RAPID EARLY DESIGN SPACE EXPLORATION USING LEGACY DESIGN DATA, TECHNOLOGY SCALING TREND AND IN-SITU MACRO MODELS

Charles V.K. Thangaraj

Department of Electrical and Computer Engineering

Ph.D. Final Examination

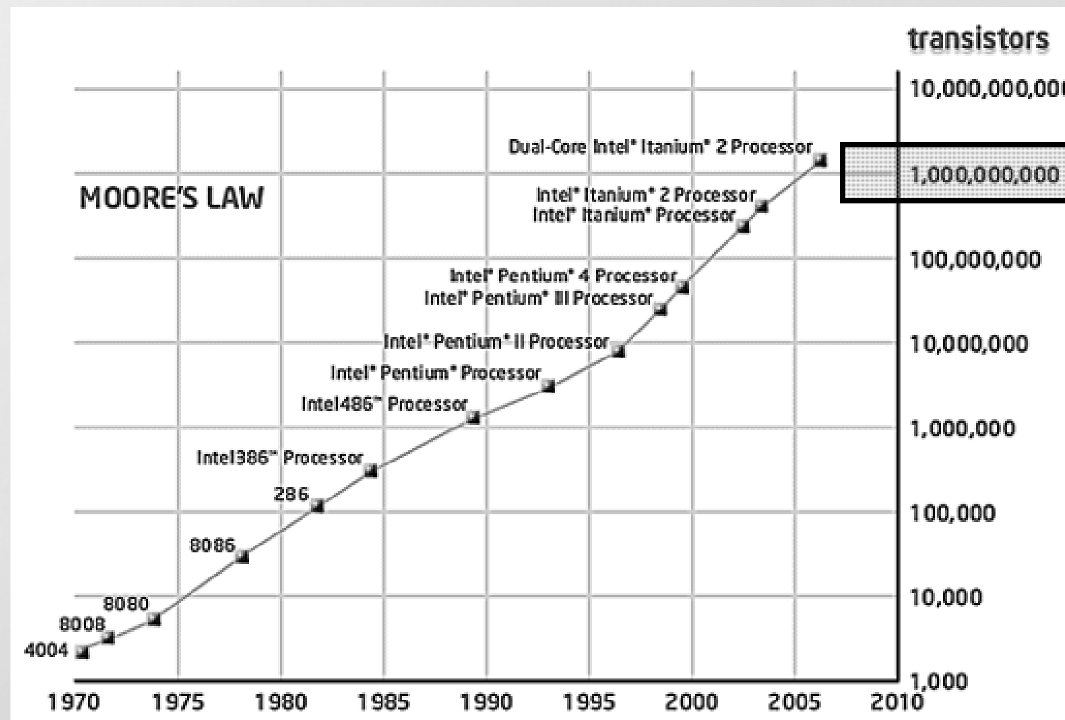
Committee Members

- Prof. Tom Chen
- Prof. George Collins
- Prof. Antony Maciejewski
- Prof. Phillip Chapman

Presentation Outline

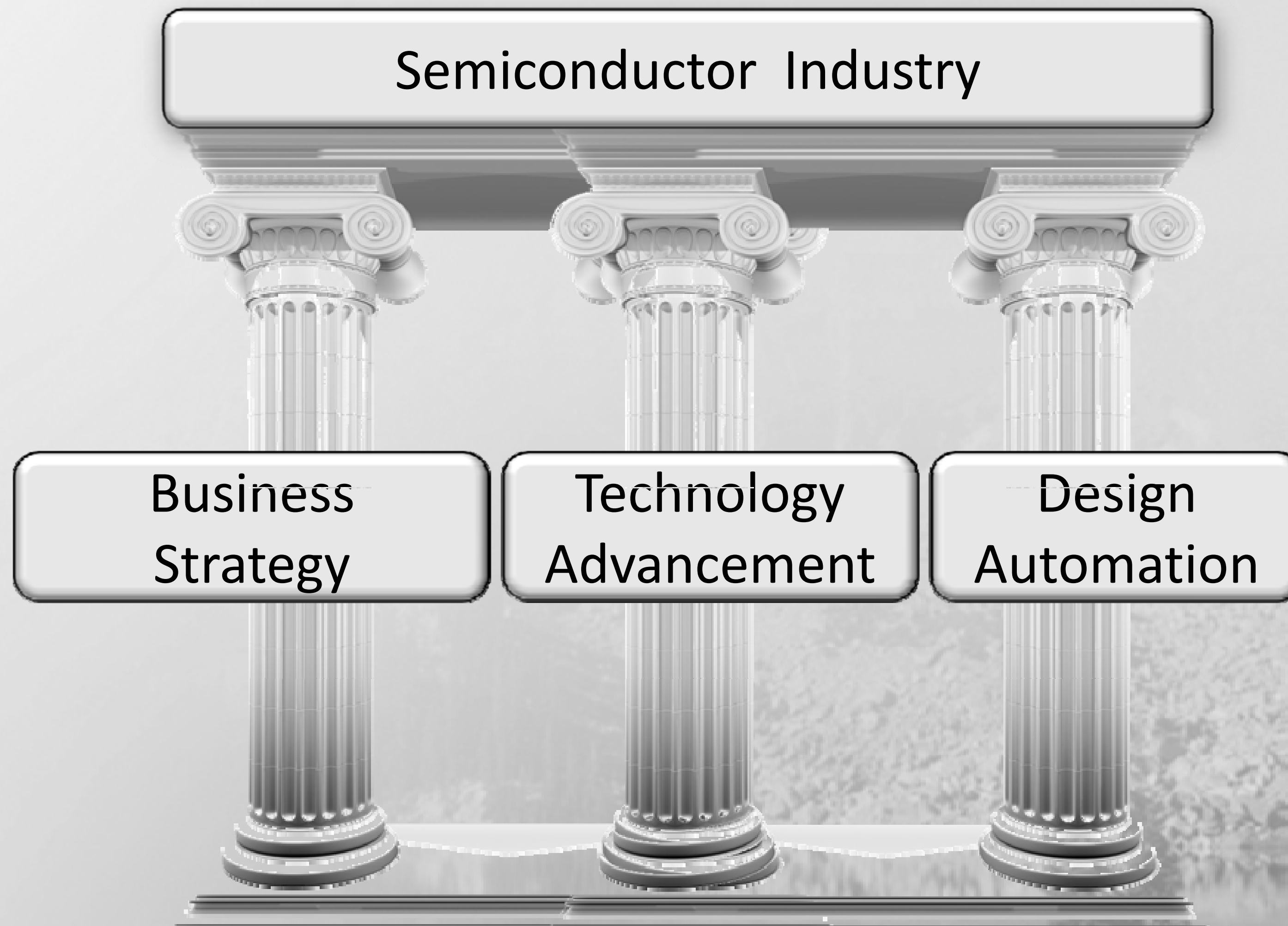
- Introduction
- Motivation & Objective
- Proposed Approach
- Experimental Setup
- Results & Discussions
- Conclusion & Future Work

Moore’s Law – 1965 to ?



- No. of devices and operating frequency double every 18 months
  - faster FETs, higher device density, higher performance, lower cost
  - Will this trend continue?

Survival of the Semiconductor Industry



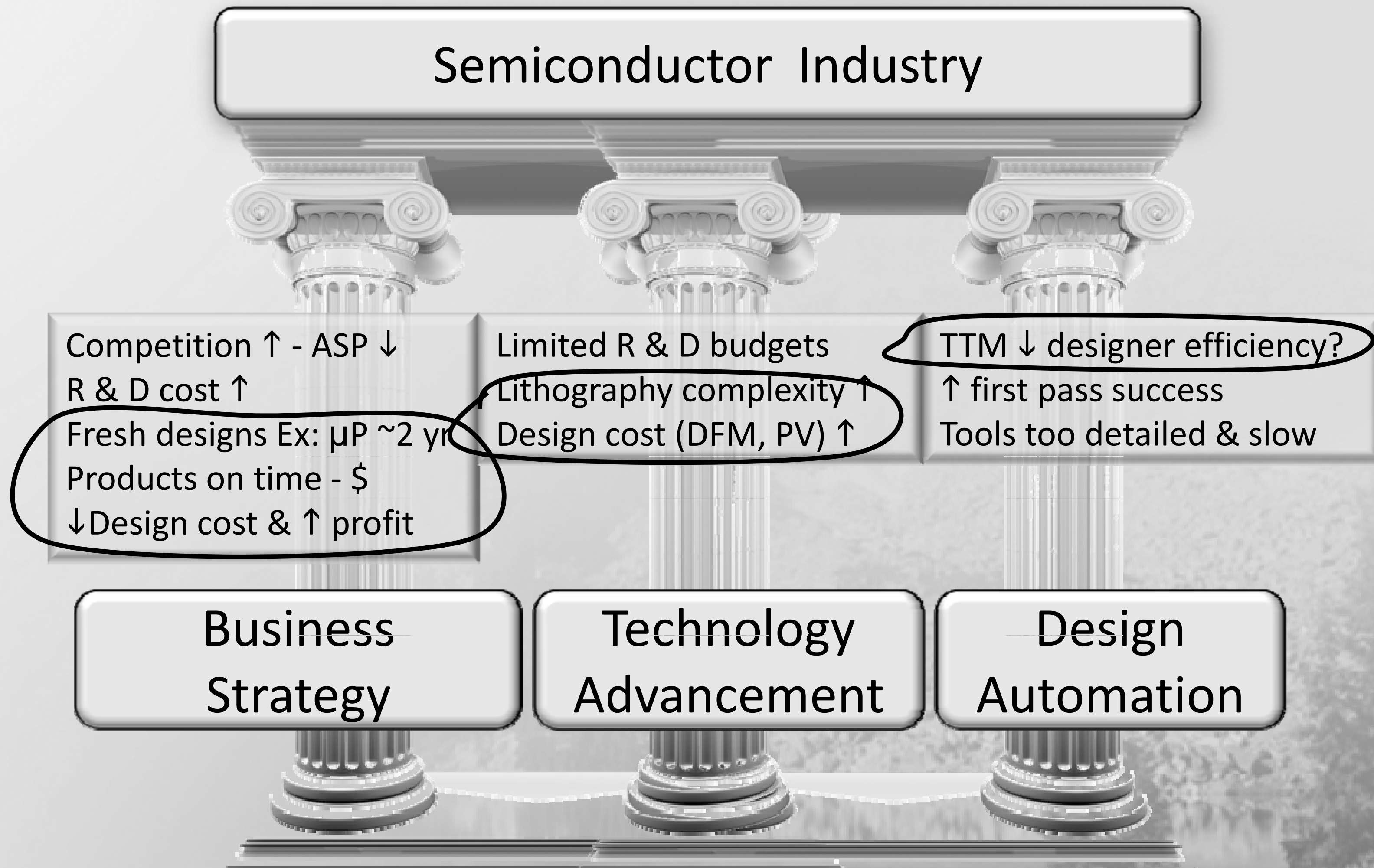
Semiconductor Industry Trends & Challenges: Business

- Business trends
  - 10X revenue growth in the past decade
  - Growth to US $700 B from US $200 B in the next decade
  - Newer & emerging markets are opening up
    - Auto electronics, medical, GPS, intelligent appliances etc.
- Business challenges
  - Capital overhead - exponentially increasing fab costs
  - R&D overhead - process development expenditure increasing
  - Design cost reduction to max profit - min NRE (non recurring engineering expenditure)
  - Competition – price point, product quality and time-to-market schedule pressures
  - Opportunity cost – need to maintain market segment share in high volume low cost segment
  - Need to innovate to stay in business
- Engineering decisions are being increasingly influenced by business needs !!
  - Optimize designs for specific market segments i.e. design target trade-offs
    - Cost/area–performance–power–reliability–yield

Semiconductor Industry Trends & Challenges: Tech & EDA

- Technology challenges influencing design convergence
  - Vdd – Vt gap closes -- leakage power, SRAM stability, computation reliability, noise etc
  - Increase in transistor density -- thermal instabilities, yield issues
  - Interconnect bottleneck
  - Manufacturing process variation (Vt, L, tox) – design guard banding
  - Sub-wavelength lithography
    - DFM rules explosion, OPC, PSM, OAI, I-litho, EUV-litho, SRAF
- EDA tool challenges influencing design optimization
  - Existing EDA tools
    - Aimed at speeding conventional design process - obsolete methodologies
    - accurate, very specific and detailed -- less flexibility
  - Limited use of system level design modeling & optimization
    - Concoct such a tool with existing tools
    - computationally expensive
    - very limited opportunity for design space exploration
    - Interoperability overhead, time consuming approach
- A design team should meet business need, overcome challenges and achieve design convergence on time !!

The Challenge

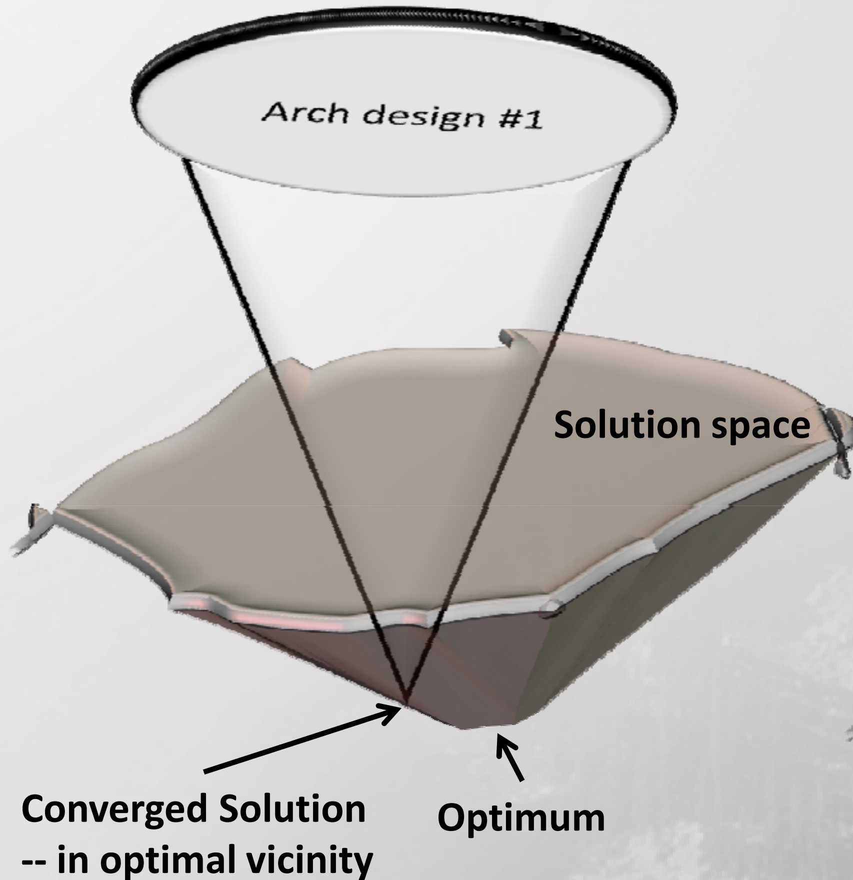


Presentation Outline

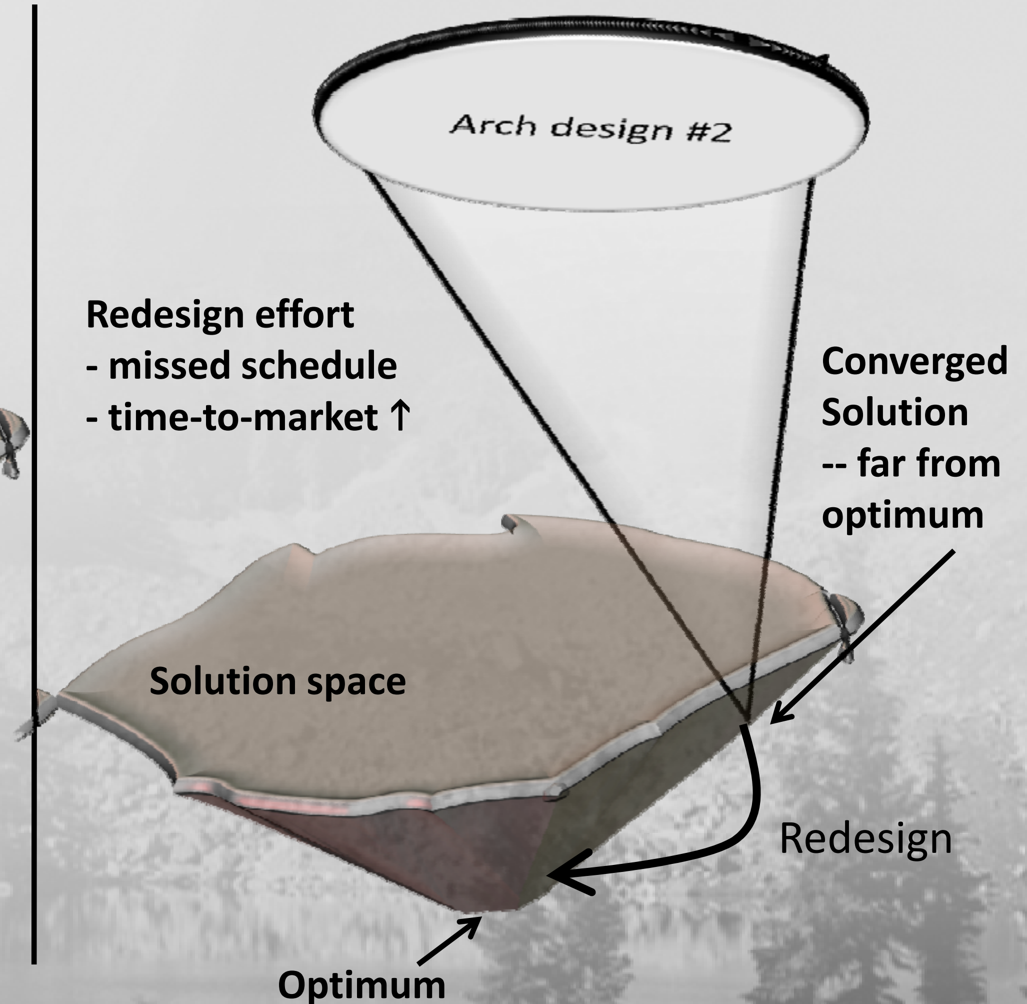
- Introduction
- Motivation & Objective
- Proposed Approach
- Experimental Setup
- Results & Discussions
- Conclusion & Future Work

Correct Early Stage Design Plan

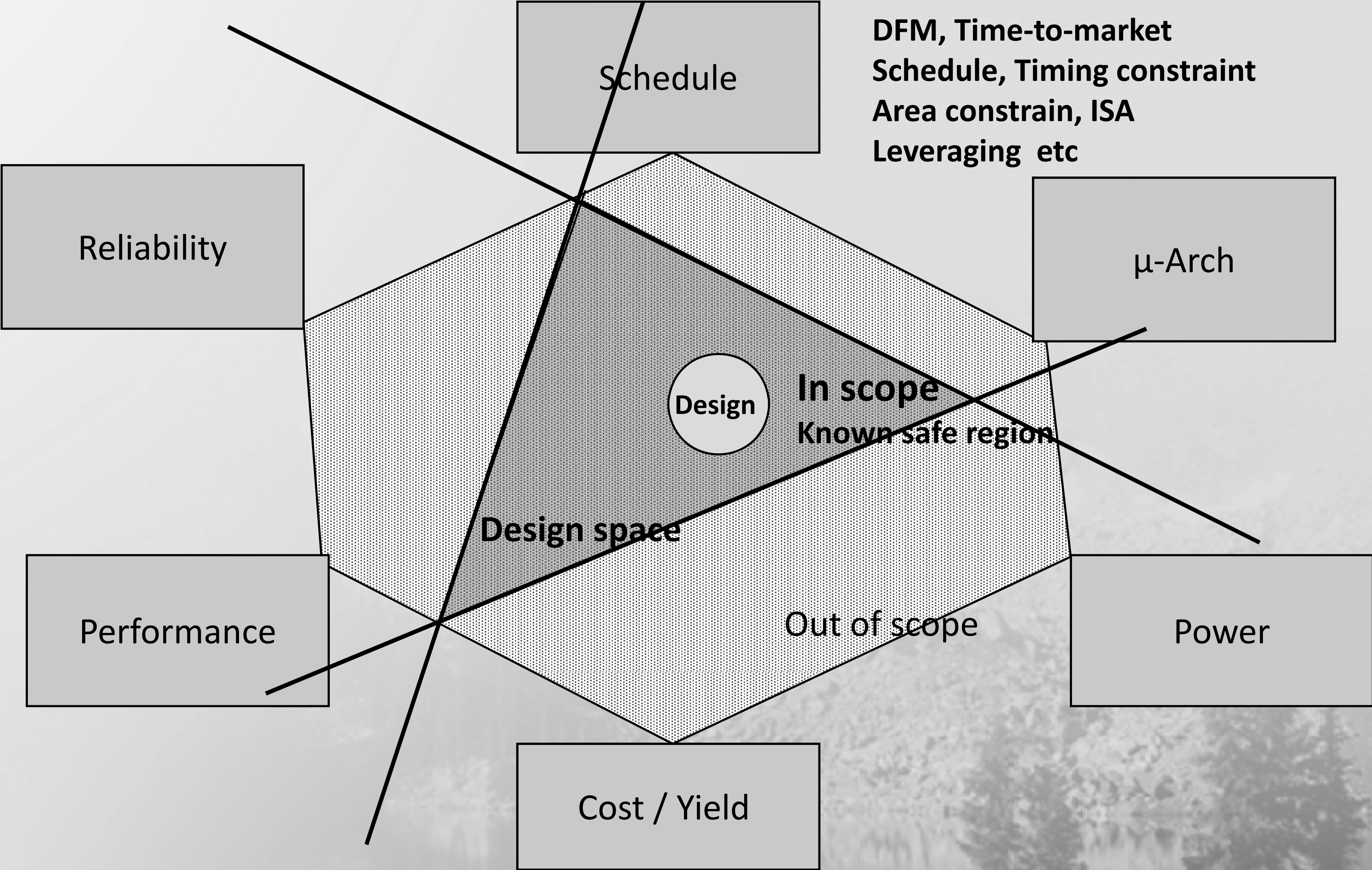
System level optimization considering *low level implementation details*



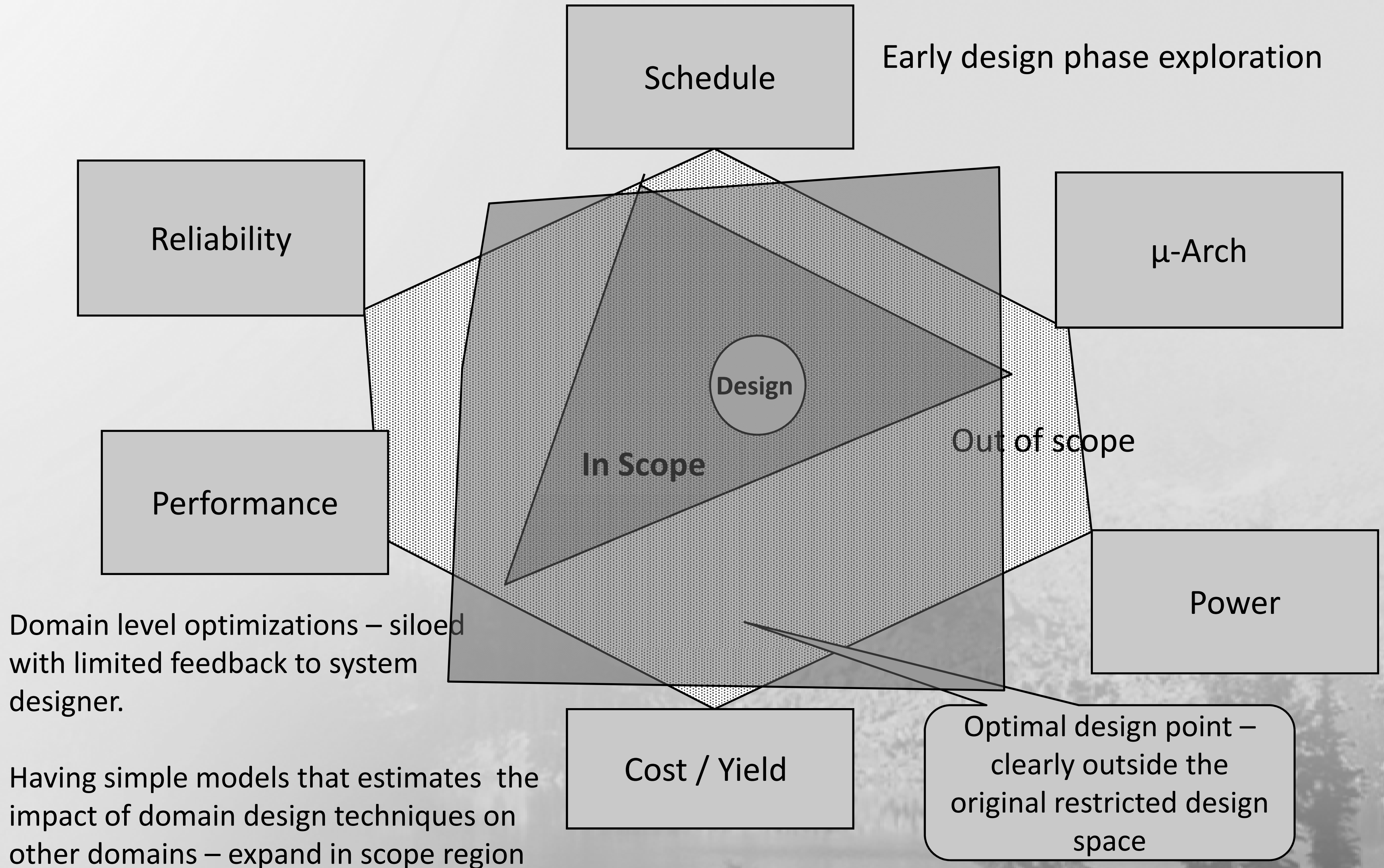
System level optimization NOT considering *low level implementation details*



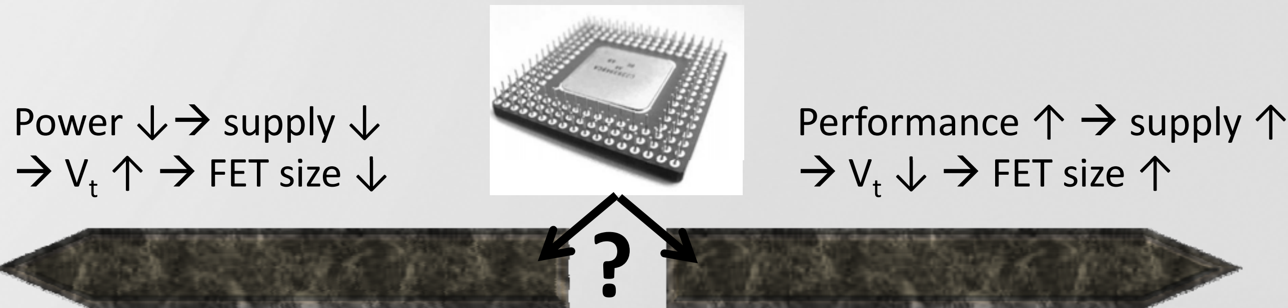
Current Approach



A Better Approach



Competing Design Goals



- Nano-meter CMOS
  - Power consumption
  - Process variation
  - High litho/mask costs – fewer re-spins
- Get it right the first time – key to success of modern high performance designs
  - Correct early stage design plan
- Early design phase, design target tradeoff analysis considering low level implementation details
  - Improves design convergence.
  - Guarantees time-to-market
  - Helps in avoiding costly redesigns in later design stages

Problem Statement And Research Objective

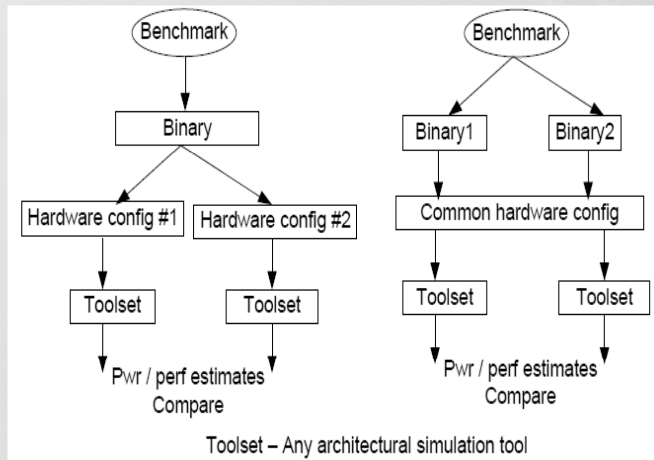
- To develop a modular system level modeling methodology
  - Simple yet effective for trade-off analysis and design optimization
  - Considers low level implementation details
  - Able to handle large designs i.e. scalable
- To develop analytical design target prediction models and module descriptors with inter-domain (design target) impact estimation capability
  - Estimate system dynamic and leakage power
  - Estimate system performance
- Software development to implement the methodology and tool flow
  - Enable quick design evaluation and design space exploration
  - Utilizing macro-model generation and in-situ SPICE simulations
- Experimentally demonstrate & validate the proposed methodology

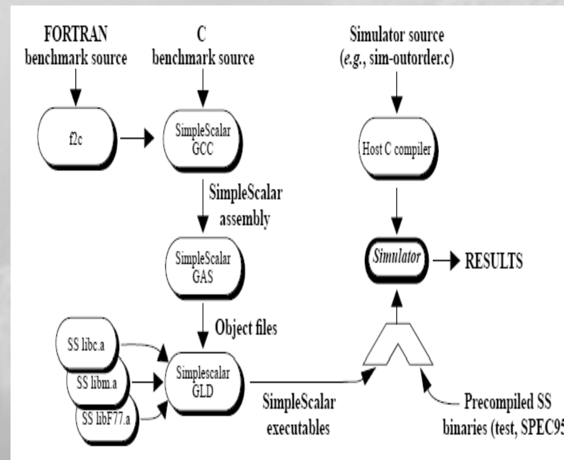
Presentation Outline

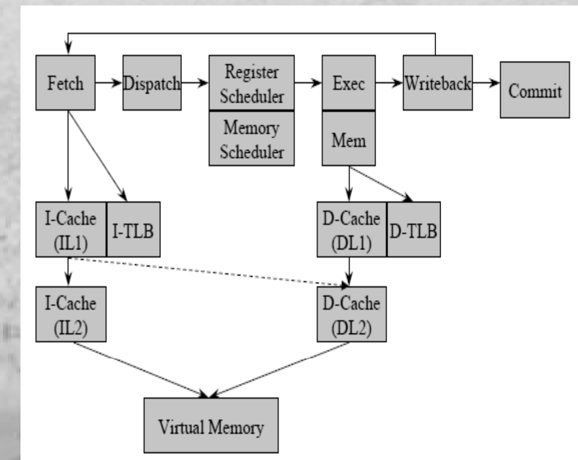
- Introduction
- Motivation & Objective
- Proposed Approach
- Experimental Setup
- Results & Discussions
- Conclusion & Future Work

Existing Tools For System Level Design Optimization

- Two broad categories
  - Computation engine focused
    - * SimplePower, SimpleScalar, Wattch, AccuPower and PowerTimer
  - ASIC focused
    - * BACPAC – most widely know and other developmental in-house tools
- Computation engine focused tools, model an underlying computation architecture and emulate code execution on the modeled architecture.
- They collect activity rates, instruction execution rates, miss rates, and prediction efficiency to estimate cycles per instruction (CPI) and other performance metrics







Existing Tools For System Level Design Optimization

- ACIC focused tools, estimate power and performance based on low level physical design parameters and key process parameters.
- BACPAC which is based on low-level physical design parameters attempts to "re-create" the design in a bottom-up manner for analysis.

Critical path delay = (Total global wire length *times* unit length delay) + (# of gates in critical path × gate delay) * (1 + 0.05 + 0.1).

$$Power_{short_circuit} = \#\text{ of gates } \times 1/2 \times T_{short_circuit} \times I_{peak} \times V_{dd} \times f \times alpha$$

$$P_{dyn_tot} = P_{blocks} + P_{global} + P_{clock} + P_{memory} + P_{IO_pads}.$$

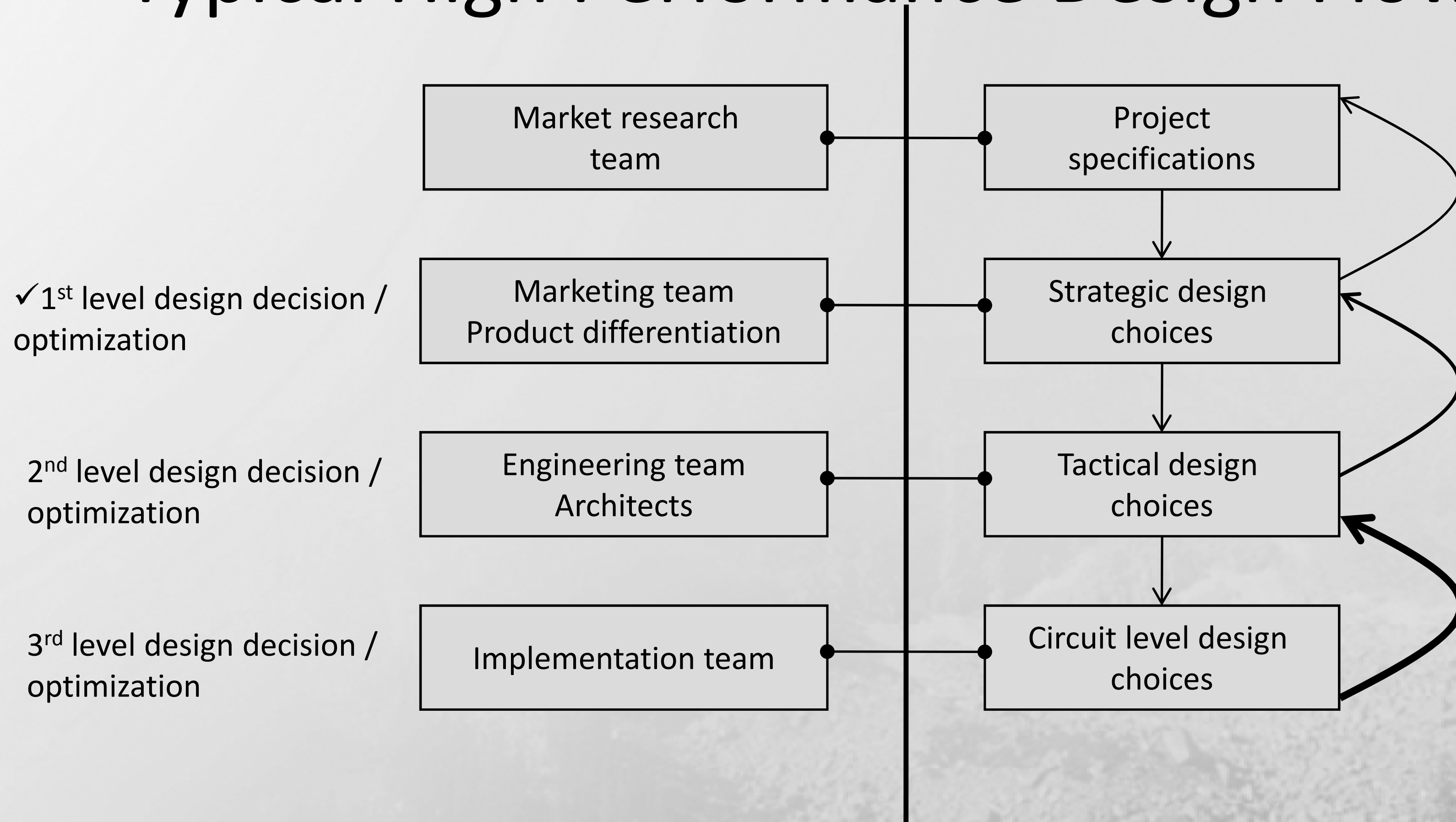
$$Power_{leakage} = W_{device} \times 10\mu A/\mu m \times 10^{-V_t/95mV}.$$

Existing Tools – Shortcomings

- Computation architecture optimization without physical implementation constraints do not offer any benefits in achieving design convergence in power, performance, area etc.
  - SimplePower uses pre-characterized power tables to estimate power.
    - unsuitable for DSE with changing low-level implementation details
  - Clock network power not included in SimplePower. Up to 30% of the total power.
  - AccuPower requires complete layout information. nonexistent during early design phase
  - Wattch & PowerTimer have a constant “hold” power equation to estimate leakage power.
    - leakage power is increasing exponentially in nanometer CMOS
  - PowerTimer is very slow
    - Power macros need to be re-characterized with every change in low-level implementation

- Optimization tools for physical implementation based on low-level physical design parameters are limited in their ability to explore the system design space.
  - BACPAC’s uses equi-partition of designs
    - not suitable for highly modular microprocessors and high performance designs.

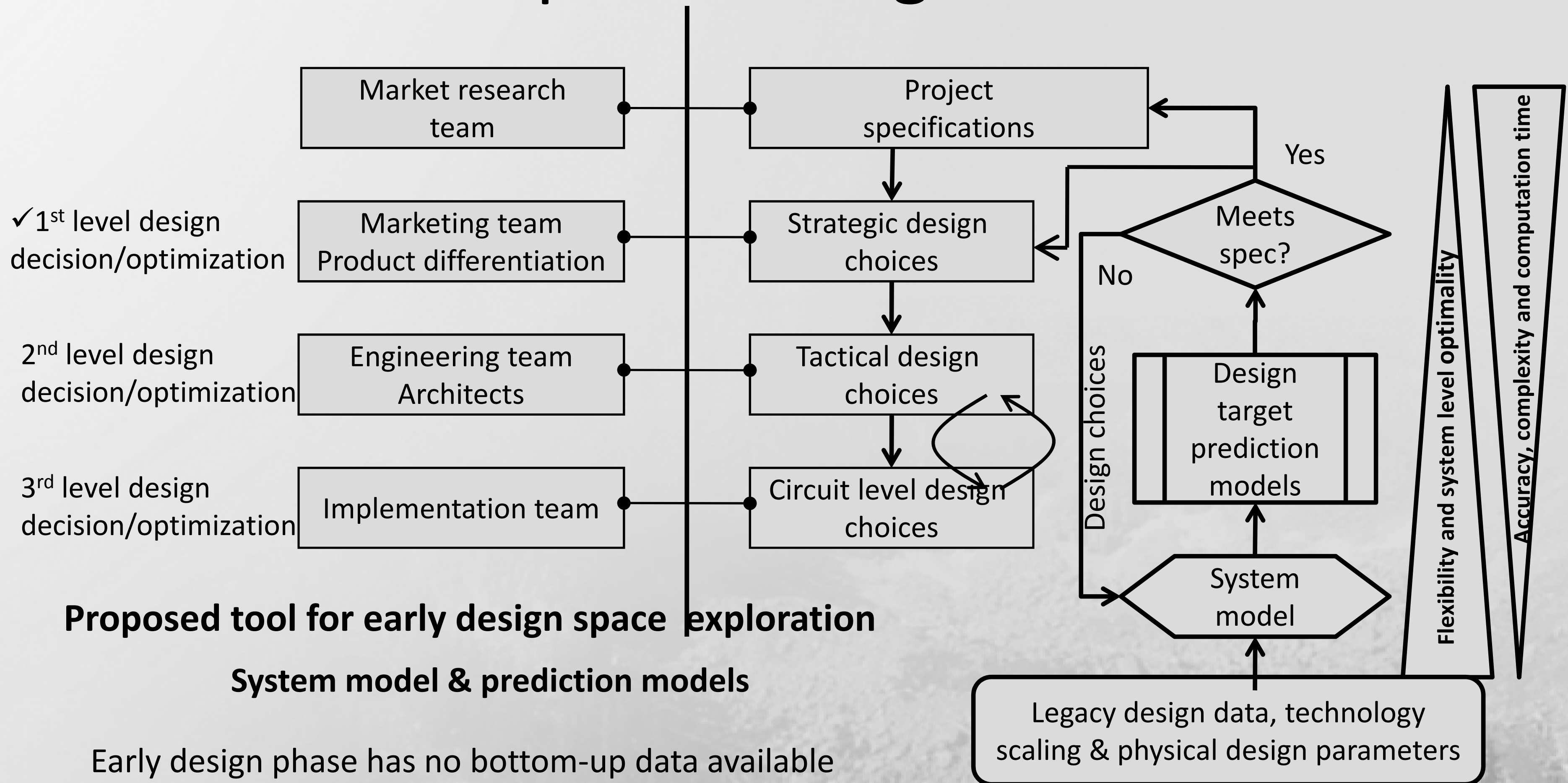
Typical High Performance Design Flow



Early design phase has no low level implementation (bottom-up) data available

Early design decision are made without bottom-up data for validation
→Increasing probability of a) circuit level design changes in the later design stages
→ b) product delivery delay or compromising product competitiveness

Proposed Design Flow



Early design phase has no bottom-up data available

DSE to set project specification

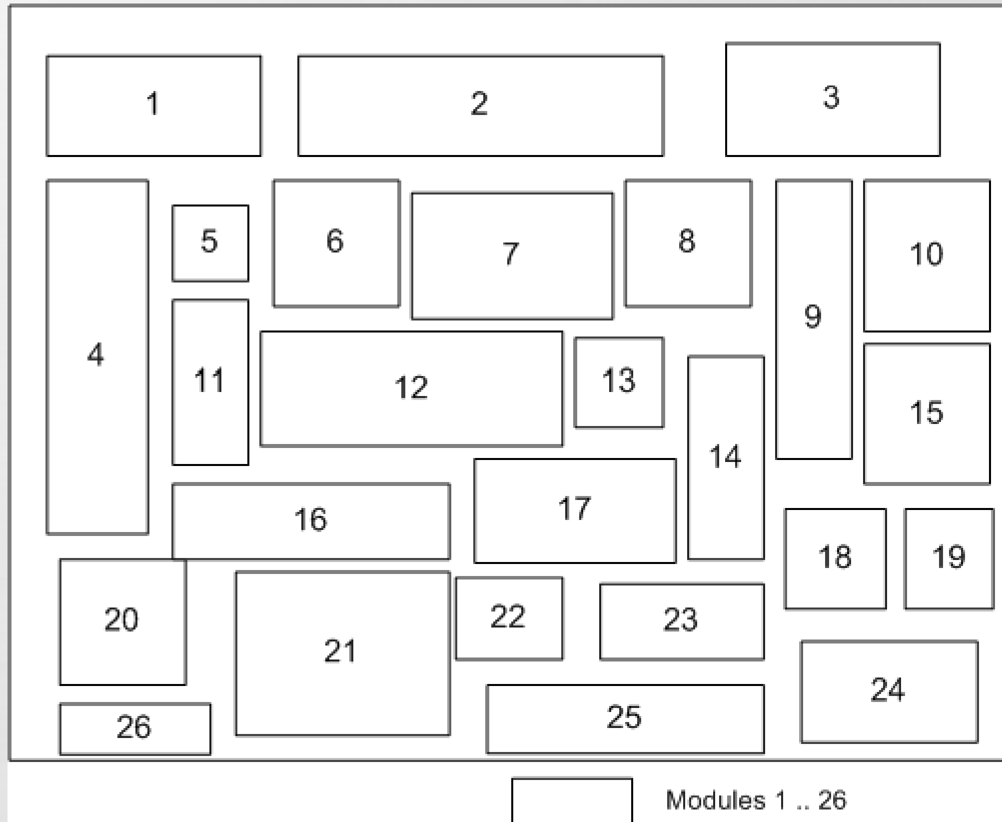
** legacy design data

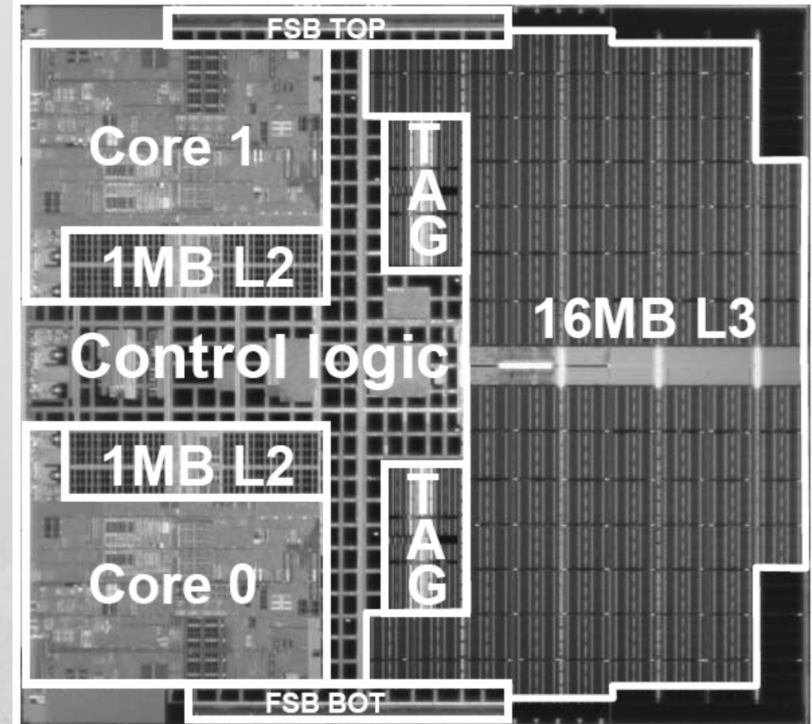
** scaling trends

** physical design data (abstracted low level implementation details)

Leads to Correct Early Stage Design Plan

System Modeling Methodology





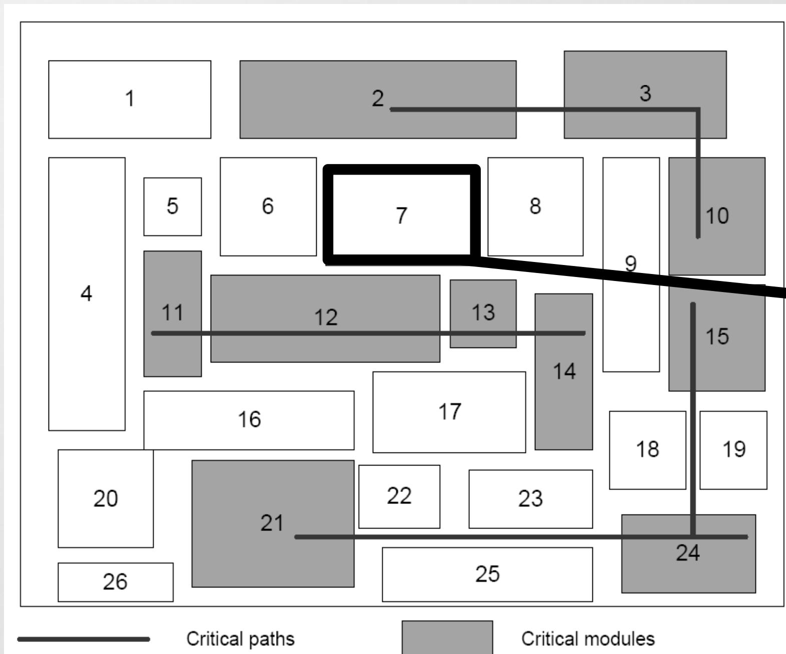
A generic system with 26 functional units or modules.

Leveraged designs ~ legacy information readily available.
- ** # of module, sizes
- ** critical modules and paths

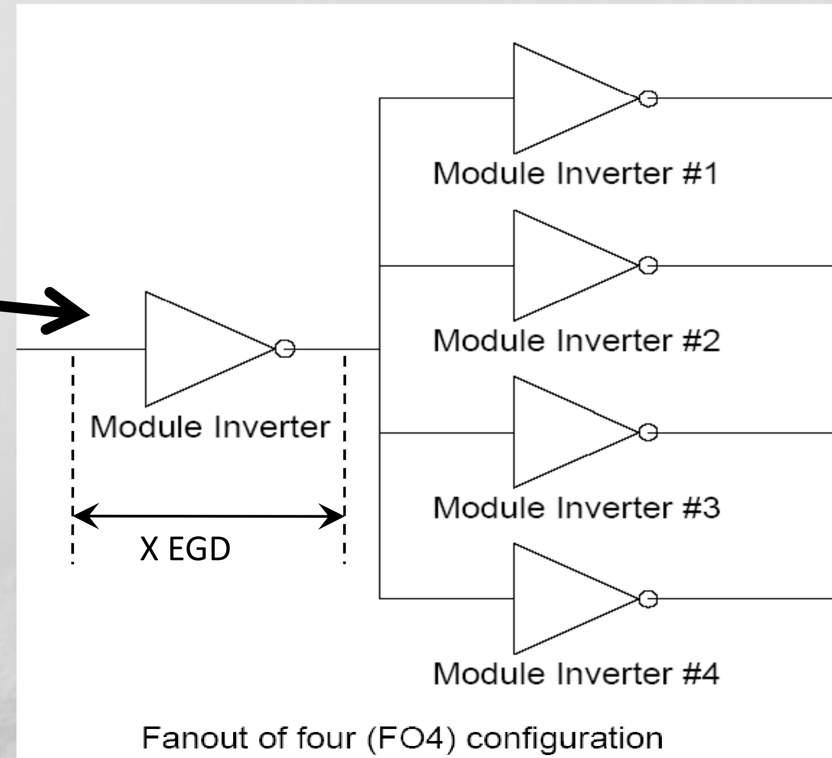
Die picture of an actual microprocessor implemented in a 100 nm process. Shown at 15th Int. Conf. on VLSI Design, 2002

Therefore a system is a collection on modules.

System Modeling Methodology



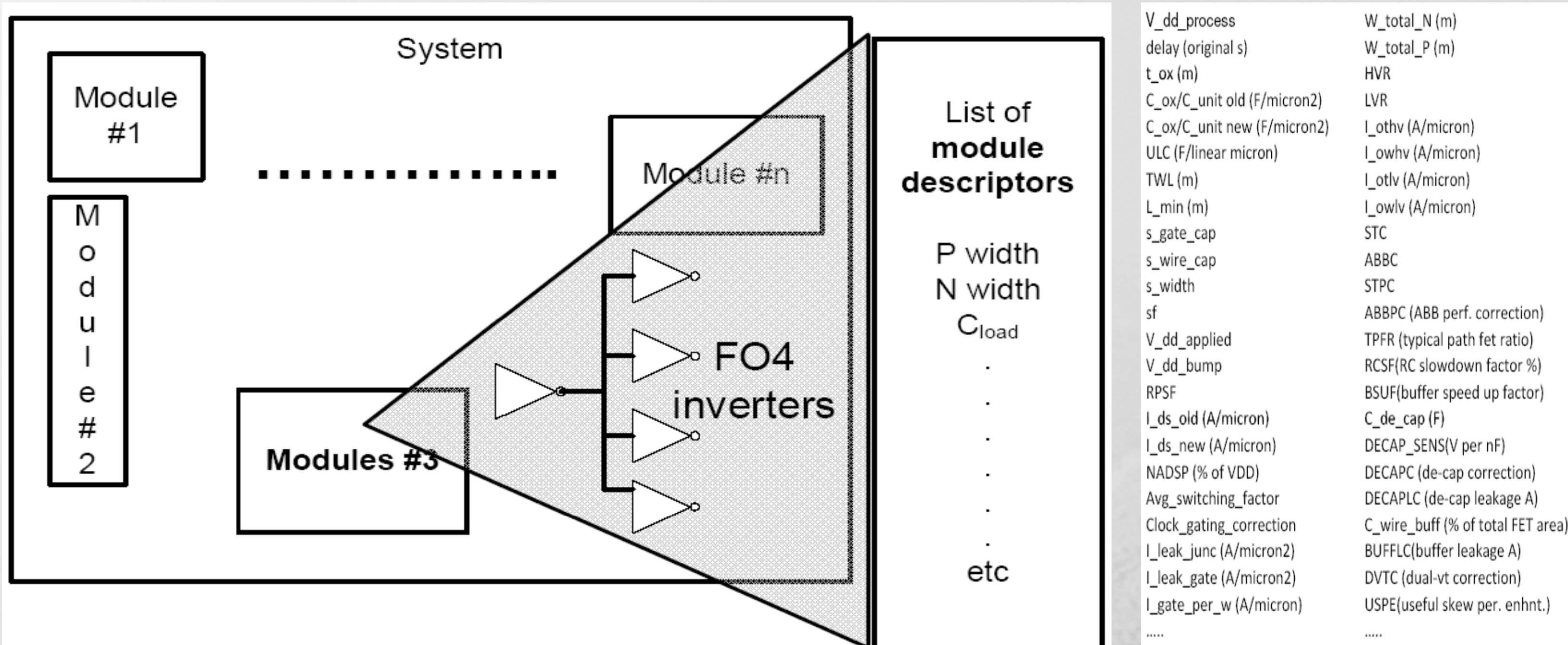
Module critical path delay = X EGDs



- Module ~ fanout-of-4 equivalent logic gate (inverter) configuration
- **Simple yet effective abstraction

  **Module inverter size proportional to total module N & PFET size

System Modeling Methodology



Each module is described by a set of descriptors (descriptor vector) ~ 60

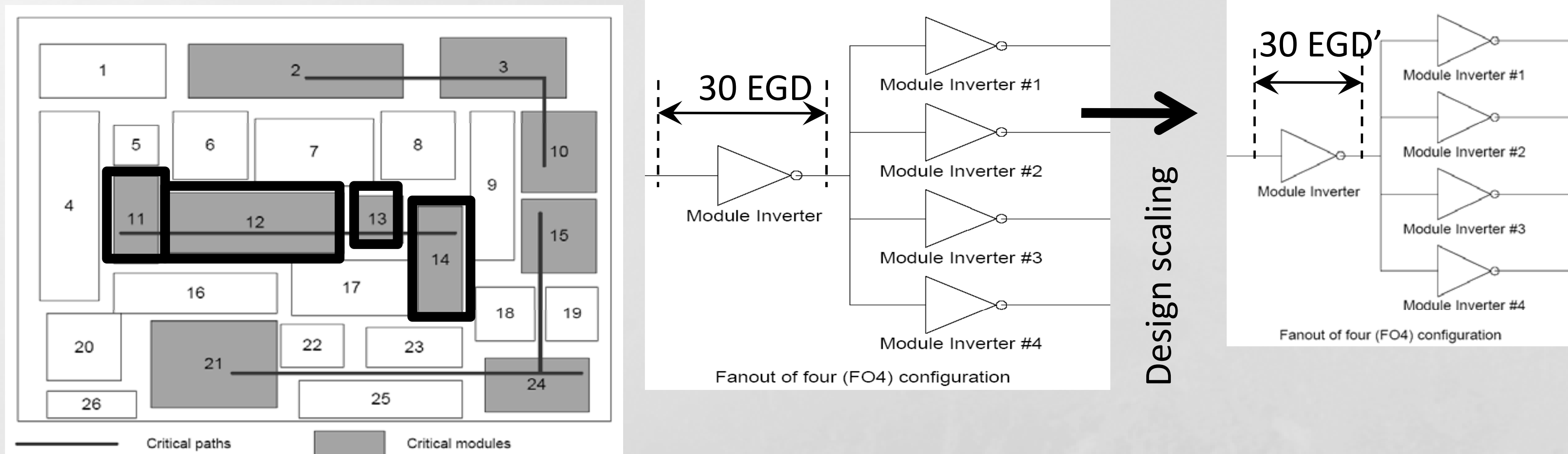
**Process constants (t_{ox}, L_{min}, unit gate cap, ULC etc)

**Physical sizes and other legacy data (P_{width}, N_{width}, C_{load}, TWL, sf, TPFR etc)

**In-situ SPICE simulations (STC, ABBC, DVTC, STPC, ABBPC)

A collection of descriptor vectors, forms the system model for early DSE.

Performance Modeling and Prediction



Module performance is given in EGDs ~ technology independent metric

Critical path delay $_{legacy}$ = $EGD_{11} + EGD_{12} + EGD_{13} + EGD_{14}$

** Example EGD_{11} = x NAND + y NOR + z INV + p BUFF delays expressed in EGD = 30 EGD

** Say, critical path delay = 100 EGD

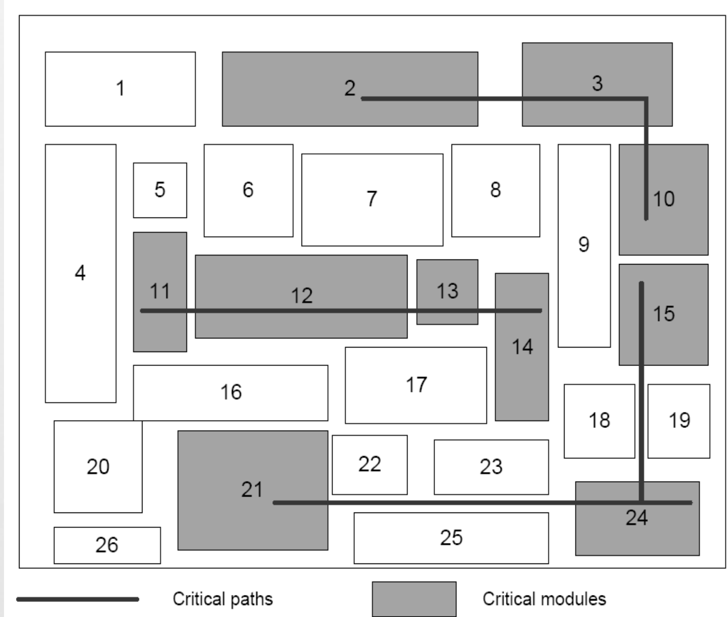
Critical path delay $_{scaled}$ = $EGD_{11} + EGD_{12} + EGD_{13} + EGD_{14}$

**Critical path delay of the scaled design = 100 * EGD’

System critical path delay is the max path delay among all identified critical paths

$$f_{new} = 1 / (\text{system critical path delay})$$

Power Modeling and Prediction



For each module *(1..26)* :

W_N, W_P -- estimted from legacy data

$C_{dyn} = C_{gate} + C_{interconnect}$

$P_{dynamic} = V^2 f_{new} C_{dyn}$

$P_{leakage} = P_{gate} + P_{junc} + P_{sub\text{-}threshold}$

$P_{module} = P_{leakage} + P_{dynamic}$

End Loop

Total power = estimated dynamic + estimated leakage power.

Total power estimated is the sum of the individual power estimates

$$P_{system_dymanic} = \Sigma P_{dynamic}$$

$$P_{system_leakage} = \Sigma P_{leakage}$$

$$P_{system_total} = \Sigma P_{module}$$

Analytical Design Target Prediction Models

(More later)

- Module power equations

$$P_{dyn} = C_{new} \times V^2_{dd_new} \times f_{new} \times RPSF$$

$$C_{new} = \{C'_{orig} \times ((s_{gate_cap} \times frac_{fet}) + ((1 - frac_{fet}) \times s_{wire_cap}))\} + C_{wire_buff}$$

$$f_{new} = (f_{predict}) \times ASF \times CGF$$

$$P_{leak} = \{[(V_{dd_bump} \times I_{off}) + (V_{dd_bump} \times (I_{gate} + BUFFLC)) + (V_{dd_bump} \times I_{junc})] \times ((1 - ASF) \times \varphi)\} + [V_{dd_bump} \times DECAPLC]$$

- Where,
  1. RPSF – library redesign power saving factor
  2. ASF – average switching factor
  3. CGF – clock gating factor
  4. BUFFLC – buffer leakage correction factor
  5. DECAPLC – decap leakage correction factor
  6. ϕ – sleep transistor and back bias correction

- Module performance equations

$$f_{predict} = \frac{((f_{old} \times HVR) + (f_{old} \times LVR \times \boxed{DVTC})) \times DIF}{(TPFR \times FSF) + ((1 - TPFR) \times RCSF_{\ddagger})} \times X$$

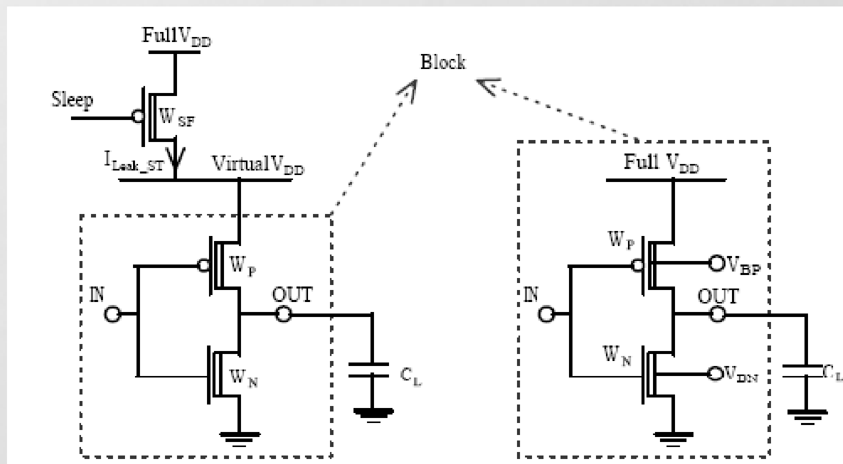
$$RCSF_{\ddagger} = \frac{RCSF}{BSUF}$$

$$X = \boxed{STPC \times ABBPC} \times USPE$$

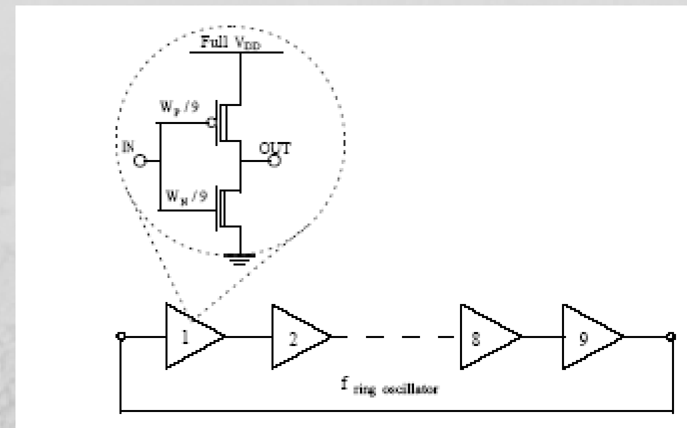
- Where,
  1. HVR – high-V_t FET ratio
  2. LVR – low-V_t FET ratio
  3. f_{old} – 1/ original module critical path delay
  4. DVTC – dual V_t performance correction
  5. DIF – device (I_{ds}) improvement factor
  6. TPFR – typical path FET ratio (delay)
  7. FSF – FET slowdown factor (due to environment)
  8. RCSF – RC (interconnect) slowdown factor
  9. BSUF – buffer speed up factor (wire delay)
  10. STPC – sleep transistor perf. correction
  11. ABBPC – adaptive body bias perf. correction
  12. USPE – useful skew performance correction

In-situ Macromodel Generation

- The macromodels are generated when a design choice is applied to a module.
  - Sleep transistors insertion
  - Back body biasing
  - Supply voltage-V_{dd} changes
  - Low-V_t transistors for critical path speedup
  - Process technology changes through SPICE model



Experiment to obtain ABBC, ABBPC, STC and STPC factors



Experiment to obtain DVTC factor

  - The effect of V_{dd} scaling on leakage currents and FET performance is estimated using a-priori analytical approximations on the target process technology.

Iterative Design Space Exploration

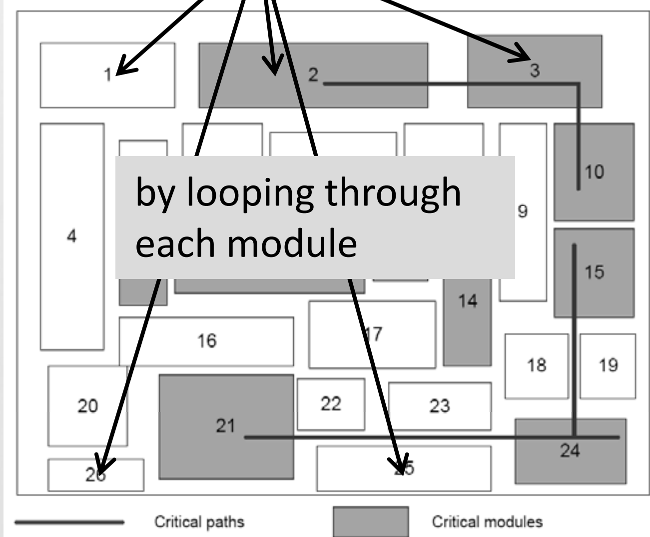
List of P circuit level design choice applicable to modules

$Choice_1$

$Choice_2$

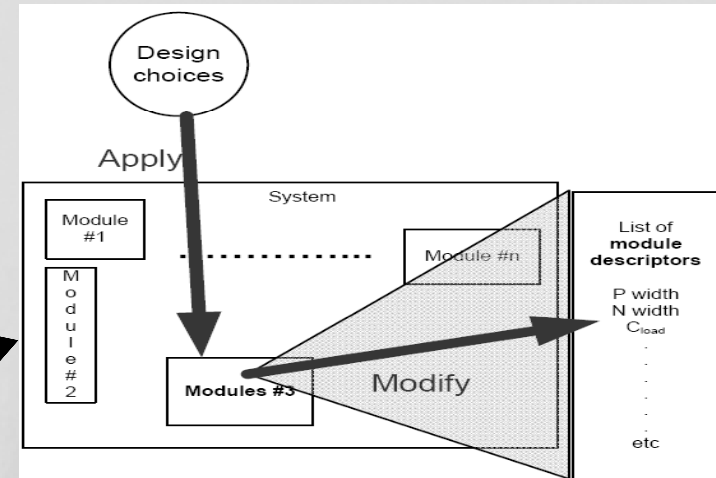
..

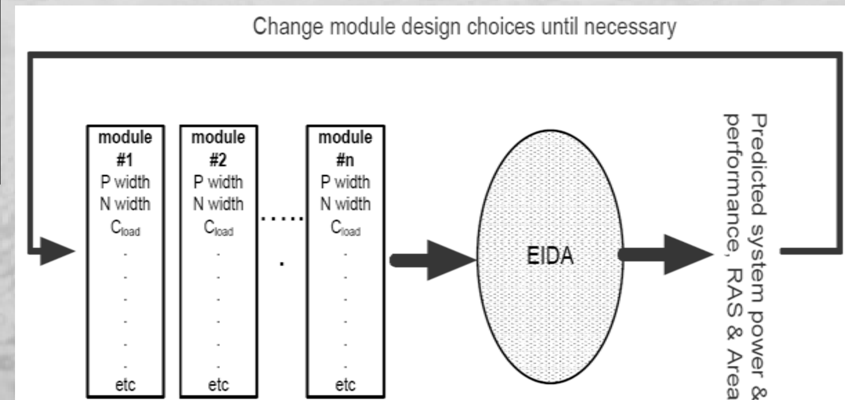
$Choice_P$



by looping through each module

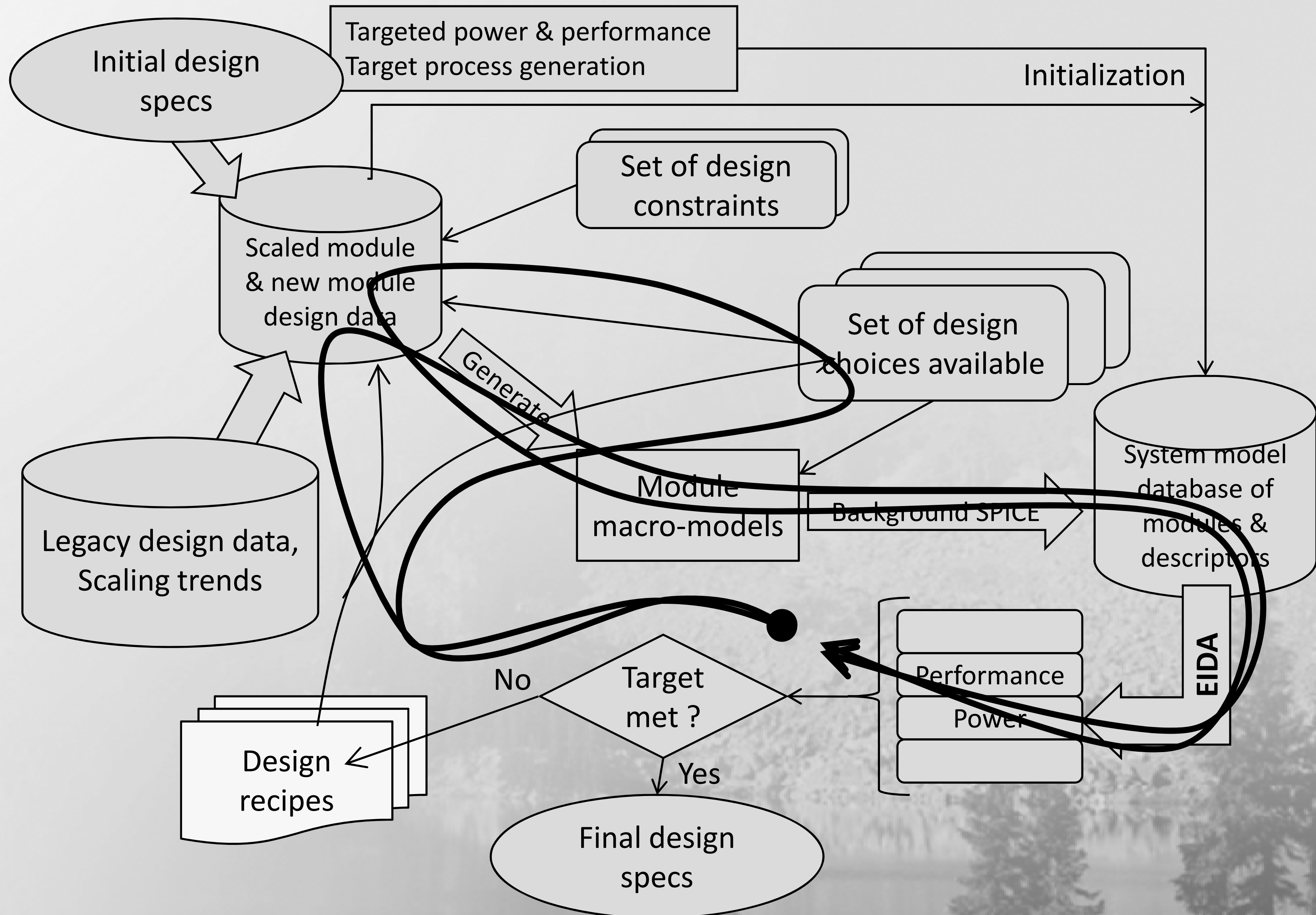
Module	Choice
#1	12
#2	7
#3	8
..	..
#24	5
#25	10
#26	3





Each module has been assigned one (of P) design choice

Iterative Design Space Exploration



Presentation Outline

- Introduction
- Motivation & Objective
- Proposed Approach
- Experimental Setup
- Results & Discussions
- Conclusion & Future Work

Experimental Plan

- Experiments in Technology Node Migration
  - Verifying proposed methodology
- Design Target Prediction Accuracy
  - Validating the prediction models
- Evolutionary Algorithm Based Design Space Exploration
  - Prove scalability
  - DSE demonstration

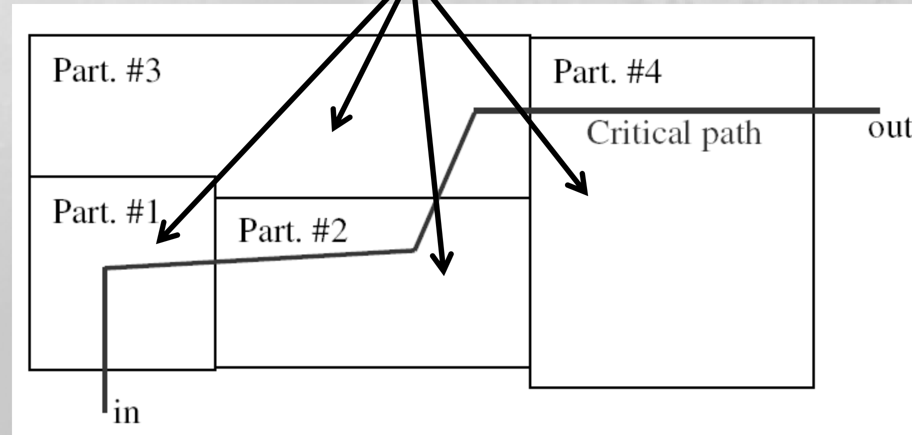
Experiments in Technology Node Migration

- Methodology verification
  - Technology migration from180 nm TSMC process to 130 nm PTM process
    - ISCAS85 C5315 ~ 178 inputs, 123 outputs, 2406 logic gates
    - ISCAS85 C6288 ~ 32 inputs, 32 outputs, 2406 logic gates
    - ISCAS85 C7552 ~ 207 inputs, 108 outputs, 3512 logic gates
    - ISCAS89 S9234 ~ 36 inputs, 39 outputs, 211 DFF, 5597 logic gates
    - ISCAS89 S13207 ~ 62 inputs, 152 outputs, 638 DFF, 7951 logic gates
    - ISCAS89 S15850 ~ 77 inputs, 150 outputs, 534 DFF, 9772 logic gates
    - ISCAS89 S38584 ~ 38 inputs, 304 outputs, 1426 DFF, 19253 logic gates
    - ISCAS89 S38417 ~ 28 inputs, 106 outputs, 1636 DFF, 22179 logic gates
  - 4 partitions or modules per circuit
  - Identify critical path by Pathmill
  - Apply known power & performance design recipies
  - Observe and verify expected trends

Experiments in Technology Node Migration

- Apply the design choices to the partitions and evaluate the relative merit of the various design choice assignments or recipes on the design.
- Design choices
  - Lowering V_{dd} by 200 mV to reduce power consumption
  - Elevating V_{dd} by 200 mV to improve performance
  - Using Low V_t FETs in critical path to improve performance
  - Sleep transistor insertion to reduce idle or leakage power
  - Forward body biasing of critical path transistors to improve performance
  - Reverse body biasing of transistors to reduce leakage power

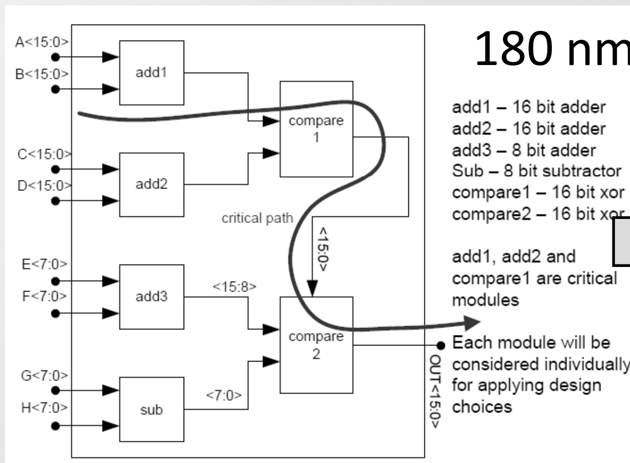
Chosen independently

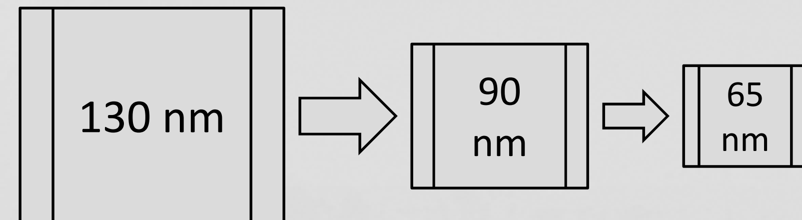


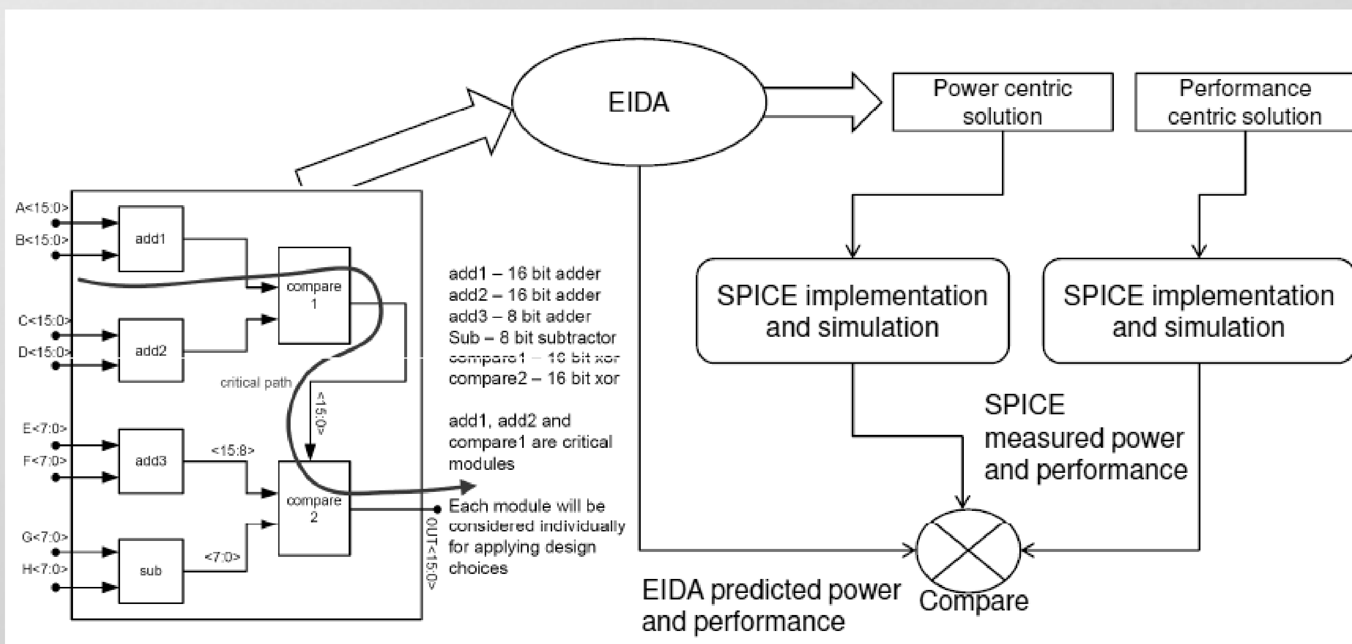
Design Target Prediction Accuracy

- Performed on a test circuit
  - Manageable size for simulations
- Successive scaling (straight porting)
  - Legacy 180 nm implementation
  - 180 nm to 130 nm to 90 nm to 65 nm
  - Compare SPICE measurements and EIDA predictions
  - No additional design choices applied to ported design
- DSE of test circuit in 32 nm
  - Apply design choices and combinations
  - Find power and performance centric solutions
  - Implement in SPICE and compare with EIDA predictions

Design Target Prediction Accuracy





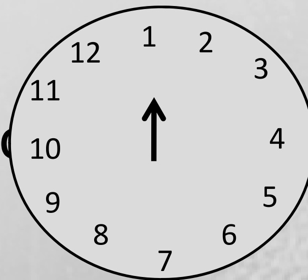


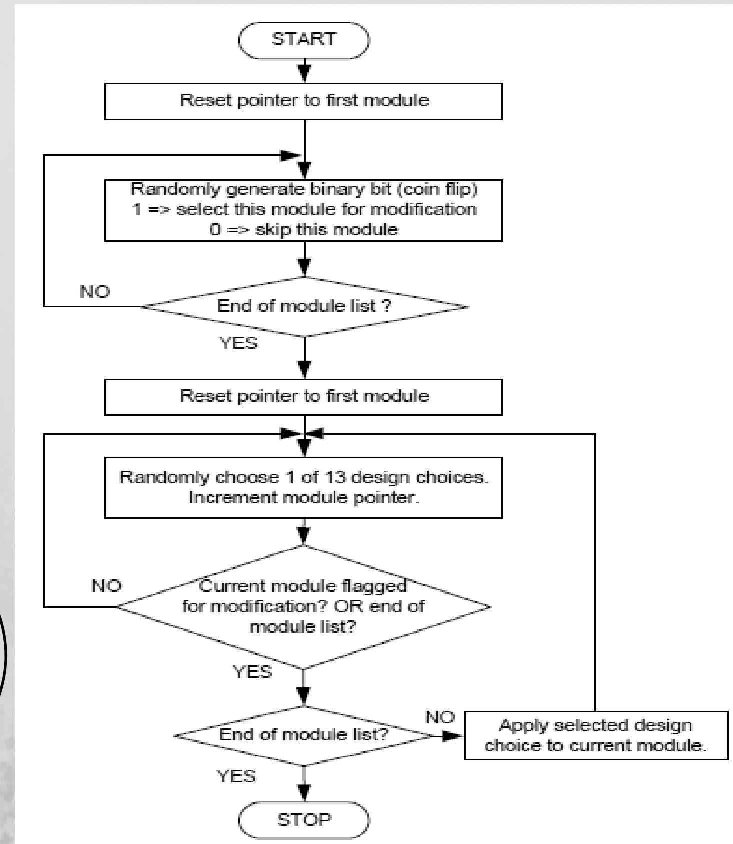
Evolutionary Algorithm Based Design Space Exploration

- μP based design in 32 nm
  - 65 nm to 32 nm porting
  - Pareto analysis (DSE)
    - Randomizer (guided random walk) DSE
    - EA based DSE
- 2 ISCAS circuits (s38584 and s38417)
  - 180 nm to 130 nm
  - EA based DSE
  - Scalability & replacement scheme validation

Randomizer (guided random walk) DSE

- Baseline technique, proves suitability of pareto analysis
- 12 seed design recipes
- Randomized algorithm applied on seed recipes to generated 240 additional recipes
- Generated pareto-fr
- Found power & performance centric recipes





0	1	0		1	0
Module01	**Module02**	**Module03**	**……**	**Module25**	**Module26**
ST	~~ST~~ ↑V_{dd}	Dual-V_t	……	~~ABBRB~~ ST	ST

EA based DSE

- Fixed population size – 100
- Uniform crossover
- Mutation probability 1%
- Two random chromosomes selected for crossover
- Dominant child replacement
- One replacement per iteration
- Binary encoded chromosomes
- Initial population is a non-dominant set

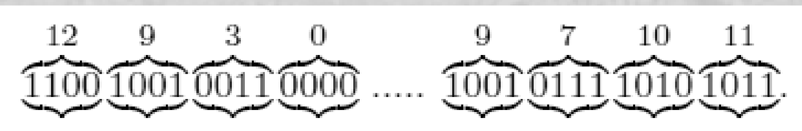
(a) Chromosome – length of 26 (genes)

#1	#2	#3	#4	…….	#23	#24	#25	#26

Design choices (gene values)
0 – No change to the original design
1- Reduce Vdd by 100 mV
.
.
12 – Insert sleep Transistors

(b) A valid chromosome with 26 gene values

12	9	3	0	…….	9	7	10	11



EA based DSE

- C - criteria (objective) vector
- S_i - a generic chromosome
- F_i – chromosome's fitness vector

$$C \in \mathbb{R}^n \, and \, C = [C_1, C_2, ..., C_n]^T$$

$$S_i \in \mathbb{R}^m \, and \, S_i = [S_{i1}, S_{i2}, ..., S_{im}]^T$$

$$F_i \in \mathbb{R}^n \, and \, F_i = [F_{i1}, F_{i2}, ..., F_{in}]^T$$

- Split vector C into three such as;

$$C = [C_1, C_2, ..., C_\vartheta, C_{\vartheta+1}, C_{\vartheta+2}, ..., C_\lambda, C_{\lambda+1}, C_{\lambda+2}, ..., C_n]^T$$

- Criteria 1 … varpi are min, varpi+1 … lambda are max and lambda+1 … n are don't care
- Let S_1, S_2 & F_1, F_2 be two chromosomes and their fitness vectors, respectively.
- Chromosome S_1 dominates S_2, if;
- 50000 iterations to get PF

$$F_{1j} > F_{2j} \quad \forall \quad j : 1 \ldots \vartheta$$

$$F_{1k} > F_{2k} \quad \forall \quad k : (\vartheta + 1) \ldots \lambda$$

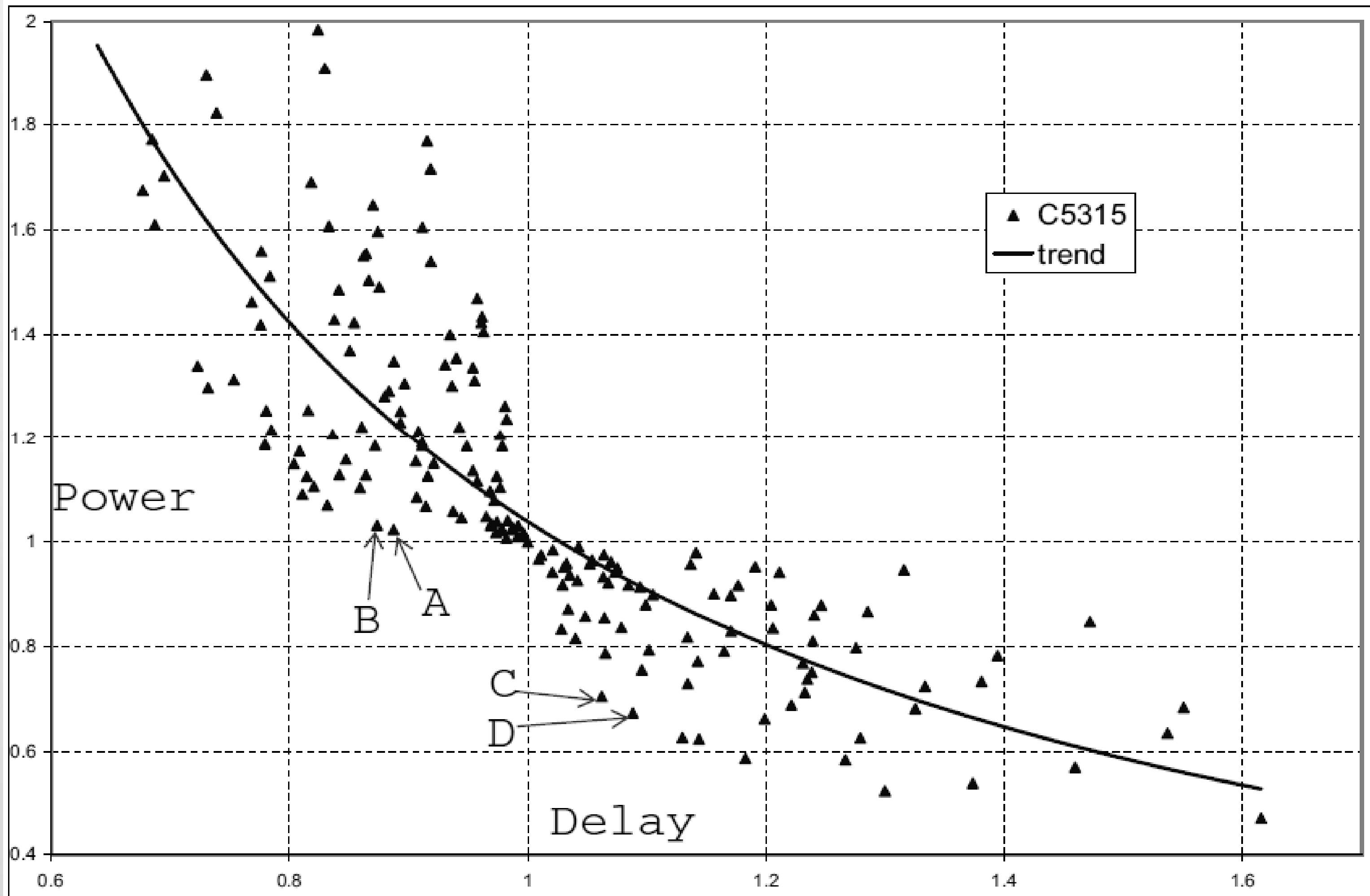
Experimental Plan - recap

- Experiments in Technology Node Migration
  - Verifying proposed methodology
- Design Target Prediction Accuracy
  - Validating the prediction models
- Evolutionary Algorithm Based Design Space Exploration
  - Prove scalability
  - DSE demonstration

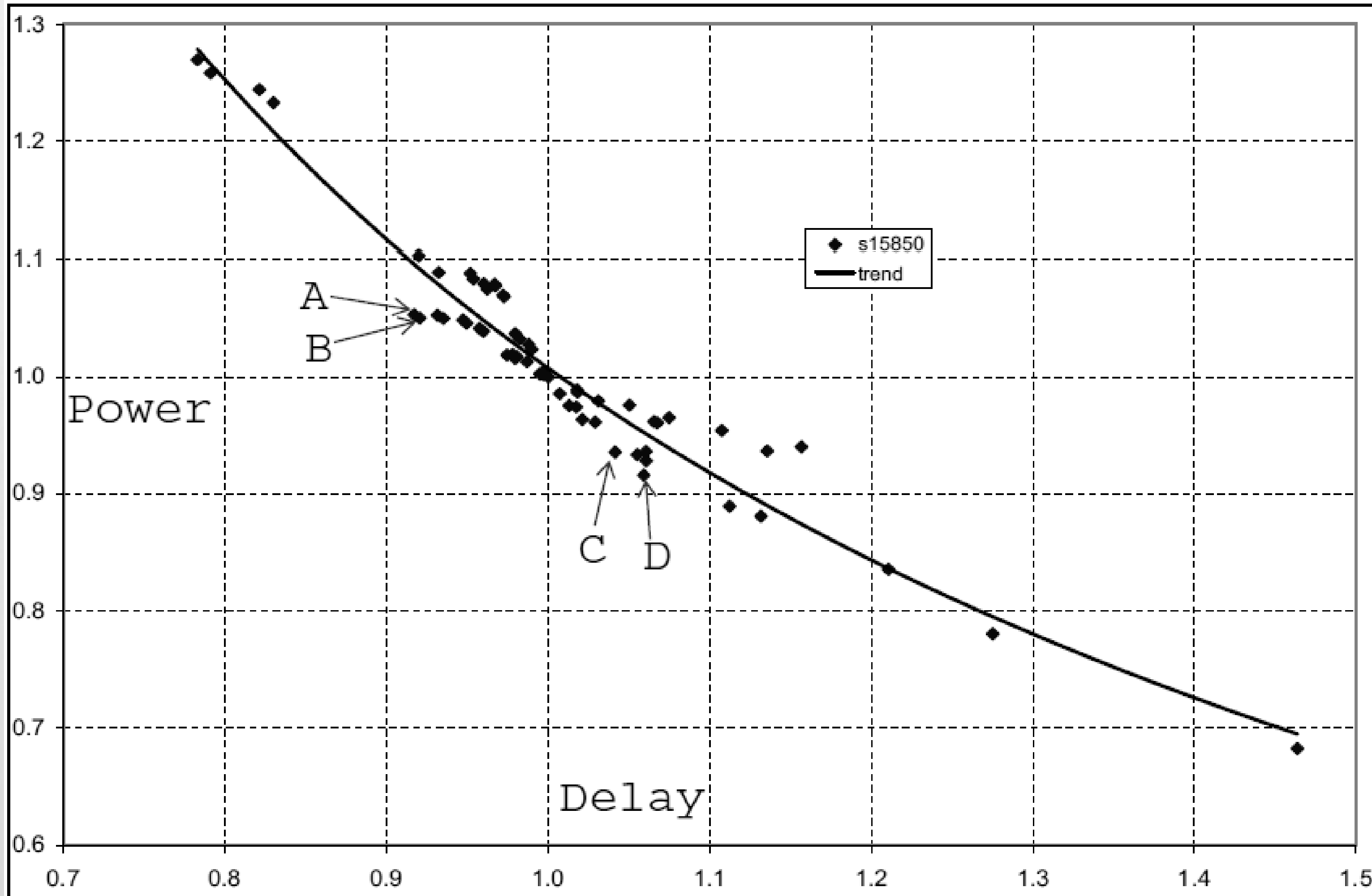
Presentation Outline

- Introduction
- Motivation & Objective
- Proposed Approach
- Experimental Setup
- Results & Discussions
- Conclusion & Future Work

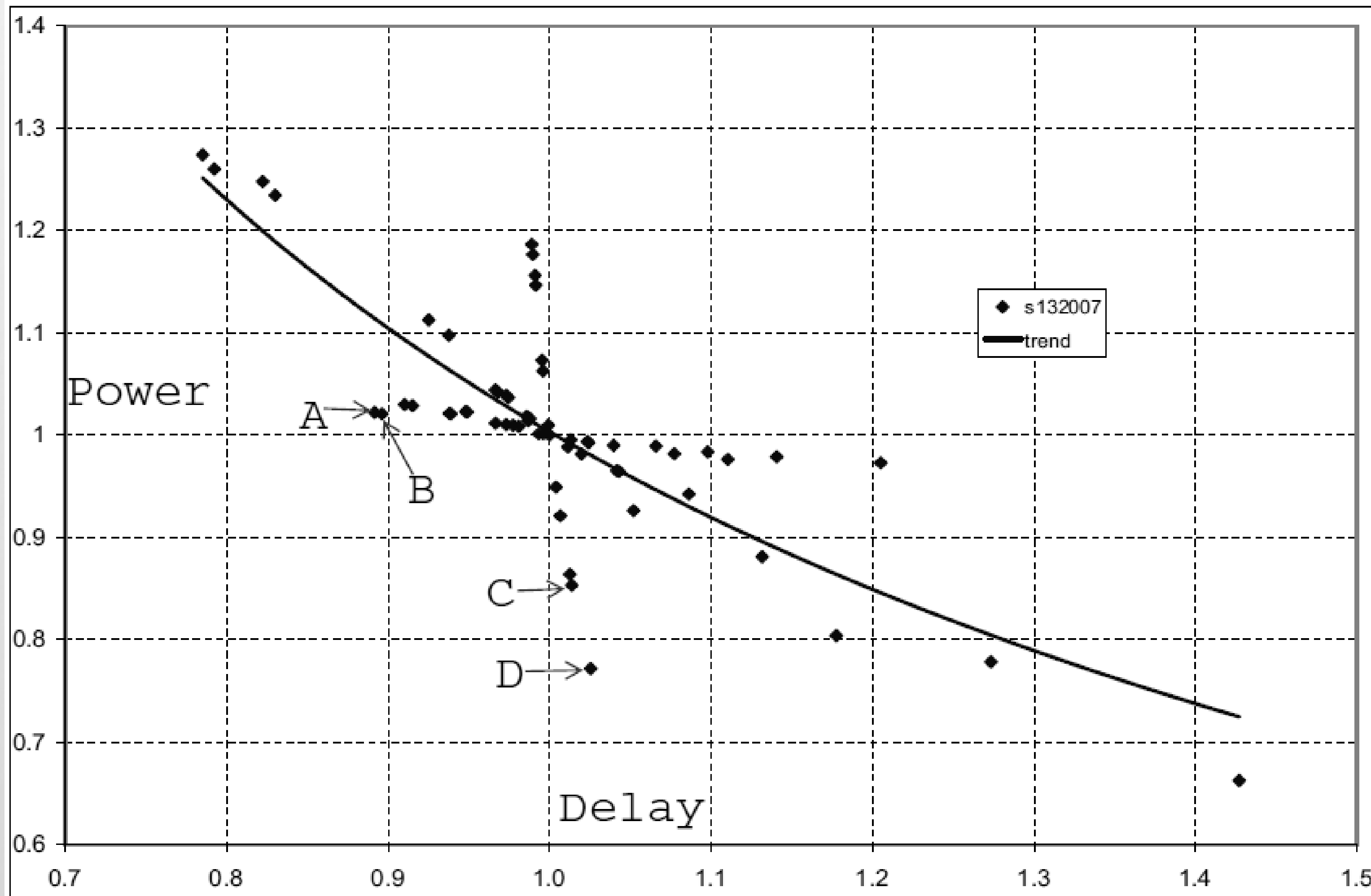
Results of Experiments in Technology Node Migration



Results of Experiments in Technology Node Migration

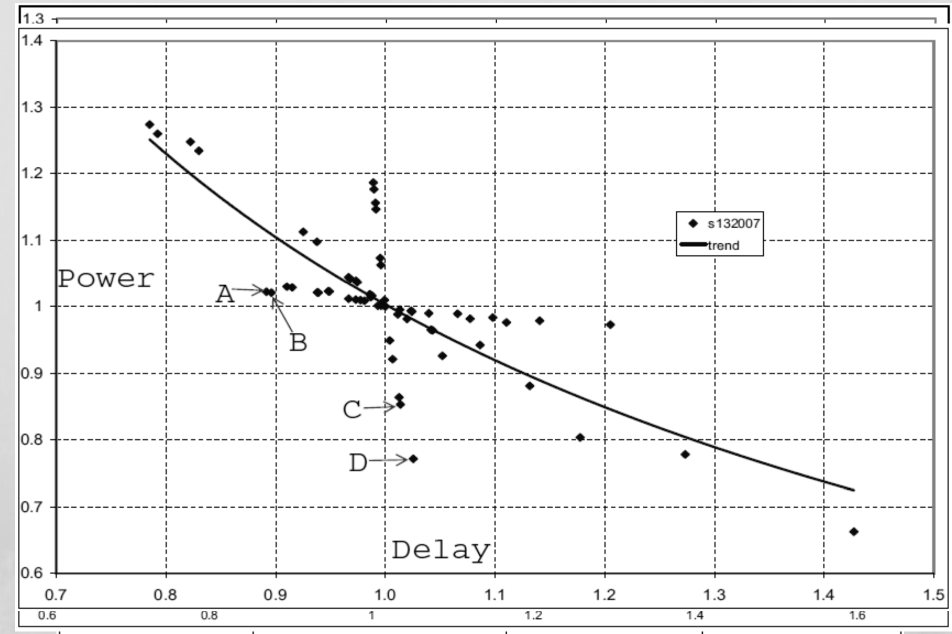


Results of Experiments in Technology Node Migration



Results of Experiments in Technology Node Migration

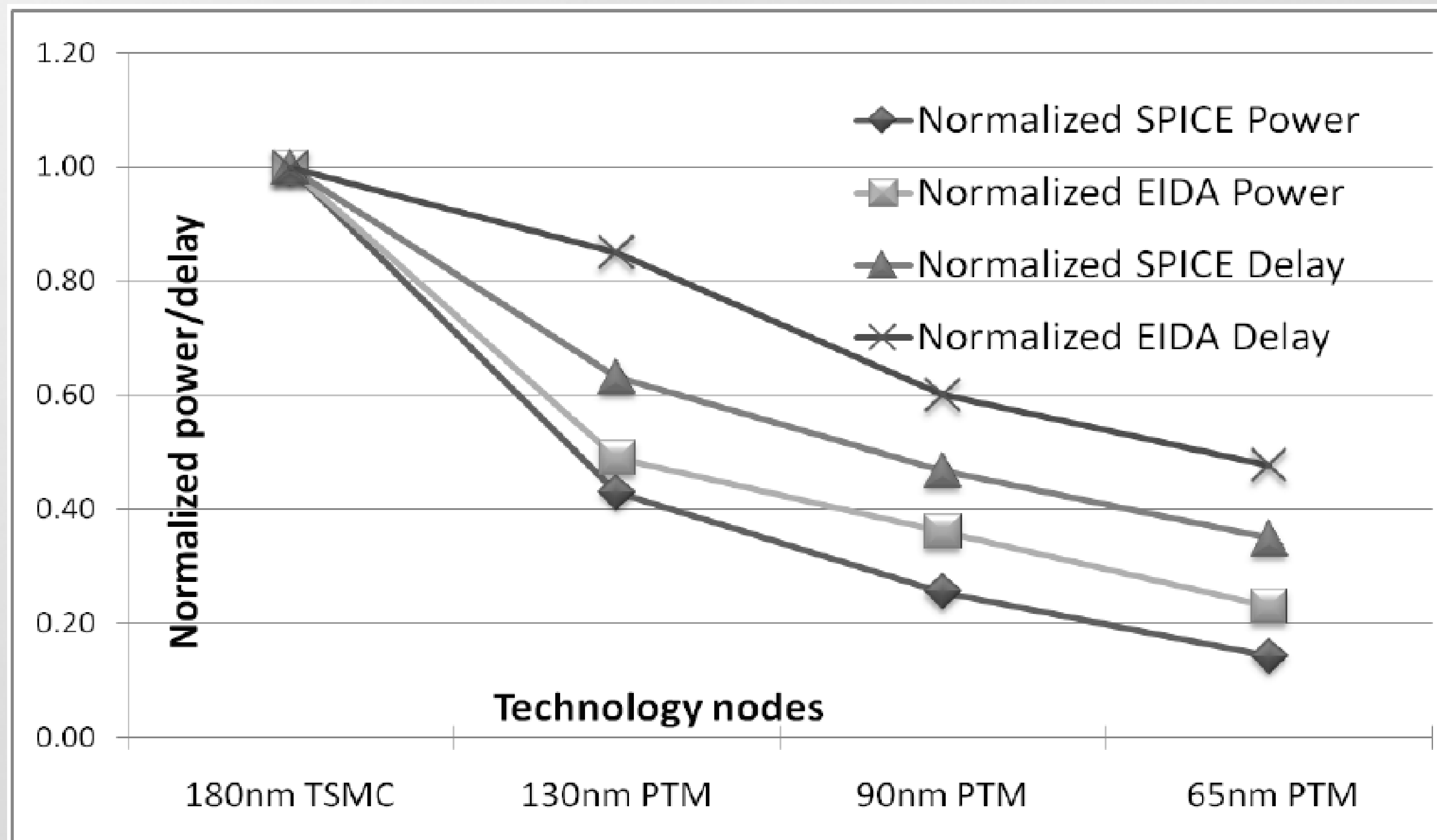
- Expected power – performance trends were verified
- All circuits except two (S15850,S9234) lead to solutions optimizing both power and performance.
- S15850,S9234 all modules contributing equally to both system power and system critical path delay. Poor plot spread lead to fewer assignments dominating other assignments.
- In circuits S132007 and S38417, one module (#4 in S132007 and #3 in S38417) contributed less than 3% to the system critical path delay. Seen in plots.
- Our results predicted 2.5-6X reduction in leakage power with ST. An industrial study showed 7X reduction [82].
- Our results predicted 3-6% reduction in leakage power with ABB. Similar to an industrial study [83].



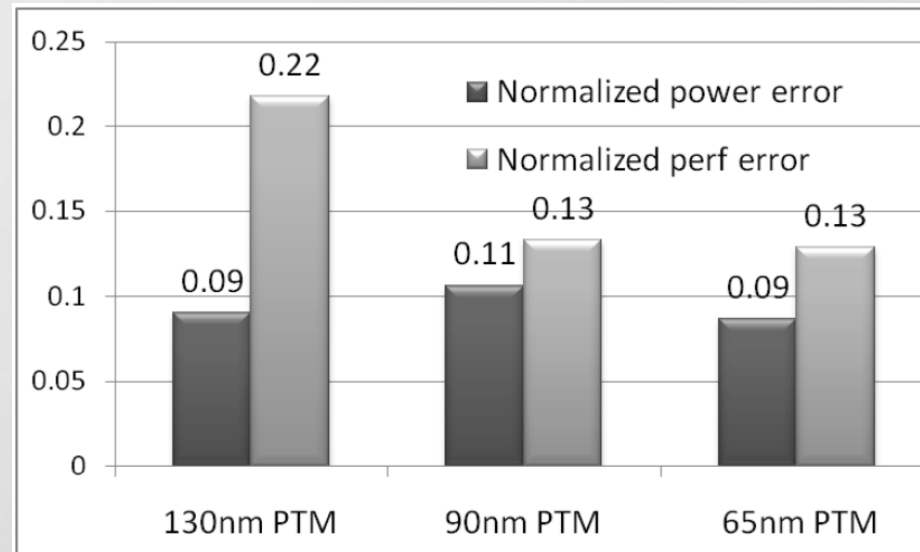
Power centric designs:
↓power by 7% - 32% for a 0% - 9% ↓ in performance
Performance centric designs:
↑perf. by 11.25% - 17% for a 2% - 3.85% ↑ in power

This experiment verifies the proposed methodology.

Results of Design Target Prediction Accuracy – Successive Scaling



Results of Design Target Prediction Accuracy – Successive Scaling, DSE at 32 nm

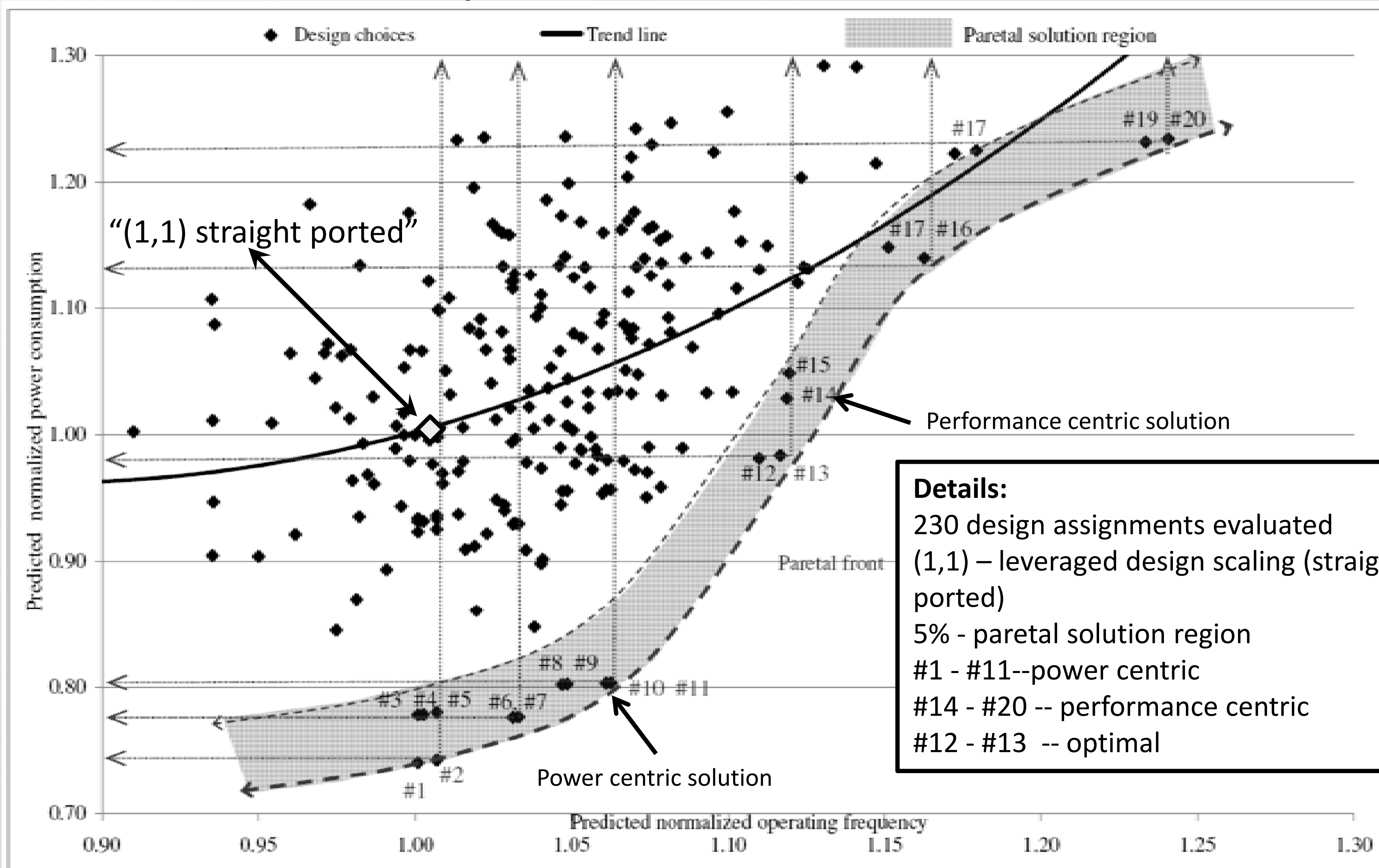


Measurement	EIDA	SPICE	% error
straight-ported design	1.7924 mW	1.6084 mW	-11.4%
straight-ported design	392 MHz	451 MHz	13.1%
power-centric design	1.0456 mW	0.937 mW	-11.5%
power-centric design	406 MHz	368 MHz	-10.3%
performance-centric design	1.5523 mW	1.375 mW	-12.9%
performance-centric design	442 MHz	513 MHz	13.8%

Results of Design Target Prediction Accuracy

- Successive technology scaling trends are consistent with two industrial results [84] and [85].
- Power prediction errors ranged from 9% to 11% with respect to SPICE. Power prediction errors are well controlled given the high level nature of the system model used in EIDA.
- Performance prediction errors ranged from 13% to 22% with respect to SPICE. This is mainly due to the unavailability of detailed layout interconnect parasitic values included in the SPICE netlist.
- Existing tools using Thevenin based flow for timing analysis yield errors between 10-15% [89].
  - These tools typically operate on detailed transistor level design details
- Therefore, power and performance prediction errors in the range of 9 to 22% compared to SPICE, without any bottom up data is acceptable.
  - Makes the usage of EIDA for high level design tradeoffs practical

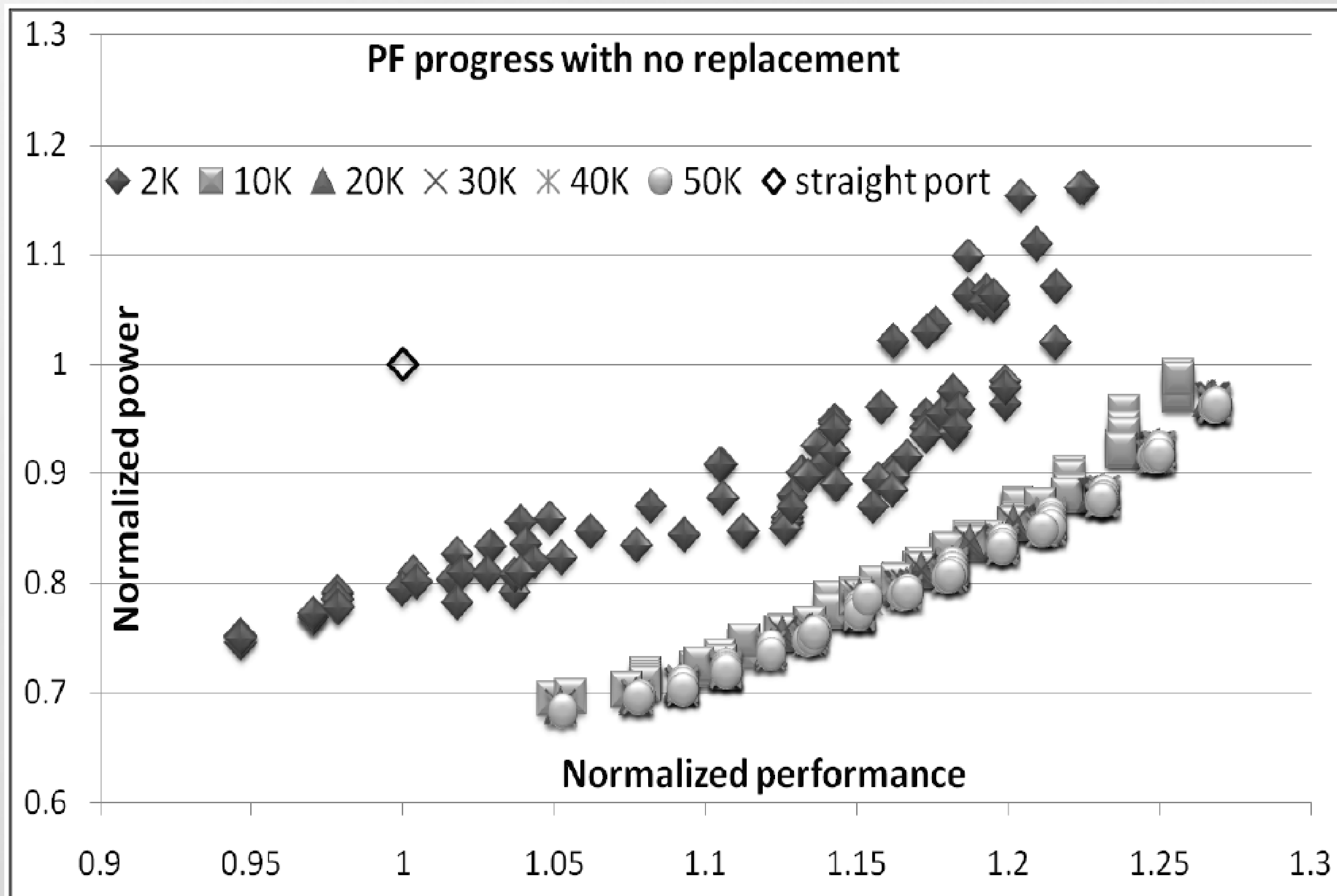
Results of Randomizer (guided random walk) DSE – Pareto Front



Results of Randomizer (guided random walk) DSE

- Performance centric solution (#13) achieves 11.7% increase in performance while saving 2% power
- Power centric solution (#11) achieves 19.6% power savings while increasing performance by 6.2%
- Pareto analysis is suitable for DSE
- Sparse pareto-front due random nature of recipe generation
  - Need to improve generated pareto-front, increasing the no of recipes may not be a good approach
- Randomizer does not track non-dominant recipes
  - Need evolutionary algorithms

Results of EA based DSE

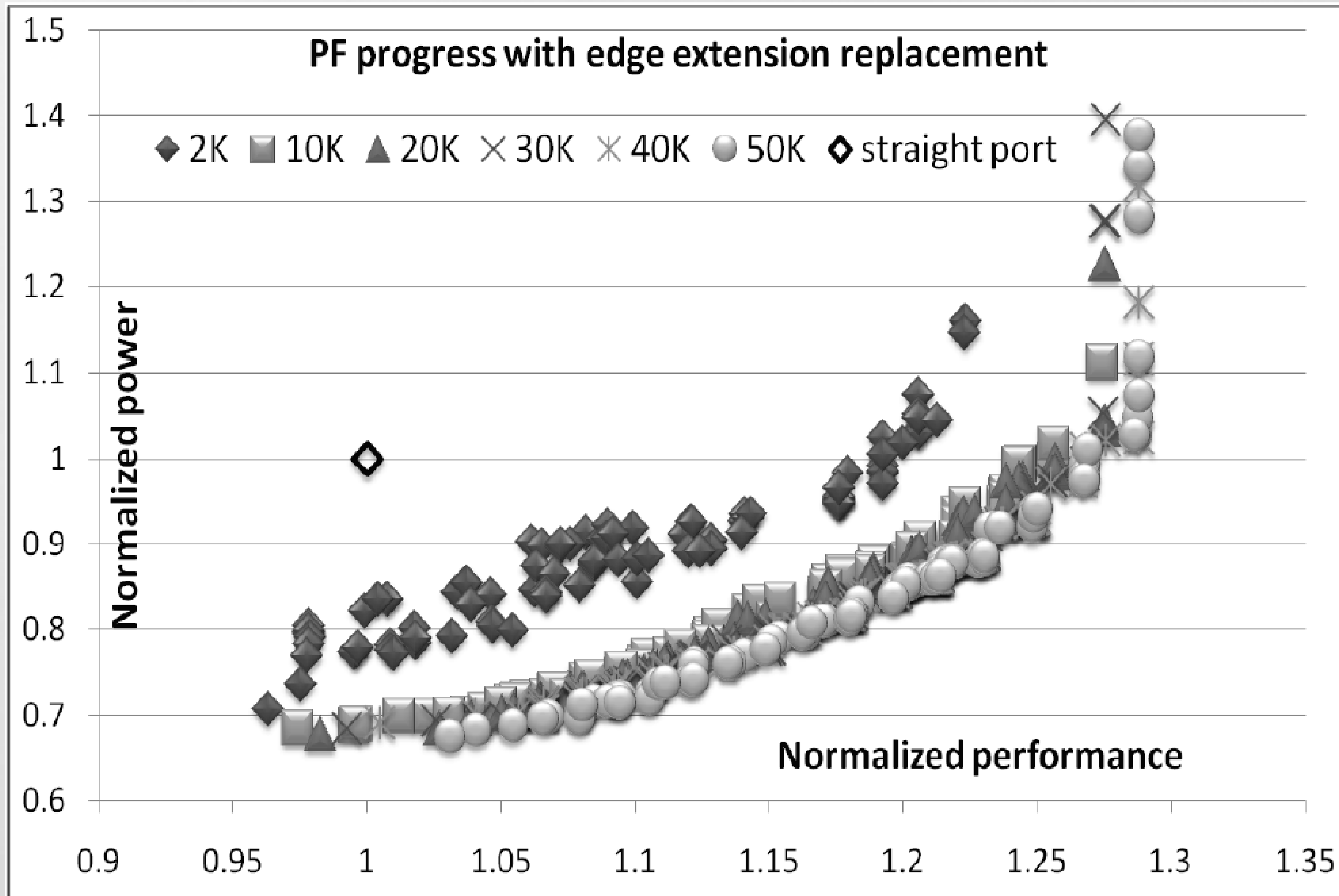


1) This PF is better than Randomizer generated PF
2) But with a Pop size of 100, the PF appears crowded
3) PF shows progressive improvement with # of iterations

Results of EA based DSE – Replacement

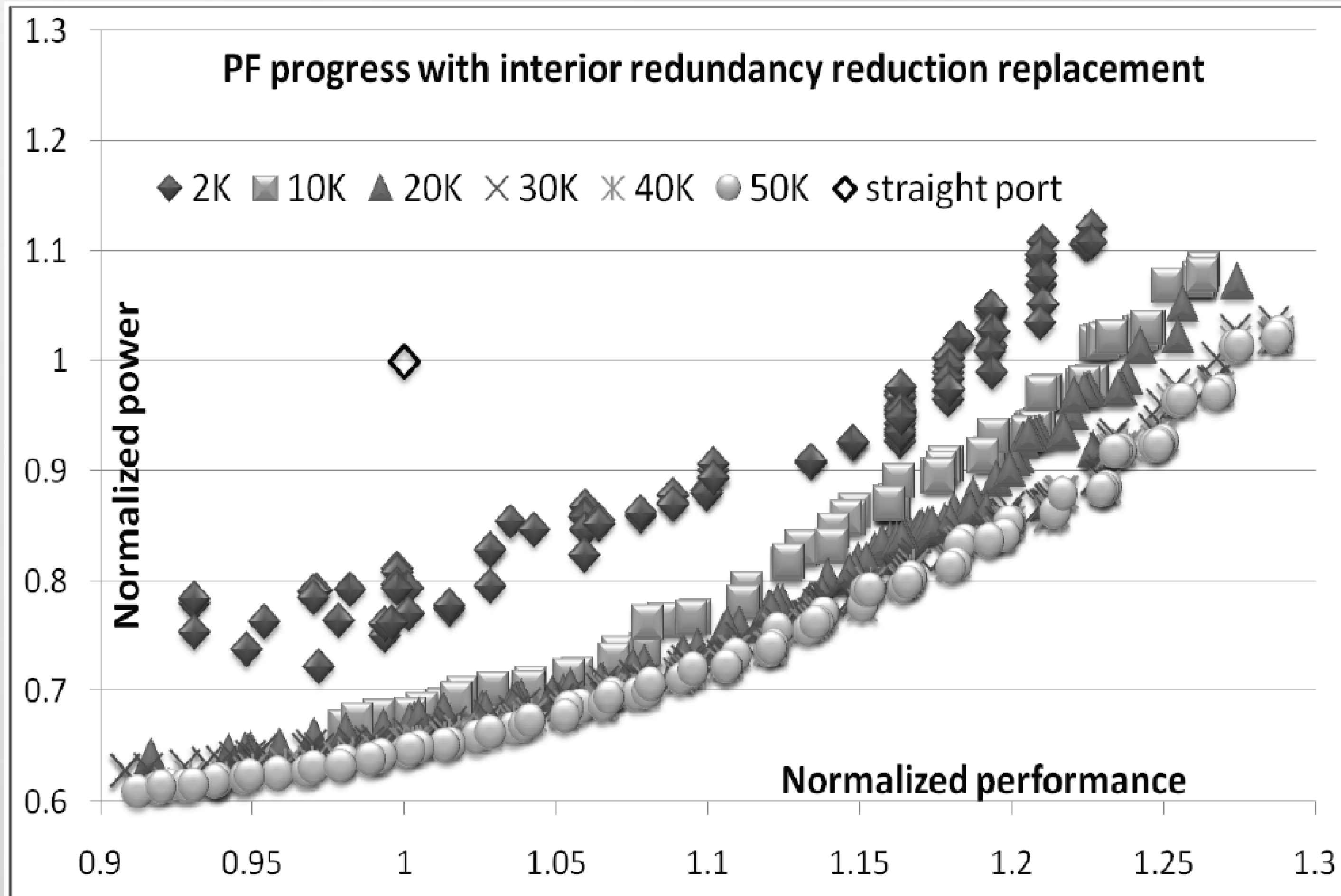
- A newly generated chromosome is defined as an "outside-pareto" chromosome if;
  - No chromosome in current population dominates it
  - It does not dominate any chromosome in the current population
  - Such a chromosome will be discarded in the standard iteration
- Replacement identifies such chromosomes and forces them into the current population, by replacing an existing chromosome. Two ways to replace;
  - Edge extension replacement scheme (EER)
  - Interior redundancy reduction replacement scheme (IRRR)
- EER replaces the closest chromosome in the current population
- IRRR replaces the most redundant chromosome in the current population (typically in the interior of the PF away from the edges)

Results of EA based DSE - EER

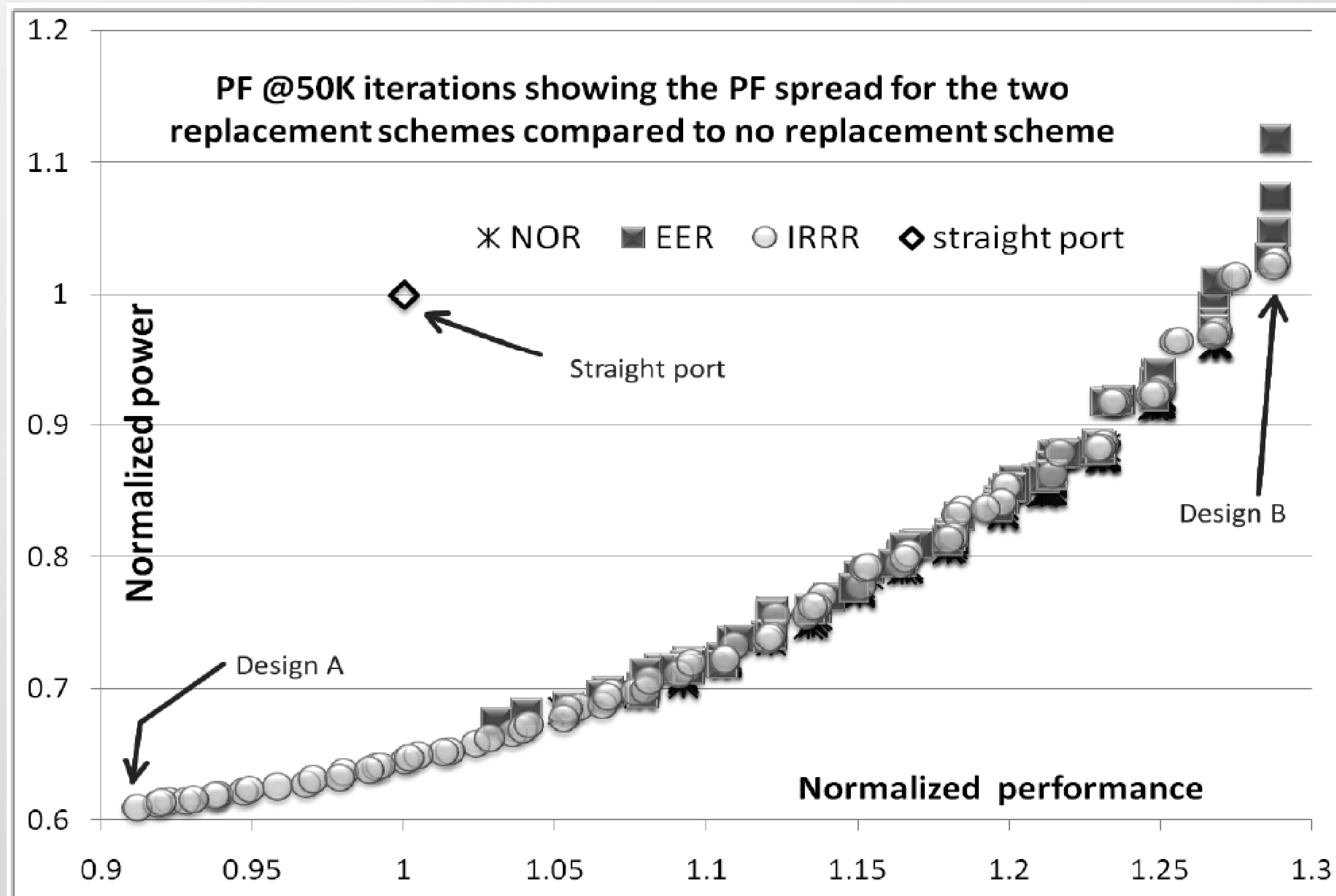


1) This PF is better than the PF with no replacement
2) The PF appears crowded at the edges

Results of EA based DSE - IRRR



Results of EA based DSE – All Schemes



1) IRRR scheme produces the best PF
2) Need a way to quantitatively compare the PFs

Results of EA based DSE – FOMs

- Let p – population size
- F – fitness vector of size lambda
- Criteria 1 … varpi are min, varpi+1 … lambda are max
- PF points are normalized and are positive
- Then;

$$FOM_SQ = Median(\left[\frac{\phi\nu}{\chi\nu}\right]); \forall \quad \nu : 1 \ldots p$$

$$\chi\nu = \prod_{j=1}^{\vartheta} F_{\nu j}$$

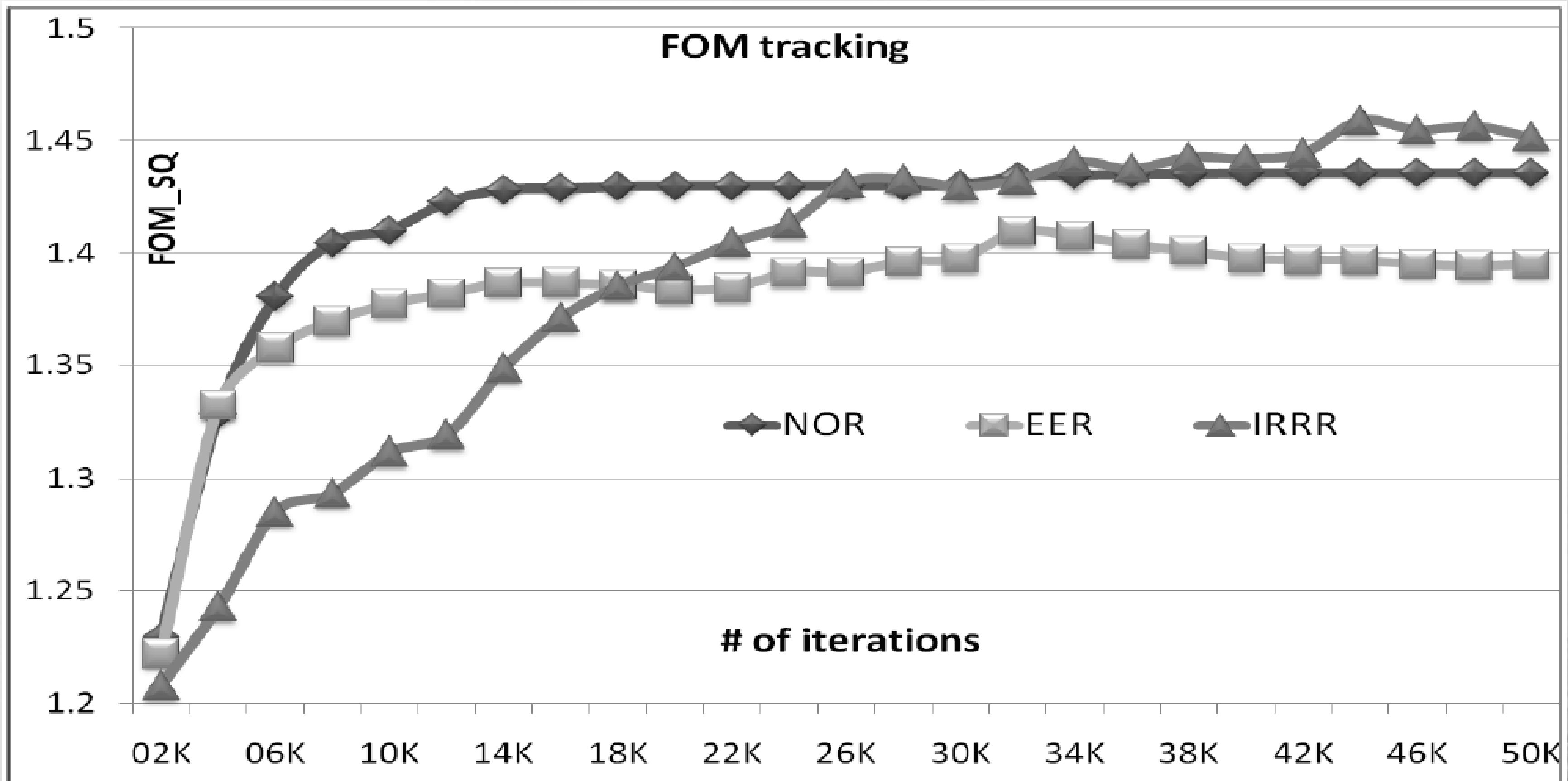
$$\phi\nu = \prod_{k=(\vartheta+1)}^{\lambda} F_{\nu k}$$

$$FOM_SS = \prod_{x=1}^{\lambda} \sigma_x$$

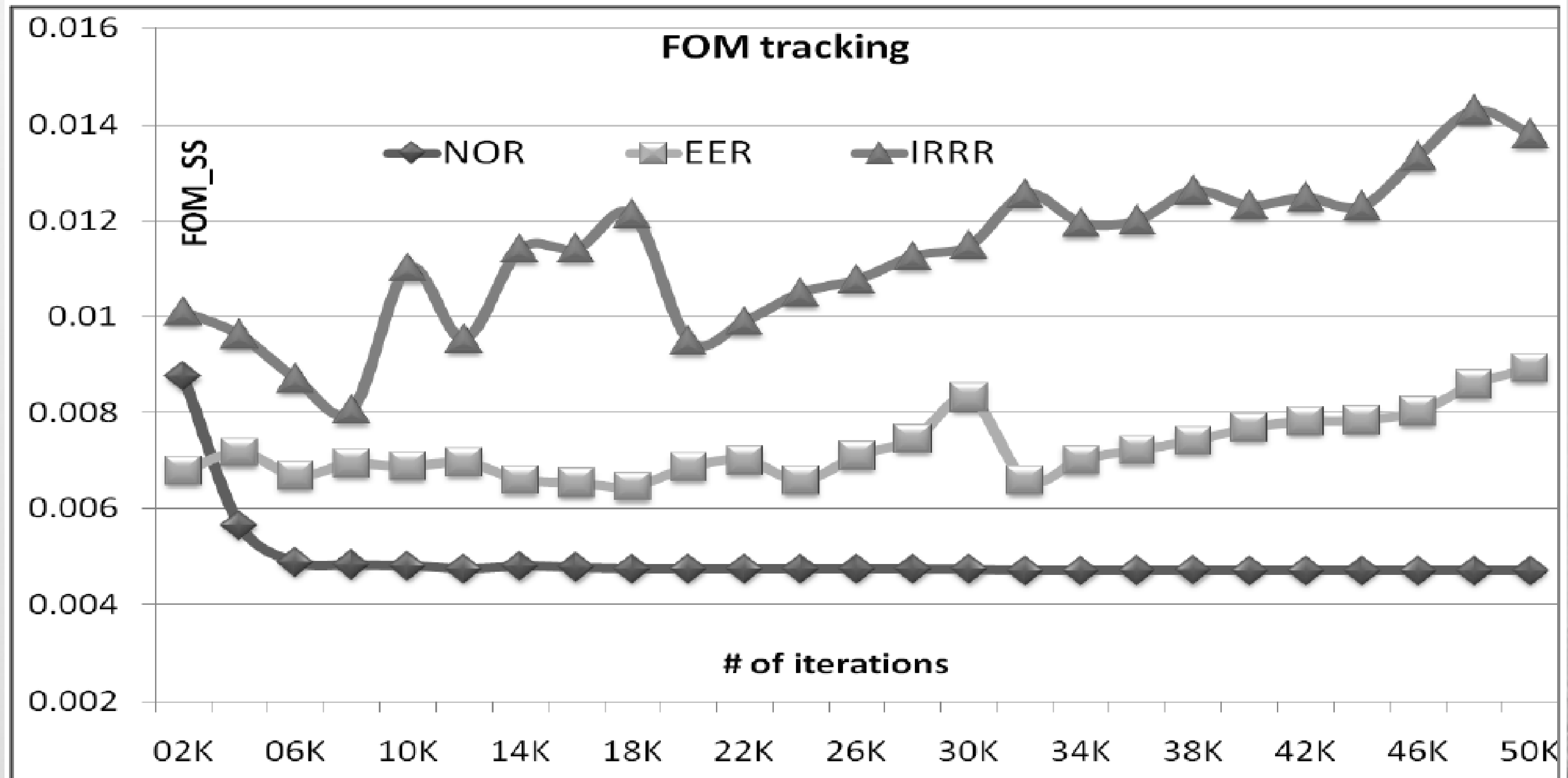
$$\sigma_x = Std\,Dev(F_{1x}, F_{2x}, \ldots, F_{px})$$

- FOM_SQ - measure of the quality of the PF, larger values are better
- FOM_SS - measure of the spread of the PF – larger values are better

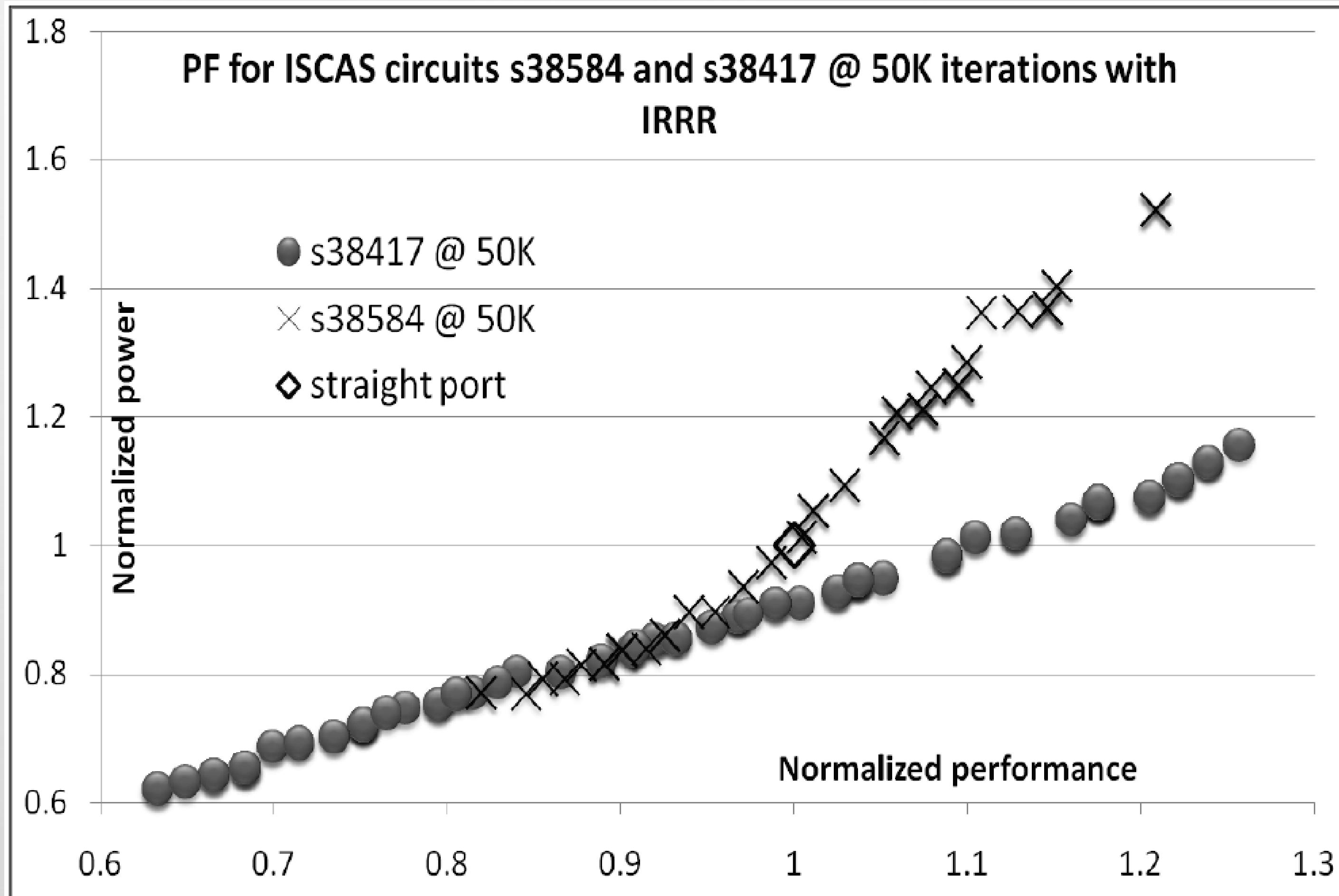
Results of EA based DSE – FOM_SQ



Results of EA based DSE – FOM_SS

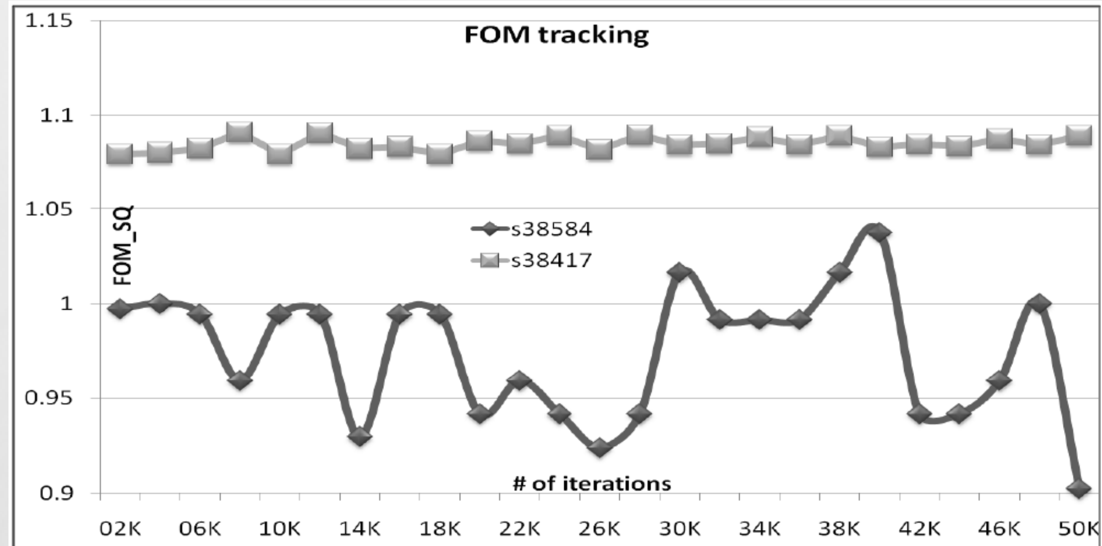


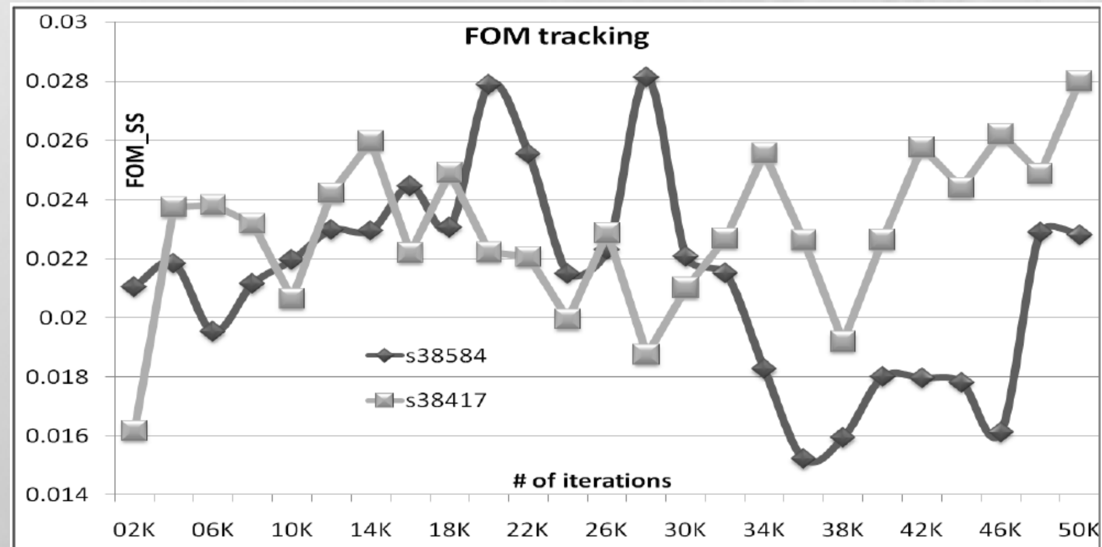
Results of DSE on ISCAS Circuits with IRRR



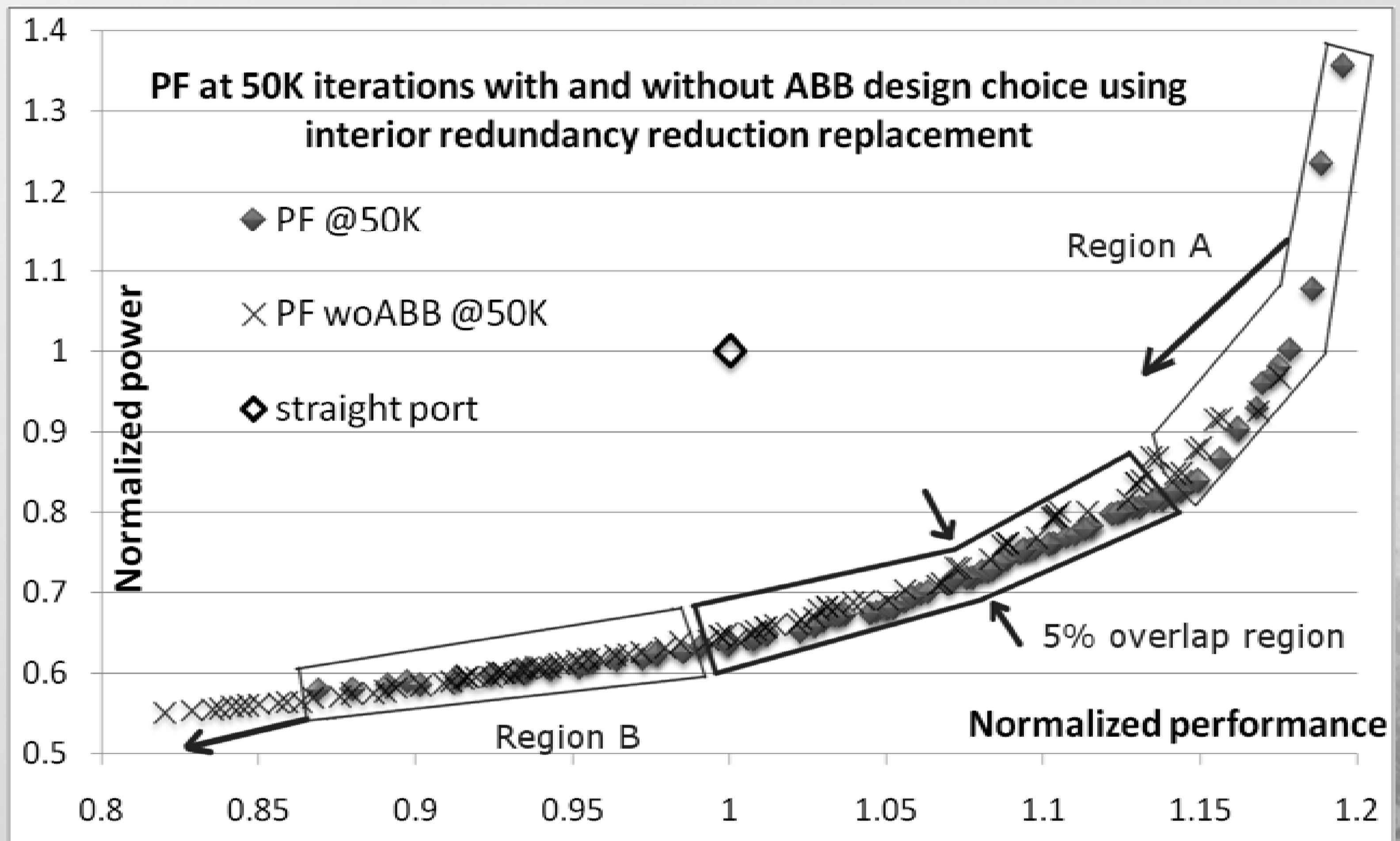
The Pareto-front for s38417 circuit was expected to be better of the two since one module in s38417 contributed less than 3% to the critical path delay.

Results of DSE on ISCAS Circuits with IRRR





Discussion: Impact of ABB Design Choice Solution Complexity



Discussion: Prediction Model Complexity

Descriptor / Technology	Power Sensitivity			Perf. Sensitivity		
	130nm	90nm	65nm	130nm	90nm	65nm
η (Table 3.2)	2.23	2.19	2.23	0.59	0.70	0.66
DECAP_SENS (Eqn 3.7)	1.68	1.79	1.75	0.74	0.79	0.77
β (Eqn 3.31)	-0.96	-0.68	-0.65	-0.25	-0.30	-0.21
γ (Eqn 3.32)	-0.96	-0.68	-0.65	-0.25	-0.30	-0.21
δ (Eqn 3.33)	-0.96	-0.68	-0.65	-0.25	-0.30	-0.21
α (Eqn 3.30)	-0.94	-0.67	-0.64	-0.25	-0.30	-0.21
USPE (Eqn 3.25)	0.91	0.94	0.87	1.00	1.00	1.00
RPSF (Eqn 3.24)	0.91	0.94	0.87	n/a	n/a	n/a
ϵ (Eqn 3.34)	-0.82	-0.47	-0.48	-0.09	-0.07	-0.02
Y (Eqn 3.35)	-0.65	-0.53	-0.60	-0.22	-0.27	-0.25
X (Eqn 3.35)	-0.63	-0.51	-0.60	-0.20	-0.25	-0.25
BSUF (Eqn 3.24)	0.49	0.56	0.48	0.54	0.60	0.55
RCSF (Eqn 3.1)	-0.48	-0.55	-0.48	-0.53	-0.59	-0.55
sf (Eqn 3.14)	-0.03	-0.02	-0.06	n/a	n/a	n/a
n/a - not applicable						

Descriptor	Power Sensitivity			Perf. Sensitivity		
	130nm	90nm	65nm	130nm	90nm	65nm
η (Table 3.2)	2.38	2.46	2.42	0.69	0.91	0.80
RPSF (Eqn 3.24)	0.92	0.94	0.89	n/a	n/a	n/a
USPE (Eqn 3.25)	0.92	0.94	0.89	1.00	1.00	1.00
β (Eqn 3.31)	-0.89	-0.72	-0.63	-0.31	-0.41	-0.28
γ (Eqn 3.32)	-0.89	-0.72	-0.63	-0.31	-0.41	-0.28
δ (Eqn 3.33)	-0.89	-0.72	-0.63	-0.31	-0.41	-0.28
α (Eqn 3.30)	-0.87	-0.71	-0.63	-0.31	-0.41	-0.28
ϵ (Eqn 3.34)	-0.77	-0.54	-0.49	-0.18	-0.22	-0.12
Y (Eqn 3.35)	-0.61	-0.57	-0.59	-0.26	-0.36	-0.31
X (Eqn 3.35)	-0.59	-0.55	-0.59	-0.24	-0.34	-0.31
DECAP_SENS (Eqn 3.7)	0.51	0.61	0.58	0.11	0.17	0.14
BSUF (Eqn 3.24)	0.47	0.52	0.46	0.51	0.55	0.52
RCSF (Eqn 3.1)	-0.46	-0.51	-0.45	-0.50	-0.55	-0.51
sf (Eqn 3.14)	-0.04	-0.03	-0.04	n/a	n/a	n/a
n/a - not applicable						

- Some model parameters are difficult to obtain. Eg. ETA, BUFFLC, BSUF & USPE
- Sensitivity analysis – least sensitive parameters can use default values
- For example, consider DECAP_SENS parameter – how the designer decided its value?
- Find sensitivity reducing de-coupling capacitance by half and increasing DECAP_SENS by 30%
- DECAP_SENS's sensitivity rank drops => designer may use default value
- Simplifies the analytical prediction model => modeling flexibility and complexity reduction

Discussion: Impact of Evolutionary Algorithm

- The EA with IRRR is relatively simple compared to state-of-the-art EA
- The focus of the work at this time is to establish the applicability of the proposed design framework for design space exploration; hence the simple EA with IRRR is used.
- FOM_SS & FOM_SQ are unique compared to the existing metrics to evaluate PFs
  - Only one non-dominated set is needed to obtain the FOMs unlike many existing metrics
  - Does not require the knowledge of the absolute PF to compare PFs
- The IRRR scheme is very effective and yet simple to implement, unlike some techniques such as the NBI technique.
- There is room for improving the EA technique implemented.

Presentation Outline

- Introduction
- Motivation & Objective
- Proposed Approach
- Experimental Setup
- Results & Discussions
- Conclusion & Future Work

Results Summary

- The technology node migration experiment establishes the feasibility of the proposed methodology.
  - For these circuits, DSE uncover designs that offer 7-32% power reduction with 0-9% performance degradation or 10.31-17% performance improvement with 2-3.85% power penalty
- Successive scaling and DSE of the test circuit provided error estimates which are comparable to results in literature.
  - Power and performance errors ranged from 9 to 22%
- EA based DSE of a microprocessor based design with IRRR yielded Pareto optimal solutions that optimized both power and performance.
  - improved the straight-port microprocessor based design by reducing power by 40% with 9% performance impact or by improving performance by 29% with 2.5% power penalty.

Concluding Remarks

- The key contributions of the proposed methodology and tool are;
  - The ability to uncover pareto optimal, complex and non-intuitive system designs
  - The ease of performing high level tradeoff and what-if analysis
  - The potential to expose any underlying design risks very early in the design phase
  - Design complexity reduction by generating solution without complex design choices such as ABB with a bounded power and performance impact.

- Thus by incorporating the proposed methodology as part of a standard system design flow;
  - The likelihood for design convergence is greatly improved
  - The chances of meeting time-to-market schedule is improved

Limitations and Future Work

- Analytical prediction models
  - Power & performance currently modeled
  - Yield, chip area & reliability => ↑EIDA's value as a design tool
- Process variation impact
  - Statistical inputs and outputs => ↑EIDA's value as a design tool
  - Macromodel based critical path delay calculation including clock and signal uncertainties
- System model and module descriptors
  - Special libraries, RF, Analog modules do not follow standard scaling trends
  - Need to incorporate these modules into the system model
  - New descriptors and macro models
- Evolutionary algorithm
  - NSGA-II or SPEA2 Vs IRRR
  - FOM comparisons

Questions?