

Virtual Prototyper (ViPro): An Early Design Space Exploration and Optimization Tool for SRAM Designers

Satyanand Nalam, Mudit Bhargava, Ken Mai,
Benton H. Calhoun



UNIVERSITY
of VIRGINIA

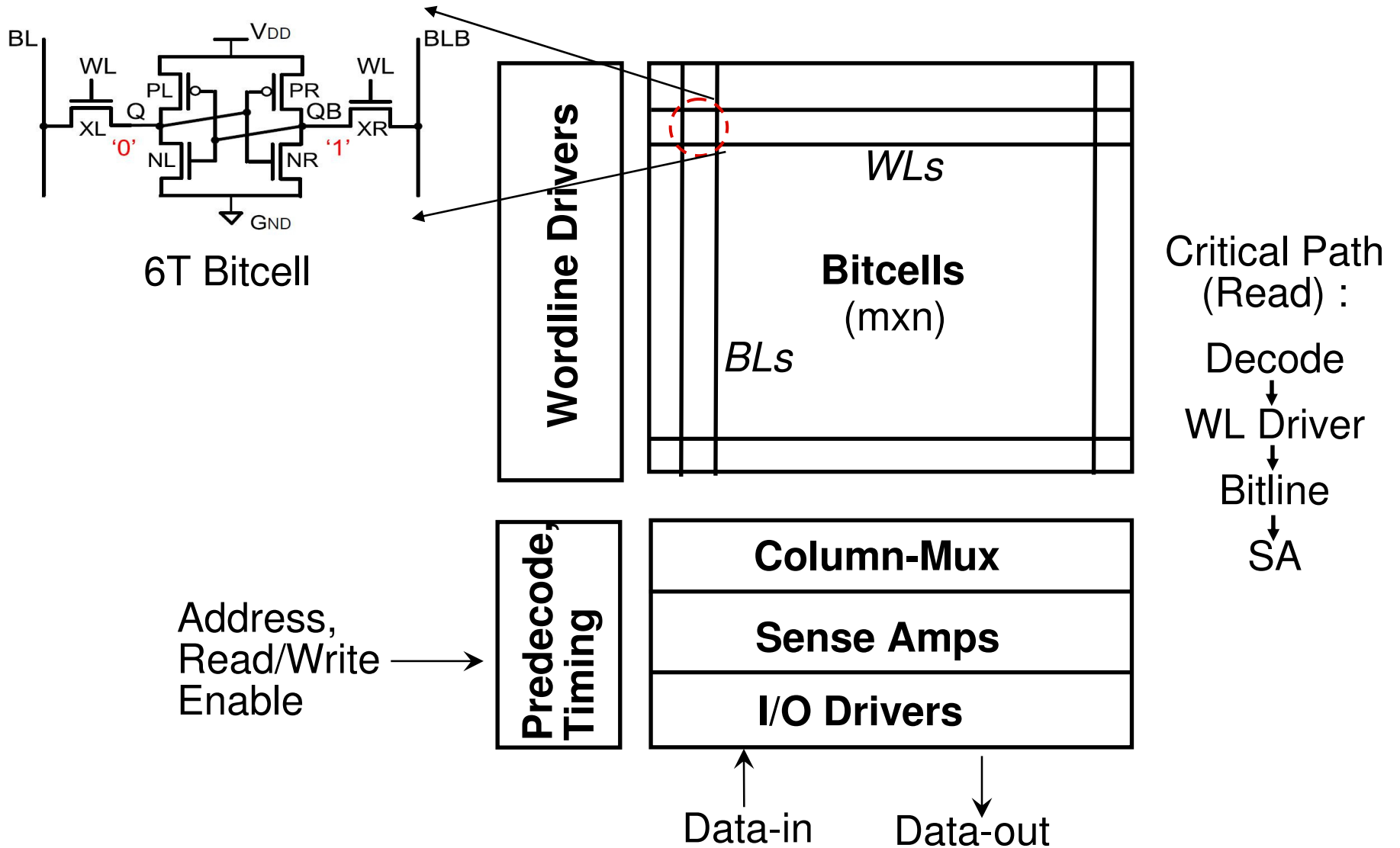
SCHOOL OF ENGINEERING
AND APPLIED SCIENCE

Carnegie Mellon

Overview

- Introduction and motivation
- Flow/Design methodology
- Tool components
 - ⊙ Characterizer
 - ⊙ Hierarchical SRAM model and “Meta” Compiler
- Usage Example
- Conclusions

SRAM – single bank macro



Optimal SRAM design

How to combat
SRAM scaling
challenges? →

Alternative bitcells

R/W Assists

Process
innovations

Architectural
innovations

→ What method(s)
result in optimal
SRAM?

- Energy/Power
- Performance
- Yield
- Area

Virtual Prototyping tool (ViPro)

1. Base-case SRAM



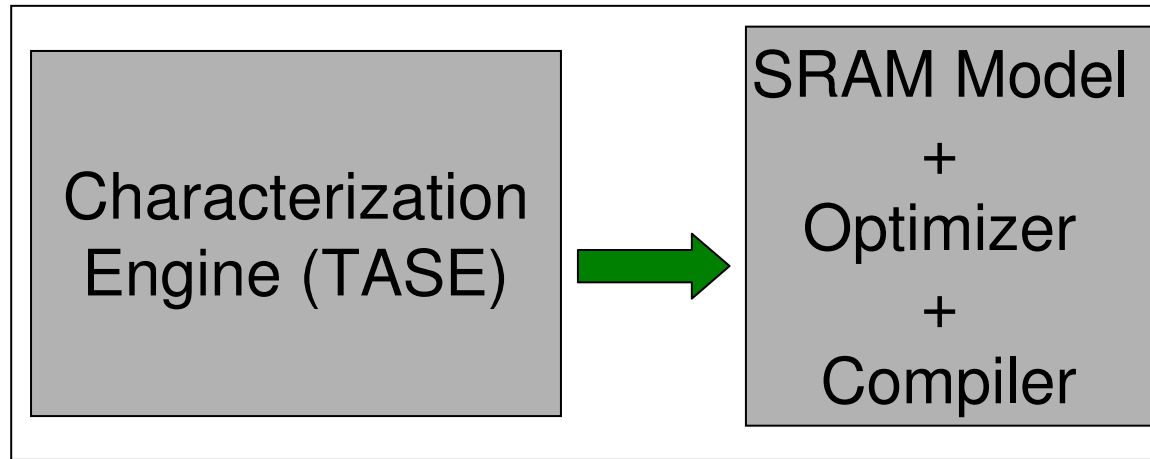
2. Quick re-optimization for rapid design space exploration – by using variable levels of detail



Key Idea: Iterative optimization through designer intervention

Design Methodology using ViPro

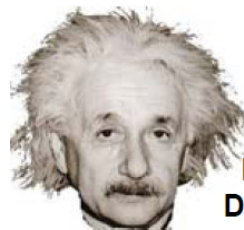
ViPro



INPUTS

- Tech details
- Specs
 - Architectural
 - Circuit components
- Constraints
 - E,D
 - Bitcell area
 - Noise margins

Iterative Design



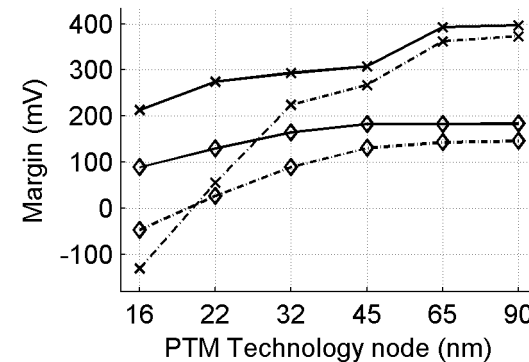
Expert Designer

OUTPUTS

- SRAM virtual prototype
 - Top-level metrics
 - Trade-off curves

Technology Agnostic Simulation Environment (TASE)

Observe trends, “port” circuit analysis



[Nalam et al, ICCD '09]

TASE

Use 1: Stand-alone
→

Use 2: Part of ViPro
→

Characterize SRAM components in terms of Power and performance

TASE: Example template (Spectre)

```
//=====
// Half cell
//=====
subckt HALFCELL (IN OUT BL WL VDD VBP VSS VBN)
  MP (OUT IN VDD VBP) PU_TRANSISTOR width=wpu length=lpu
  MN (OUT IN VSS VBN) PD_TRANSISTOR width=wpd length=lpd
  ...
  ...
  VU (U 0) vsource dc=pvln
  VWL (WL 0) vsource dc=pvdd
  ...

```

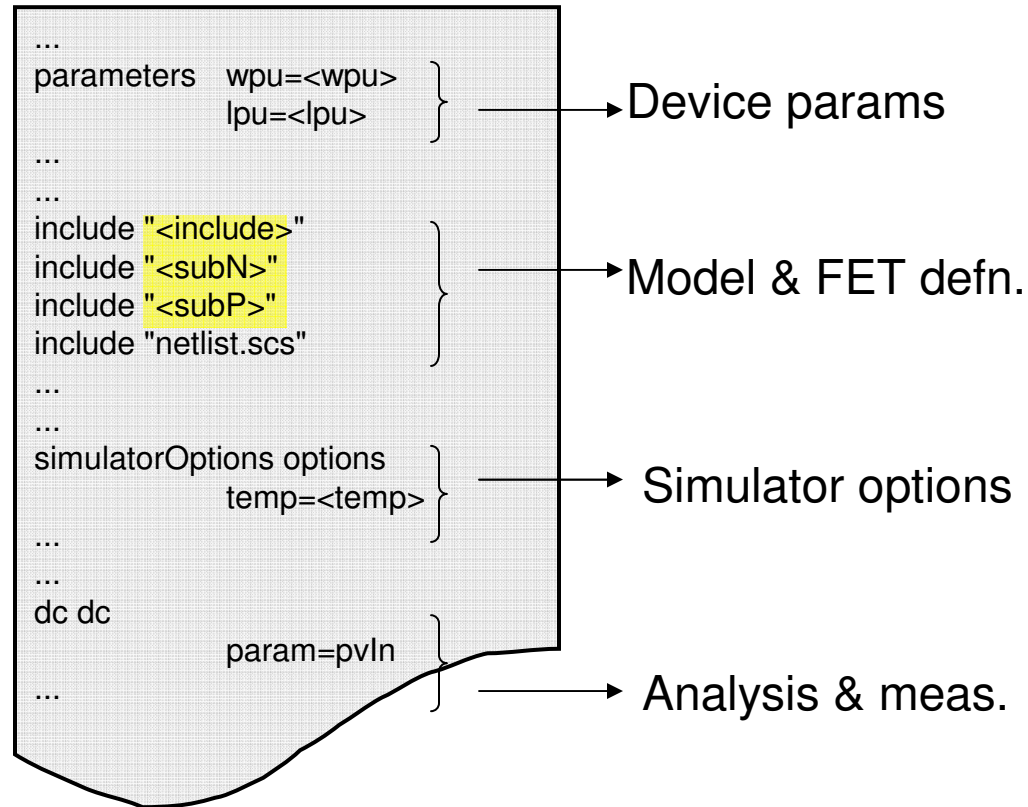
Stimuli, ic {

Generic FET names

Sub-component template netlist

TASE: Example template (Spectre)

XML-Style
Parameter Tags



Technology-agnostic characterization sweeps

TASE Example Execution file

```
...
<lpu>      100n
<wpu>      200n
<pvdd>     1.0
<subN>     ../../../../template/ptm90/subN.scs
<subP>     ../../../../template/ptm90/subP.scs
<include>  ../../../../template/ptm90/include.scs
...
...
#####
# TEST EXECUTION SELECTION
#####
<scs>
IDVD_N
...
</scs>
<ocn>
...
</ocn>
```

Netlist parameters {

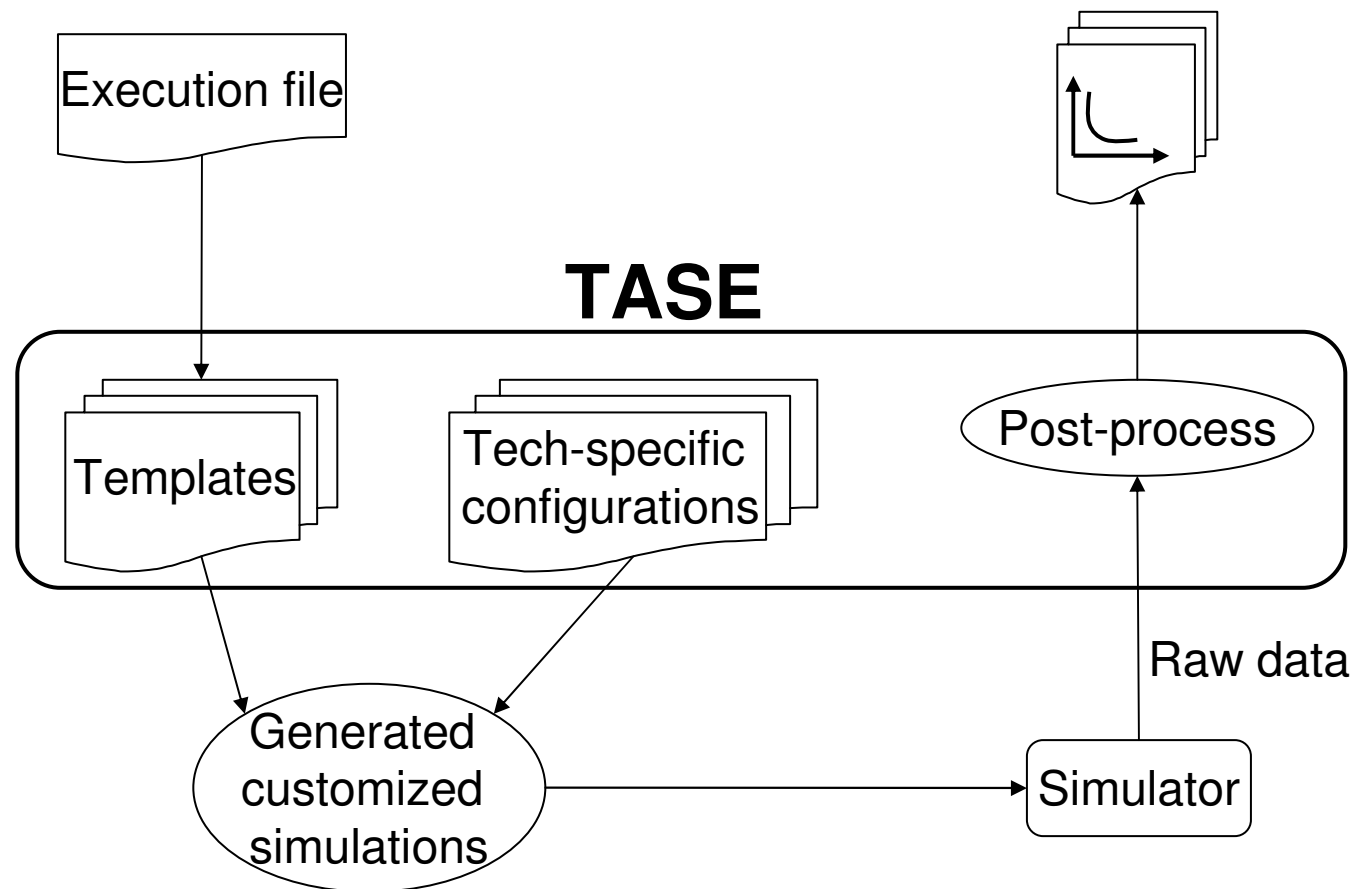
Model files {

Simulation set {

Tech-specific parameter tag defns. in exec. file

TASE: Characterization Flow

Sub-component
characterization data



Hierarchical SRAM model

Global variables: #R,
#C, col-mux, word-size



SRAM top-level

Decoder

WL Driver

Bitcell

I/O

SA



Sub-component metric info:

- TASE characterization
- User inputs
 - E,D Guesstimates
 - Analytical – CV^2

Variable
levels of
detail

Hierarchical SRAM model

- MATLAB implementation
- Component objects
 - Properties
 - Local – E.g. SA offset
 - Global/Inherited – E.g. number of rows
 - Methods
 - E, D wrapper functions
 - TASE characterized output
 - User provided guesstimates/ analytical expressions
- Co-optimization using global properties inherited from a parent class.
- Example in next slide.

Co-optimization example

Bitcell object:

Local property – device width

Inherited property - **#rows**

$E/D = f(\text{local \& inherited})$

SA object:

Local property – input device width, offset

Inherited property - **#rows**

$E/D = f(\text{local \& inherited})$

- Capture bitcell, SA dependencies
- Co-optimize #rows
 - Change SA input device width to adjust offset
 - Increase bitcell pass-gate width to improve drive

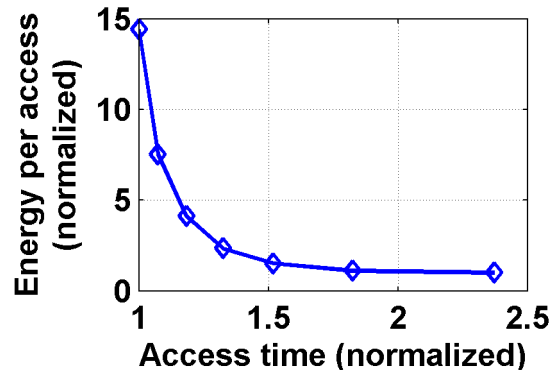
Optimization details

- $E_{\text{SRAM}} = \sum E_{\text{components}}$
- $D_{\text{SRAM}} = \sum D_{\text{CPath}}$
- Yield model (J. Wang, ESSCIRC '07) and bitcell area model incorporated
- Optimization objectives and constraints – SRAM Energy per access and access time (E, D)
- Optimization knobs - #rows, #columns of bitcell array
- Brute force search for optimal solution
- Framework extensible:
 - Introduce more knobs – local and global
 - Introduce more objectives – area and yield
 - Use more powerful optimization tools

Usage Example

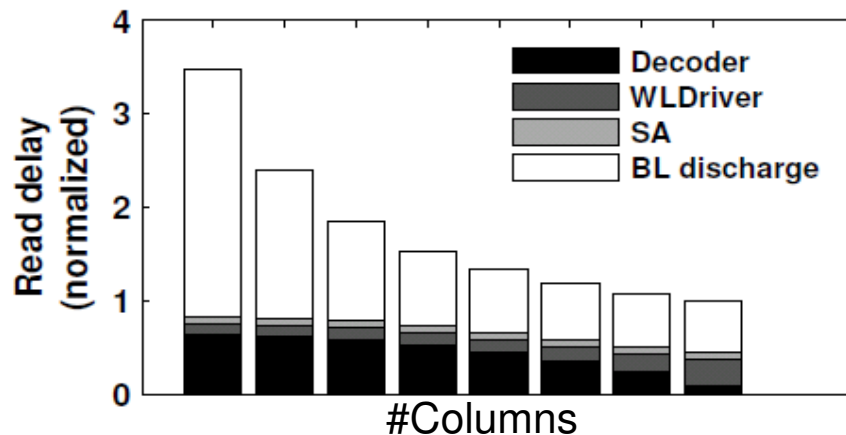
Steps 1,2:

1. Process and Sub-component characterization (TASE)
2. Generate Base-case prototype



Outputs:

1. Trade-off (e.g E-D) curves
2. Cross-component metric breakdown



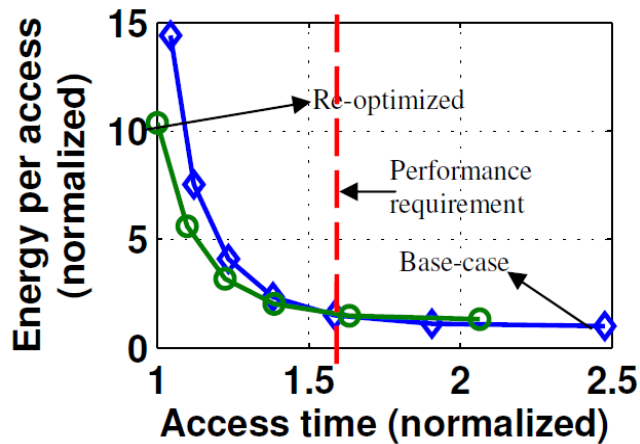
Usage Example

Step 3: Design space Exploration

- Change bitcell
 - ⊙ Larger PD for more I_{READ}
- Use faster SA – still in early stages of design
 - ⊙ 24% lower delay
 - ⊙ 15% higher offset
- Inputs provided with variable levels of detail:
 - ⊙ Modified bitcell template added to TASE library
 - ⊙ Guesstimates of SA delay and offset input to SRAM model

Usage Example

Step 4: Re-run tool for re-optimization



Access time < 1.6
⇒ New design
optimal

Step 5: Continue exploration and re-optimization or stop

Conclusions

- Presented an early SRAM design space exploration tool
 - Technology-agnostic
 - Base-case generation
 - Variable levels of detail
 - Re-optimization
- Virtual prototyping methodology extensible to any design space exploration