FSiC (Free Silicon Conf.) 2025に行ってきたよ

広島大学 西澤真一

nishizawa@hiroshima-u.ac.jp

Free Silicon Conf. (FSiC)

FSiC2025
Free Silicon Conference

- オープンソース集積回路設計 (Open-Source Silicon) の会議
  - https://wiki.f-si.org/index.php/FSiC2025
    - 2025/7/2-4，Frankfurt an der Oder (ドイツ)
    - Leibniz-Institut für innovative Mikroelektronik: IHP開催
  - 設計，設計環境，EDA，ファンディング，教育，標準化を議論
    - 基本的には各OSSプロジェクトの中心人物が発表
    - 欧州大学の修士論文の発表も
- 私は開発しているEDAに関して発表を行った
  - キャラクタライザ：セルの電力遅延を抽出するツール

Program (Day 1)

Digital design and logic-synthesis

- 9:10, Matthias Jung (University of Würzburg), *DRAM simulation with the simulator DRAMSys*
- 9:30, Edward Bingham (Broccolimicro), *The potential for asynchronous circuits to bridge the hardware / software divide*
- 9:50, Marcel Walter (TU Munich), *aigverse: Toward machine learning-driven logic synthesis*
- 10:10, Tjark Petersen (DTU), *Towards open-source functional verification methodologies*
- 10:30, Andreas Krall (TU Wien), *OpenVADL: An open source implementation of the Vienna Architecture Description Language*
- 10:50, Kari Hepola (Tampere University), *Kactus2*
- 11:10, Oscar Gustafsson (Linköping University), *B-ASIC: a framework for simulation and implementation of static DSP algorithms*
- 11:30, Joonas Multanen (Tampere University), *OpenASIP: Co-processor co-design using open source tooling*
- 11:50, Jasper Homann and Guillermo Payá Vayá (TU Braunschweig), *Comprehensive functional verification of a configurable N-pipeline-stages RISC-V-based softcore processor using the PATARA framework* (lightning talk)
- **12:00-13:30, lunch break**

On-going FOS silicon projects

- 13:30, Leo Moser (formerly Efabless), *Greyhound: A RISC-V SoC with tightly coupled eFPGA on IHP SG13G2*
- 13:50, Simon Dorrer (Johannes Kepler University (JKU) Linz), *An open-source adaptive event-based ADC for bio-signal acquisition in 130nm CMOS*
- 14:10, Daniel Schultz (ElemRV), *ElemRV - Open source RISC-V microcontroller*
- 14:30, Zachary Kohnen and Alex Alvarado (Eindhoven University of Technology), *Manchester decoder of a home thermostat's wireless protocol in the Tiny Tapeout 07 shuttle* (lightning talk)
- 14:40, Ghaith Al Sabagh (Johannes Kepler University (JKU) Linz), *Open-source radar chip* (lightning talk)
- 14:50, Matthew Venn (YosysHQ), Tiny Tapeout update - demographics, new foundries and going to space

Foundries, PDKs and standard-cell libraries

- 15:10, Sergei Andreev (IHP Microelectronics), *IHP Open PDK: development status updates and looking ahead*
- **15:30-16:00, coffee break**
- 16:00-18:30, cleanroom and labs tour (3 stations, 3 groups, max 20 people/group)
- **18:30-20:00, dinner (barbecue & chips at IHP)**

- デジタル回路設計および論理合成：学生による設計事例紹介が多かった
- Free and Open-Source Siliconプロジェクト紹介
- iHPのセルライブラリ開発報告

Program (Day 2)

AIST Solutions 岡村さん

Foundries, PDKs and standard-cell libraries (part II)

- 9:00, Jun-Ichi Okamura (AIST Solutions), *5W1H: Open-source PDK from the perspective of Japanese foundries aka -other side of the moon-*
- 9:20, Dietmar Warning (IHP Microelectronics), *Verilog-A models in IHP OpenPDK for a modern SiGe RF process*
- 9:40, Marcus Mellor (Infinitymdm), *CharLib: an open-source standard-cell library characterizer*
- 10:00, Shinichi Nishizawa (Hiroshima University), *libretto: An open-source library characterizer for open-source VLSI design*

西澤

- 10:20, Tim Edwards (formerly Efabless, Open Circuit Design), *Update on the Efabless "Frigate" next-generation harness chip, the "Panamax" padframe design, and results from the "Chipalooza"* (lightning talk)
- 10:40, Charlotte Nägle (Bielefeld University), *Open subthreshold standard-cell library for energy-efficient digital circuits* (lightning talk)

Analog flow, transistor modelling and circuit simulation

- 11:00, Arpad Buermen (University of Ljubljana), *Recent developments in the Verilog-A circuit analysis kernel*
- 11:20, Deni Alves (LCI/UFSC & TIMA/Grenoble INP/UGA), *ACM2 – A MOSFET model bridging design and simulation*
- 11:40, Volker Mühlhaus (Mühlhaus Consulting & Software GmbH), *User friendly workflow for RFIC EM simulation using openEMS*
- **12:00-13:30, lunch break**
- 13:30, Martin Köhler (Johannes Kepler University (JKU) Linz), *KLayout-PEX – Parasitic Extraction Tool for KLayout*
- 13:50, Felix Salfelder (Gnucap MixedSignals), *Progress on Verilog-AMS support in Gnucap*
- 14:10, Leo Moser (formerly Efabless), *Update on CACE* (lightning talk)
- 14:20, Tobias Kaiser (TU Berlin), *ORDeC: A text-driven analog IC design platform*
- 14:40, Frans Skarman (Linköping University and Munich University of Applied Sciences), *Surfer - an extensible and snappy waveform viewer*
- 15:00, Holger Vogt (University Duisburg), *ngspice - status update, and degradation simulation*

Hardware security

- 15:20, Sebastian Haas (Barkhausen Institute), *A secure hardware to enable trustworthy computing*
- **15:40-16:10, coffee break**
- 16:10, Christian Schulze (Agentur für Innovation in der Cybersicherheit GmbH (Cyberagentur)), *Ecosystem verifably secure IT--provable cybersecurity*
- 16:30, Simon Klix (Max Planck Institute for Security and Privacy), *An introduction to hardware reverse engineering and HAL*

Economic sustainability and hardware licences

- 16:50, Melanie Rieback (RadicallyOpenSecurity), *Steward ownership and open silicon*
- 17:10, Joaquin Matres Abril, Troy Tamas, Sebastian Goeldi, Floris Laporte, Jan David Fischbach and Matthew Mckee (GDSFactory), *An open core model for EDA*
- 17:30, Michael Weinberg (OSHWA), Martin Häuer (Open Source Imaging Initiative e.V., Martin-Luther Universität Halle-Wittenberg), *Panel: The current landscape of free hardware — licensing and beyond.* Moderator: Panos Alevropoulos (FSF volunteer)
- 18:00-19:00, shuttle to the city from IHP
- 19:00-20:00, city tour
- **20:00, dinner at Taverna Athos** (at own expense)

- ファウンドリ，PDK，ライブラリ
- アナログ回路設計環境
- ハードウエアセキュリティ
- 経済的持続性，ハードウエアライセンス

Program (Day 3)

Policy, EU projects and funding opportunities

- 9:00, Krzysztof Herman (IHP Microelectronics), *One year of experience with IHP OpenMPW shuttles: a review*
- 9:20, Tina Tauchnitz (VDI/VDE-IT), *Update on German Microelectronics Design Initiative*
- 9:40, Norbert Herfurth (IHP Microelectronics), *Towards industrial-grade designs with open-source EDA: The DI-FLOWSPACE and DI-SIGN-HEP Approach*
- 10:00, Gerhard Kahmen (Scientific director at IHP Microelectronics), *acatech IMPULS study: Open-source design tools for sovereign chip development (DI-QDISC)* (lightning talk)

Back-end design tools

- 10:10, Tim Edwards (formerly Efabless, Open Circuit Design), *IHP Open PDK integration with Magic, Netgen, and LibreLane*
- 10:30, Leo Moser, Mohamed Gaber (formerly Efabless), *LibreLane: Looking to the future*
- 10:50, Philippe Sauter and Thomas Benz (ETH Zurich), *An open-source power simulation flow using OpenROAD*
- 11:10, Joaquin Matres Abril, Troy Tamas, Sebastian Goeldi, Floris Laporte, Jan David Fischbach and Matthew Mckee (GDSFactory), *GDSFactory+, the all-in-one solution for chip design*
- 11:30, Philippe Sauter and Thomas Benz (ETH Zurich), *ArtistIC: An open-source toolchain for top-metal IC art and ultra-high-fidelity GDSII renders*
- 11:50, Peter Thoma (Frankfurt University of Applied Sciences), *OpenTwin: An open-source framework for workflow orchestration with enhanced simulation and data management*
- 12:10, Noam Cohen (KeplerTech.io), *Simplifying and accelerating EDA tools development with the NajaEDA Python library*
- 12:30, Andrew Kahng (OpenROAD Initiative), *The OpenROAD project: status and paths forward*
- **13:00-13:30, lunch break** (lunch bags to go)

Teaching and education

- 13:30, Zihao Yu and Xiaoke Su (Institute of Computing Technology, Chinese Academy of Sciences), *"One Student One Chip" initiative: Learn to build RISC-V chips from scratch with MOOC*
- 13:50, Xueyan Zhao, Hao Wang and Yuchi Miao (Institute of Computing Technology, Chinese Academy of Sciences), *Making Open Silicon Design Everywhere: Using Cloud-based Open Agile EDA Platform*

Standards

- 14:10, Dzmitry Pustakhod (Eindhoven University of Technology (TU/e)), R. Broeke, X. Leijtens, J. Matres Abril, P. Dumon, A. Schoenau, O. Abdeen, Y.D. Gupta, S. Latkowski, *Photonic PDKs using openEPDA open standards: From foundry data to EPDA tools*
- 14:30, Norbert Herfurth, Arnd Weber, Steffen Reith (IHP Microelectronics), *The Transparent Reference Fab: A scalable, open blueprint for European semiconductor sovereignty*
- 14:50, Christophe Alexandre (Naja), *Open Source interchange format and data structures*
- 15:10, Matthias Köfferlein (KLayout), *It's time to reinvent the wheel - a plead for disruptive changes in the file format department*
- 15:30, conclusions

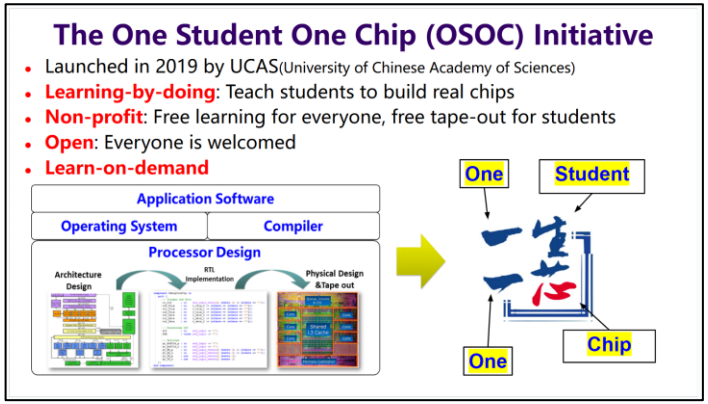
- EUでのプロジェクトおよびファンディング
- バックエンド(物理設計)EDA
- 教育
- 標準化

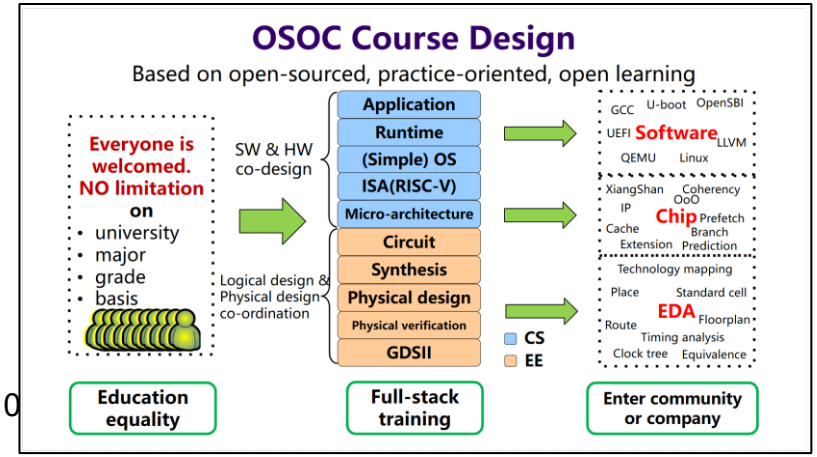
efabless閉鎖

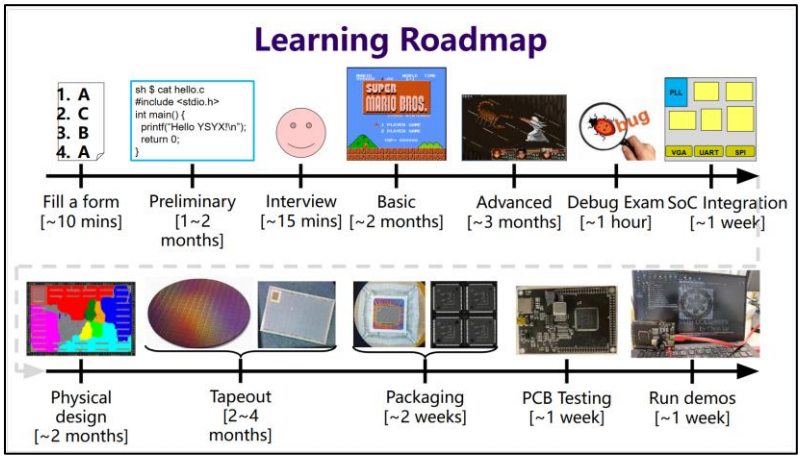
- 3月に(米)efablessが閉鎖されたことも大きい話題に
- Google+Skywater+efablessに代わり，欧州がOSSを牽引する決意を表明するなど
  - efablessが開発したOpenLaneの後継としてコミュニティーベースのLibreLane(OpenLane 2)が開発されたことを知った

中国的回路人材育成：一生一芯プロジェクト

- 学部各個人でRISC-Vプロセッサを作るプロジェクト
  - SW/HWコデザイン+チップの物理実装まで
  - チップは製造し，PCB実装後発送，オンラインで発表
  - 教材，TA，全てオンライン，12000アカウント@2025

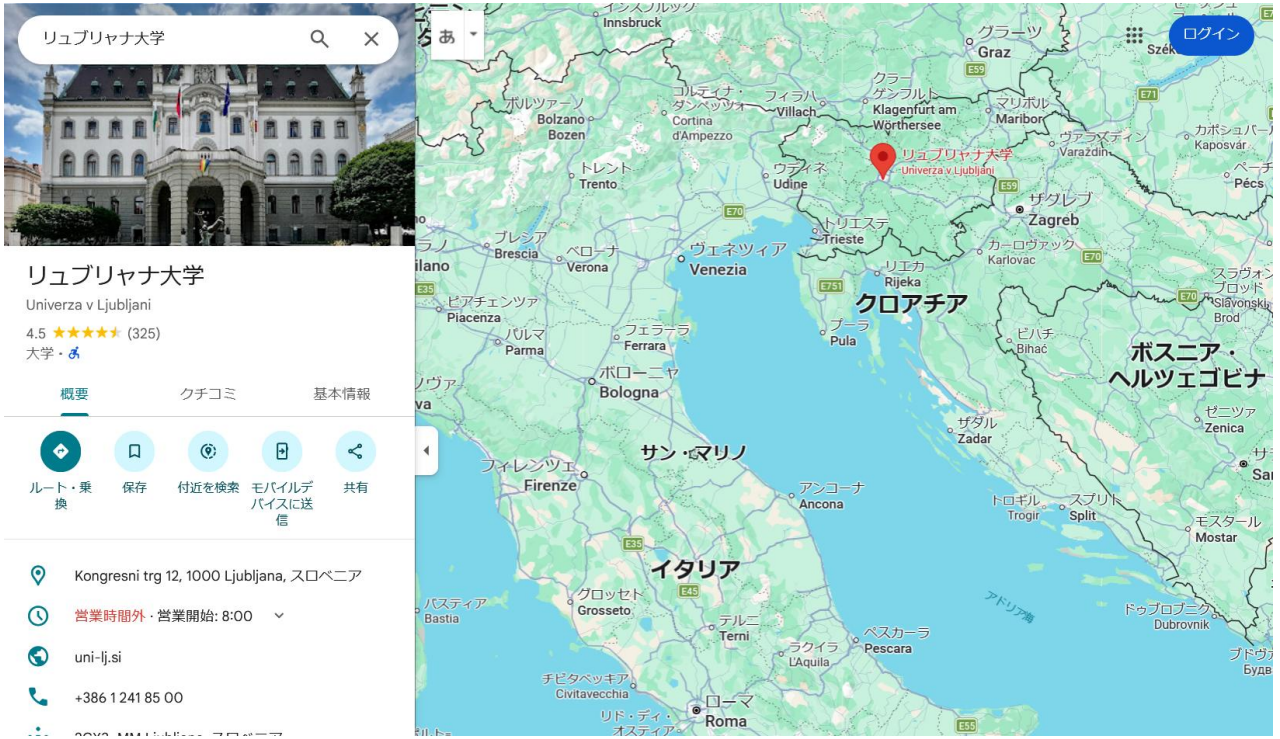






FSiC2026

- FSiC2026はUniversity of Ljubljana(スロバキア)で開催
  - 投稿は基本的に自薦，4/1までにショートサマリ投稿
  - https://wiki.f-si.org/index.php?title=FSiC2026
  - アジアからの発表が少ないので投稿すると喜ばれる(かも)
- FSiC2025での発表資料に続く



libretto: An open-source library characterizer for open-source VLSI design

Shinichi Nishizawa, Hiroshima Univ., nishizawa@hiroshima-u.ac.jp

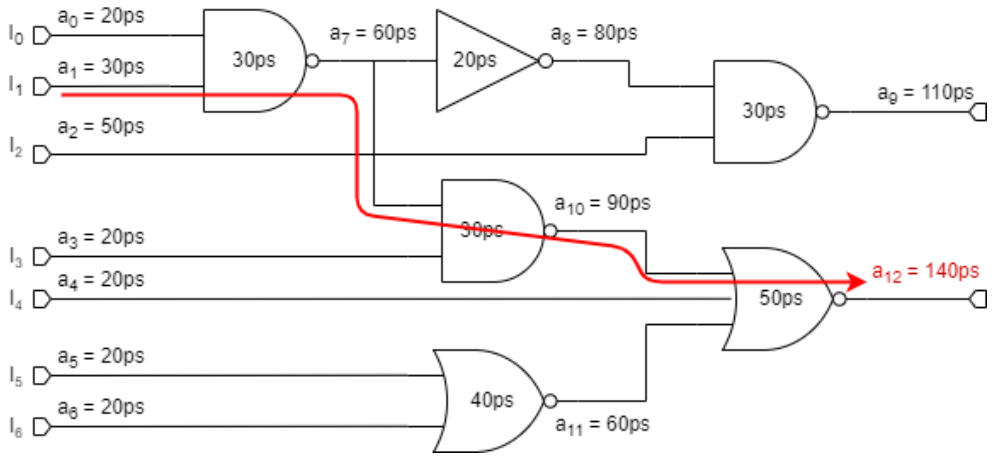
広島大学

ISHI-Kai (Japan)

Open-Source EDA supporters (Japan)

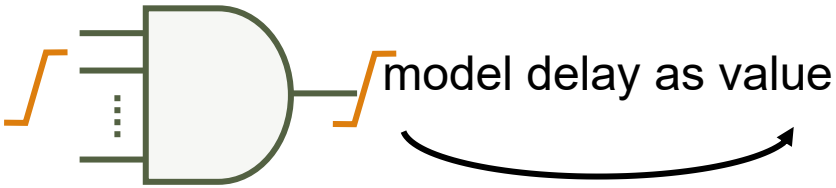
Digital circuit design and STA

- Static Timing Analysis (STA): estimation of path delay
  - Calculate max/min path delay integrating the cells delay
    - Max delay of i-th node: : $a_i = \max_{j \in \mathrm{fanin}(i)}\{a_j\} + t_{\mathrm{pd},i}$
  - Need both timing information (.lib) calculation engine (STA)
    - STA: OpenSTA (Open-ROAD), sta (yosis)
    - .lib (Liberty): By characterizer



Characterizer

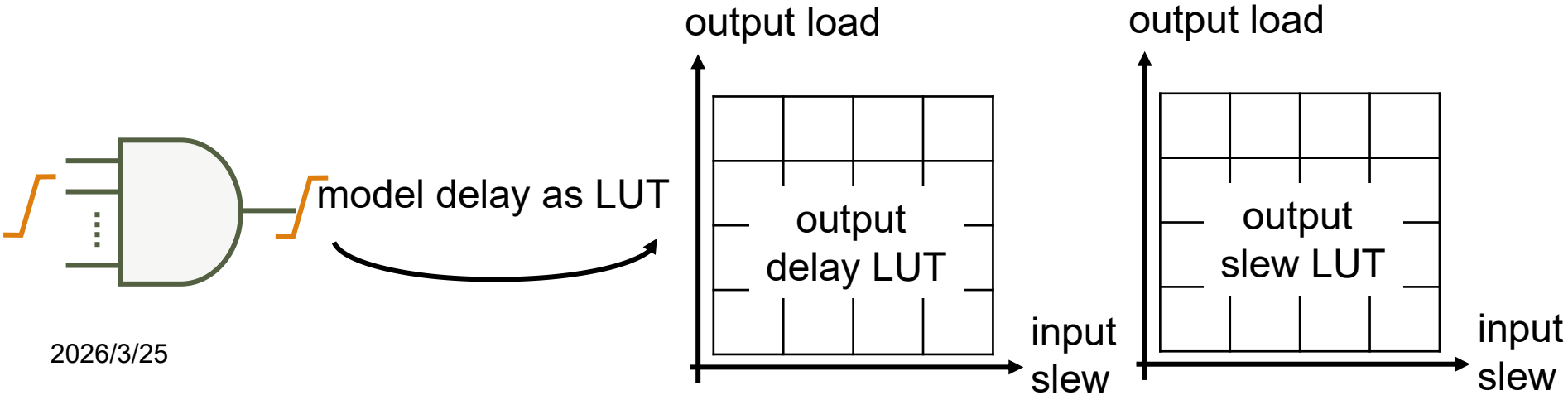
- Simulate and extract delay and power of std. cell
  - Prop delay: t_{pd} = xx ps. Trans delay: t_{td} = xx ps
    - Arrival time: $a_i = \max_{j \in \mathrm{fanin}(i)}\{a_j\} + t_{\mathrm{pd},i}$



Prop delay: t_{pd} = xx ps.
Trans delay: t_{td} = xx ps

Characterizer

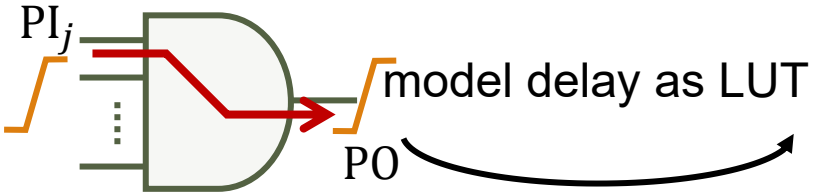
- Simulate and extract delay and power of std. cell
  - Prop delay: t_{pd} = xx ps. Trans delay: t_{td} = xx ps
    - ✗ Arrival time: $a_i = \max_{j \in \mathrm{fanin}(i)}\{a_j\} + t_{\mathrm{pd},i}$
- Delay and energy are the function of input slope, output load
  - Arrival time: $a_i = \max_{j \in \mathrm{fanin}(i)}\{a_j\} + t_{\mathrm{pd},i}(t_{\mathrm{pd,PI}_j}, C_{\mathrm{load},i})$

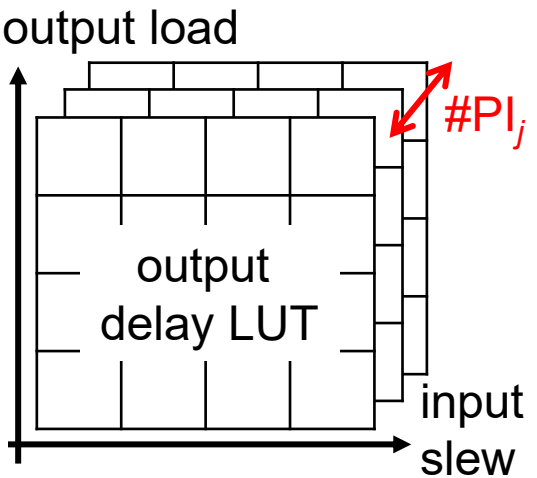


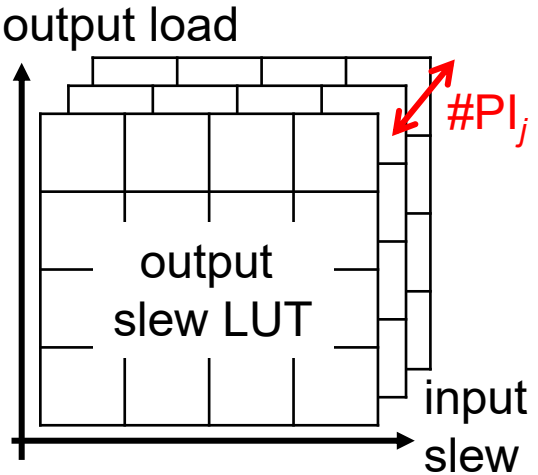
Characterizer

- Simulate and extract delay and power of std. cell
  - Prop delay: t_{pd} = xx ps. Trans delay: t_{td} = xx ps
    - ✗ Arrival time: $a_i = \max_{j \in \mathrm{fanin}(i)}\{a_j\} + t_{\mathrm{pd},i}$
- Delay and energy are the function of input slope, output load
  - ✗ Arrival time: $a_i = \max_{j \in \mathrm{fanin}(i)}\{a_j\} + t_{\mathrm{pd},i}(t_{\mathrm{pd,PI}_j}, C_{\mathrm{load},i})$
- Each primary input (PI) has different delay and energy
  - Arrival time: $a_i = \max_{j \in \mathrm{fanin}(i)}\{a_j\} + t_{\mathrm{pd},i,\mathrm{PI}_j}(t_{\mathrm{pd,PI}_j}, C_{\mathrm{load},i})$

Characterizer automates simulation & lib. creation







Related works

- Open-source characterizers w/ free SPICE are available
  - LibreCell (lctime) [1]：Widely used, well known
    - Uses “template .lib” to specify char. condition
    - Pro. tool (need to read/write .lib by engineer)
  - CharLib [2]：
    - Uses YAML to specify characterize condition
    - Needs PySpice backend
    - Supports multithread
  - And many open characterizers that uses commercial SPICE

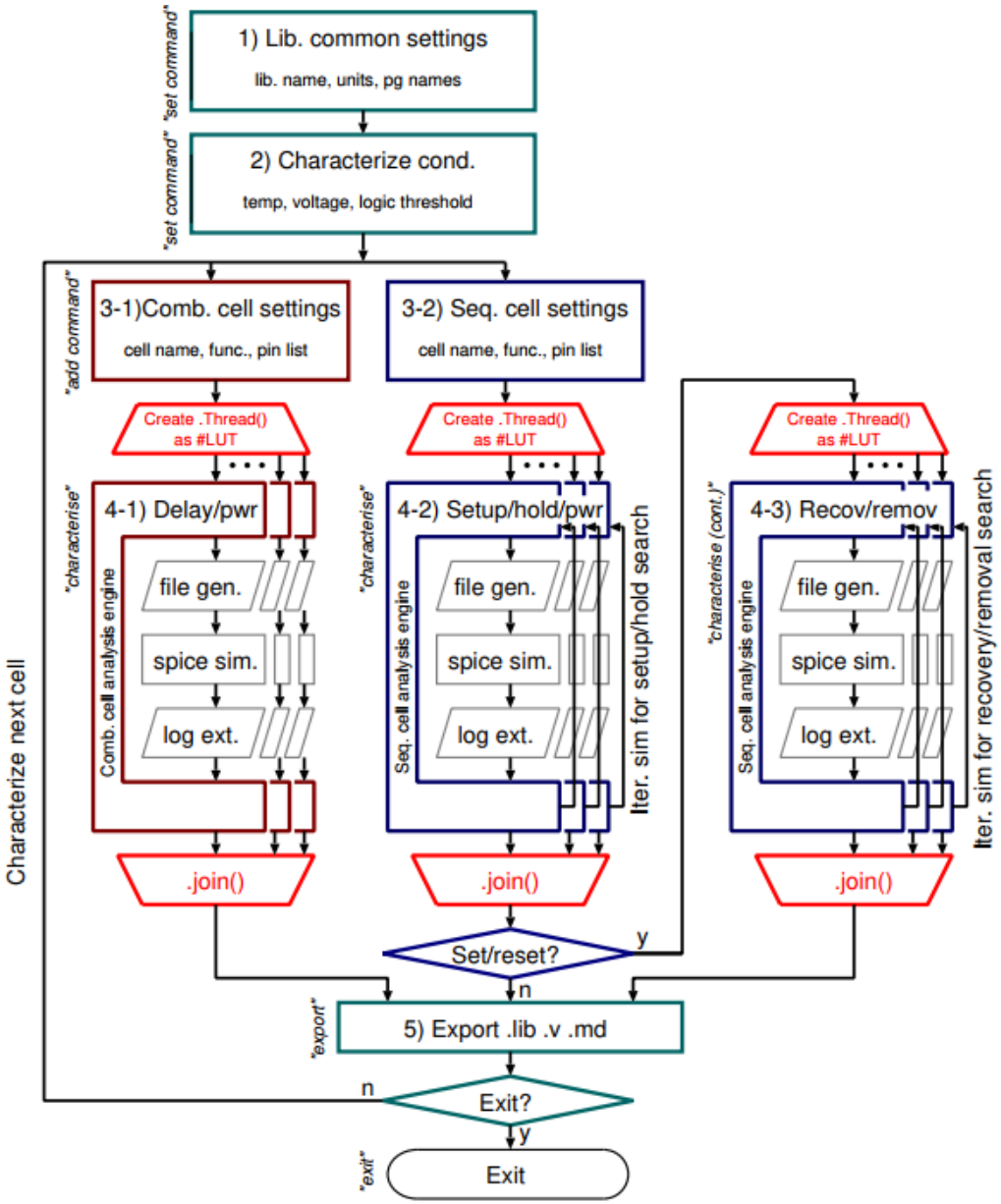
[1] T. Kramer, “lctime.” https://codeberg.org/librecell/lctime
[2] J. E. S. Jr., “CharLib.” https://github.com/stineje/CharLib.

Proposed characterizer (libretto)

- Open-source characterizer
  - Language: Python3
  - Simulator: ngspice, hspice (for comparison)
- Advantage
  - Characterize both combinational and sequential cells
    - Support Flip-Flops w/ pos/neg clock and async. set/reset
  - Easy to add functions
    - Two analysis engines: for combinational and sequential
      - Do not prepare engines for each logic function
      - Use truth table to handle different logic functions
  - Nothing new. No advantage over commercial tools
    - But open and free

Operation flow

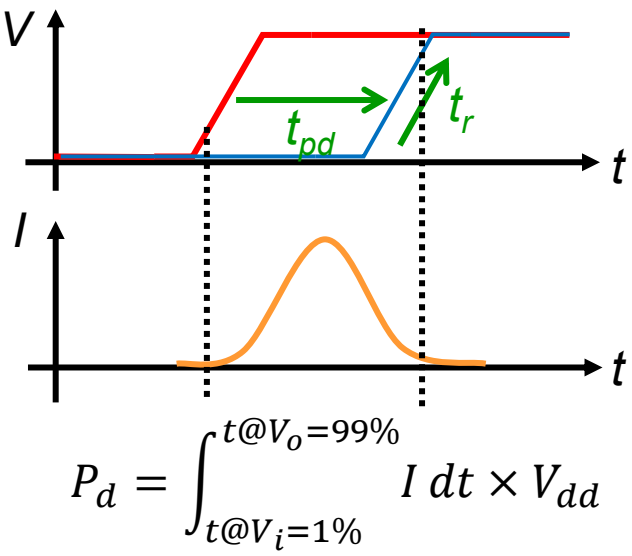
- Setup library and cells
- Branch to comb. or seq.
  - Generates test bench based on the target logic function
  - Launch simulator
    - Multiple slew/load cond. in multi-thread
- Seq. cell need iterations
  - Find min. delay by setup/hold search



Operation flow.

Characterize: combinational cell

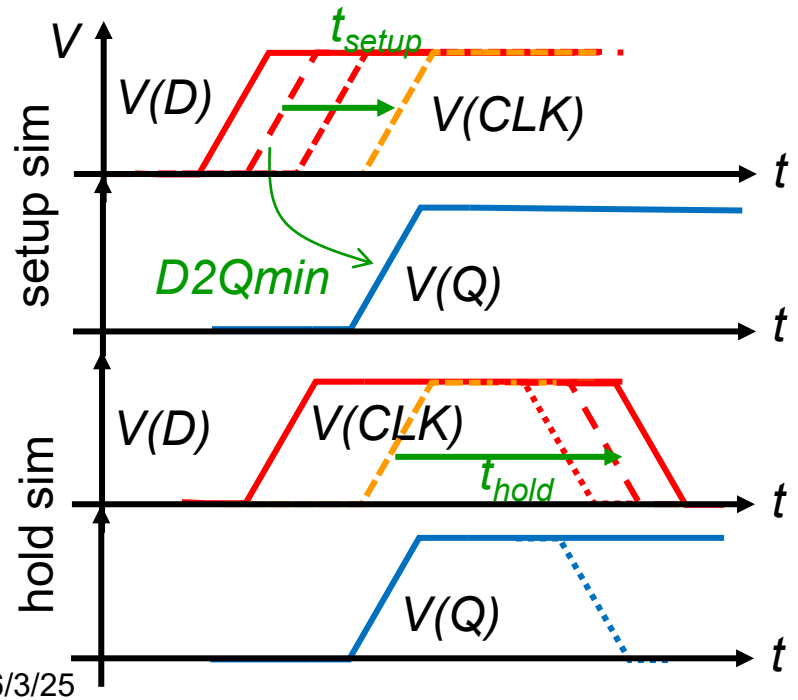
- Propagation delay: input 50% to output 50%
- Transition delay: output 20% to output 80%
- Dynamic power[*1] : integrate current from input start 1% to output end 99%
- Static power : integrate current at the beginning of sim[*2]
- Simulation times: time_step, end_time
  - Set by designer, or use “auto”

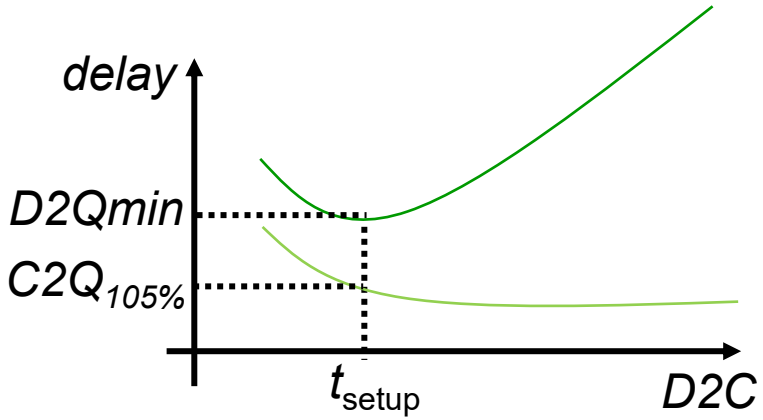


* Parameters can be changed by designer.
*1: This should be converted to energy
*2: Does not meas. input dependency

Characterize: sequential cell delay

- C2Q delay, t_{setup}, t_{hold} : calc. by min. D2Q delay
  - Change D2C, find min. D2Q
    - D2C = t_{setup}, C2Q = D2Q – D2C
  - Set D2C = t_{setup}, change C2D and find min C2D
    - Min. C2D = t_{hold}





Two def. for setup
1. D2C when C2Q increase 3~5%
2. D2C when D2Q is minimum

libretto use def. of 2.

Library setting

- Common setting for lib.
  - Library name
  - Prefix, suffix of cells
  - Units (volt, current, cap.)
  - Power name
  - Temperature
  - Voltage
  - Logical threshold
  - High/low threshold
  - Operation directory
  - Simulator
  - …

```
# common settings for library
set_lib_name          ROHM180
set_dotlib_name       ROHM180.lib
set_verilog_name      ROHM180.v
set_cell_name_suffix ROHM180_
set_cell_name_prefix _V1
set_voltage_unit V
set_capacitance_unit pF
set_resistance_unit Ohm
set_current_unit mA
set_leakage_power_unit pW
set_energy_unit fJ
set_time_unit ns
set_vdd_name VDD
set_vss_name VSS
set_pwell_name VPW
set_nwell_name VNW
```

```
# characterization conditions
set_process typ
set_temperature 25
set_vdd_voltage 1.8
set_vss_voltage 0
set_pwell_voltage 0
set_nwell_voltage 1.8
set_logic_threshold_high 0.8
set_logic_threshold_low 0.2
set_logic_high_to_low_threshold 0.5
set_logic_low_to_high_threshold 0.5
set_work_dir work
set_simulator /usr/local/bin/ngspice
set_run_sim true
set_mt_sim true
set_supress_message false
set_supress_sim_message false
set_supress_debug_message true
set_energy_meas_low_threshold 0.01
set_energy_meas_high_threshold 0.99
set_energy_meas_time_extent 10
set_operating_conditions PVT_3P5V_25C
```

Setting for each cell

- Characterize conditions
  - Add cell
  - Input slope (in array)
  - Output load (in array)
  - Netlist
  - Timestep*. sim. end*
- Flip-Flop needs
  - Clock slope*
  - Setup unit time*
  - Hold unit time*

```
## add circuit
add_cell -n ROHM18INVP010 -l INV -i A -o Y -f Y=!A
add_slope {0.1 0.7 4.9}
add_load  {0.01 0.1 1.0}
add_area 1
add_netlist rohmlib/ROHM18INVP010.sp
add_model rohmlib/model_rohm180.sp
add_simulation_timestep auto
characterize
export
```

Inverter

```
## add circuit
add_flop -n ROHM18DFP010 -l DFF_PCPU -i DATA -c CLK
-o Q -q Q QN -f Q=IQ QN=IQN
add_slope {0.1 0.7 4.9}
add_load  {0.01 0.1 1.0}
add_clock_slope auto
add_area 1
add_netlist rohmlib/ROHM18DFP010.sp
add_model rohmlib/model_rohm180.sp
add_simulation_timestep auto
add_simulation_setup_auto
add_simulation_hold_auto
characterize
export
```

Flip-Flop

Command example: logic function, in/out pins, storage, logic expression, slew/cap index, simulation time step

* Parameters can use auto set

Netlist gen. and simulation (comb. cell)

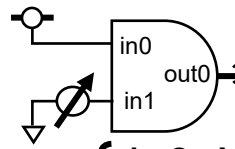
- *characterizeFiles*() : def. of logic func., its truth table
- *runCombInnOutm*() : set input/output pin setting
- *runSpiceCombDelay*() : generate netlist, run spice

```
292 def characterizeFiles(targetLib, targetCell):
293   print ("characterize\n")
294   os.chdir(targetLib.work_dir)
295             ...
308
309   elif(targetCell.logic == 'AND2'):
310     print ("AND2\n")
311                       ## [in0, in1, out0]
312     expectationList2 = [['01','1','01'],
313                         ['10','1','10'],
314                         ['1','01','01'],
315                         ['1','10','10']]
316     return runCombIn2Out1(targetLib, targetCell, expectationList2,"pos")
317
```

Truth table of AND2
in0=1
in1=1→0
out0=1→0

characterizeFiles()
logic: INV, AND2, …

*rumCombIn**2**Out**1***()



…

set { in0: V_{DD}
in1: PWL
out0: .meas

runSpiceCombDelay()
Launch sim. accum. res.

Use a function for netlist gen., spice run, analysis (*runSpiceCombDelay*())

- Pin is analyzed, connected proper voltage sources and spice measure statements

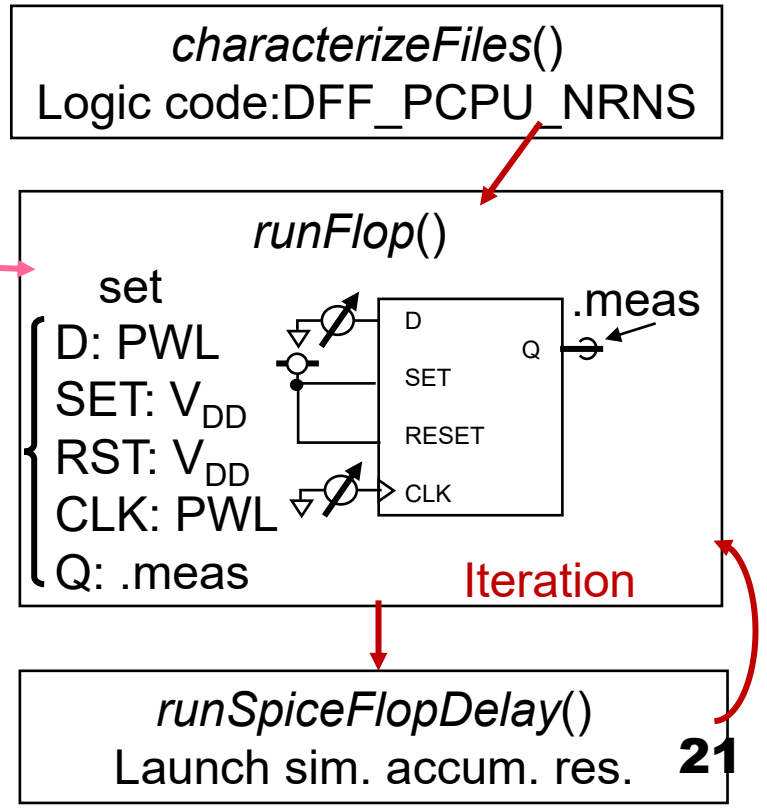
Netlist gen. and simulation (seq. cell)

- *characterizeFiles*() : def. of logic func., its truth table
- *runFlop*() : set input/output pin setting
- *runSpiceFlopDelay*() : generate netlist, run spice

```
elif(targetCell.logic == 'DFF_PCPU_NRNS'):
  print ("DFF, positive clock, positive unate, async neg-re
  ## D1 & C01 -> Q01 QN10
  ## D0 & C01 -> Q10 QN01
  ## S01      -> Q01 QN10
  ## R01      -> Q10 QN01
  ##                  [D,   C,  S,   R,    Q]
  expectationList2 = [['01','0101','1', '1', '01'],\
                      ['10','0101','1', '1', '10'],\
                      ['0','0101','01', '1', '01'],\
                      ['1','0101', '1','01', '10']]

  ## run spice deck for flop
  return runFlop(targetLib, targetCell, expectationList2)
```

Truth table of code: DFF_PCPU_NRNS
(pos. clk, pos. unate, neg. rst, neg. set)
D=0→1, SET=1, RST=1
CLK= 0→1 → 0→1
Q=0→1



Registered logic family

- Support simple logic functions and several Flip-Flops

Family	Logic function
Inv/Buf	Inverter, Buffer
NAND	NAND2, NAND3, NAND4
NOR	NOR2, NOR3, NOR4
AND	AND2, AND3, AND4
OR	OR2, OR3, OR4
And-Or-Inv.	AOI21, AOI22
Or-And-Inv.	OAI21, OAI22
Exclusive	XOR2, XNOR2
Selector	SEL2

DFF code	clk	porality*1	set	rst
DFF_PCPU *2	pos.	pos.		
DFF_PCNU	pos.	neg.		
DFF_NCPU	pos.	neg.		
DFF_NCNU	neg.	neg.		
DFF_PCPU_NR	pos.	pos.		neg.
DFF_PCPU_NRNS	pos.	pos.	neg.	neg.

*1: Pos.: in/out are same direction (H/H,L/L). neg. in/out are opposite (H/L,L/H)
*2: D-Flip-Flop w/ pos. clock edge, positive polarity

Evaluation setup

Cent OS 7.8
Ryzen 2990wx 3GHz 32core
MEM 96GB, SSD 3TB

- Use commercial 180-nm for evaluation

	Proposed		**PrimeLib**	
Simulator	ngspice		hspice	
#parallel	1	49 (#index)	32	64
Condition	Typical (TT, 1.8V, 25℃)			
Slew (ns)	0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4			
Capacitance (pF)	0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64			
Device Under Test	INVx1, NAND2x1, NAND3x1, NAND4x1 NOR2x1, NOR3x1, NOR4x1, DFF (posedge)			
Runtime	57.9 h (1x)	4.00 h (**14.5x**)	92.0 s (2265x)	106 s (1966x)

- Large runtime: need more parallelism, poor search algorithm.
- Separated timing and power sim. is also issue (ngspice do not support nesting of .meas)

Result

- No difference in simulator (ngspice vs HSPICE)
- Two characterizers show different result (PrimeLib vs libreto)
  - Delay of comb. cells might be acceptable (Max error in prop.: 0.5%, trans: 24%)
  - Intl. energy has large error: (Max 418%)
  - Seq. has problem (C2Q delay: 1125%, 2407% pessimistic)
    - Setup/hold interdependence (?)

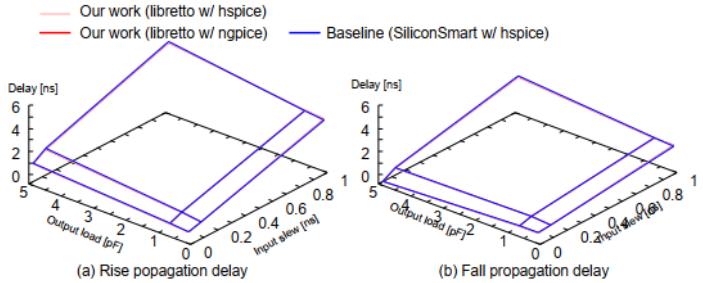


Fig. 6: Propagation delay of Inverter.

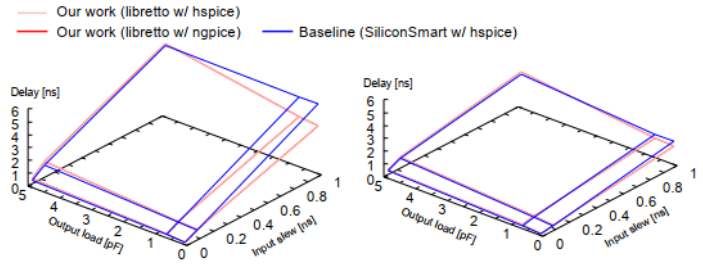


Fig. 7: Transition delay of Inverter.

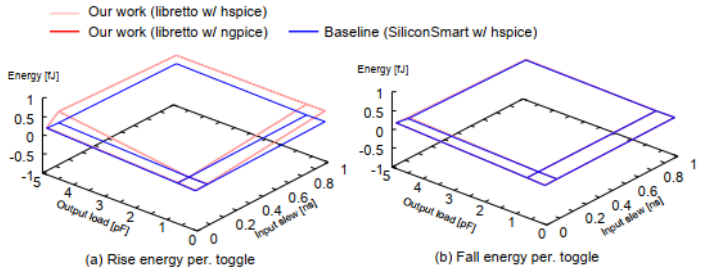


Fig. 8: Internal energy of Inverter.

TABLE II: Capacitance and leakage power of Inverter.

Characterizer	**libretto**		SiliconSmart
Simulator	ngspice		hspice
Leakage power [pW]	9.862	9.862	9.862
Input capacitance [fF]	4.120	4.063	4.599

Conclusion

- Open characterizer w/ open simulator
  - Generates timing/power library as .lib
  - Used for timing analysis (simulation, STA)
- Supports both combinational and sequential cells
- Evaluate delay, energy, and performance
  - Slow processing speed (1/215),
  - Combinational cell: delay and energy acceptable (?)
  - Sequential cell: large gap
- Checked by LibraryCompiler Synopsys
- Users: 1 in Japan, 1 in Itally
  - Let me know if you use libretto (motivate me!)

- 以上