LaDS
Laboratory of Dependable Systems

設計驗證工作室

Design Verification Team

黃鐘揚 教授

Prof. Chung-Yang (Ric) Huang

Department of Electrical Engineering

National Taiwan University

2007/11/16

Lab Profile

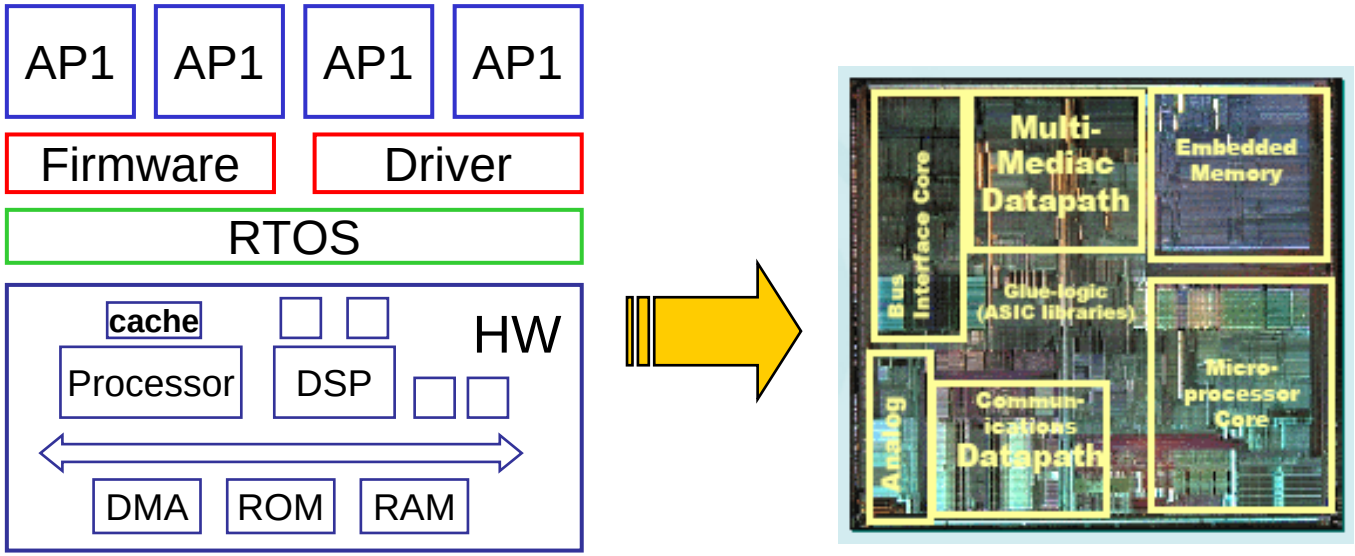
- Founded 2004.02
- Lab info
  - Office: EE-II 444
  - Lab: BL 627 (即將搬至 EE 353)
  - PTT: NTUGIEE_ric
  - Website: http://dvlab.ee.ntu.edu.tw/
  - To find me…
    - ric@cc.ee.ntu.edu.tw
    - ric2k1 @ ptt, ptt2, msn, skype, …
    - 02-3366-3644

3 Major Research Directions

1. SoC design methodology and verification
   - 國科會國家型 SoC 計畫
   - 思源科技 (SpringSoft) 產學合作 (1)
   - 經濟部科專計畫 (已結案)
   - Other cooperation: 工研院 (ITRI)…
2. Formal-assisted design optimization and debugging
   - 國科會兆級晶片系統計劃
   - 思源科技 (SpringSoft) 產學合作 (2)
   - 國科會個人型計畫 (已結案)
   - 友達光電 (AUO) 產學合作
3. Next generation constraint satisfiability and optimization engine
   - 益華電腦 (Cadence) 經濟部科專計畫 (expected starting Q4 2007)
   - 國科會計畫 (申請中)
   - Other cooperation: 中研院…

(1) SoC Design Methodology and Verification

- Think ---
  - What is System-on-Chip (SoC)?
  - What is a system? wii, ps-3, IPTV, video phone… etc



Design Component Paradigm Shifts

Where are you when you are graduated?

2000s

DSP RAM MPEG4 RF

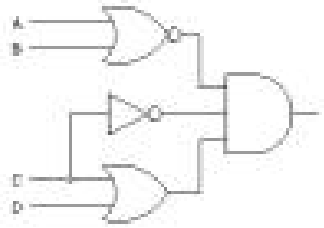
IP

1990s

```
always @ (posedge ck)
  count <= count + 1;
```

RTL

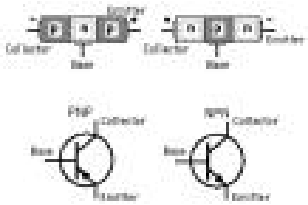
1980s



Logic gates

Logic design automation

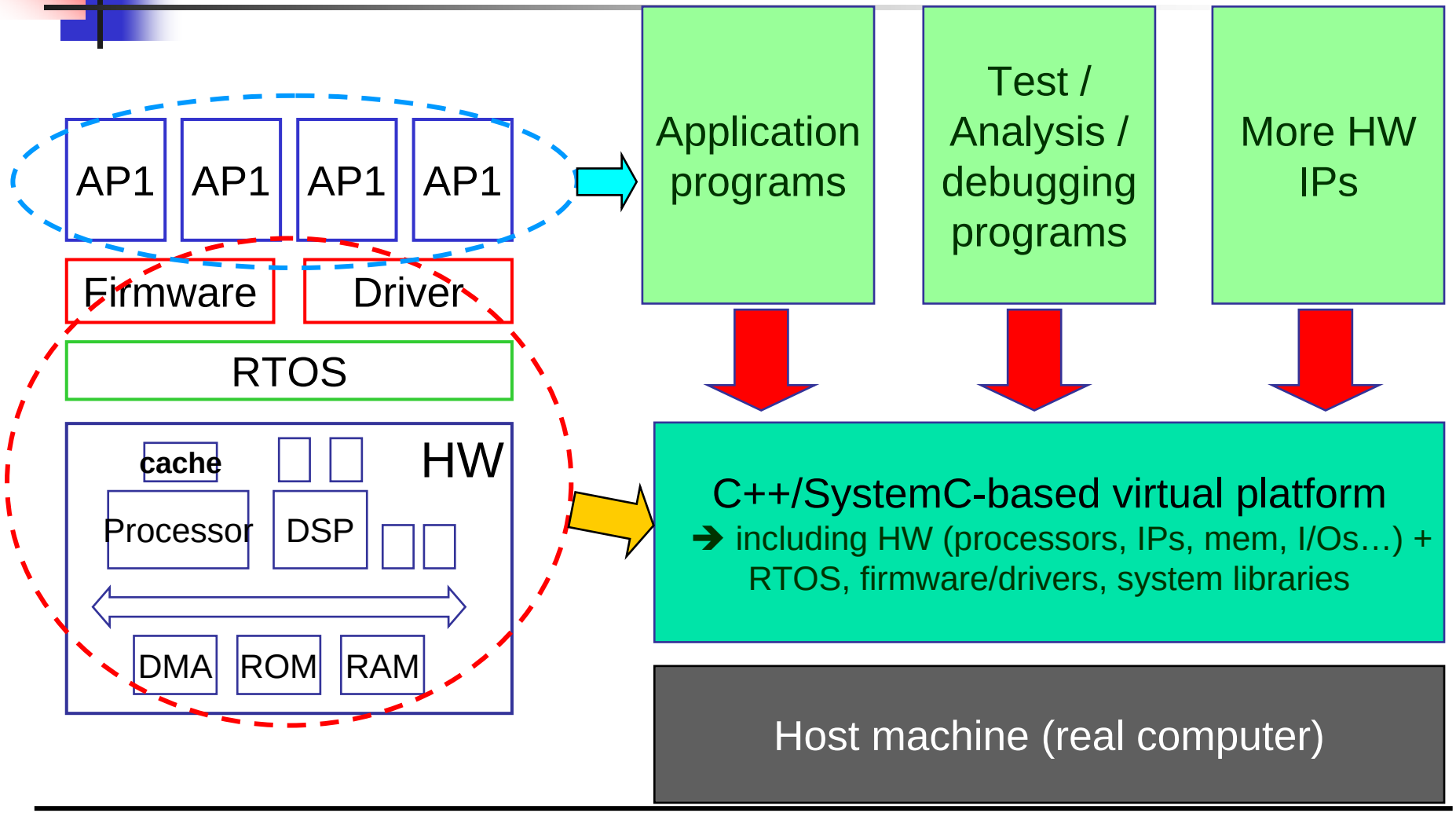
1970s



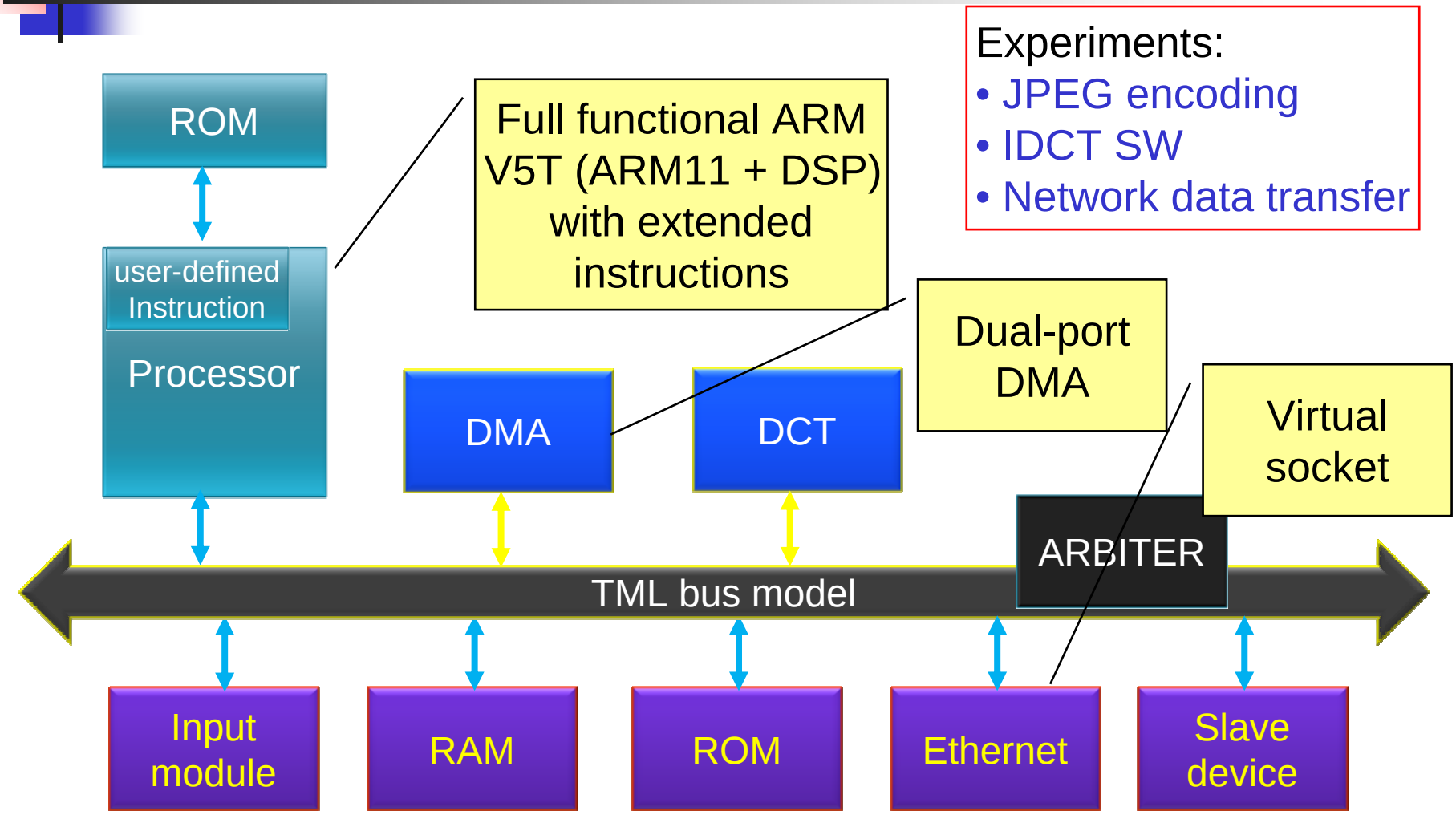
Transistors

Layout design automation

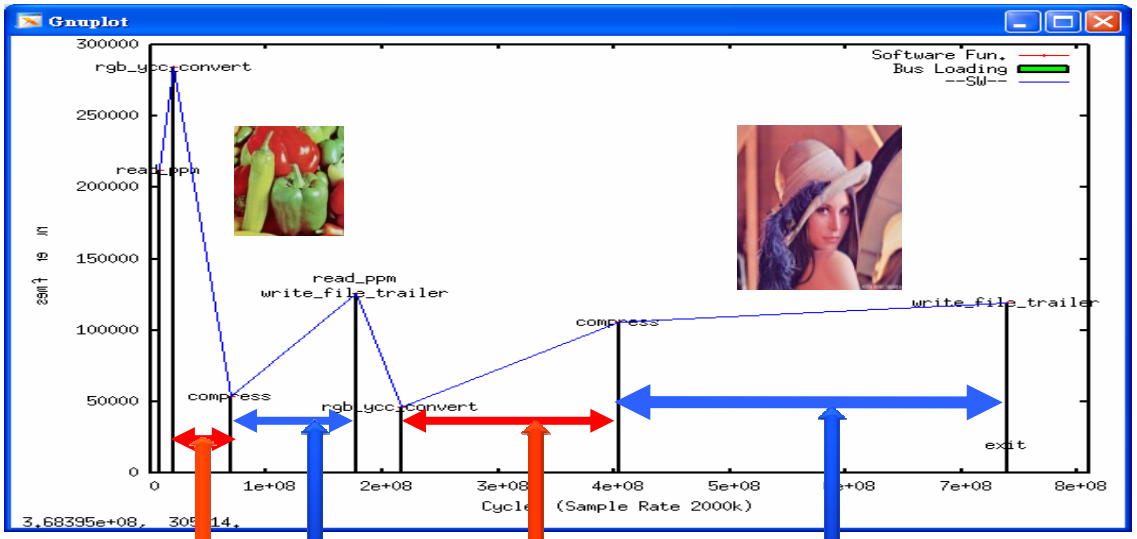
Virtual (HW) Platform

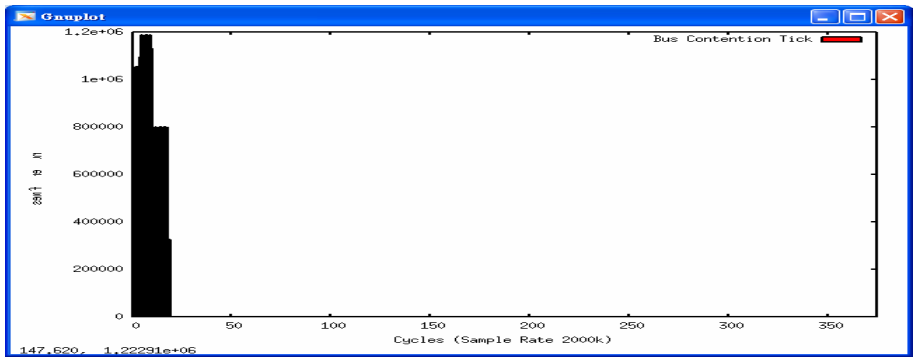


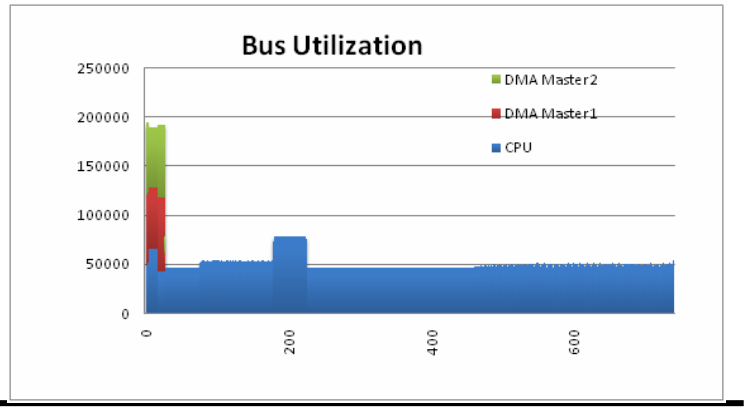
QuteVP: our virtual platform



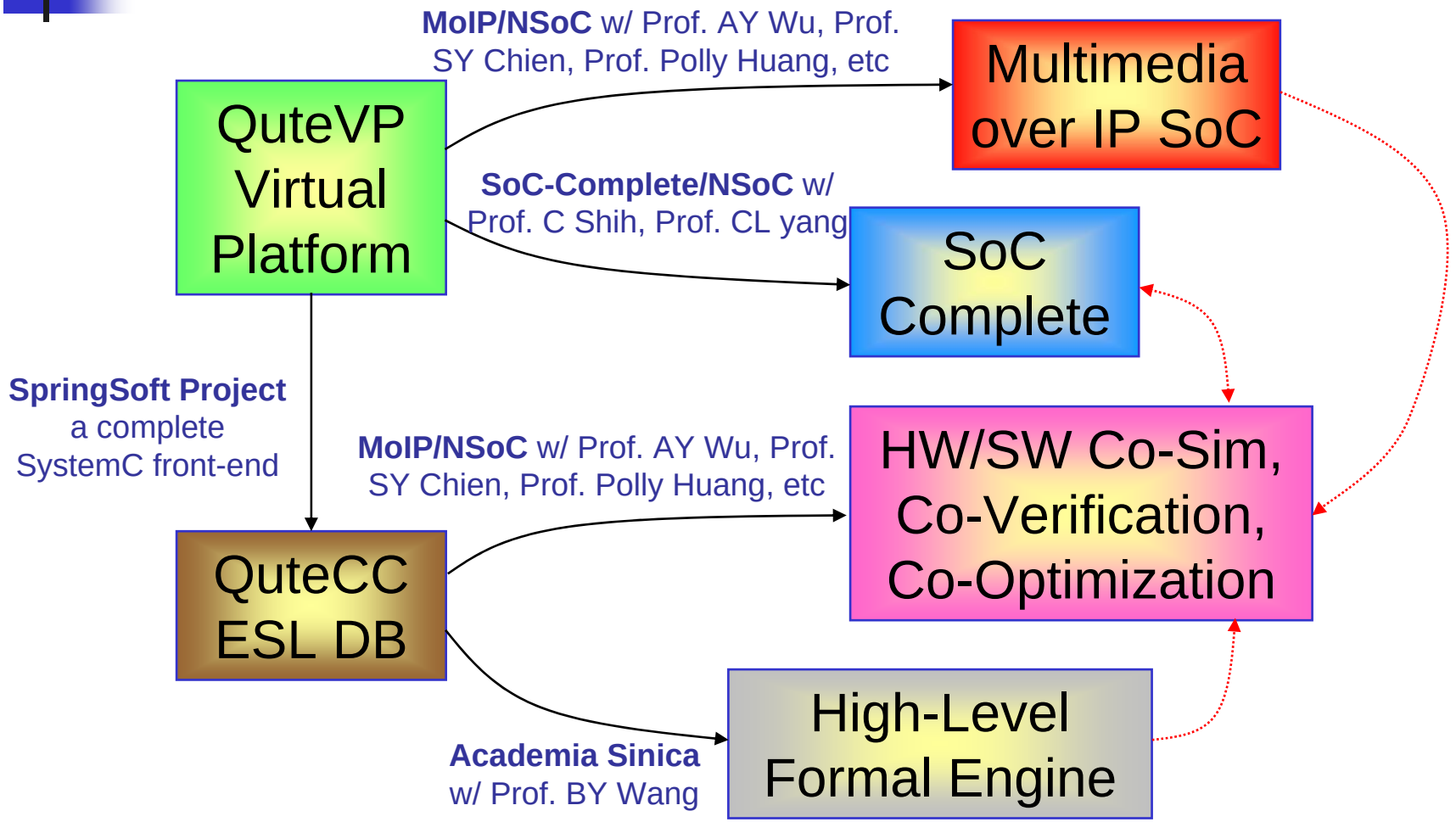
QuteVP: Design Performance Analysis







Our SoC Design/EDA Research



(2) Formal-Assisted Design Debugging and Optimization

1. Designing circuit is now like writing programs
2. Every program has bugs
3. The more complex the design is, the more bugs there will be and the more difficult it is to find the bug
4. The complexity of modern designs grows exponentially

➔ The effort to find and fix bugs is increasing tremendously

Is it really so difficult?

- Yes, the design is too complex

➔ Can you fix the real causes of the bug?

- Not really, if we can READ designer' mind and systematically verify the implementation

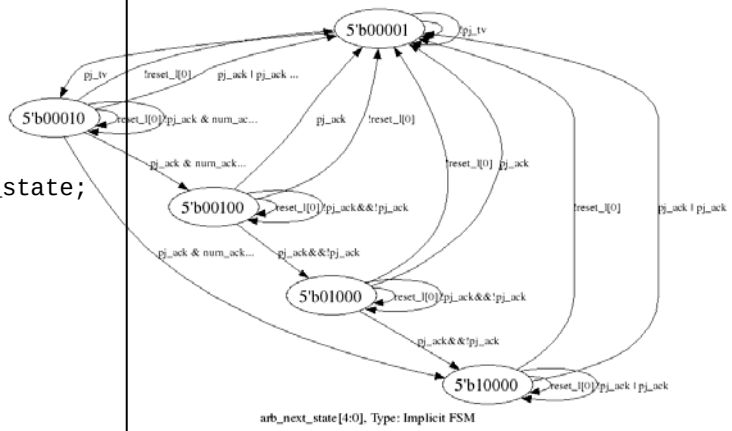
➔ Artificial Intelligence (AI)??

➔ What human can do, the computer can do it million times faster

Key Techniques

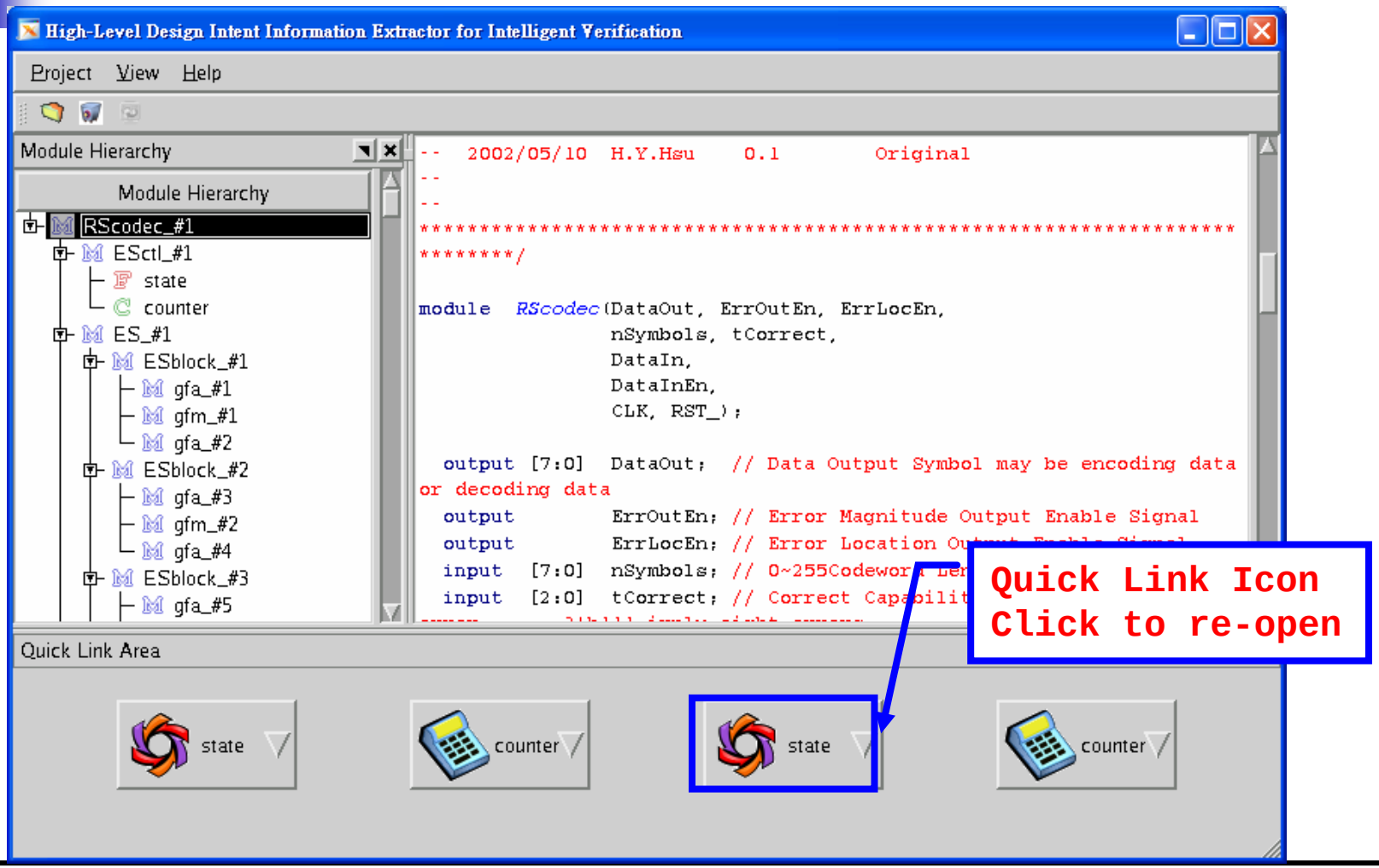
1. Design Intent Extraction (read designer's mind)
   - Extract FSM, counter, etc from RTL Verilog

```
 3 assign arb_next_state[4:0] = arbiter(arb_state[4:0], ..);
 4 ff_sre_4 arb_state_reg (.out(arb_state[4:1]),
 5         .din(arb_next_state[4:1]), .clk(clk),
 6         .reset_l(reset_l), .enable(1'b1));
 8 ff_s arb_state_reg_0 (.out(arb_state[0]),
 9         .din(~reset_l|arb_next_state[0]), .clk(clk));
11 function  [4:0] arbiter; input [4:0] cur_state; reg [4:0] next_state;
14 parameter IDLE  = 5'b00001, REQ_ACTIVE = 5'b00010,
15     FILL3 = 5'b00100, FILL2 = 5'b01000, FILL1 = 5'b10000;
16 begin   case (cur_state)
18    IDLE: if (pj_tv) next_state = REQ_ACTIVE;
20          else next_state = cur_state;
22    REQ_ACTIVE: ...   FILL3: ...  FILL2: ...  FILL1: ...
26    default: next_state = 5'bx;
27    endcase
28    arbiter[4:0] = next_state[4:0];
```

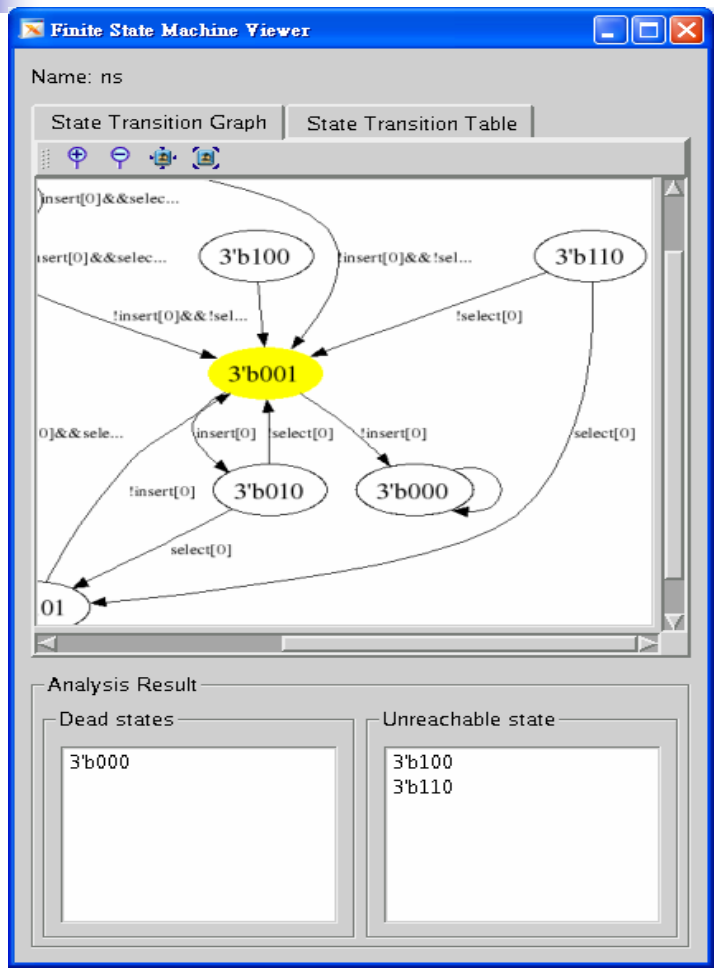


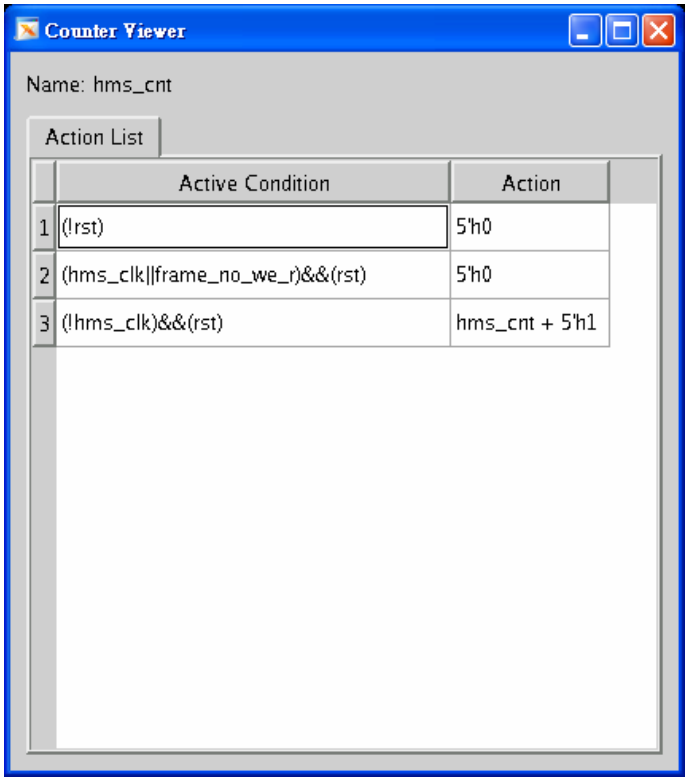
2. Powerful constraint satisfiability solver

FSM/Counter Extraction Tool

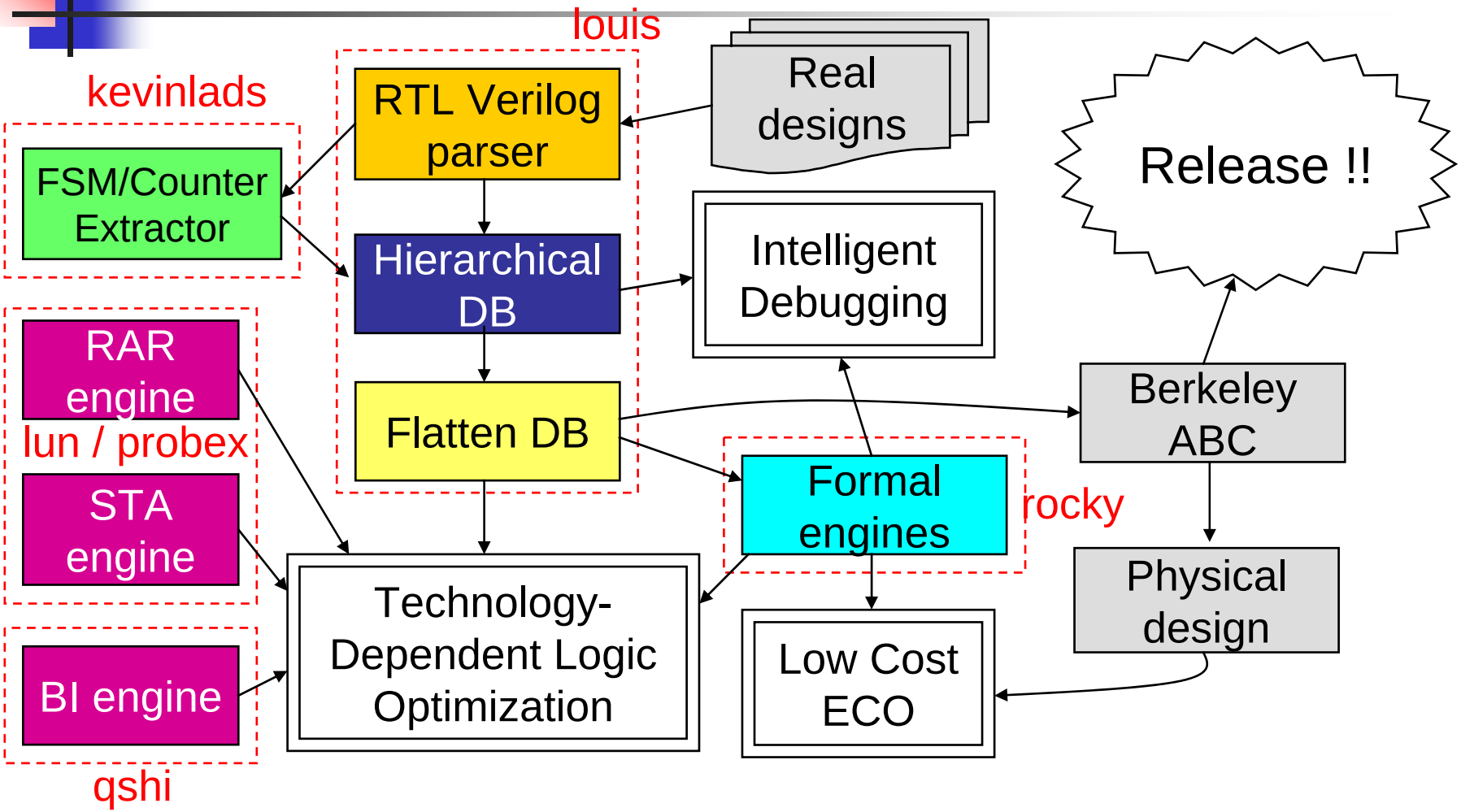


FSM/Counter Extraction Tool



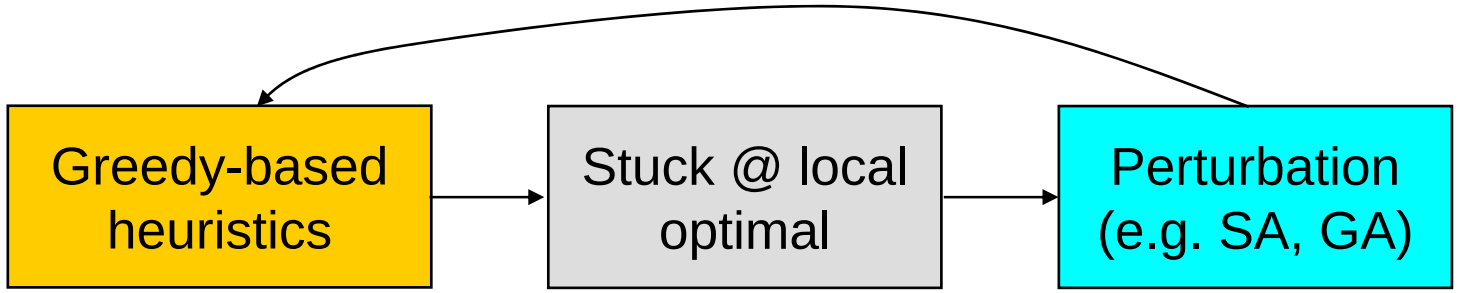


Formal-Assisted Design Debugging and Optimization Roadmap

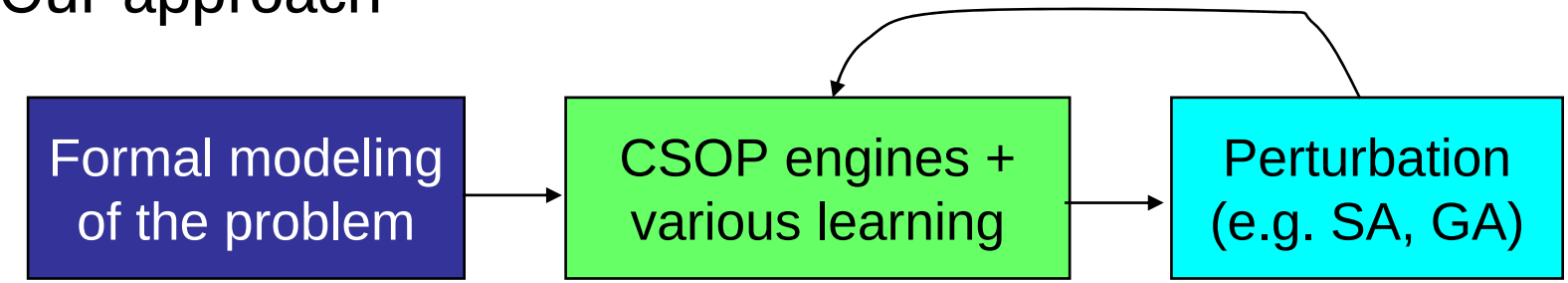


(3) Next generation constraint satisfiability and optimization engine

- Most of the problems in EDA are NP-complete or NP-hard
- Traditional approach



- Our approach



Formal Modeling

- Transform designs and problems in mathematic or logic form
  - Boolean logic (&, |, !...)
  - Arithmetic operators (+, -, *, /…)
  - Axioms…
- Solve it or prove it !!
  - SAT
  - BDD
  - ILP
  - SMT
  - Theorem prover

Weekly Study Groups

1. Formal engine
   - Problem modeling
   - SAT, Pseudo Boolean, ILP, LP, CSP,... etc

   ➔ How to create an EDA-specific formal engine

   ➔ http://140.112.42.185/~thlin/engine-study-group-index.htm
2. Front-end verification
   - Design analysis, validation, debugging,
   - Formal-driven synthesis & optimization

   ➔ How to intelligently utilize formal engines

   ➔ http://140.112.42.185/~louius/fe-study-group-index.htm
3. SoC / ESL design
   - HW/SW co-design, co-verification
   - System-level design methodologies

   ➔ Future design and verification paradigm

   ➔ http://140.112.42.185/~sam/esl-study-group-index.htm

Related courses for my researches

- 大學部
  - 資料結構與程式設計 (Ric, 2008 Spring)
  - 演算法 (Prof. Yao-Wen Chang)
  - EDA 導論
  - …
- 研究所
  - 系統晶片驗證 (Ric, 2008 Spring)
  - 自動化合成與驗證 (Prof. Jie-Hong Jiang)
  - …

Will you be one of us?

- If you…
  - 喜歡挑戰
  - 喜歡寫程式
  - 可以自動自發
  - 喜歡邏輯的思考

➔ 歡迎來找我談!!

➔ 應該還有一個名額…

實驗室工號

工號	姓名	加入年月	Title
0	黃鐘揚	2004.02	老闆
1	林庭豪	2004.02	博 3
2	葉護熹	2004.05	博 3
3	魏士貴	2004.05	碩士
4	胡敏寬	2004.05	碩士
5	林宗茂	2004.06	碩士
6	黃俊輔	2004.06	碩士
7	林長志	2004.09	--
8	張啓文	2004.12	碩士
9	李智群	2004.12	碩士
10	李晨豪	2004.12	碩士
11	蔡詩蘅	2004.12	博 2
12	洪星智	2005.04	碩士
13	劉惠芳	2005.04	碩 2
14	葉昱甫	2005.06	博 3
15	湯凱富	2005.06	博 3
16	吳濟安	2005.11	碩 2
17	粘敬佳	2005.11	碩 1
18	黃紹倫	2005.12	博 2
19	周俊男	2005.12	博 2
20	吳鎧竹	2005.12	碩 2
21	李沅龍	2006.04	碩 2
22	胡啓政	2006.04	碩 2
23	饒瑞晞	2006.11	碩 1
24	葉鎮丞	2006.11	碩 1
25	許智仁	2006.11	碩 1
26	林坤輝	2006.11	碩 1
27	蔡曉傑	2006.11	碩 1
28	賴昭宇	2007.09	RA

博班: 7, 碩班: 11, 研究助理: 2, 專題生: 4

For more information...

- To find me
  - Office: EE-II 444
  - ric@cc.ee.ntu.edu.tw
  - ric2k1 @ ptt, ptt2, msn, skype, ...
  - 02-3366-3644
- Lab:
  - BL 627 (即將搬至 EE 353)
  - PTT: NTUGIEE_ric
  - Website: http://dvlab.ee.ntu.edu.tw/