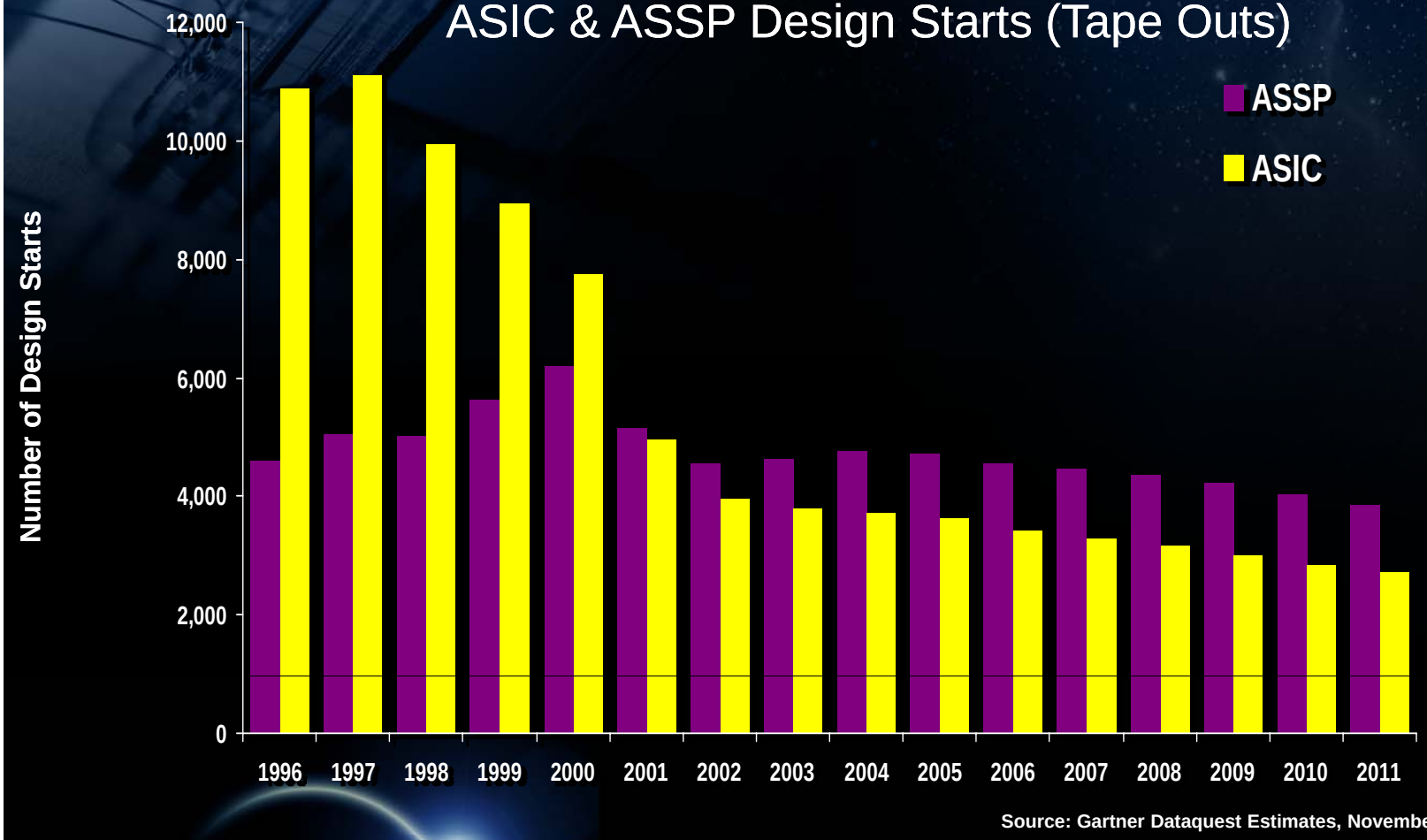
Placement Challenges for Structured ASICs

Herman Schmit

VP of Technology

eASIC Corporation

Custom IC Design Starts Decreasing



Source: Gartner Dataquest Estimates, November 2007

eASIC®

* Only 250 design starts projected in 2030! (source eASIC)

Causes of the Decline of Design Starts

- Costs:
  - Masks
  - EDA tools
  - Complexity, Intellect
- Verification effort is huge
  - Escalating costs lead to combined “super” chips, that further escalate verification costs
- These do not get any better in the future…
  - Process variation, manufacturability, etc.
- The contract is broken… Reasonable sized teams can’t make chips

eASIC®

Why a New ASIC?

- FPGAs cannot close the performance/power gap
- ASSPs cannot provide the customization required to differentiate products

- Do you need to specify 100s of layers to get customization you want?
- Can you get other advantages of FPGAs and ASSPs:
  - Preverified interfaces, IP, etc.

- eASIC Solution:
  - User customizes **one** via layer → cheap
  - All other mask costs are amortized over all customers for that standard part
  - Simplified flow
  - Manufacturability thru regularity
  - Reduced turn-around-time

eASIC®

Nextreme Family: 90nm

	eCells	Approx Gates	BRAM storage	PLLs	User IO
NX750	55,296	750K	864Kb	4	298
NX1500	100,352	1.5M	1.5Mb	6	450
NX2500	169,984	2.5M	2.7Mb	8	584
NX4000	276,480	4.0M	4.3Mb	8	742
NX5000	358,400	5.0M	5.6Mb	8	790

eASIC®

Differences from the ASIC EDA Problem

- Nothing fundamentally new, just a new mix of ingredients

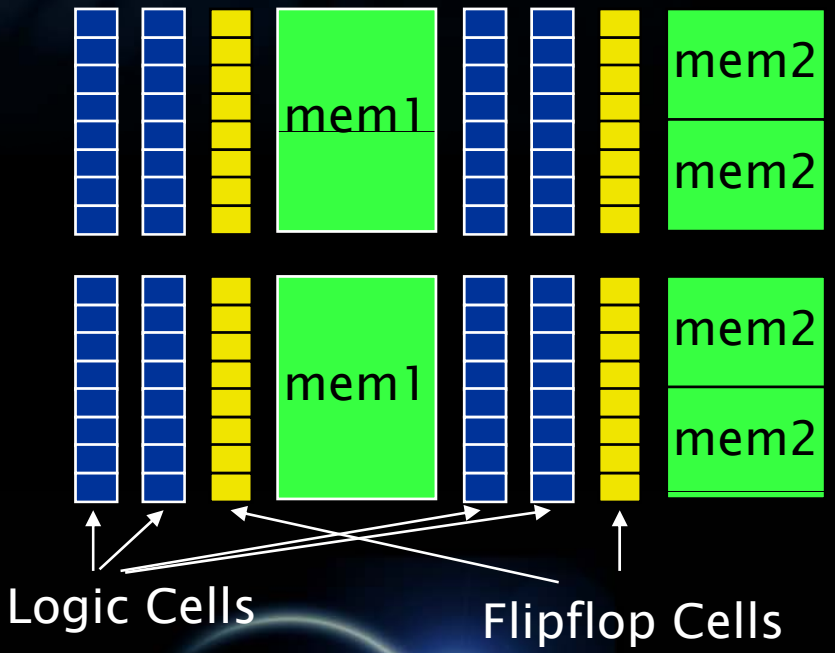
- Logic Synthesis and Technology Mapping
  - Small size LUTs — ASIC/FPGA like
- Placement and Buffering
  - Number and size of place-able objects — ASIC like
  - Legalization due to site compatibility — FPGA like
  - Legalization due to intrinsic resources (clocks) — FPGA like
  - Buffering needs to be done, but pre-allocated — Unique

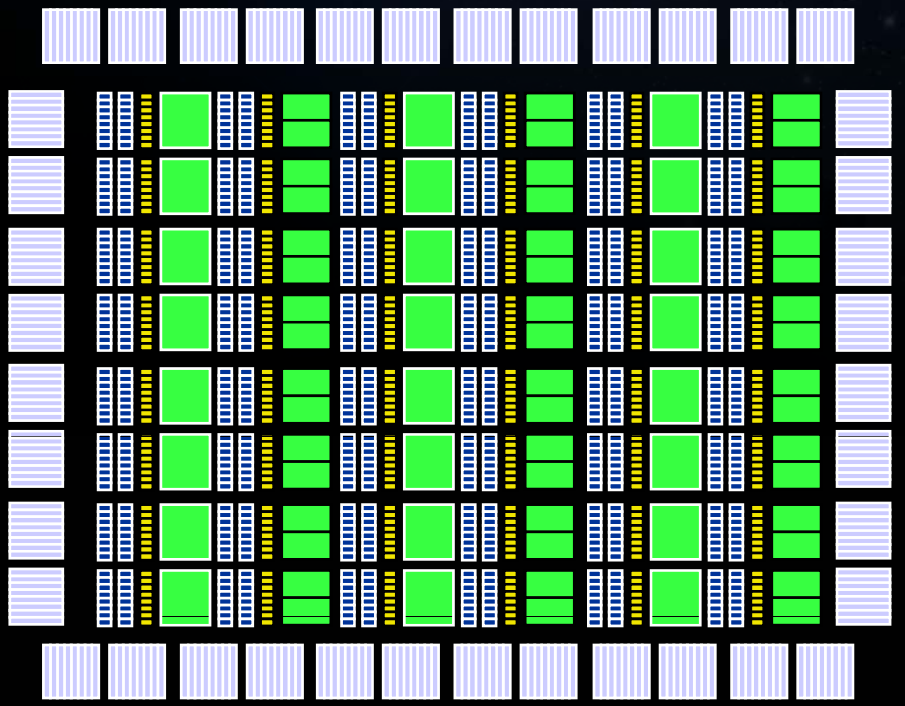
  } Focus
- Routing
  - “Embedding” like FPGAs, but with much more flexibility

eASIC®

Placement Legalization: Site Compatibility

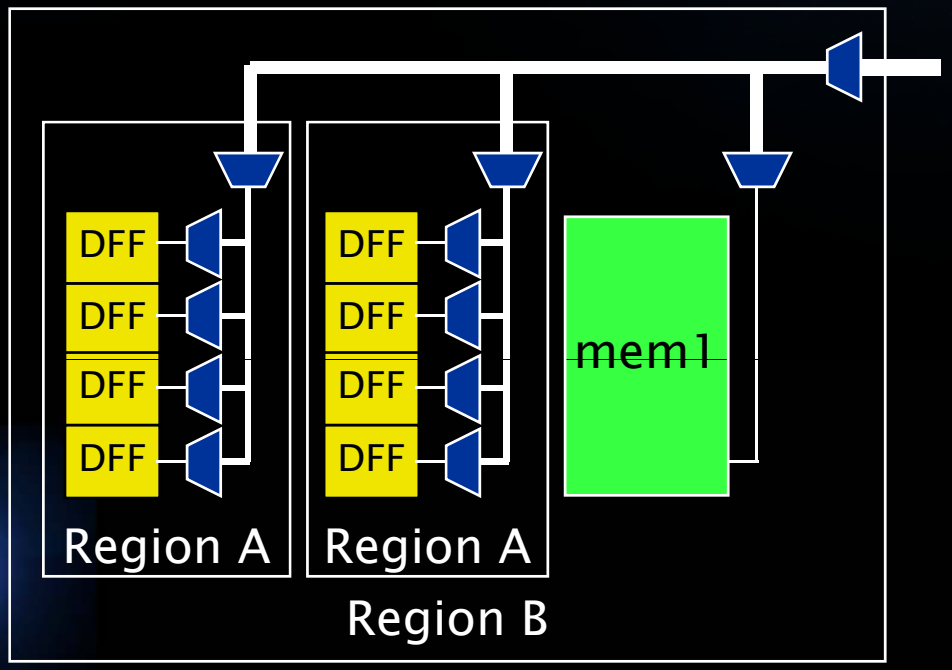
- Different cell types: logic, flip-flops, memories, buffers, IO and system resources (PLLs, DLLs, etc)
- Instances must go exactly on compatible site





Placement Legalization: Intrinsic Resources

- Clocks (and resets) are distributed globally with down-selection at different physical locations
  - Usually hierarchical, logically and physically
- Each region can have N clocks selected from among the surrounding regions



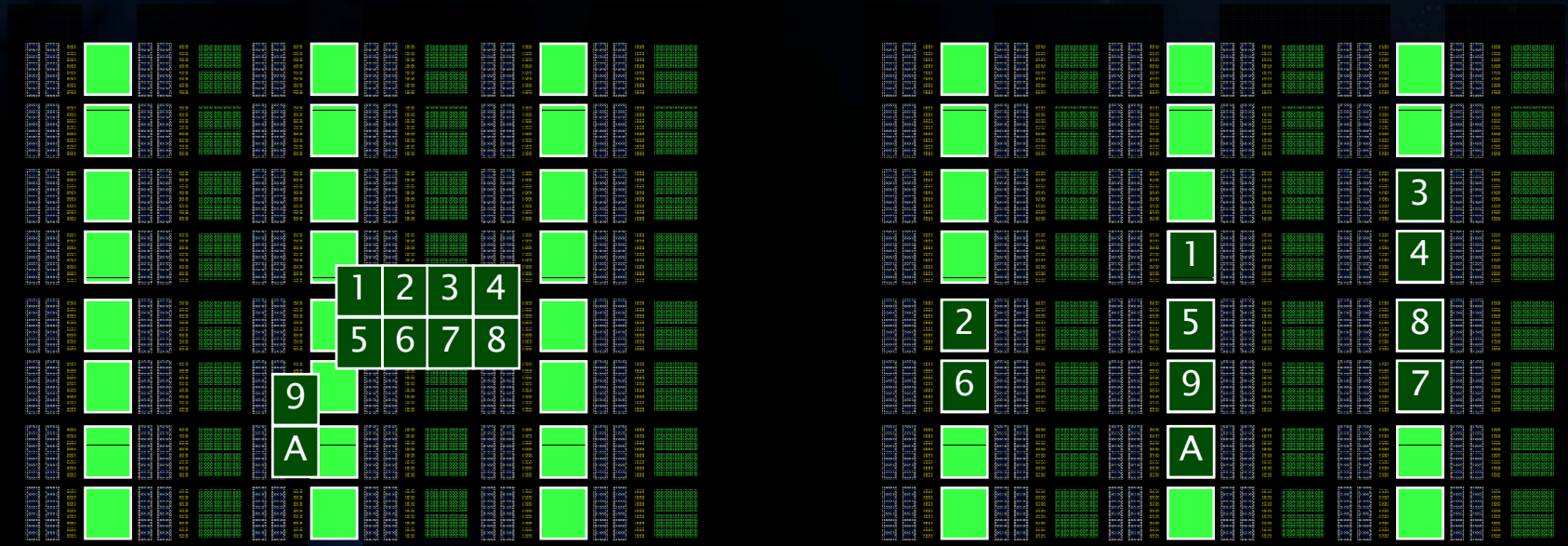
Our Current Solution

- Using adapted version of Magma ASIC tools
  - Use ASIC physical synthesis thru global placement
  - Local heursitics to move objects to legal solution
    - “Optimal” global place to legal site degrades results
- Symptoms of the heuristics
  - The Smear
  - The Yank
  - The Tangle

eASIC®

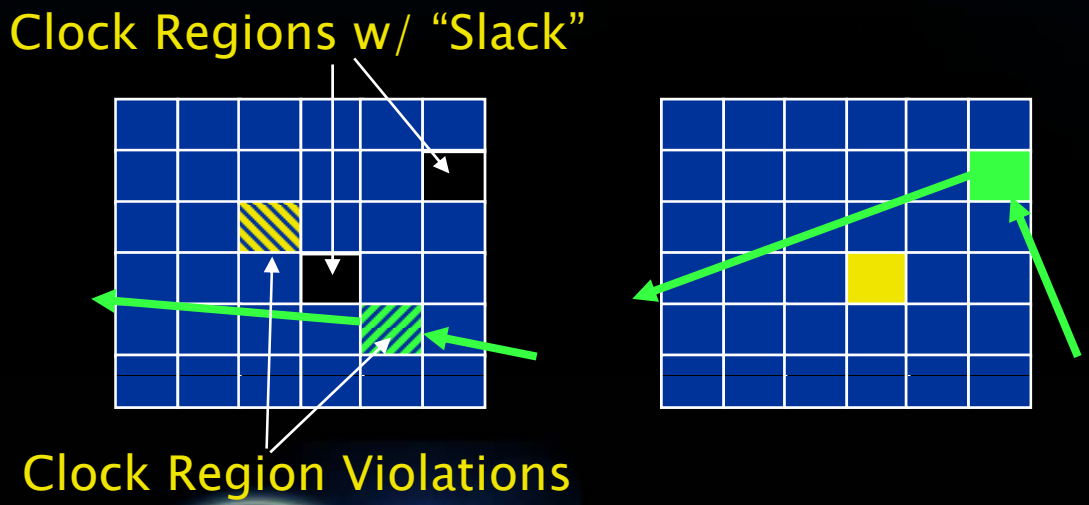
Symptom 1: The Smear

- Global placement resolved overlap but not site legality
- Getting from the no-overlap placement to legal placement…



Symptom 2: The Yank

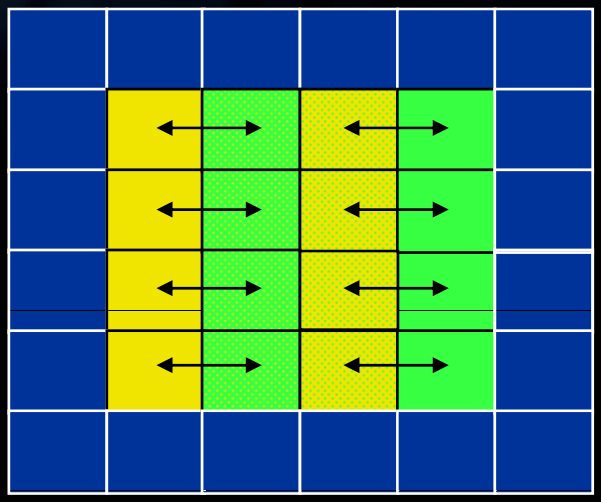
- Clock Legalization takes advantage of unallocated sites first
- Some elements moved significantly from their original location
  - Impact: Timing degradation

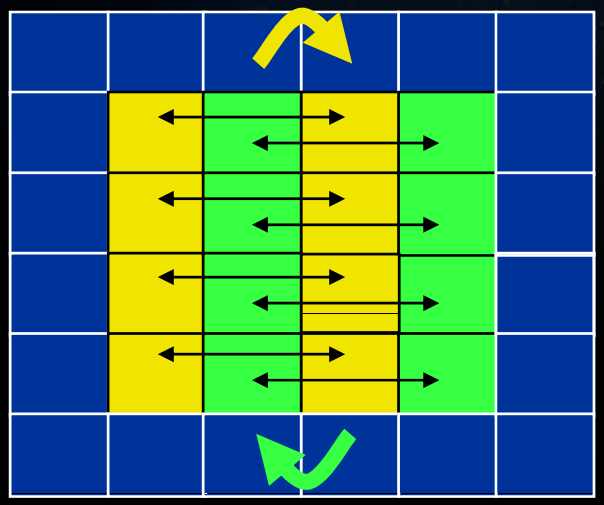


Available sites for violators may not be nearby

Symptom 3: The Tangle

- Routability Impact of Clock Legalization





How to Solve These Problems?

- Option A: Improve the Architecture
  - Build in much more flexibility so that
    - StdCell Solution maps much better to the Structured Solution
    - More clock domains per region
  - Problem: Hard to do with hard blocks (memories, IOs, etc)
  - Problem: chicken-and-egg
    - Few user designs or tools when the architecture is finalized
  - Problem: simple designs pay for the complexity of hard designs
- Option B: Improve the Software
- Our next generation will do both, carefully…

eASIC®

Software Improvements

- Obviously: improve the quality/scope of the optimization

- Option 1: Flow optimization
  - Use recipes of existing techniques to improve results
  - Getting this right requires time or luck
- Option 2: New Formulation
  - Eureka!
  - Getting this also requires time or luck

- Time or luck is not available, diversify our investment
  - Get the research community involved
  - Cast the problem, provide examples, incentivize the solution
  - Using “parallel engineering” we hope to have an edge to build a better placer in a shorter time

eASIC®

Casting the Problem

- Using the Nodes/Nets infrastructure for placement
- Supplementing with a set of files that present the limiations of the architecture

- .props file:
  - Corresponds to nodes, provides type and “color” information to correspond to clock domain
- .regions file:
  - Provides regional constraint information on top of .pl file

Properties File: *.props

```
EASIC props 1.0
# Created   :
# User      :
# first section defines the properties classes
PropertiesNumber : 4
PropClass
Name : clock_domain
Value : clock_1
Value : clock_2
Value : system_clock
EndPropClass
PropClass
Name : reset_domain
Value : reset_1
Value : reset_2
EndPropClass
PropClass
Name : type
Value : edff
Value : bram
Value : ecell
Value : reg_file
Value : eio_pad
EndPropClass
#end of prop declaration section
EndProps
```

```
#second section of the file contains a list of
    nodes names
#associated with properties and their values
NodesNumber :123
Node
Name : o0
Prop :  clock_domain Value : clock_1
Prop :  reset_domain Value : reset_1
Prop :  set_domain Value : set_dff
Prop :  type Value: dff
EndNode
Node
Name : o1
Prop :  clock_domain Value : clock_2
Prop :  reset_domain Value : reset_1
Prop :  type Value: dff
EndNode
…
…
#end of node list
EndNodes
```

Regions File: *.regions

```
eASIC regions 1.0
# Created   :
# User      :
#edff_column area definition
PropArea
Name        : edff_column
Width                   : 1
Height                  : 64
Property : type                     Values : edff
Property :reset_domain  NumColors : 1
Property :set_domain                NumColors : 1
EndPropArea

#edff_block  area definition
PropArea
Name        : edff_area
Property : clock_domain             NumColors : 4
#list of instances
#instantiate edff column 1
PropAreaInst            : edff_column : 0  : 0
#instantiate edff column 2
PropAreaInst            : edff_column : 0  : 1
EndPropArea
```

```
#bram area definition
PropArea
Name        : bram_area
Width                   : 20
Height                  : 60
Property : clock_domain             NumColors : 2
Property : type                     Values : bram
EndPropArea


#ecell area definition
PropArea
Name        : ecell_area
Width                   : 30
Height                  : 64
Property : type                     Values : ecell
EndPropArea
```

Regions File: *.regions

```
# group level area definition
PropArea
Name        : group_area
Property : clock_domain              NumColors : 8

#list of instances
#instantiate bram area
PropAreaInst            : bram_area : 0 : 0
#instantiate firs ecell area
PropAreaInst            : ecell_area : 0 : 20
#instantiate second ecell area
PropAreaInst            : ecell_area : 0 : 50
#instantiate edff_block area
PropAreaInst            : edff_area : 0 : 80
#instantiate reg_file area
PropAreaInst            : reg_file_area : 0 : 84
#end prop area “group_area”
EndPropArea


#cluster level area definition

#top level chip area definition

#instantiate top chip area
PropAreaInst            : top_chip_area : 0 : 0
```

Benchmarks

- Initial release: 5 – 7 designs
  - 2-3 focus on Site Legality, with a single clock
  - 3-4 focus also include multiple clocks
  - Placeable objects: up to 1.5M objects, 400 RAMs
  - Clock Domains: Up to 35 domains
  - Regions files for each benchmark will be provided
  - Still trying to improve the synthesis flow to get more appropriate cell counts
- Final Release will include total of 10 benchmarks

eASIC®

Benchmark Sample

Name	eCells *	eDFFs	BRAMs	Regfiles	Clk/Rst Domains
etens2	912K	53K	192	0	1
etens4	1.8M	106K	384	0	1
sdvt1					
sdvt2					

eASIC®

Incentives: ePrize1

- Incentive to the research team able to achieve the best result in the timeframe
- Requirements for the prize:
  - Registration by May 15
  - Check in of intermediate results in September
  - Publishable/re-usable source code for winning solution
    - Exact terms of the license TBD
- Criteria: similar to previous placement experiments
  - HPWL
  - CPU Penalty Factor from ISPD 2006 Contest
  - Complete legality of final placement

ePrize1: Size Matters

- Google Lunar X Prize: $20,000,000
- Netflix Prize: $ 1,000,000
- Android Developer Contest: $ 200,000 per app

- eASIC ePrize1: $ 30,000

Contest Site and Timelines

- Site: easic.com
- Registration open: April 15
- Initial Contest Rules: May 1
- Initial Data Release: May 1
- Contest Checkpoint: Sept 15
- Final Results Submissions: Nov 5
- Award announced at ICCAD

Roadmap for the future

- Plan: ePrize2
- Vision:
  - Build real placers of real netlists on a real timing analysis capable framework
- We are working with the Open Engines initiative to build a placement layer on top of OpenAccess
- Looking for collaborators, participants, reviewers

Summary

- Structured ASIC EDA problems are an amalgam of ASIC/FPGA problems
- Legalization, but MUCH larger
  - Not adequately researched… yet
- Opportunity for fame, fortune, and perpetual gratitude

eASIC®