独自IPの評価手法を構築し、拠点利用者に提供できるよう整備する。その利用法のマニュアル等を電子的に作成しHPにて公開する。	SoC設計・評価プラットフォームを構築し拠点利用者に提供。プラットフォームの利用方法をHPにて公開した。	◎	28nmプロセス向けSoC設計・評価プラットフォームおよび12nmプロセス向けSoC設計・評価プラットフォームを構築した。構築したプラットフォームを活用しAI-OneおよびAI-Twoに搭載した各独自IPの早期動作確認および早期デモンストレーションを可能とした。

Appendix

1) 標準システム回路

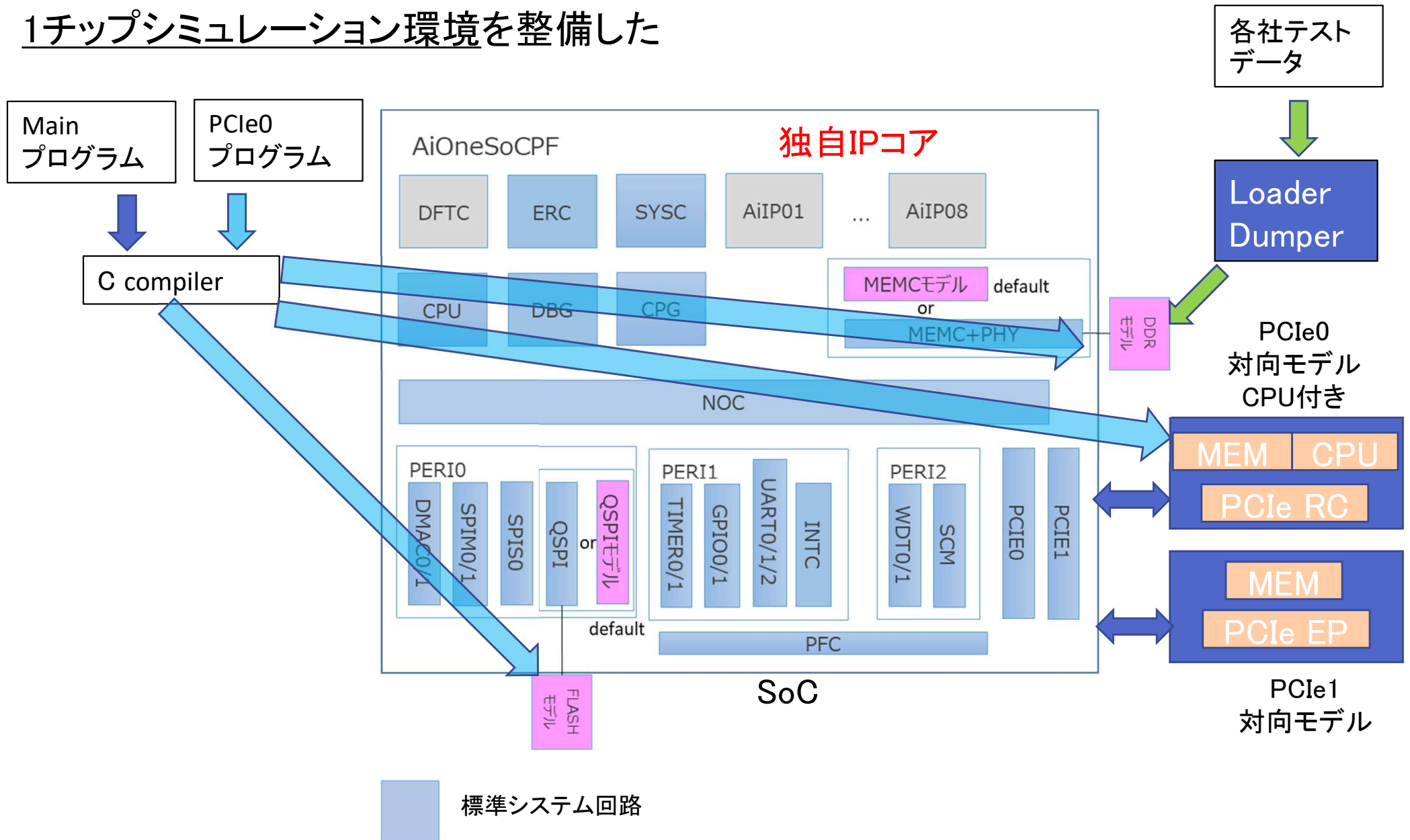
評価プラットフォームで整備した標準システム回路

- (1) CPU: デュアルコア構成
- (2) チップ内ネットワーク(NOC)
- (3) テスト回路: チップ内部の論理回路や内蔵メモリの故障検出・リペア機能
- (4) LPDDR4: 低消費電力型ダイナミックメモリインタフェース回路。
- (5) PCIe: PCなどの標準的な高速インタフェース
- (6) Peripheral: チップのデバッグ機能、System Boot可能なFlashメモリインタフェース、シリアルインタフェース、パラレルインタフェース、チップ全体制御(SYSC)、割込み制御、DMAコントローラ、タイマー、ウォッチドックタイマー

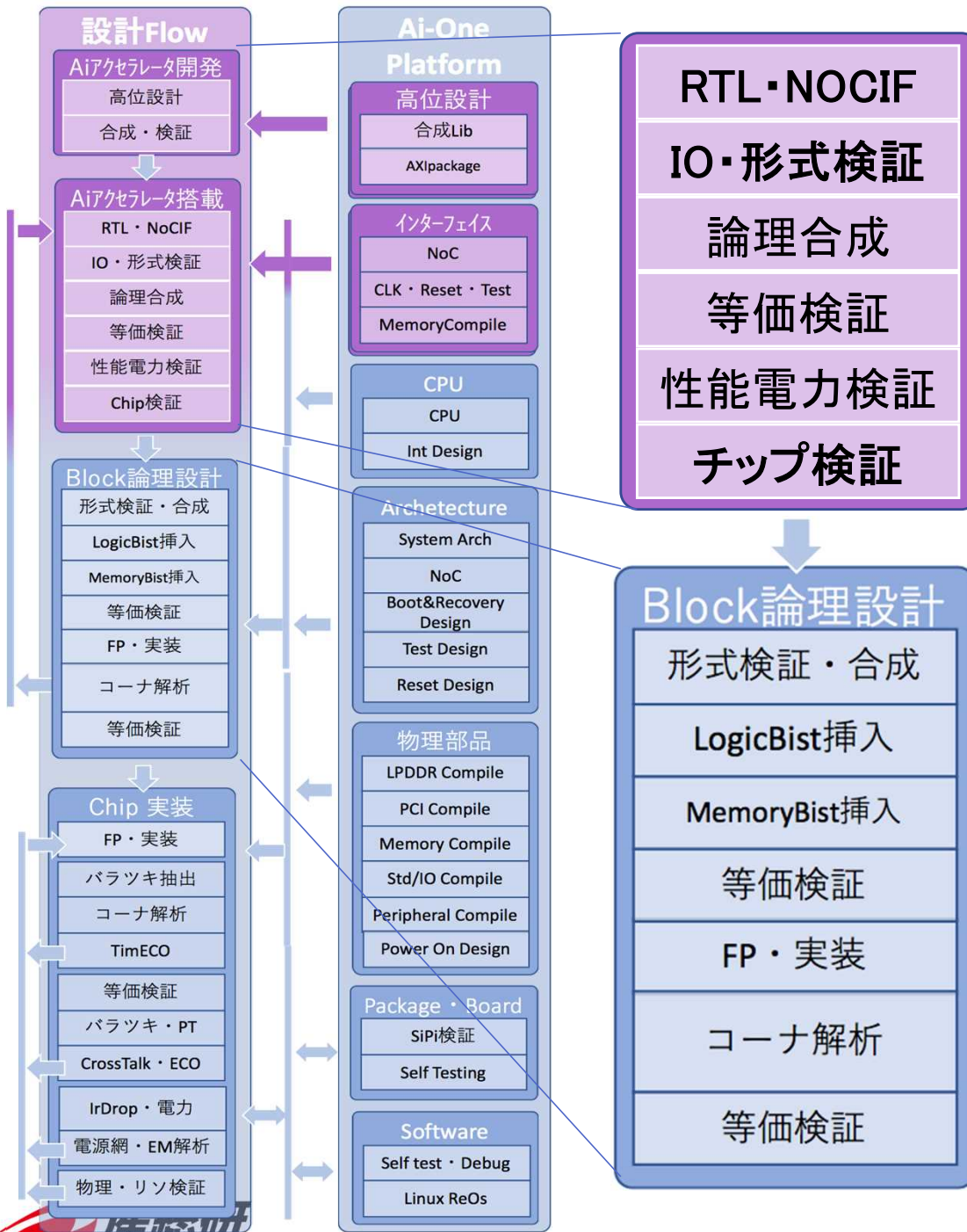


2) 1チップシミュレーション環境

独自IPコアをSoCと組合せたときの機能、性能の評価・検証を行うため
1チップシミュレーション環境を整備した



3) AIチップ向け独自IPの設計・評価フロー



独自IPコア搭載

独自IPコアの記述がC++の場合

- 高位合成ツールでRTL化
 - FPGAのライブラリは書換
- SoCへのインテグレーションのために
- インターフェイス仕様と使用するメモリ仕様の提出
 - Iocheck, SpyGlassチェックを実施、修正
 - 合成、等価検証
 - 論理シミュレーション検証による機能および性能検証

Block論理設計

- LogicBist挿入、MemoryBist挿入では、独自IPコアやNoC等にテスト回路設計を行う
- フロアプラン (FP)・実装では、各ModuleのFPと実装設計を行う
- コーナ解析でタイミング解析を実施以上を行い各独自IPコアのLSI化可否を評価・検証

FP: フロアプラン

独自IPコアのSoCへの搭載

NOC (Network on Chip)

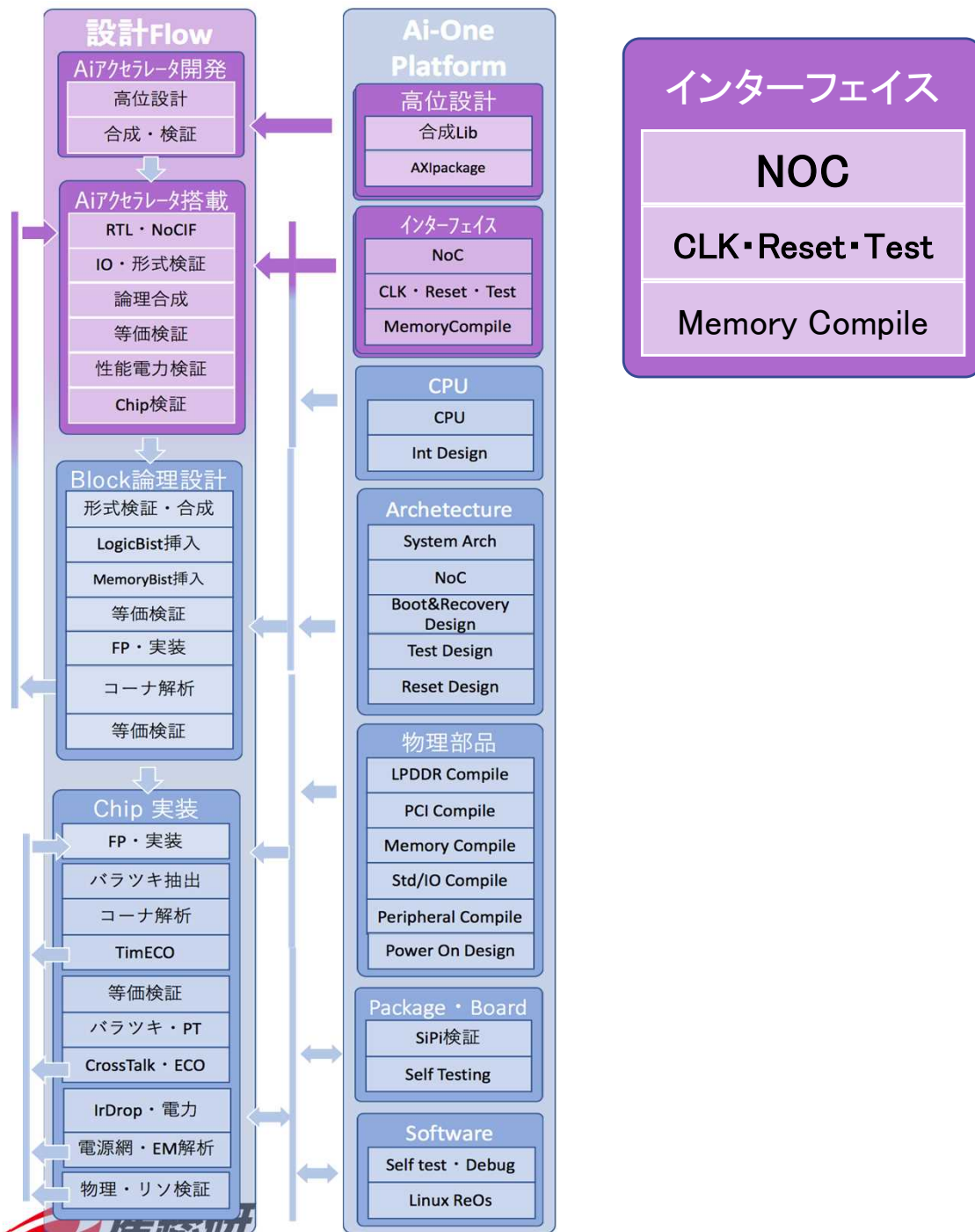
- 独自IPコアのNOCインターフェイス仕様要求に合わせてNOCのコンフィギュレーションを修正
- コンフィギュレーションに合わせたNOC RTLを生成しSoCチップに組み込み

CLK・Reset・Test

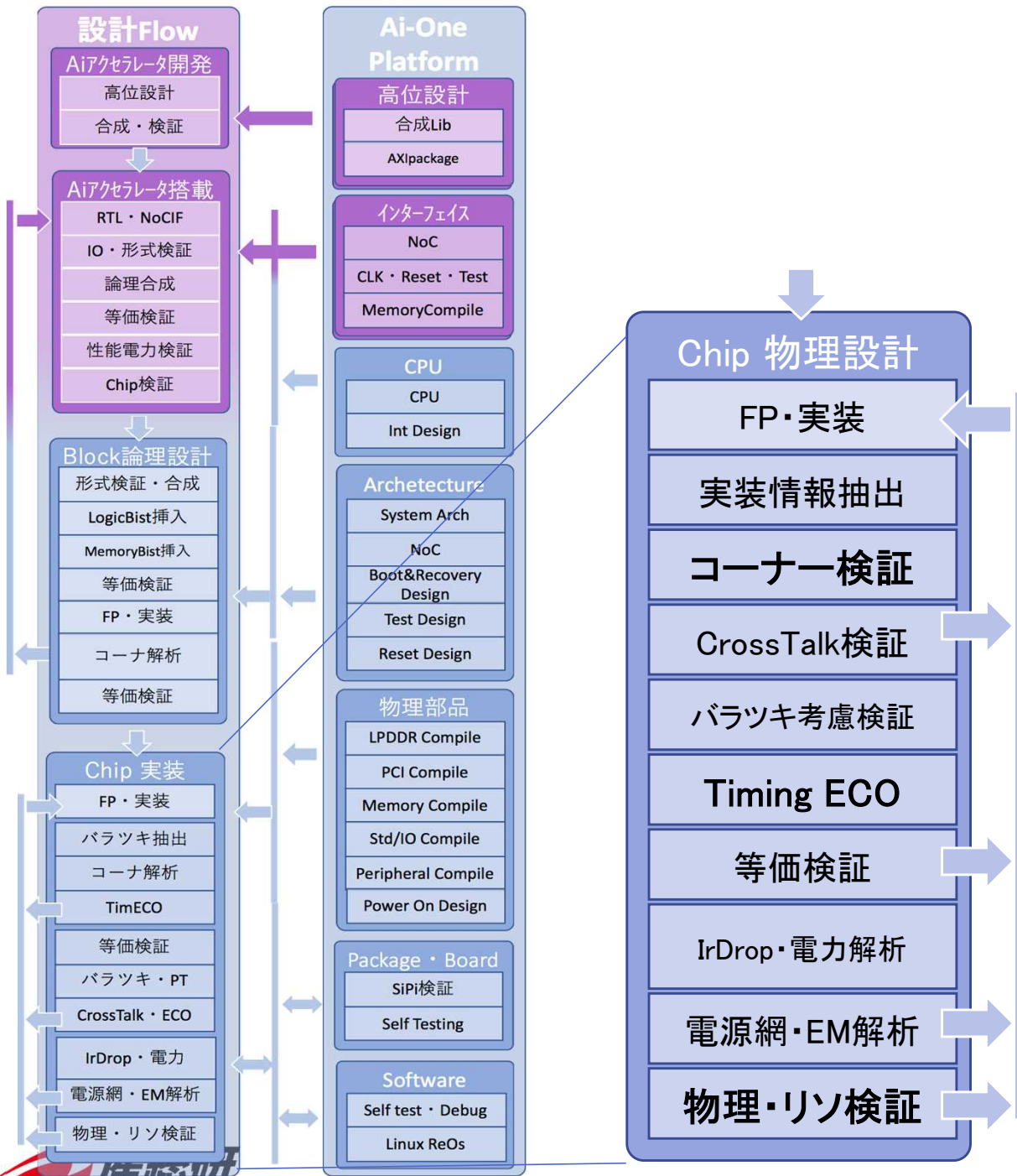
- 独自IPコアのCLK仕様、Reset仕様に合わせてSoCチップのシステムコントローラIPを修正
- 修正したNOC、システムコントローラを組み合わせシミュレーションによる接続検証を実施し、検証環境をリリース
- SoCチップテストコントローラに独自IPコアの論理テスト(LBIST)仕様およびMemory Compileで生成されたメモリテスト(MBIST)仕様を追加

Memory Compile

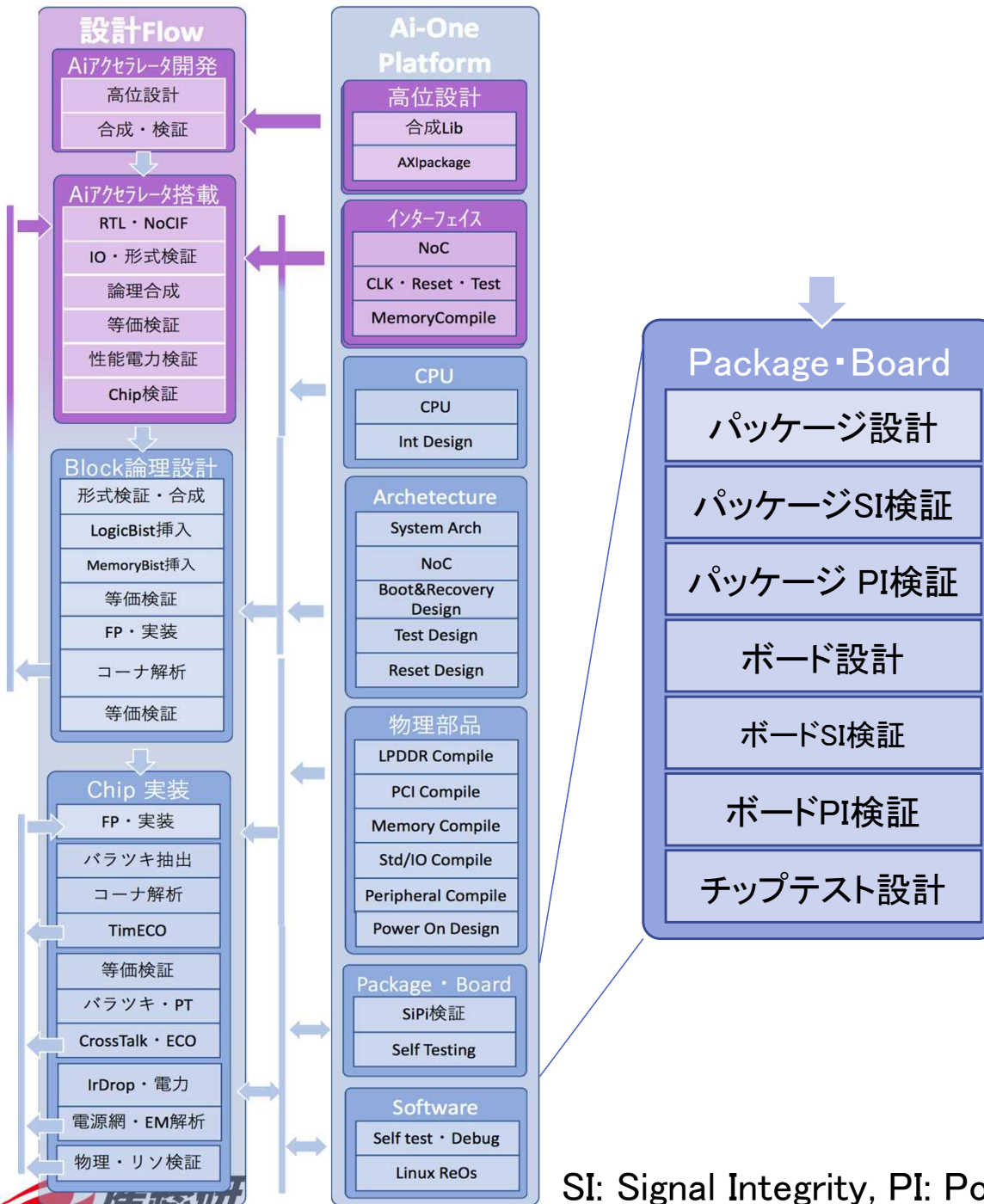
- 独自IPコアの内蔵メモリ要求仕様に合わせてSRAM Memory Compilerを用いてSRAMを生成。
- 生成されたメモリの設計データをSoC設計環境にリリース



3)アプリケーション向けSoC設計フロー(続)



- タイミング検証のコーナー検証において使用する電圧、温度、プロセスコーナーの組合せは対象とするプロセスごとに適正なものを使用する。微細プロセスになるに従い組合せ条件数、特にホールドマージンチェックの条件数が増加する。
- Timing ECOでタイミング検証の対策を実施。
- 物理・リソ検証はFABから要求される検証条件、ツールがプロセスごとに異なるため、FABのガイドラインに合わせ適正なものを使用する。



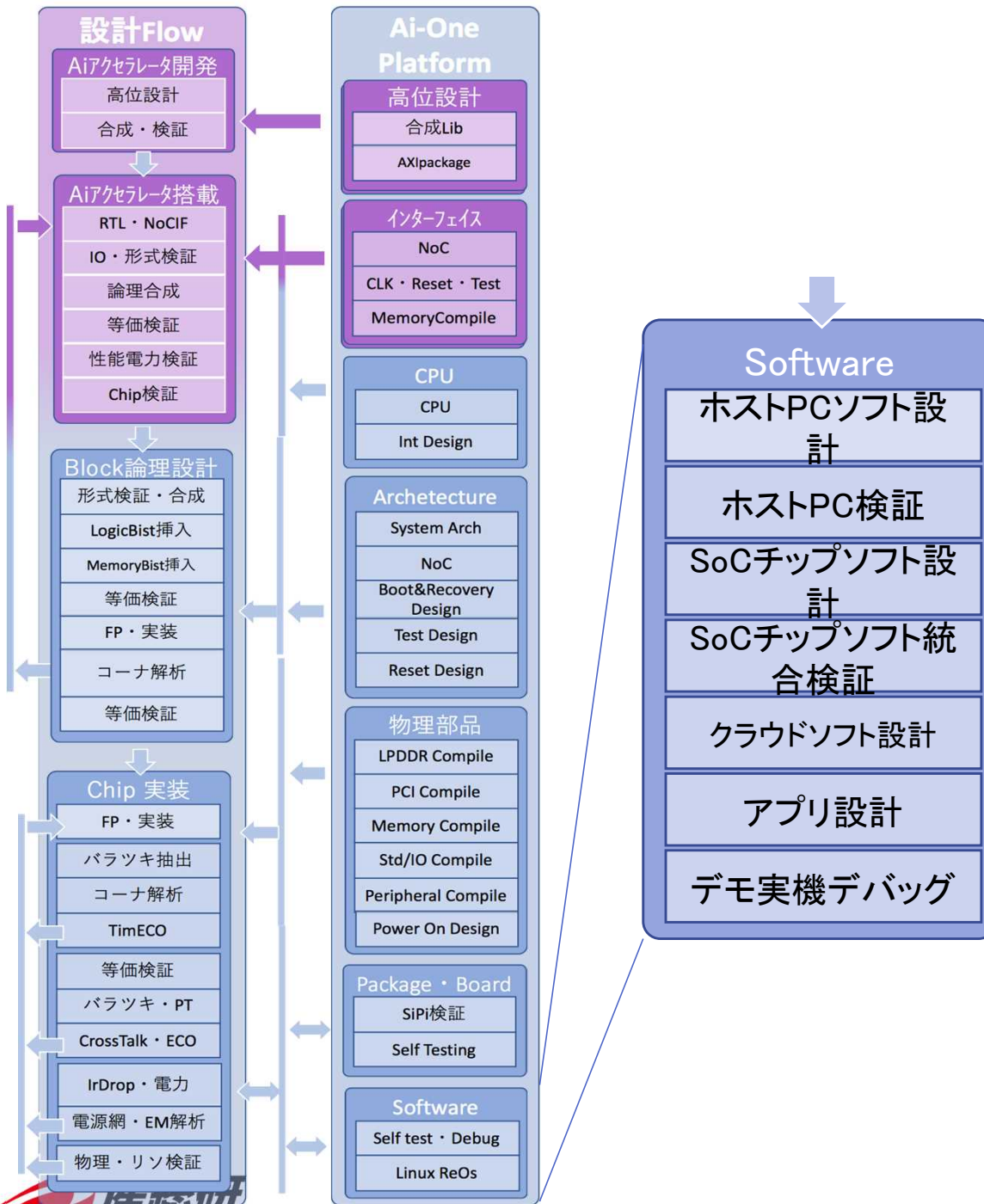
パッケージ設計

- パッケージのSI検証は高速インターフェイスであるLPDDR4, PCIeを対象にSPICEシミュレーションを実施。ボードのデータはSパラメータモデルを利用
- パッケージのPI検証はLPDDR4用1.2V, PCIe用1.8V, コアロジック電源の3つに分けそれぞれに対してSPICEシミュレーションを実施

ボード設計

- ボードのSI検証はLPDDR4, PCIeを対象にSPICEシミュレーション実施。SoCおよびメモリはIBIS, SPICEモデルを利用。ボードは、Sパラメータデータを利用。
- ボードのPI検証はボードの実抽出データとLCRの部品モデルデータを利用。
- チップセルフテストおよび結果の表示機能とJTAGによるテスト機能を実現

3)アプリケーション向けSoC設計フロー(続)



- ソフトウェアのデバッグ、検証に1チップシミュレーション環境を利用しシフトレフトを実現する
- ホストPC側のドライバなどの開発におけるデバッグ、検証をチップ試作前に行うためエミュレータのVirtual PC機能を用いたデバッグ、検証を実施する
- 1チップシミュレーション、エミュレータのVirtual PC機能を活用し、評価チップのテストプログラム、開発環境およびデモソフトウェアの事前デバッグを実施する

OTPを用いた拠点サーバへの接続と各種ツールの拠点サーバでの実行により機密保持と完全リモートワークを実現

マイルストーン (日付はAI-One)

RTLリリース前後 2020.04

- ・マイルストーン、概略日程、設計データのリリース要件の説明、共有
- ・RTLガイドライン説明
- ・NOCインターフェイス仕様、Iolistによるインターフェイス情報の収集
- ・SRAM要求仕様書によるSRAM生成用コンフィギュレーションデータ収集と物量の把握
- ・SpyGlass, DesignCompiler, FormalityによるIPの設計データ検証方法の説明とリリース前実施項目依頼

RTL Pon (RTL Power on) 2020.07

- ・1chip 論理シミュレーション環境の提供
- ・物理インプリメンテーションの初期実行結果のレビューとAIアクセラレータに対するフィードバック

DF (Dirty Final) 2020.10

- ・CF向け論理検証項目の説明と依頼
- ・物理インプリメンテーションからAIアクセラレータへのフィードバック
- ・拠点での各種検証結果のフィードバック

CF (Clean Final) 2020.12

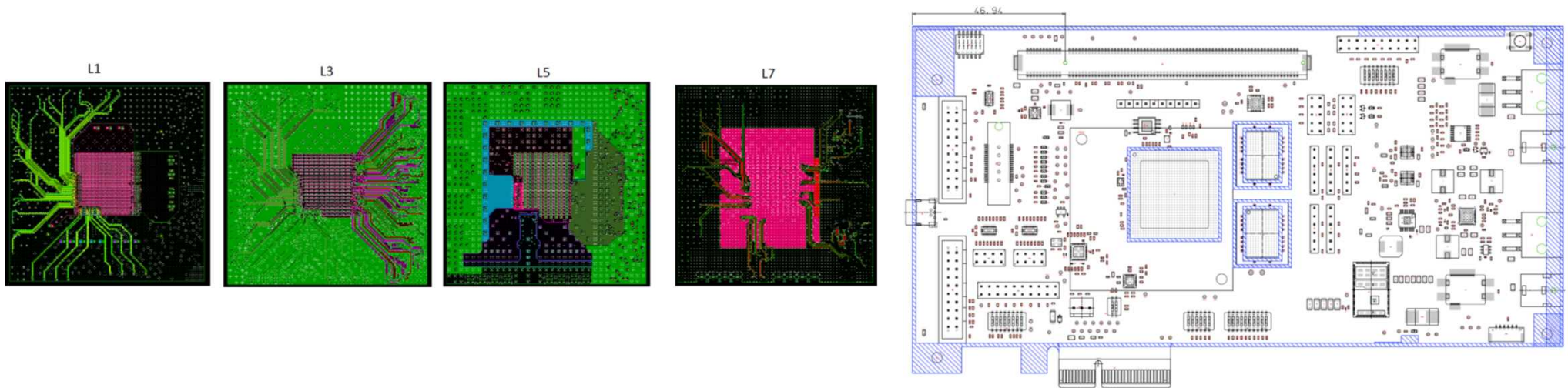
- ・ポストCF論理検証項目の説明と依頼
- ・タイミング検証結果確認項目の説明、依頼
- ・論理変更用ECOフローの説明

ECOファイナル 2021.01

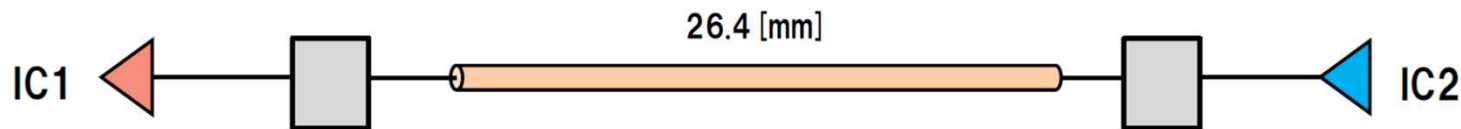
- ・サインオフ検証項目の説明と依頼

4) パッケージ設計、5) ボード設計

- パッケージ設計データはリファレンス設計データとして拠点で保有
LPDDR4, PCIeのパッケージ実装設計は各IPの実装ガイドライン活用
- シグナルインテグリティ、電源インテグリティチェックのためのモデルを使用して
信号、電源特性検証



- 評価ボードはホストPCとの接続性を考えPCIeカードサイズ
- スタンドアロンとしての評価、デモンストレーションも考慮し、設定スイッチ、外部入出力端子を設置
- ボード設計データはレイアウト情報含め拠点のリファレンス設計データに

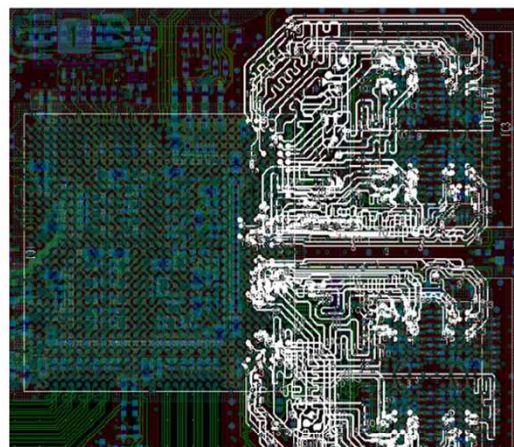


② ODT比較

- 40ohm
- 80ohm
- 240ohm
- ODT OFF

① PullDown比較

- 40ohm ($V_{OH}=V_{DDQ}/3$) (default)
- 60ohm ($V_{OH}=V_{DDQ}/3$)
- 80ohm ($V_{OH}=V_{DDQ}/3$)



① PullDown比較

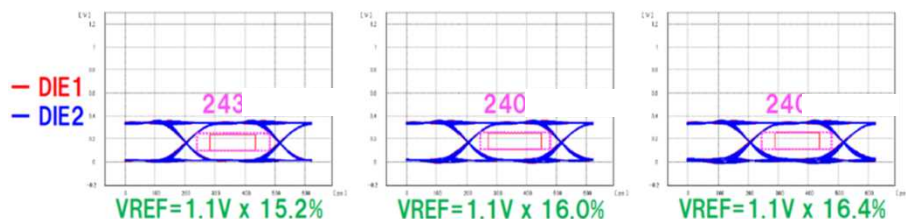
入力信号: PRBS7

IC2: PullDown=40ohm ($V_{OH}=V_{DDQ}/3$) / 60 ($V_{DDQ}/3$) / 80 ($V_{DDQ}/3$)、System ODT=40ohm
 IC1: ODT=40ohm

40ohm ($V_{DDQ}/3$)

60ohm ($V_{DDQ}/3$)

80ohm ($V_{DDQ}/3$)



パッケージの特性情報、PCB基板上の配線設計情報をもとにLPDDR4などの高速インターフェースの信号品質をシミュレーション確認。

関連するすべての信号線をシミュレーションし最適動作ポイントをあらかじめ算出、実機評価前に最適設定を探し出し、ソフトウェアによる初期化プログラムに反映させた。

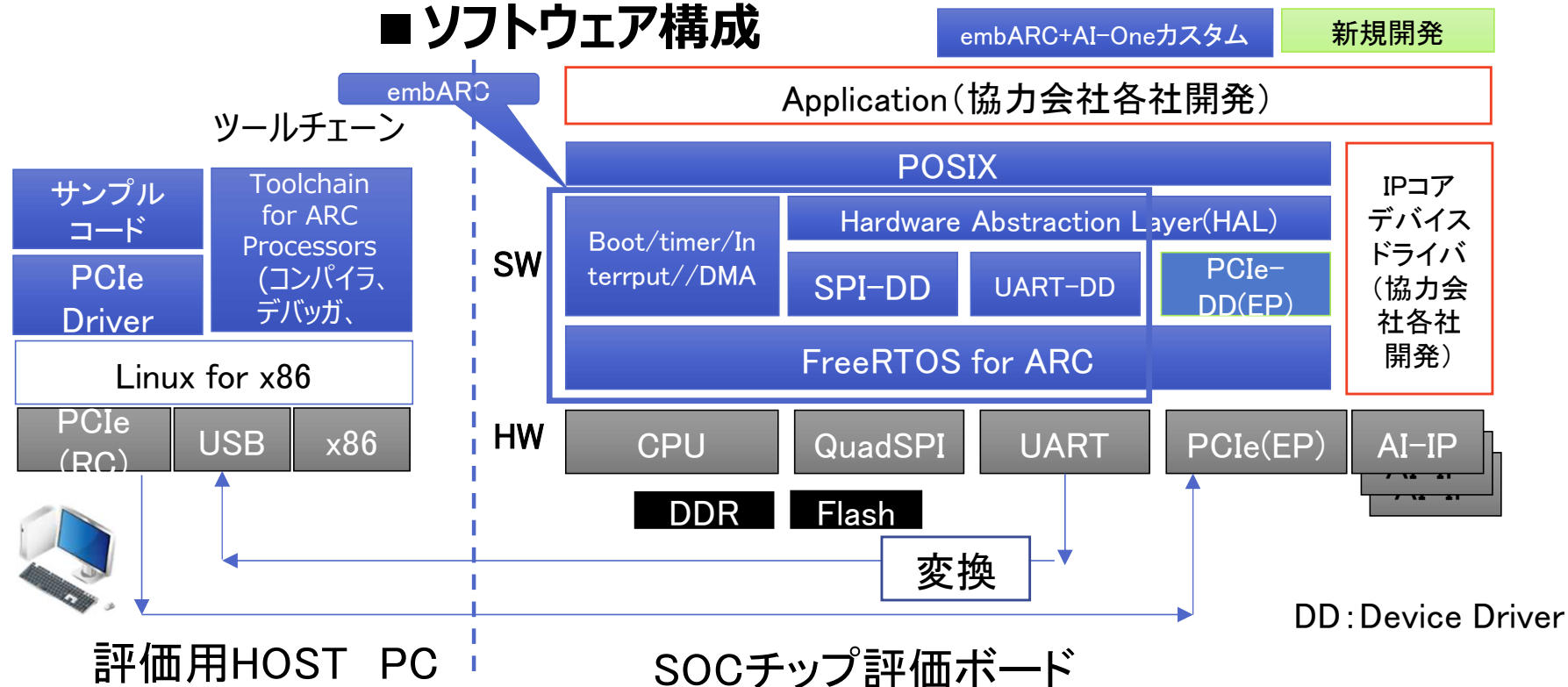
*ODT On Device Termination

6) ソフトウェア開発環境

独自IPコアの性能をシステムレベルで評価するためソフトウェアを開発する環境を整備した

- 独自IPコア向けソフトウェア開発を容易にするためLinux (Ubuntu) の走るHOST PC上でソフトウェア開発を行い、ダウンロードしてデバッグ、評価、デモを行うユースケースを実現可能とする。
- 開発コスト低減やユーザの拡張性、保守性確保のためembARC Open Software Platform (OSP) ベースで開発

■ ソフトウェア構成



苦労した点と対応

- 独自IPコアの搭載に最初想定外に時間がかかった。
- 各独自IP協力会社とオンラインの定例ミーティングを開催し、双方の疑問点、問題点を提示して議論を行い開発の加速を図った。
- プロジェクトの進捗管理ツールとしてRedmineを導入し協力会社との間及びプロジェクト内の案件の管理を行った。
- 協力会社間の独自性保全、営業機密保持のため設計拠点と各協力会社間の情報・データ授受、設計データの保持は完全に分離する管理を行った。設計データへのアクセス権の管理は厳密にボリュームを分離して行った。このためRedmineなどでは、同一の通知を別々のチケットを用いて個々の会社に連絡する場合もあった。

NEDOニュースリリース 2022.03.22

複数のAIアクセラレータを搭載した実証チップ「AI-One」の動作を確認

6種類の異なるAIアクセラレータを1チップに搭載

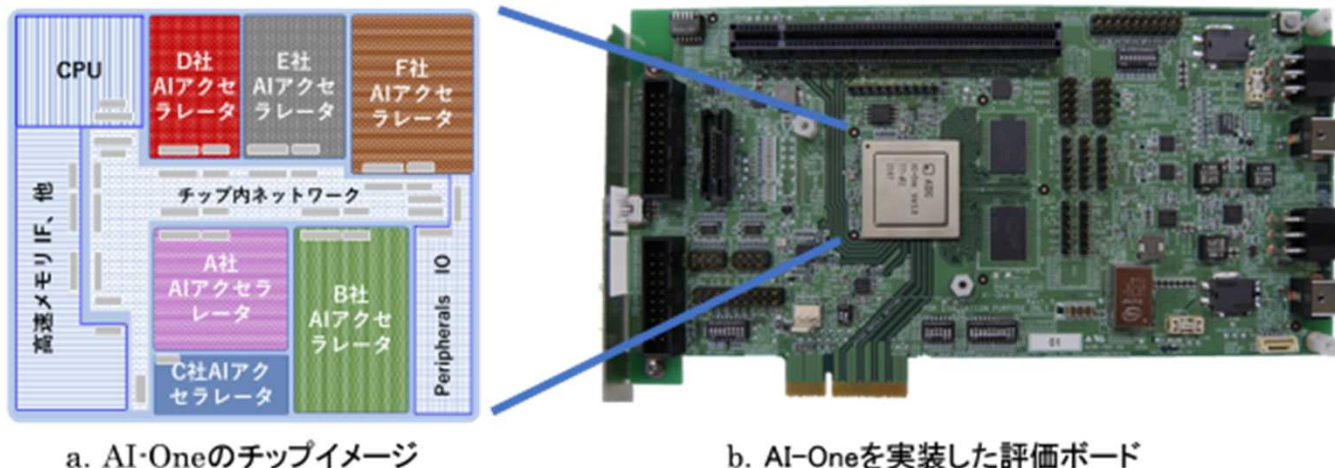
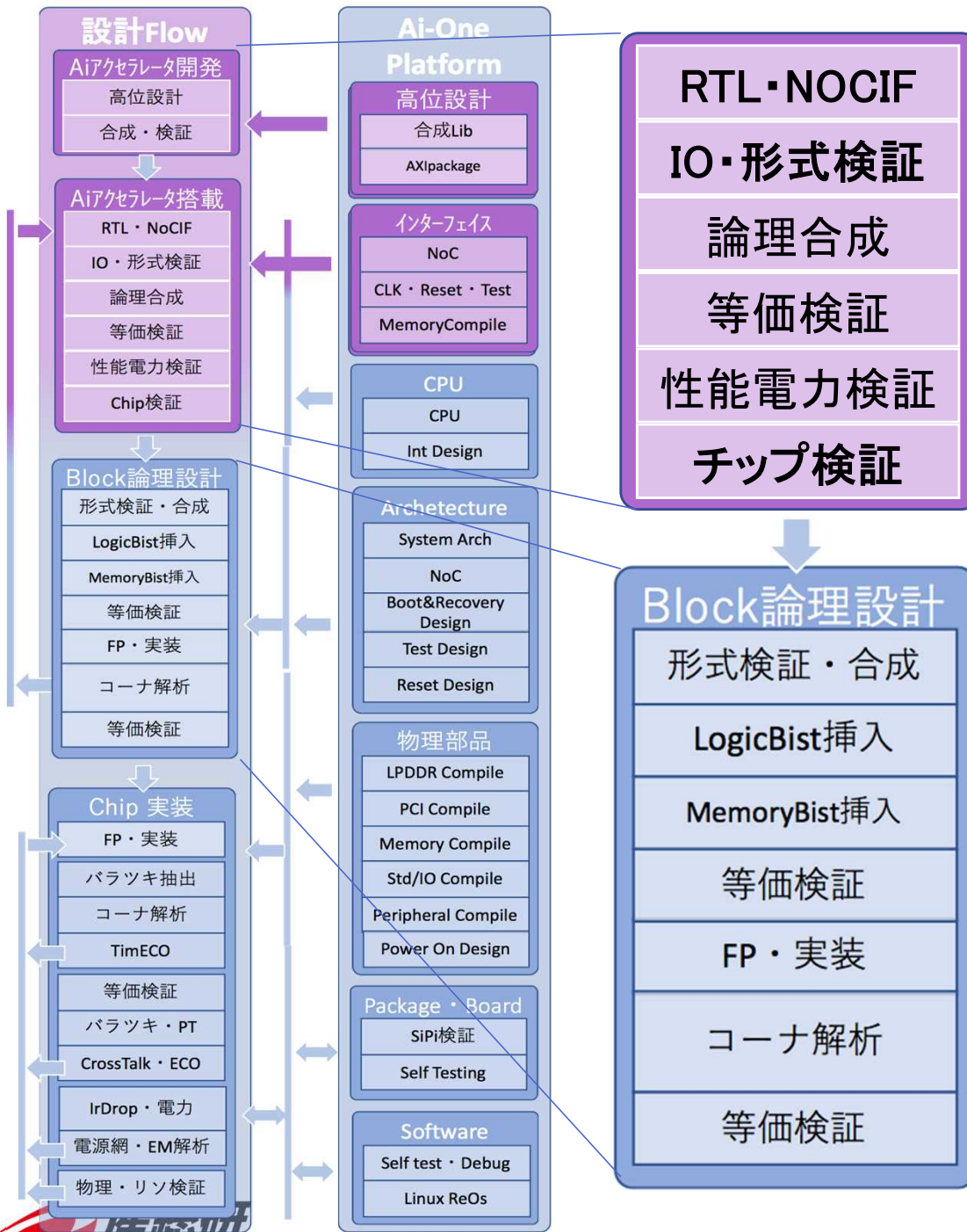


図1 AIアクセラレータ向け評価プラットフォームの実証チップAI-One

https://www.nedo.go.jp/news/press/AA5_101526.html

- Design Automation Conference 2022 サンフランシスコ
- CEATEC 2022 幕張メッセ
- Edge Tech+ 2022 パシフィコ横浜
- Design Solution Forum 2022 武蔵小杉



独自IPコア搭載

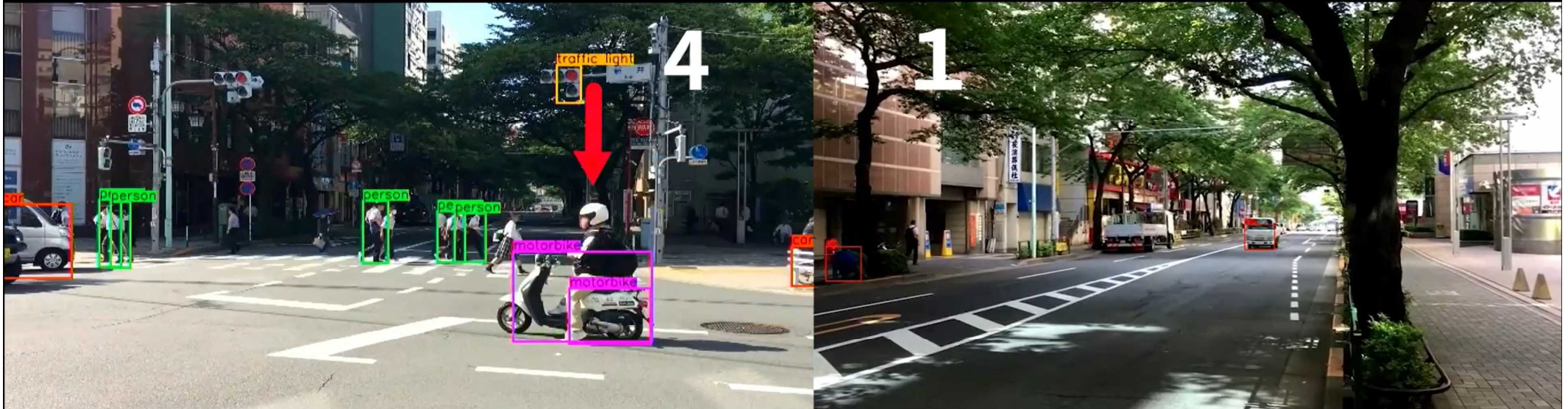


- ①-1の成果を取り込み論理エミュレータ検証による機能および性能検証の高速化 x約100 10KHz → 1MHz

Block論理設計

- フロアプラン (FP)・実装を各IP設計の初期の段階からテスト回路挿入を行わずに実施し早期のうちの問題検出と対策試行の短期化を実現
- 実装データをユーザがEDAツールを使って分析

Ai-One vs FPGA



90~100 fps
~ 3.0 W

処理能力
消費電力

20~30 fps
~ 15 W

Ai-One

Noise Reduction

Before



ノイズが除去されています

▶ 0:39 / 0:40

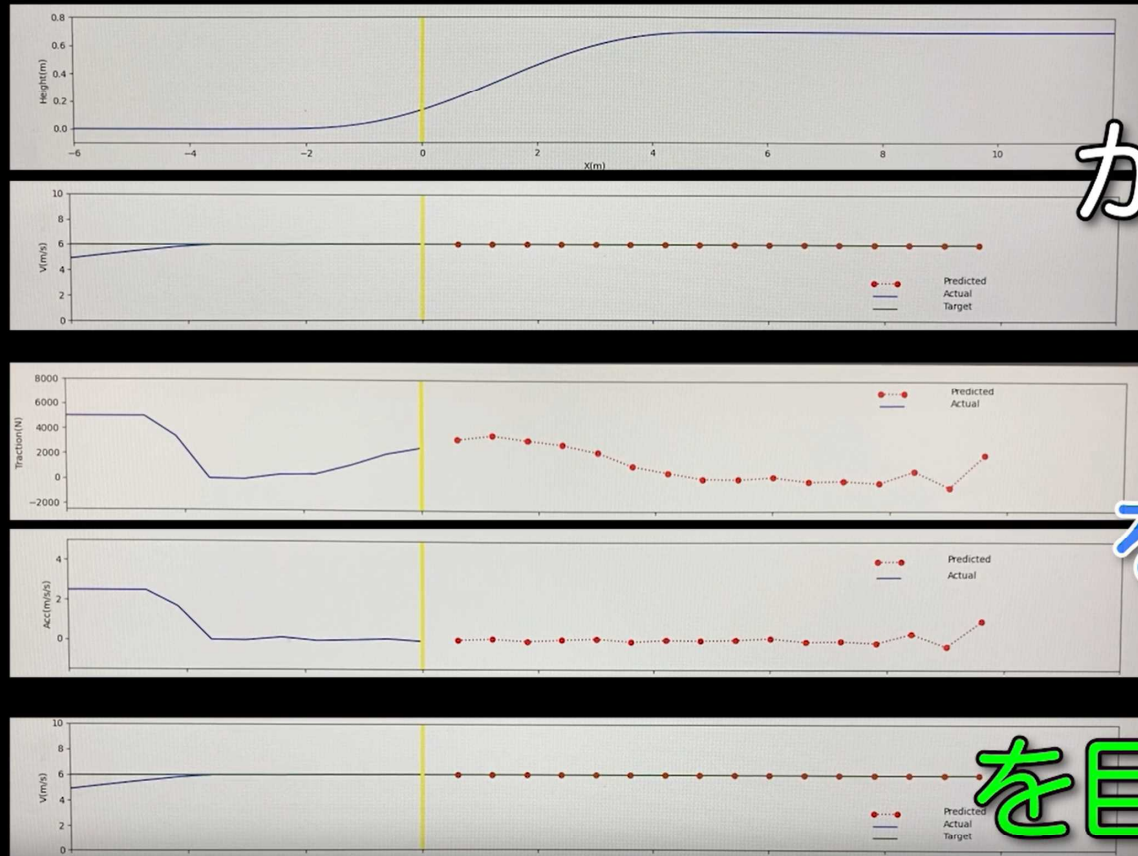
🔊 🔍 ⋮

AI-One搭載速度制御IP

勾配
目標速度

トルク
アクセル

実速度



が与えられ

を制御して

を目標速度に