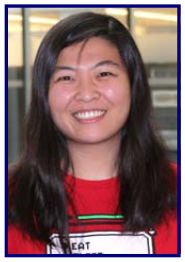
The TAU 2015 Contest

Incremental Timing and CPPR Analysis



Jin Hu
IBM Corp.
[Speaker]

Greg Schaeffer
IBM Corp.

Vibhor Garg
Cadence

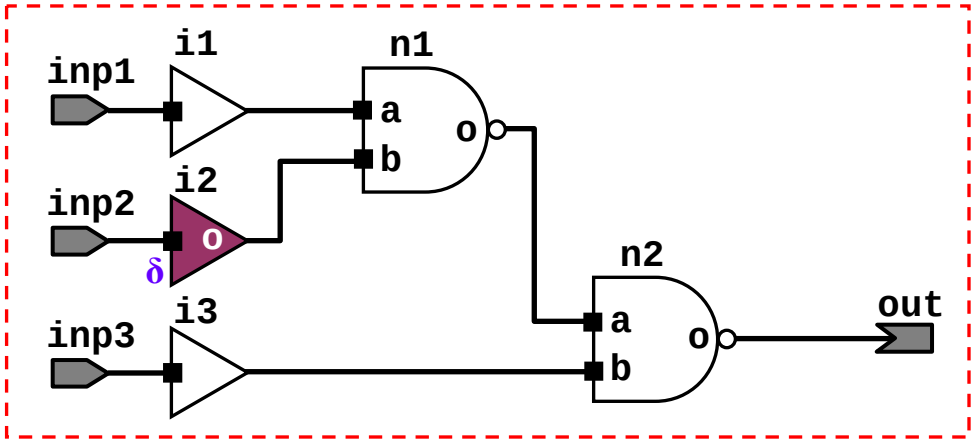
Sponsors:

Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

Incremental Timing (Arrival Time Update Example)

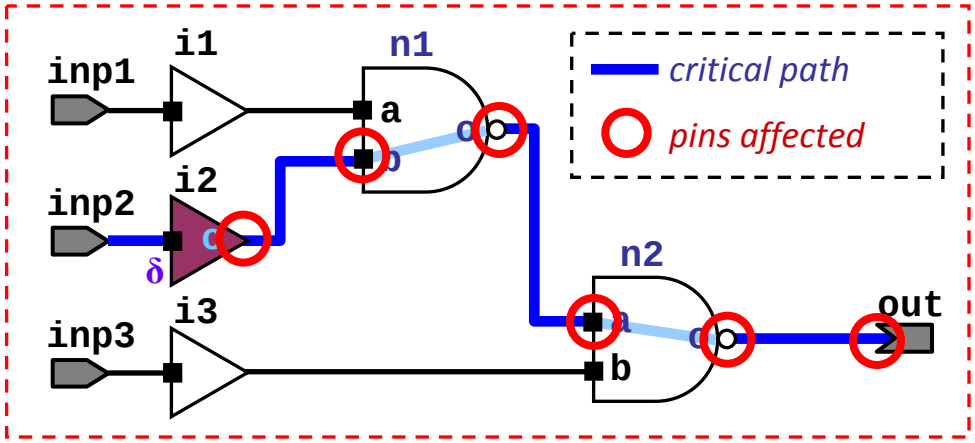


Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

Incremental Timing (Arrival Time Update Example)

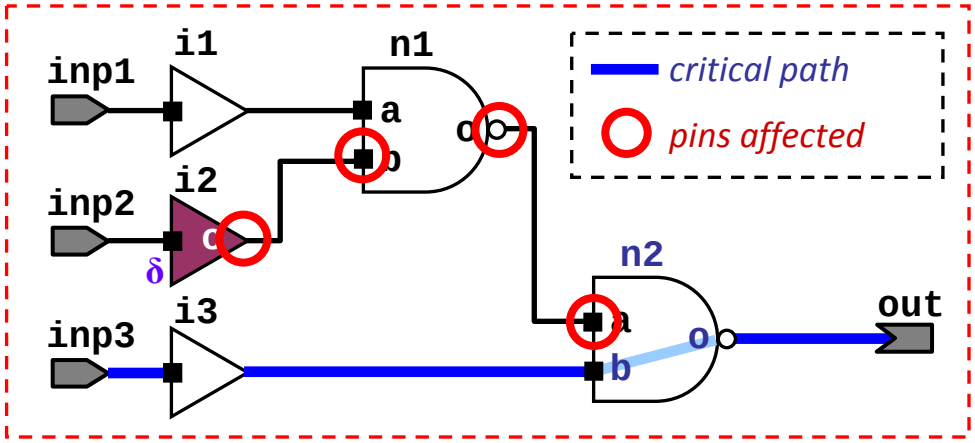


Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

Incremental Timing (Arrival Time Update Example)

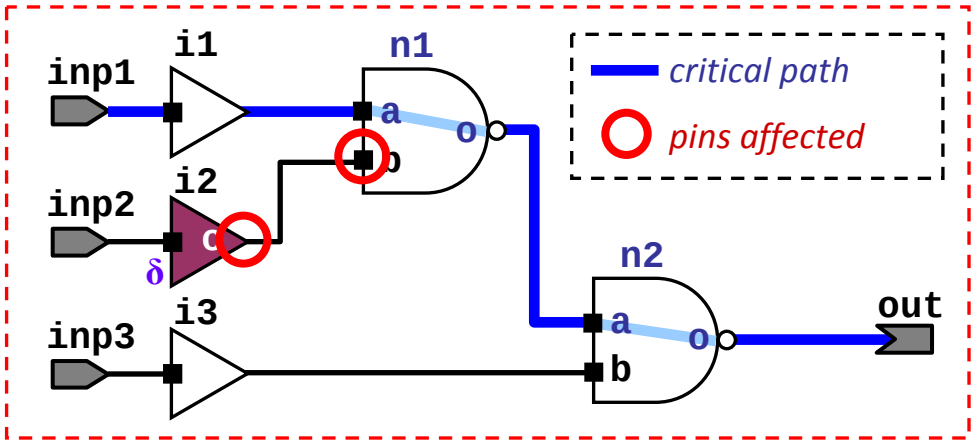


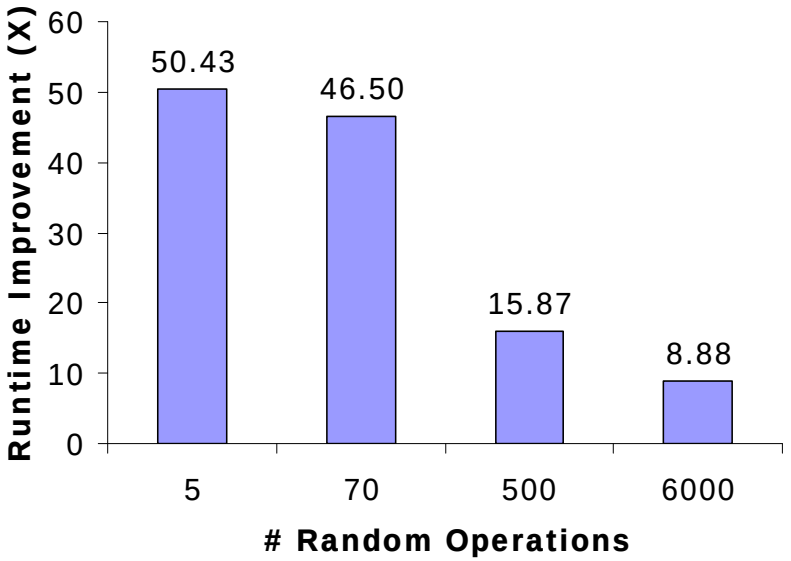
Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

Incremental Timing (Arrival Time Update Example)



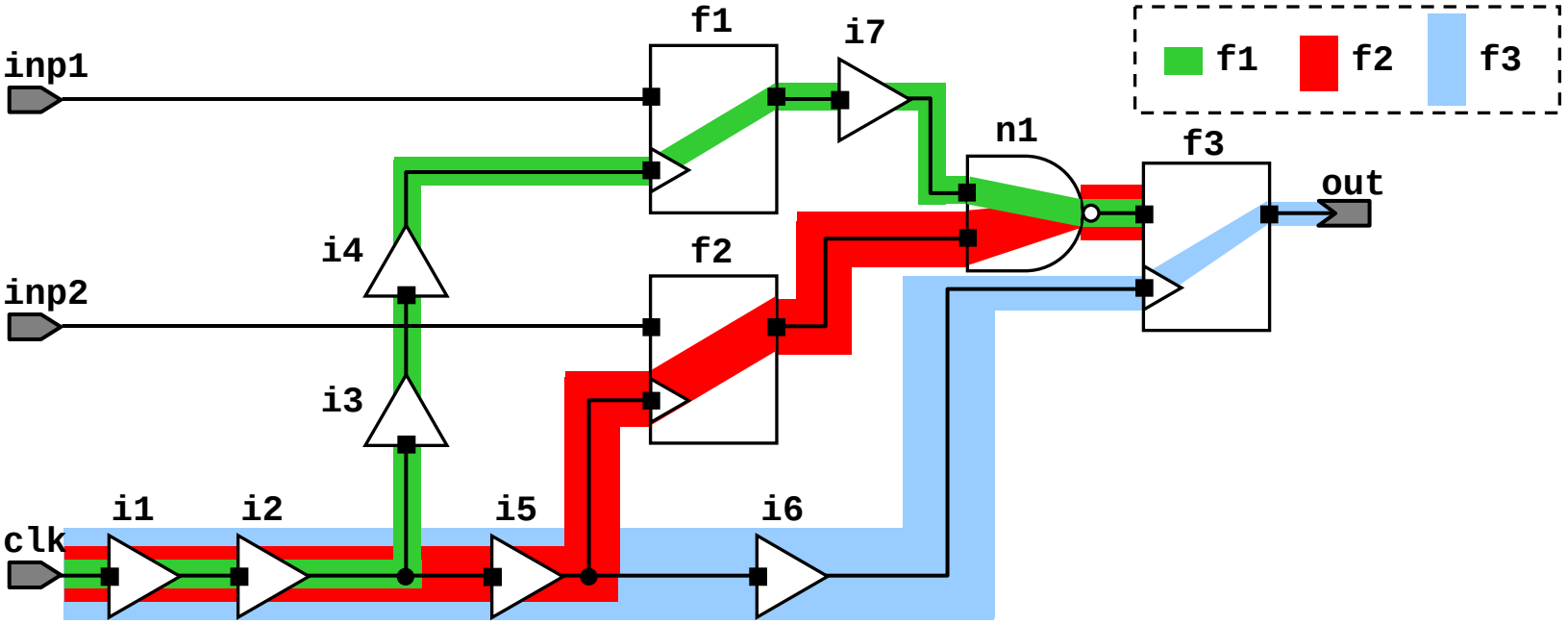


Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

Incremental CPPR (Credit Computation Update Example)

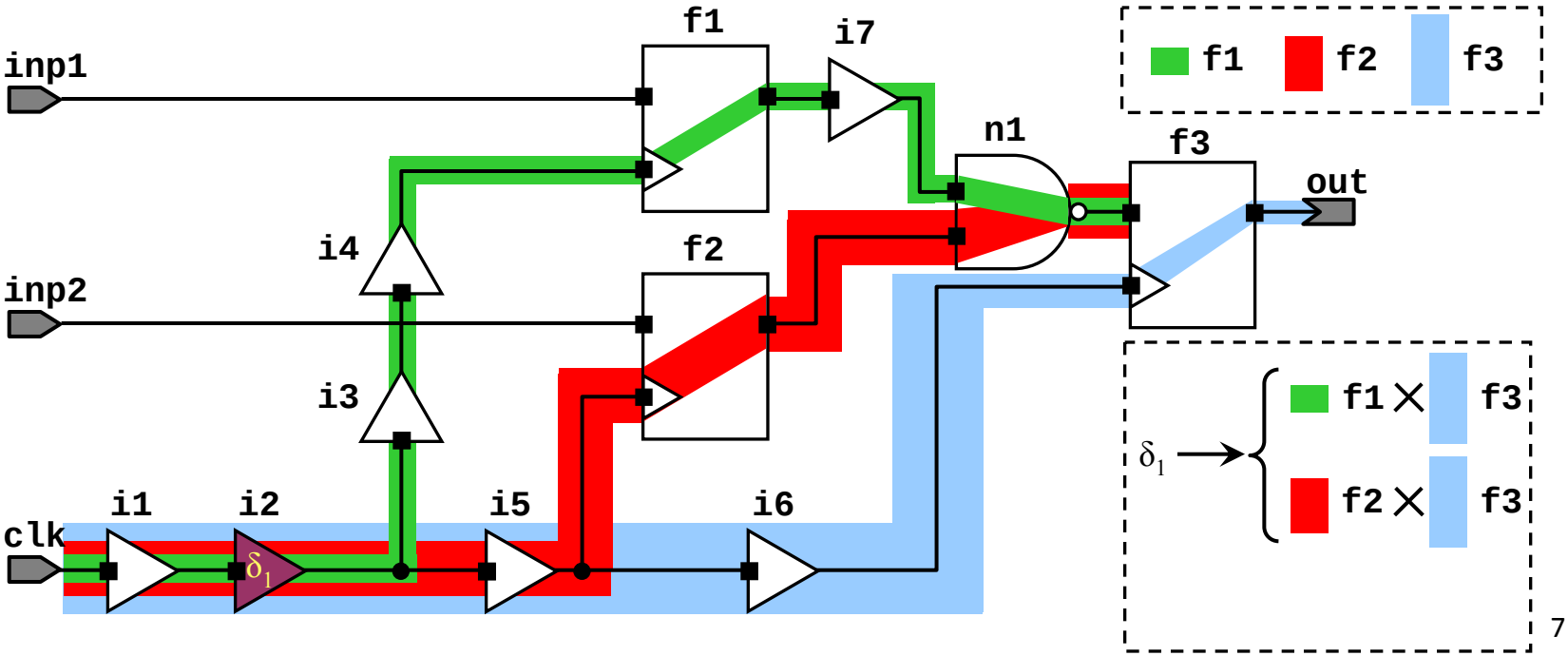


Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

Incremental CPPR (Credit Computation Update Example)

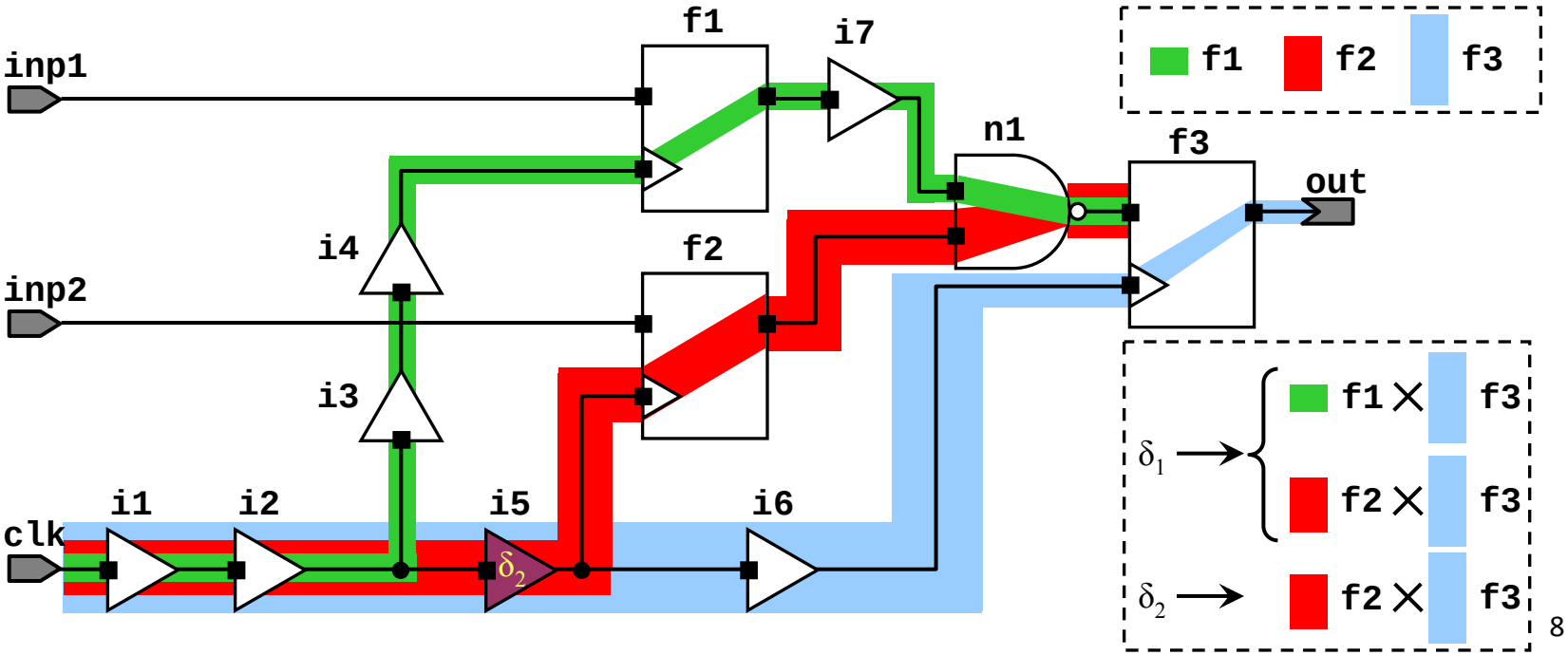


Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

Incremental CPPR (Credit Computation Update Example)



Motivation of Incremental Analysis

Performance

Faster turnaround time for timing analysis in presence of design changes

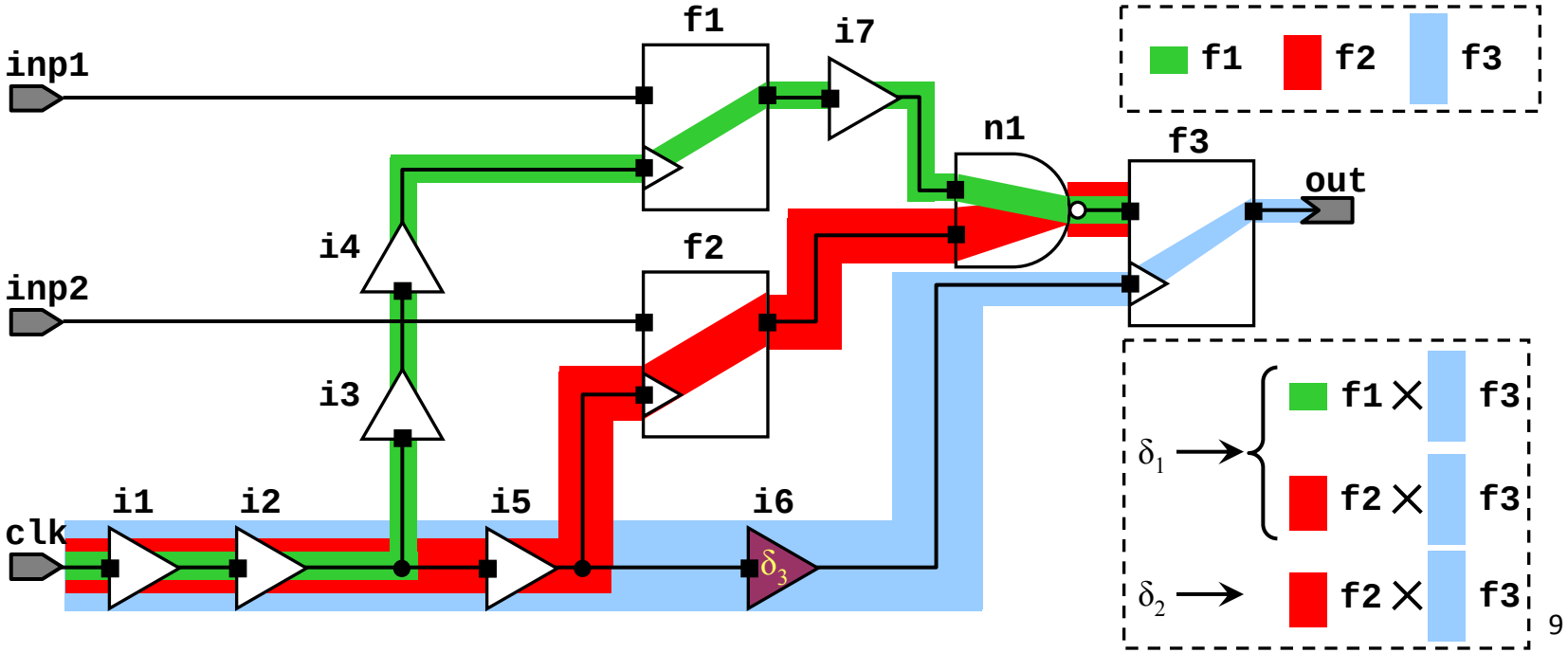
Broadened Research Scope

Enables **timing-driven** applications, such as placement and routing

Theory and Implementation

Requires both algorithmic intelligence and **agile coding** techniques

Incremental CPPR (Credit Computation Update Example)



TAU 2015 Contest: Bigger and Better

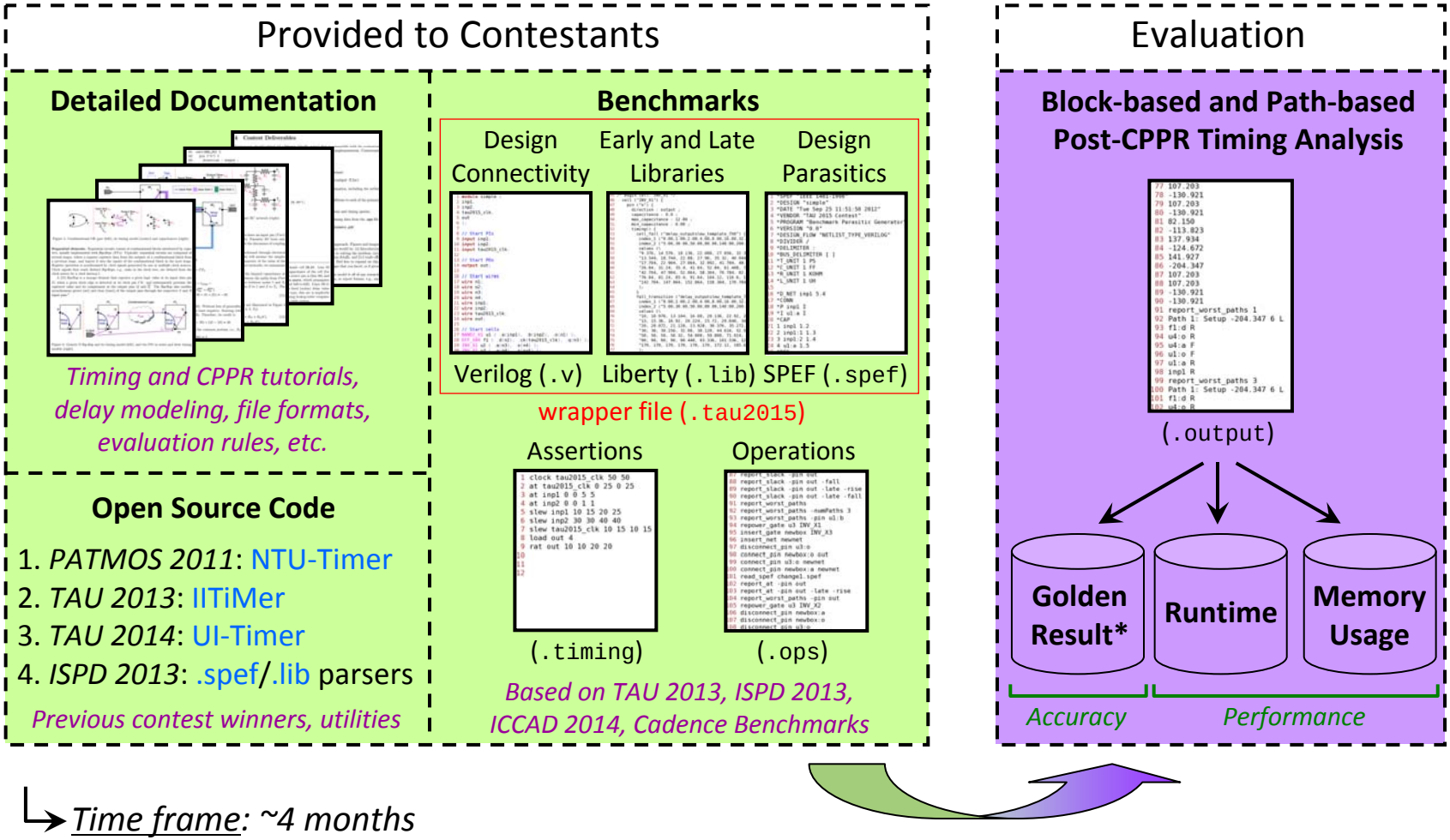
Develop a block-based **and** path-based timer that supports

Incremental Timing and **Incremental CPPR**† **Analysis**

	PATMOS'2011	TAU 2013	TAU 2014	TAU 2015
Delay and Output Slew Calculation	✓	✓		✓
Separate Rise/Fall Transitions	✓	✓		✓
Block / Gate-level Capabilities	✓	✓		✓
Path-level Capabilities			✓	✓
Statistical / Multi-corner Capabilities		✓		
Incremental Capabilities				✓
Industry-standard Input Formats				✓

†*CPPR: process of removing inherent but artificial pessimism from timing tests and paths*

TAU 2015 Contest Overview



↳ *Time frame: ~4 months*

↳ *Contest Scope: only hold, setup, RAT tests; no latches (flush segments); single-source clock tree*

using industry timer

Input Files – Design Infrastructure

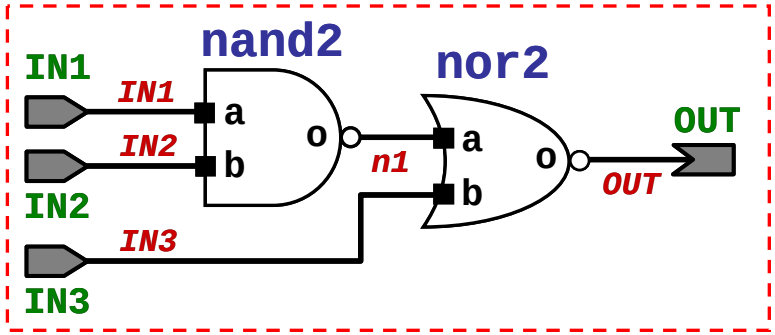
industry-standard formats

Verilog (.v): design connectivity

```
module simple  (IN1, IN2, IN3, OUT);

NAND2_X1 nand2 (.a(IN1), .b(IN2), .o(n1));
NOR2_X1 nor2   (.a(n1),  .b(IN3), .o(OUT);

endmodule
```



Input Files – Design Infrastructure

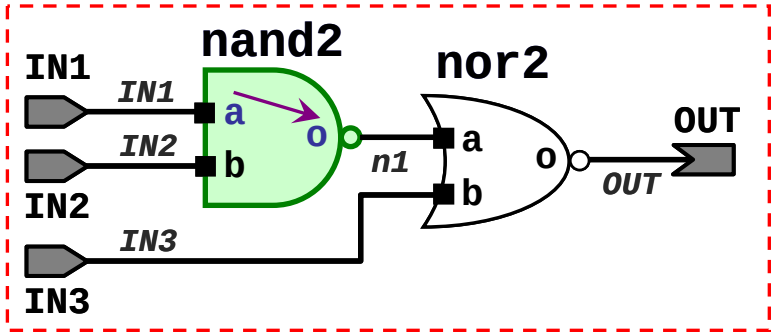
industry-standard formats

Verilog (.v): design connectivity

```
module simple  (IN1, IN2, IN3, OUT);

NAND2_X1 nand2 (.a(IN1), .b(IN2), .o(n1));
NOR2_X1 nor2   (.a(n1),  .b(IN3), .o(OUT);

endmodule
```



Liberty (.lib): gate library

```
cell ("NAND2_X1") {          // Fall-to-Rise from a → o
  pin ("o") {
    direction   : output;
    capacitance : 0.0;
    timing() {
      cell_fall ("scalar") {        // delay
        values ("26.064");
      }
      fall_transition("scalar") { // output slew
        values ("30.216");
      }
      timing_sense : negative_unate;
      related_pin  : "a";
    }
  }
}
```

Input Files – Design Infrastructure

industry-standard formats

Verilog (.v): design connectivity

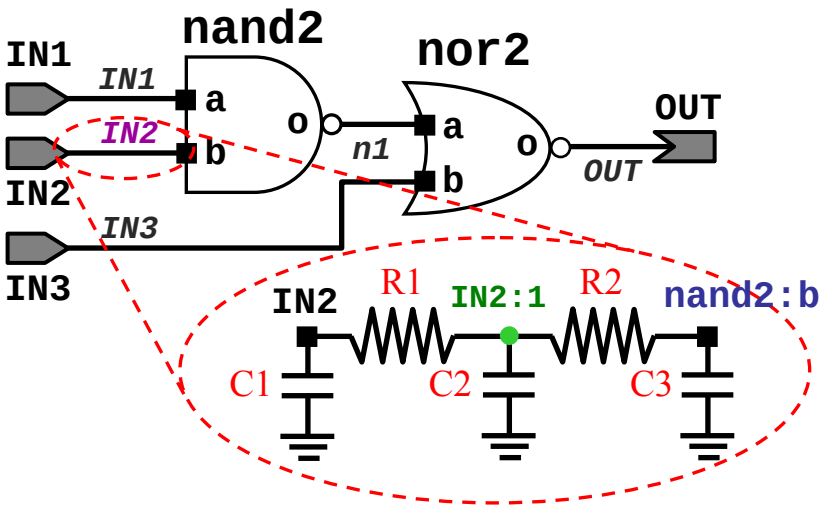
```
module simple  (IN1, IN2, IN3, OUT);

NAND2_X1 nand2 (.a(IN1), .b(IN2), .o(n1));
NOR2_X1 nor2   (.a(n1),  .b(IN3), .o(OUT);

endmodule
```

Liberty (.lib): gate library

```
cell ("NAND2_X1") {          // Fall-to-Rise from a → o
  pin ("o") {
    direction   : output;
    capacitance : 0.0;
    timing() {
      cell_fall ("scalar") {        // delay
        values ("26.064");
      }
      fall_transition("scalar") { // output slew
        values ("30.216");
      }
      timing_sense : negative_unate;
      related_pin  : "a";
    }
  }
}
```



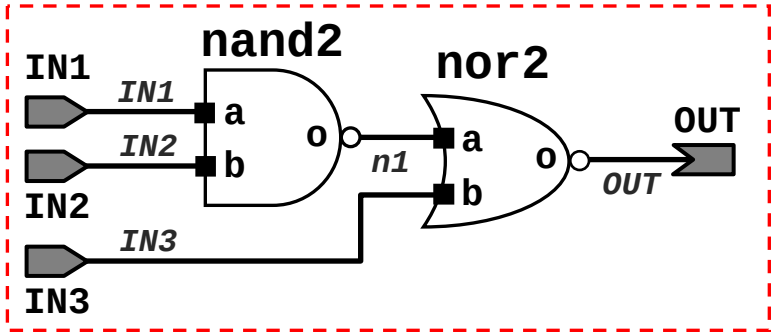
SPEF (.spef): design parasitics

```
*D_NET IN2 1.6
*CONN
*P IN2 I
*I nand2:b I
*CAP
C1 IN2      0.2
C2 IN2:1    0.5
C3 nand2:b 0.9
*RES
R1 IN2    IN2:1    1.4
R2 IN2:1 nand2:b 1.6
*END
```

Input Files – Assertions (`.timing`)

Provides initial conditions for timing

- Values specific to [Early/Late] x [Rise/Fall]
- Arrival times for each **Primary Input**
- Input slews for each **Primary Input**
- Required arrival times for each **Primary Output**
- Asserted pin capacitances for each **Primary Output**
- Clock period for the **clock source** (if design is sequential)



Assertions (`.timing`)

```
//                ER    EF    LR    LF
at     IN1        0     0     5     5
at     IN2        0     0     1     1
at     IN3        0     0    10    12
slew   IN1       10    15    20    25
slew   IN2       30    30    40    40
slew   IN3       30    30    40    40
rat    out       10    10    20    20

load   out        4
```

Input Files – Operations (.ops)

Design-level Operations

GATE-LEVEL

insert_gate **<name>** **<type>**	:	creates new gate **<name>** of type **<type>**
remove_gate **<name>**	:	removes existing gate **<name>**
repower_gate **<name>** **<type>**	:	changes power level of gate **<name>** to type **<type>**

NET-LEVEL

insert_net **<name>**	:	creates new net **<name>**
remove_net **<name>**	:	removes existing net **<name>**
read_spef **<file>**	:	changes or adds parasitics of net(s) in **<file>**

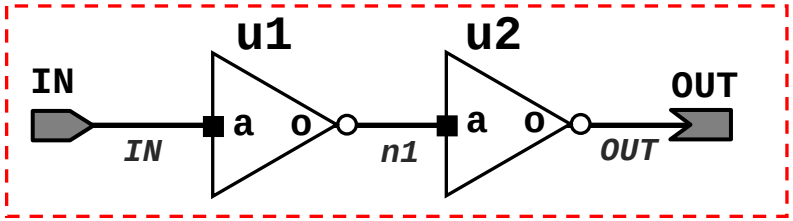
PIN-LEVEL

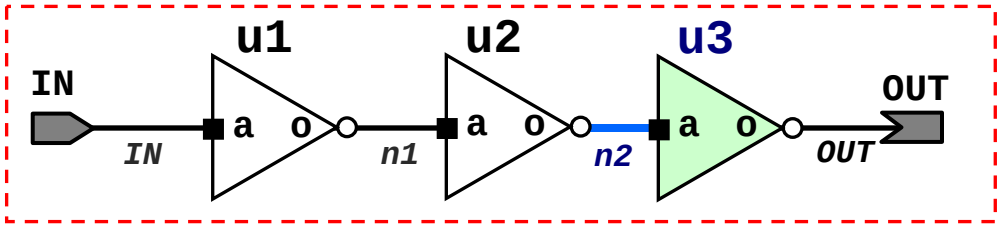
disconnect_pin **<pin name>**	:	decouples pin **<pin name>** from any net
connect_pin **<pin name>** **<net name>**	:	links pin **<pin name>** to net **<net name>**

Input Files – Operations (.ops)

Design-level Operations Example

add inverter into design



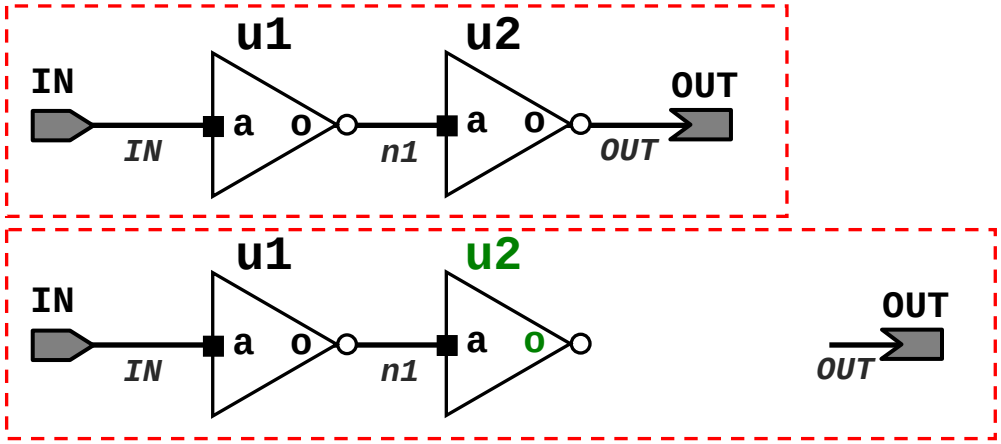


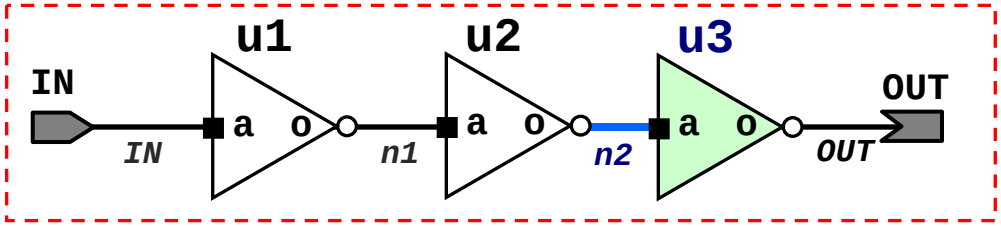
Input Files – Operations (.ops)

Design-level Operations Example

add inverter into design

```
disconnect_pin u2:o
```





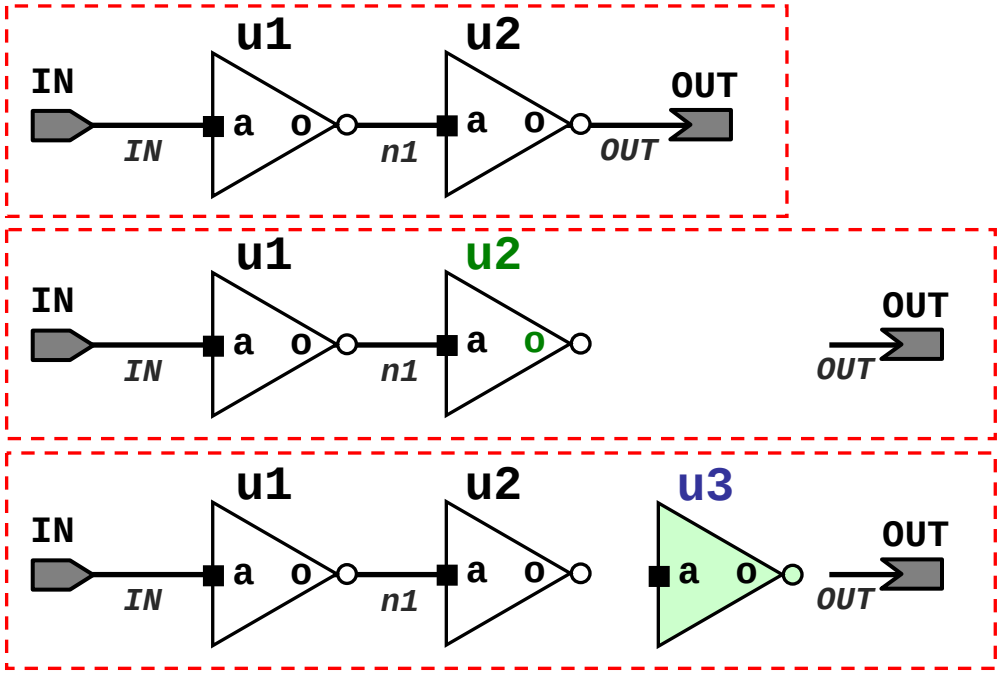
Input Files – Operations (.ops)

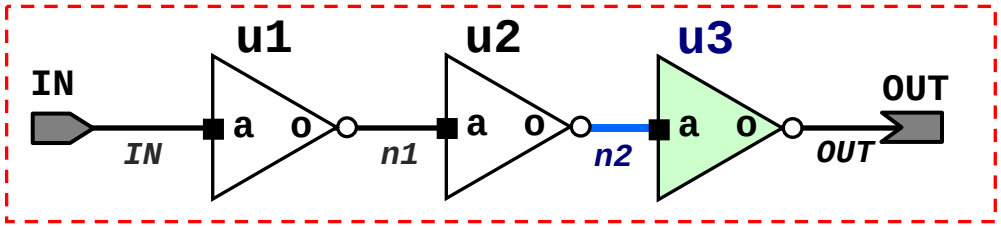
Design-level Operations Example

add inverter into design

```
disconnect_pin u2:o
```

```
insert_gate u3 INV_X1
```





Input Files – Operations (.ops)

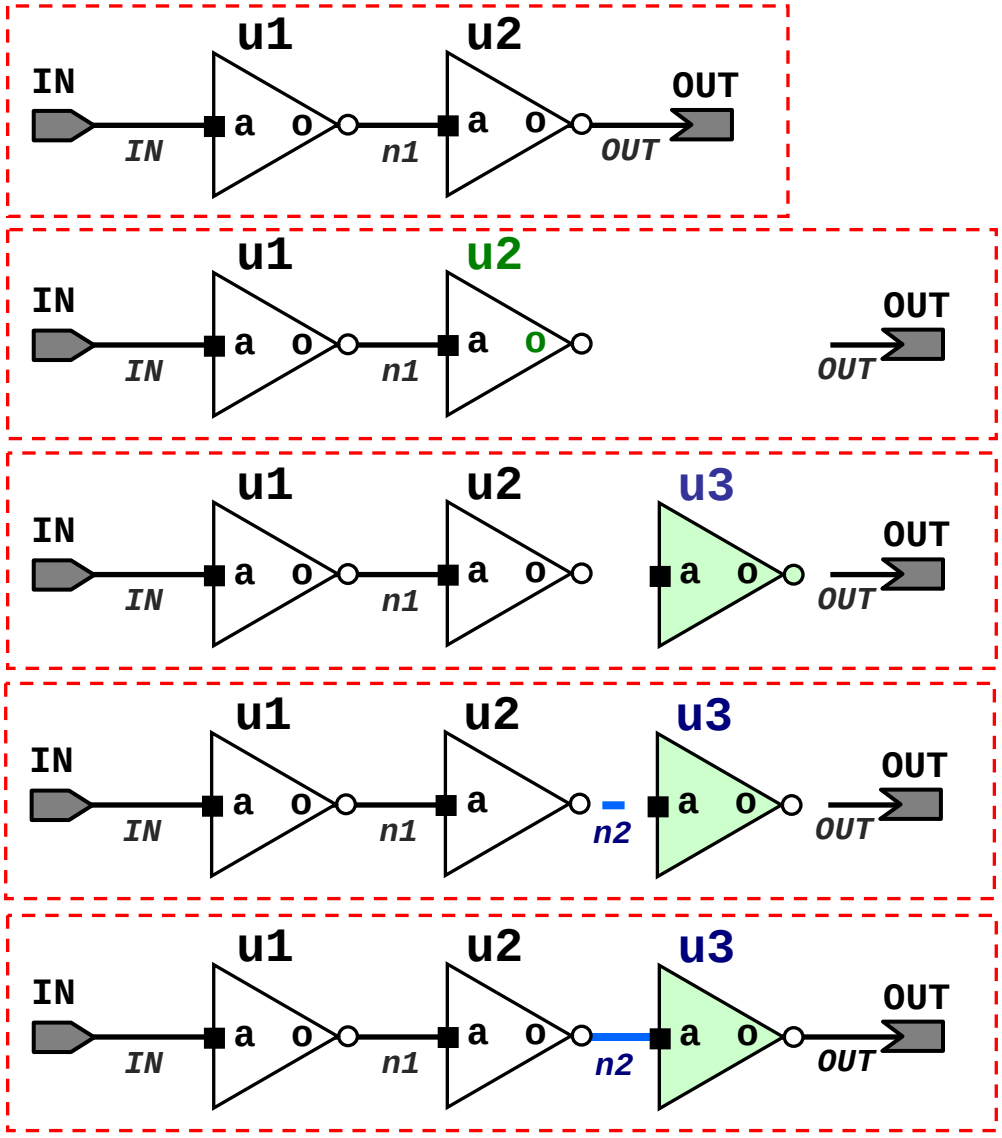
Design-level Operations Example

add inverter into design

```
disconnect_pin u2:o

insert_gate u3 INV_X1

insert_net n2
```



Input Files – Operations (.ops)

Design-level Operations Example

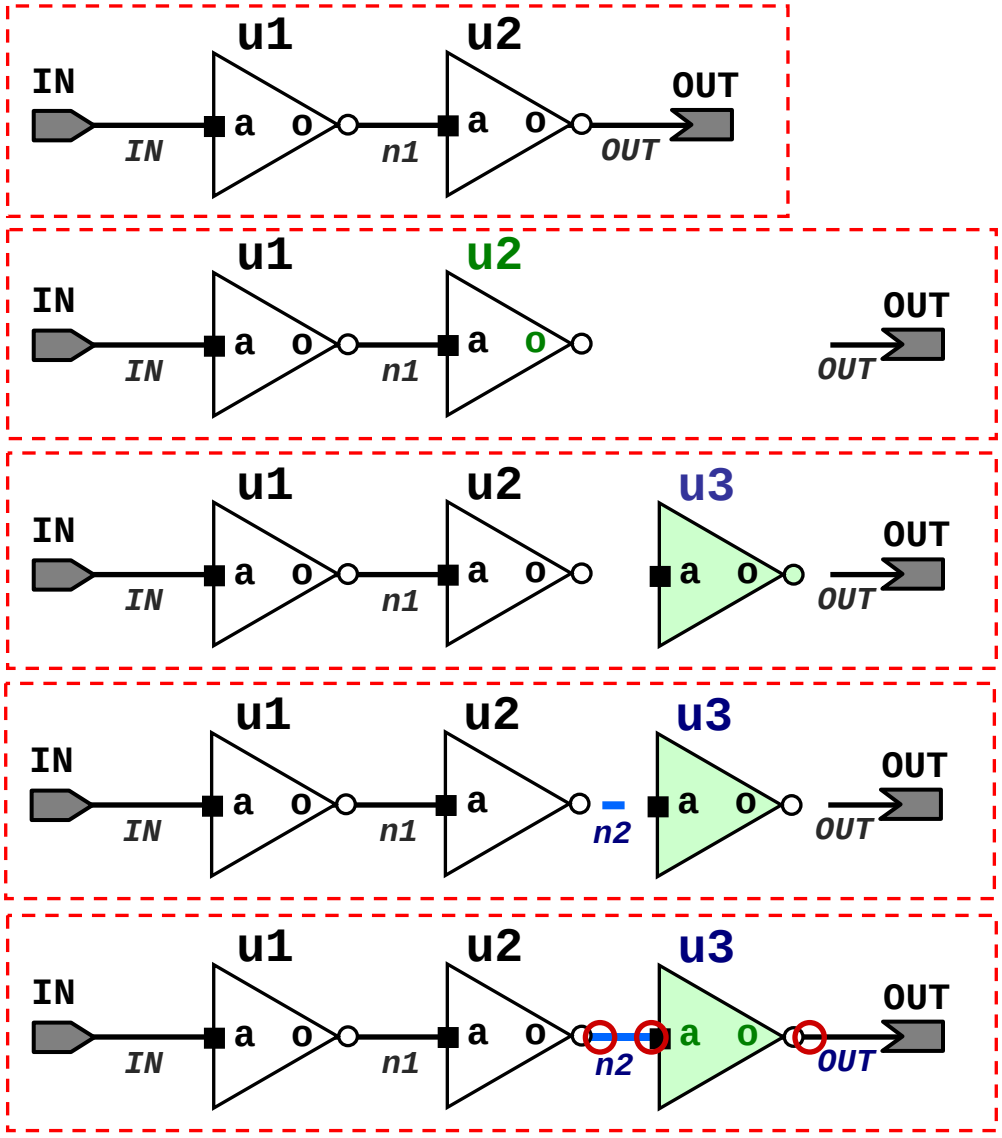
add inverter into design

```
disconnect_pin u2:o

insert_gate u3 INV_X1

insert_net n2

connect_pin u2:o n2
connect_pin u3:a n2
connect_pin u3:o OUT
```



Input Files – Operations (.ops)

Timing Queries

BLOCK-BASED

report_at **<pin name>** **[RFEL]** : prints arrival time at pin **<pin name>** for **[Rise/Fall][Early/Late]**

report_rat **<pin name>** **[RFEL]** : prints required arrival time at pin **<pin name>** for **[Rise/Fall][Early/Late]**

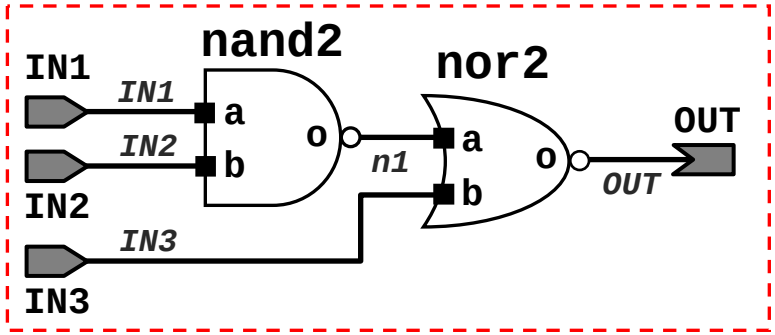
report_slack **<pin name>** **[RFEL]** : prints post-CPPR slack at pin **<pin name>** for **[Rise/Fall][Early/Late]**

PATH-BASED

report_worst_paths **[pin name]** **[n]** : prints top **[n]** paths w/worst post-CPPR slack
(*i*) in the design, or
(*ii*) through **[pin name]**

Output File (`.output`)

`report_at`
`report_rat`
`report_slack` → *single value*



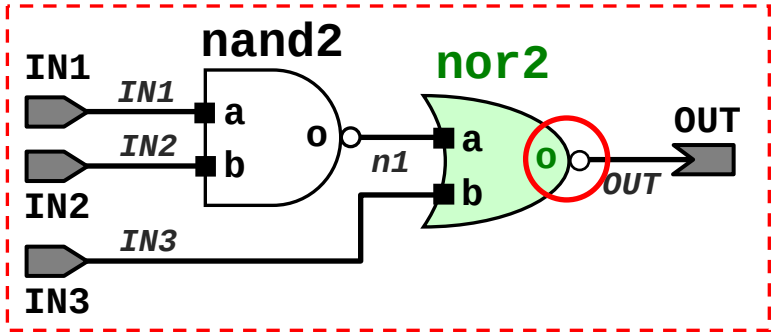
`report_worst_paths` → *Path type, post-CPPR path slack, path length, mode, path* [pin RFEL]
[RAT/Setup/Hold] [Early/Late]

Operations File

Output File

Output File (`.output`)

`report_at`
`report_rat`
`report_slack`
→ *single value*



`report_worst_paths` → *Path type, post-CPPR path slack, path length, mode, path* [pin RFEL]
[RAT/Setup/Hold] [Early/Late]

Operations File

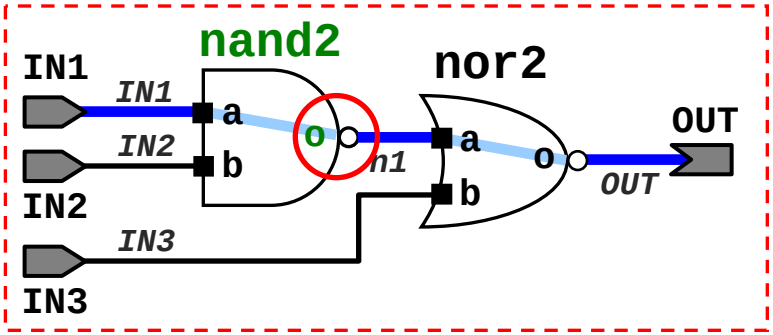
```
report_at –pin nor2:o -fall -early
```

Output File

```
15.85
```

Output File (.output)

```
report_at
report_rat
report_slack
```
→ *single value*



`report_worst_paths` → *Path type, post-CPPR path slack, path length, mode, path* [pin RFEL]

[RAT/Setup/Hold] [Early/Late]

Operations File

```
report_at –pin nor2:o -fall -early
report_worst_paths –pin nand2:o –fall -late
```

Output File

```
15.85

Path 1: RAT -14.7 6 L
OUT      R
nor2:o   R
nor2:a   F
nand2:o  F
nand2:a  R
IN1      R
```

Benchmarks: Phase 1

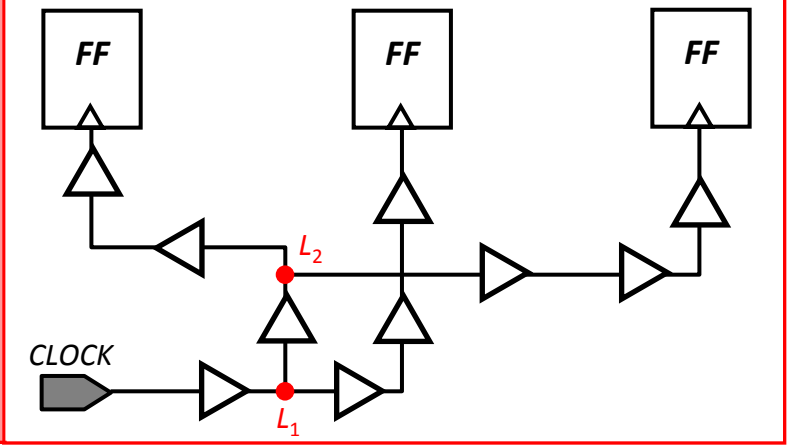
13 based on TAU 2013 v1.0 combinational benchmarks (~10^0 – ~10^3 gates)

10 based on TAU 2013 v1.0 sequential benchmarks (~10^0 – ~10^3 gates)

Design	Number of:			
	PIs	POs	Gates	Nets
c3_slack	4	3	3	7
c17	5	2	6	11
c17_slack	5	2	6	11
c432	36	7	134	170
c499	41	32	176	217
c1355	41	32	180	221
c1908	33	25	222	255
c2670	157	63	344	501
c3540	50	22	691	741
c5315	178	123	918	1.1K
c6288	32	32	1.7K	1.7K
c7552	206	107	1.1K	1.4K
c7552_slack	206	107	1.1K	1.4K
ff	2	3	11	14
s27	6	1	28	34
s344	11	11	182	193
s349	11	11	194	205
s386	9	7	177	186
s400	5	6	221	226
s510	21	7	312	291
s526	5	6	304	309
s1196	16	14	641	657
s1494	10	19	804	814

Added randomized clock tree [TAU 2014]

- **BRANCH**(*CLOCK*, *initial FF*)
- For each remaining FF
  - Select random location *L* in current tree
  - **BRANCH**(*L*,*FF*)
- **BRANCH**(`src,sink`): create buffer chain from `src` to `sink`



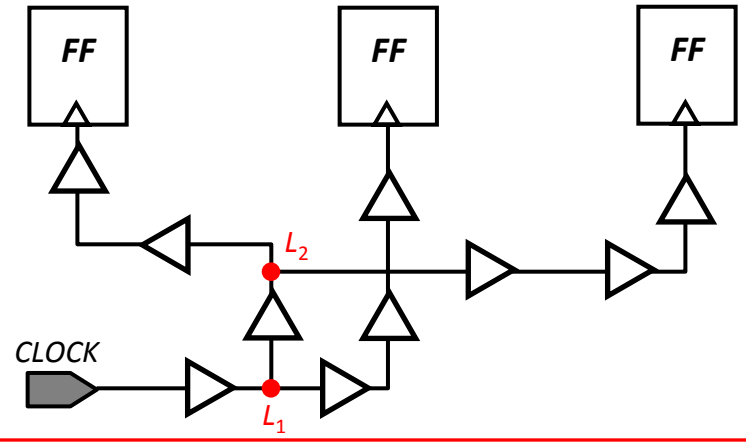
Benchmarks: Phase 1

11 based on TAU 2013 v2.0 benchmarks (~10^3 – ~10^5 gates)

Design	Number of:			
	PIs	POs	Gates	Nets
ac97_ctrl	84	48	14.3K	14.4K
aes_core	260	129	22.9K	23.2K
des_perf	235	64	105.4K	106.5K
mem_ctrl	115	152	10.5K	10.7K
pci_bridge32	162	207	19.1K	19.3K
systemcaes	260	129	6.5K	6.8K
systemcdes	132	65	3.4K	3.6K
tv80	14	32	5.3K	5.3K
usb_funct	128	121	15.7K	15.9K
vga_lcd	89	109	139.5K	139.6K
wb_dma	217	215	4.2K	4.4K

Added randomized clock tree [TAU 2014]

- **BRANCH**(*CLOCK*, *initial FF*)
- For each remaining FF
  - Select random location *L* in current tree
  - **BRANCH**(*L*,*FF*)
- **BRANCH**(`src,sink`): create buffer chain from `src` to `sink`



Benchmarks: Phase 2

7 based on ISPD 2013 benchmarks (~10^3 – ~10^5 gates)

1 based on Cadence benchmarks (~10^2 gates)

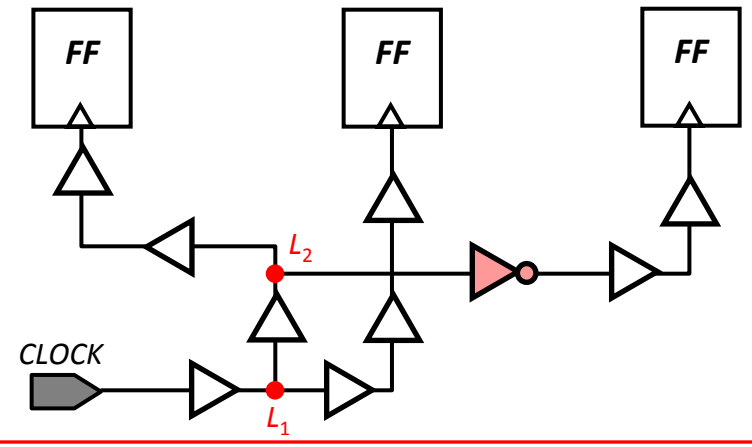
1 based on ICCAD 2014 benchmarks (~10^6 gates)

	Design	Number of:			
		PIs	POs	Gates	Nets
	cordic_ispd	34	64	45.4K	45.4K
	des_perf_ispd	234	140	138.9K	139.1K
	edit_dist_ispd	2.6K	12	147.6K	150.2K
	fft_ispd	1.0K	2.0K	38.2K	39.2K
	matrix_mult_ispd	3.2K	1.6K	155.3K	167.2K
	pci_bridge_32_ispd	160	201	40.8K	41.0K
	usb_phy_ispd	15	19	923	938
*	crc32d16N	19	32	478	495
	netcard_iccad	1.8K	10	1496.0K	1497.8K

**clock tree has inverters and buffers*

Added randomized clock tree [TAU 2014]

- **BRANCH**(*CLOCK*, *initial FF*)
- For each remaining FF
  - Select random location *L* in current tree
  - **BRANCH**(*L*,*FF*)
- **BRANCH**(`src,sink`): create buffer chain from `src` to `sink`



Operations File: Phase 1 and Phase 2

RepowerGate(numGates)

- repower_gate ×[numGates]

AddBuffer(numGates)*

- insert_gate
- insert_net
- disconnect_pin
- connect_pin
- read_spef

×[numGates]

RemoveBuffer(numGates)*

- disconnect_pin
- connect_pin
- remove_gate
- remove_net

×[numGates]

QueryTiming(numPaths)

- **report_at** all PIs, POs, all internal pins†
- **report_rat** all PIs, POs, all internal pins†
- **report_slack** all PIs, POs, all internal pins†
- **report_worst_paths** [numPaths]

*no clock tree

†verbose mode

```
QueryTiming(0);
RepowerGate(1);
AddBuffer(1);
QueryTiming(3);
RepowerGate(3);
AddBuffer(3);
QueryTiming(3);
RepowerGate(10);
AddBuffer(5);
QueryTiming(3);
RepowerGate(20);
AddBuffer(50);
QueryTiming(3);
RepowerGate(1000);
AddBuffer(5000);
QueryTiming(3);
RemoveBuffer(5);
QueryTiming(3);
RemoveBuffer(500);
QueryTiming(0);
```

Benchmarks: Evaluation

7 based on released benchmarks (~10^2 – ~10^5 gates)

3 based on Cadence benchmarks (~10^2 – ~10^5 gates)

6 based on ICCAD 2014 benchmarks (~10^5 – ~10^6 gates)

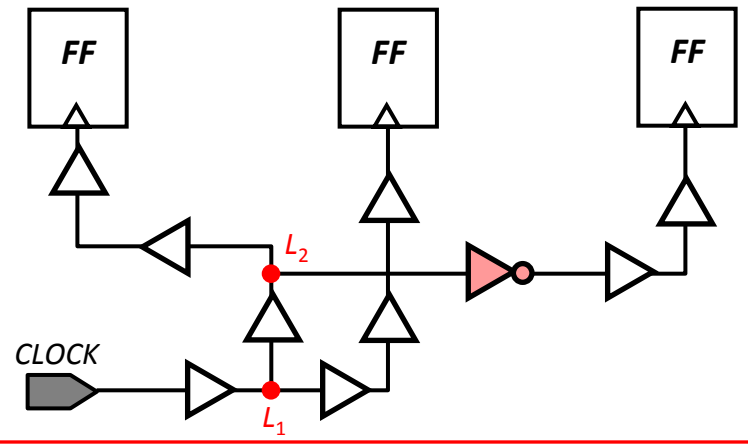
		Design	Number of: PIs	POs	Gates	Nets
		vga_lcd	89	109	139.5K	139.6K
		cordic_ispd	34	64	45.4K	45.4K
		des_perf_ispd	234	140	138.9K	139.1K
		edit_dist_ispd	2.6K	12	147.6K	150.2K
		fft_ispd	1.0K	2.0K	38.2K	39.2K
†	*	cordic_dut	80	78	3.6K	3.6K
	*	crc32d16N	19	32	478	495
†	*	softusb_navre	34	51	6.9K	7.0K
†	*	tip_master	778	869	37.7K	38.5K
†		b19_iccad	22	25	255.3K	255.3K
†		mgc_edit_dist_iccad	2.6K	12	161.7K	164.2K
†		mgc_matrix_mult_iccad	3.2K	1.6K	171.3K	174.5K
†		vga_lcd_iccad	85	99	259.1K	259.1K
		netcard_iccad	1.8K	10	1496.0K	1497.8K
†		leon2_iccad	615	85	1616.4K	1517.0K
†		leon3mp_iccad	254	79	1247.7K	1248.0K

**clock tree has inverters and buffers*

†hidden benchmark (not released)

Added randomized clock tree [TAU 2014]

- **BRANCH**(*CLOCK*, *initial FF*)
- For each remaining FF
  - Select random location *L* in current tree
  - **BRANCH**(*L*,*FF*)
- **BRANCH**`(src,sink)`: create buffer chain from `src` to `sink`



Operations File: Evaluation

QueryTimingEval(numPins,numPaths,numFFs,numPathsFF)

- **report_at** all PIs, POs, **[numPins]** internal pins
- **report_slack** all PIs, POs, **[numPins]** internal pins
- **report_worst_paths [numPaths]**
- **report_worst_paths -pin <random pin> [numPathsFF]** ✕ **[numFFs]**

```
QueryTimingEval(10000,1000,100,10);
RepowerGate(10000);
AddBuffer(10000);
QueryTimingEval(10000,1000,100,10);
RemoveBuffer(50000);
RepowerGate(100000);
AddBuffer(100000);
QueryTimingEval(10000,10000,100,10);
```

Evaluation

Accuracy (Compared to Golden Results)

PIN: all timing points in Design D

PATH: paths in Design D

Accuracy Score (Difference)

[0,0.1]	ps	100
(0.1,0.5]	ps	80
(0.5,1.0]	ps	50
(1,∞)	ps	0

Block-based Value Accuracy A(*PIN*)

- Arrival time
- Post-CPPR slack

Path-based Accuracy A(*PATH*)

- Path correctness (75%)
- Post-CPPR path slack (25%)

Accuracy of Design A(*D*)

$$A(D) = \frac{A(PIN) + A(PATH)}{2}$$

Runtime Factor (Relative)

$$RF(D) = \frac{MAX\_R(D) - R(D)}{MAX\_R(D) - MIN\_R(D)} \times R(D)$$

Memory Factor (Relative)

$$MF(D) = \frac{MAX\_M(D) - M(D)}{MAX\_M(D) - MIN\_M(D)} \times M(D)$$

Composite Design Score

$$score(D) = A(D) \times (70 + 20 \times RF(D) + 10 \times MF(D))$$

Overall Contestant Score is average over all design scores

TAU 2015 Contestants

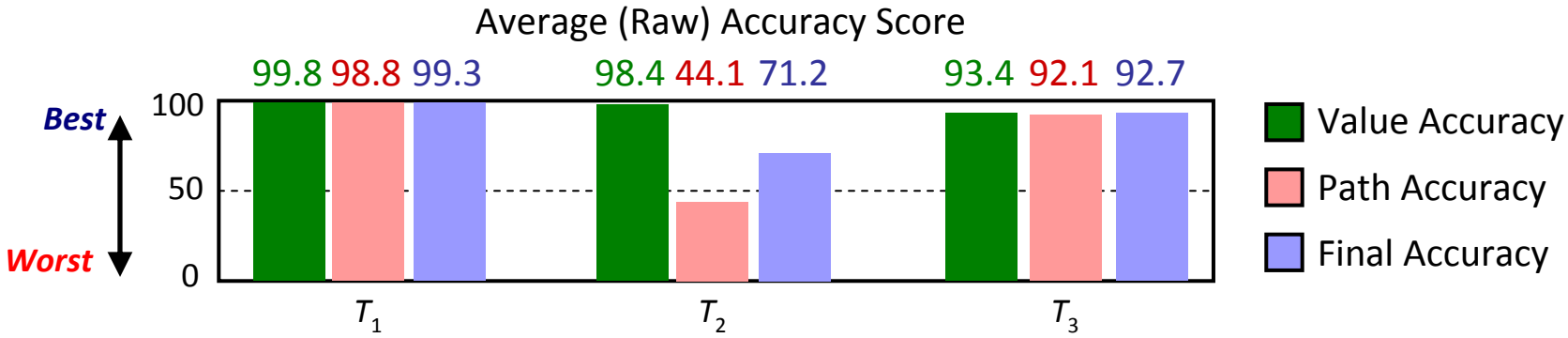
[12 Teams, 10 Universities, 6 Countries]

	University	Country	Team Name
	Georgia Institute of Technology	USA	GeorgiaTech
	University of Illinois at Urbana-Champaign	USA	UI-Timer 2.0
	Drexel University	USA	SAPPER
	Moscow Institute of Physics and Technology	Russia	School146
	University of Thessaly	Greece	U-Thessaly
	India Institute of Technology, Madras	India	Timer_IITM
	India Institute of Technology, Madras	India	iitRACE
	India Institute of Technology, Hyderabad	India	IIITimer
	Universidade Federal do Rio Grande do Sul	Brazil	UFRGS-Brazil
	National Chiao Tung University	Taiwan	iTimerC 2.0
	National Tsing Hua University	Taiwan	NN
	National Tsing Hua University	Taiwan	NTU

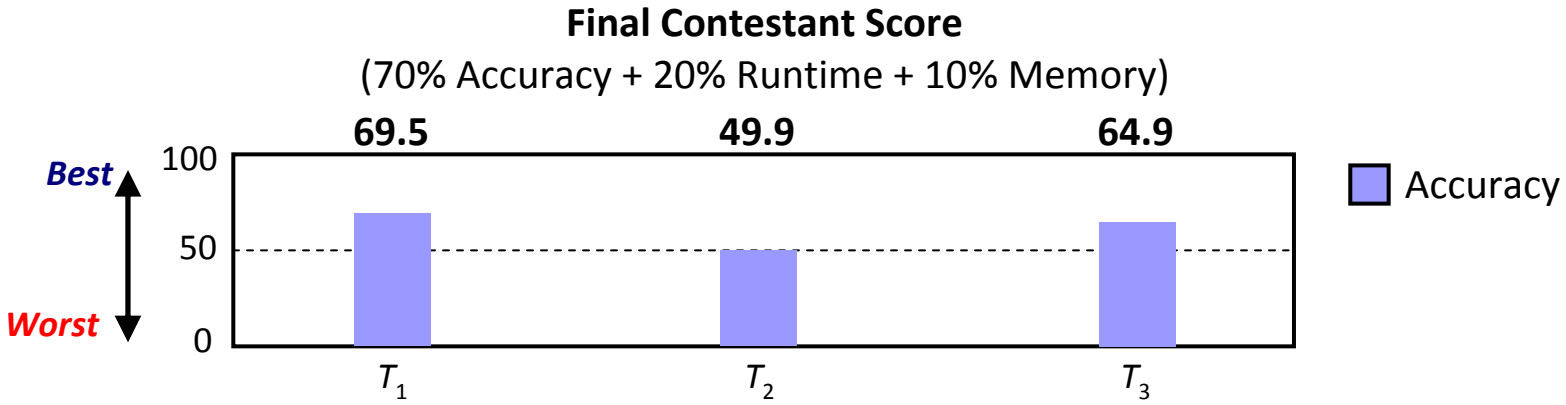
Contestant Results – Accuracy

Many entered, few survived – but excellent quality

- 12 teams down to 4 teams: top 3 team results presented
- Each team tested against 16 benchmarks – only 1 crash out of all 48 runs



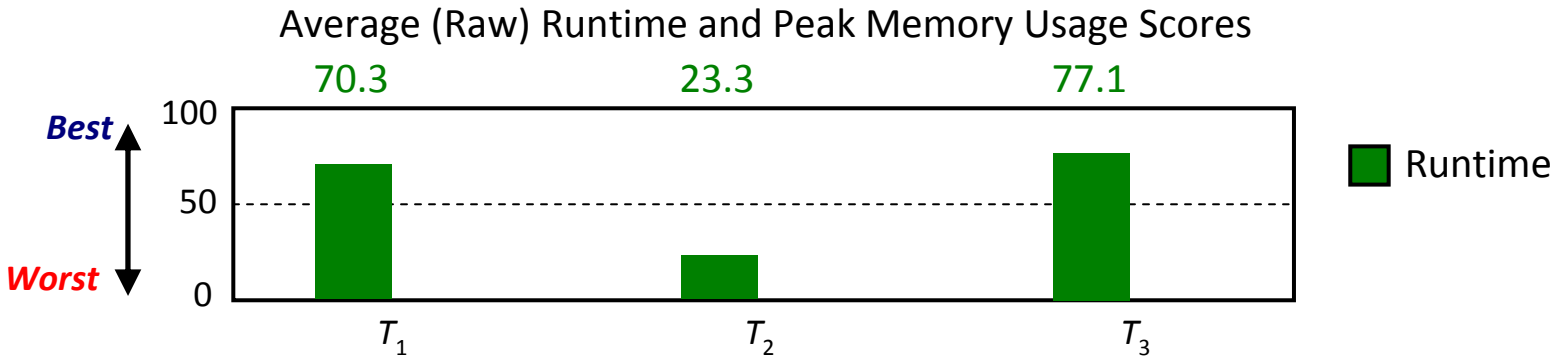
- **Perfect** value accuracy on 9 of 16 benchmarks
- Vast majority of accuracy scores **99%+**



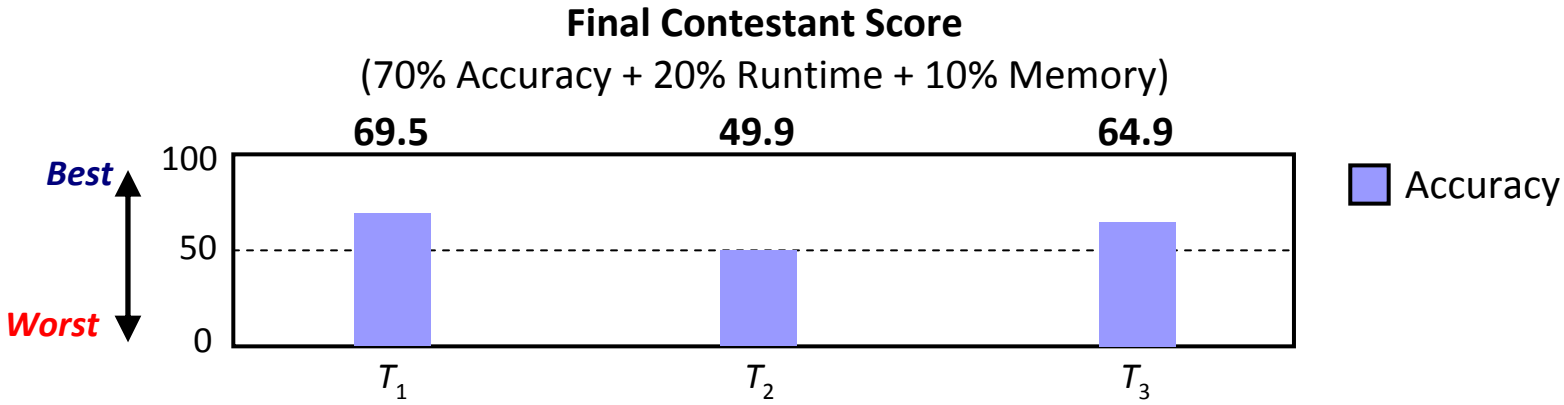
Contestant Results – Performance

Many entered, few survived – but excellent quality

- 12 teams down to 4 teams: top 3 team results presented
- Each team tested against 16 benchmarks – only 1 crash out of all 48 runs



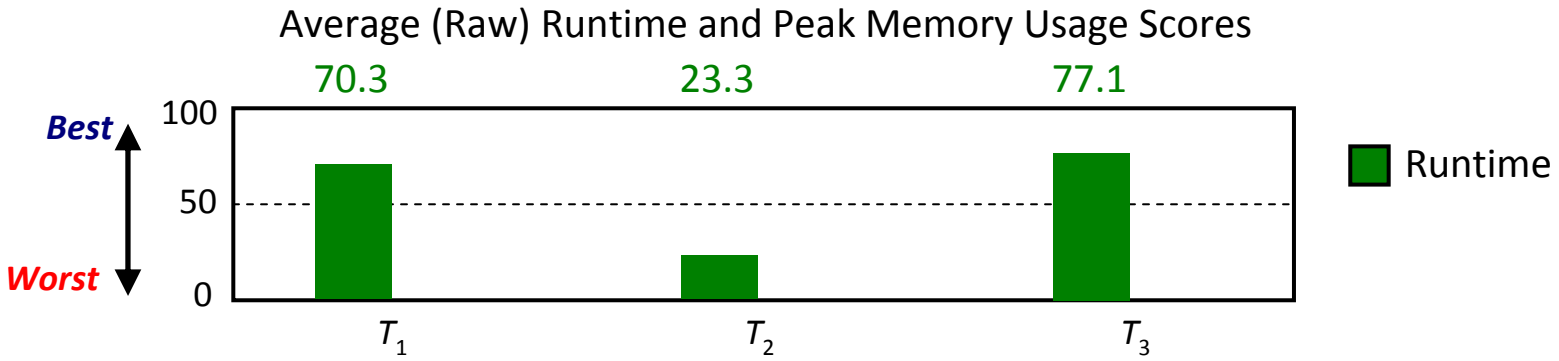
- Evaluation Machine: 16X *Intel(R) Xeon(R) CPU E5-2667 v2 @3.30 GHz*
- T_1 used 8 threads [>1 hour]; T_2 used 1 thread [~9 hours]; T_3 used 1 thread [<1 hour]



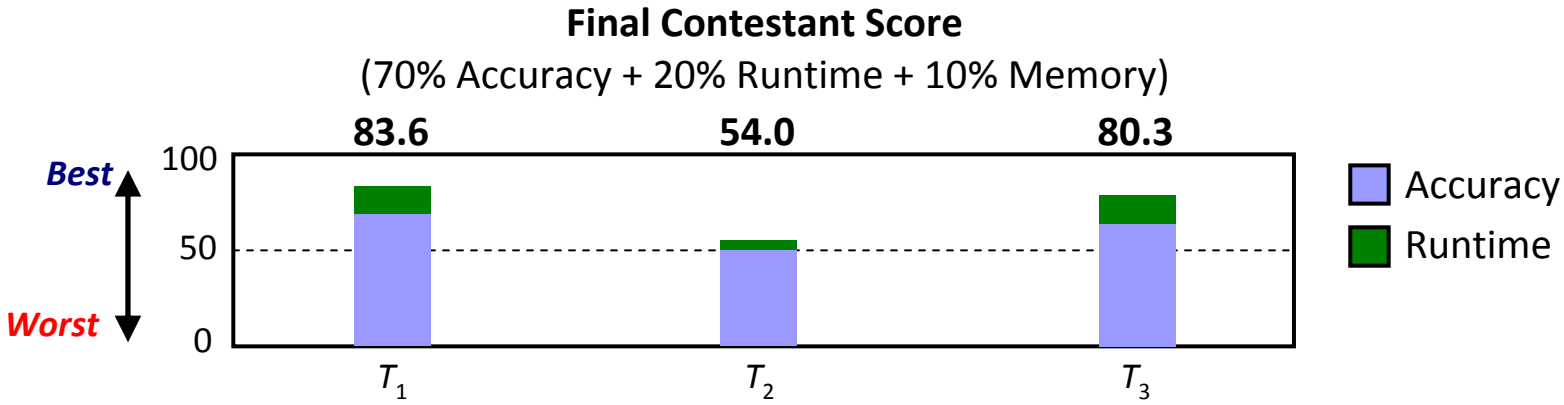
Contestant Results – Performance

Many entered, few survived – but excellent quality

- 12 teams down to 4 teams: top 3 team results presented
- Each team tested against 16 benchmarks – only 1 crash out of all 48 runs



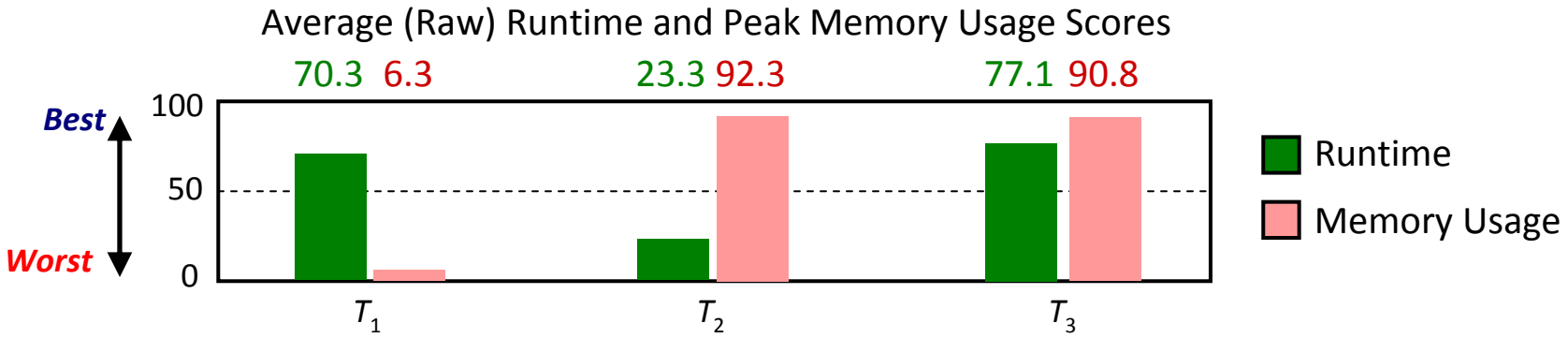
- Evaluation Machine: 16X *Intel(R) Xeon(R) CPU E5-2667 v2 @3.30 GHz*
- T_1 used 8 threads [>1 hour]; T_2 used 1 thread [~9 hours]; T_3 used 1 thread [<1 hour]



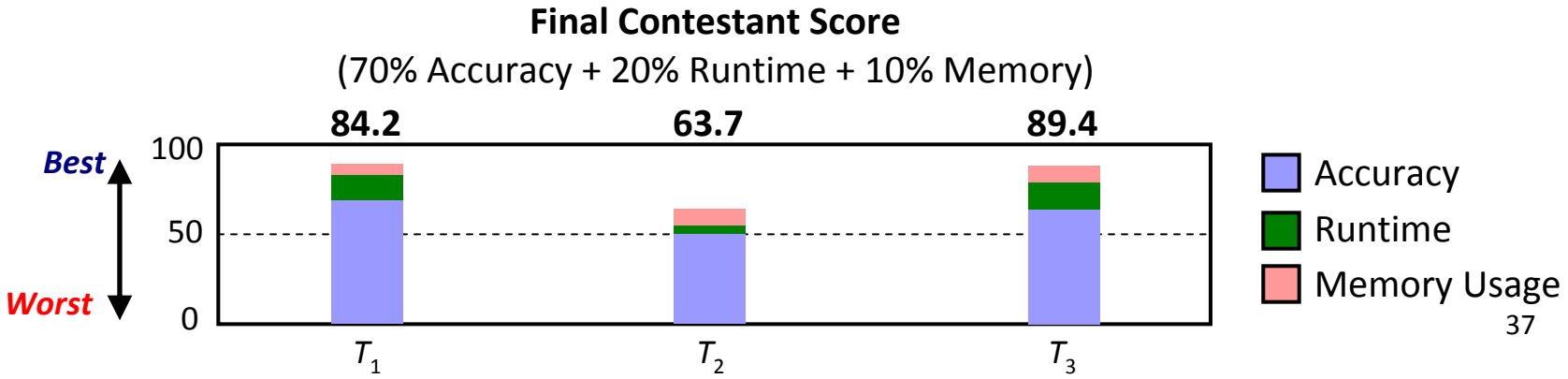
Contestant Results – Performance

Many entered, few survived – but excellent quality

- 12 teams down to 4 teams: top 3 team results presented
- Each team tested against 16 benchmarks – only 1 crash out of all 48 runs



- Evaluation Machine: 16X *Intel(R) Xeon(R) CPU E5-2667 v2 @3.30 GHz; a lot of memory*
- **leon2** benchmark: T_1 used ~350 GB, T_2 used ~10 GB, T_3 used ~30 GB



Acknowledgments

↳Greg Schaeffer, Vibhor Garg [TAU 2015 Contest Committee]

↳Debjit Sinha, Igor Keller [TAU 2015 Workshop Committee]

The TAU 2015 Contestants

*This contest would not have been successful without your **hard work** and **dedication***

and the winners are…

TAU 2015

Timing Contest on Incremental Timing and CPPR Analysis

Third Place Award

Presented to

Chaitanya Peddawad, Aman Goel,
Dheeraj B, and Nitin Chandrachoodan

For

iitRACE

IIT Madras, India

Igor Keller	*Debjit Sinha*	*Jin Hu*
General Chair	Technical Chair	Contest Chair

TAU 2015

Timing Contest on Incremental Timing and CPPR Analysis

Second Place Award

Presented to

Tsung-Wei Huang and Martin D. F. Wong

For

UI-Timer 2.0

University of Illinois at Urbana-Champaign, USA

Igor Keller	*Debjit Sinha*	*Jin Hu*
General Chair	Technical Chair	Contest Chair

TAU 2015

Timing Contest on Incremental Timing and CPPR Analysis

First Place Award

Presented to

Pei-Yu Lee, Cheng-Ruei Li, Wei-Lun Chiu,
Yu-Ming Yang, and Iris Hui-Ru Jiang

For

iTimerC 2.0

National Chiao Tung University, Taiwan

Igor Keller	*Debjit Sinha*	*Jin Hu*
General Chair	Technical Chair	Contest Chair

Contest Achievements and Reflections

- Modern (large) **benchmarks** and **infrastructure** for timing research
  - Enables extensions for timing **features** (e.g., statistical, cell modeling)
  - Support **timing-driven** applications (e.g., placement and routing)
- Open-source timers that have **block-based** and **path-based** capabilities
  - Supports **industry-standard** input formats (e.g., Verilog, Liberty)
  - High-performance and robust software – **multi-threaded** implementations
- Recognition of **talent**
  - Intelligent, **motivated**, driven (and polite!) contestants
  - Teams capable of **solving** difficult problems in a few months – not easy!
- **Open forum** for directed and **relevant** industry problems
  - Platform for **proposing** challenging problems – TAU 2016 and beyond