CharFlo-Memory!™

A Total Solution for Characterizing and Verifying Memory IP from Memory Compilers
Agenda

- Introduction
- Legend’s Products and Roadmaps
- CharFlo-Memory! Characterization Flow
- Memory Characterization with Reliability Check
- SpiceCut Technology
- Performance Optimization
- Power-Gating Solutions
- The Conclusion
- Appendix: ActiveNet Flow
Legend’s Products

◆ IP Library Characterization Products
  ● Charflo-Cell™: *Automatic Cell/IO Library Characterization*
  ● Charflo-Memory™: *Automatic Memory Characterization*

◆ IP Library Model Quality Assurance Products
  ● Model Diagnoser-Cell™:
    *Cell Library .Lib Quality Assurance and Defect Repair*

◆ Circuit Simulation Products
  ● MSIM®: Accurate-Spice Simulator
  ● Turbo-MSIM™: Fast-Spice Simulator
Tools for Semiconductor IP
Characterization and Verification

IP Macros

Custom Memory

Compiled Memory

Standard/IO Cell Library

CharFlo-Cell!

CharFlo-Custom!

CharFlo-Memory!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!

CharFlo-Cell!
CharFlo-Memory!™
Push-button Characterization Flow

- **MSL™**: MemChar SpiceCut Library, the tool for automating memory characterization with ‘.Lib-in and .Lib-out’
- **SpiceCut™**: Critical-path circuit building tool based on layout-extraction with RCs
- **MemChar™**: Memory characterization tool with optimization.
- **MSIM™**: High-accuracy circuit simulator
In-House and Commercial Memory Compilers

- Legend’s products have been successfully used for in-house memory compiler development including 40/45/65/90 nm.
- Legend provides ‘push-button’ characterization tools for

<table>
<thead>
<tr>
<th>Vendors’ Memory Compilers</th>
<th>180nm</th>
<th>130nm</th>
<th>90nm</th>
<th>65nm</th>
<th>45/40nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artisan*</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Virage</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>TSMC*</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Faraday*</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Virtual Silicon*</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Synopsys (Avanti)</td>
<td>YES</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Dolphin Technology*</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>VeriSilicon*</td>
<td>YES</td>
<td>YES</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* Legend’s customer and/or partner

N/A means indicated compilers not available

Leader in IP Characterization and Circuit Simulation
Major Applications
For Commercial Memory Compilers

- Incoming QA for high yields
  - Analyze the mismatching for re-characterization or not.
  - Perform reliability checks (e.g. glitch, signal vs noise)
- Instance re-characterization at new PVTs
  Generate accurate instance models (.lib file), instead of the interpolating or extrapolating by compiler models
- Optimizing voltage supply for low-power designs
- Silicon Failure Analysis
  Eliminate the timing issues for specific instances as possible cause of failure, and focus on other potential problems
What-if Analysis
By Using CharFlo-Memory!™

◆ The matrix of possible variations
  - PVTs (Process, Voltage and Temperature)
  - Spice models
  - Bit-cells
  - MOS device sizes (e.g. W/L) and parasitic R and C

◆ Scenarios
  - Locate minimum voltage supply of low-power designs to pass the read-margin check
  - Locate minimum voltage supply of memory banks to pass the voltage-retention check
CharFlo-Memory!™
A Complete Solution for Characterization

- For both In-house memory compiler and vendors’ memory compiler
- For both memory characterization and validation
- For both pre-sim and post-sim characterization
- For both ‘critical-path’ circuit and ‘full’ circuit characterization
- Enable automatic layout RC extraction on active-net only, which enhance throughputs tremendously
Instance Characterization
Compare with Compiler Models

Compiler timing < Actual timing
- Actual Silicon (Instance Model)
- Compiler Model (no margin)

Compiler timing >> Actual timing
- Slow Design!
- Speed Sacrifice!

Design Failure! Low Yield!

CharFlo-Memory!™ can produce actual instance model!

Legend
Leader in IP Characterization and Circuit Simulation
Bi-Section Optimization

Bi-Section model in MemChar™ is based upon multi-criterion ‘binary search’ algorithm. The convergence is controlled by the ‘BiSect Error’

\[ t_1 = \text{Start} \]
\[ t_2 = \text{Start} \]
\[ t_3 = \frac{(t_1 + t_2)}{2} \]
\[ t_4 = \frac{(t_2 + t_3)}{2} \]
\[ t_5 = \frac{(t_3 + t_4)}{2} \]

<table>
<thead>
<tr>
<th>Time</th>
<th>Goal</th>
<th>BiSect Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Fail</td>
<td>Norm(t2-t1)</td>
</tr>
<tr>
<td>t2</td>
<td>Success</td>
<td>Norm(t4-t3)</td>
</tr>
<tr>
<td>t3</td>
<td>Fail</td>
<td>Norm(t3-t2)</td>
</tr>
<tr>
<td>t4</td>
<td>Success</td>
<td>Norm(t5-t4)</td>
</tr>
<tr>
<td>t5</td>
<td>Success</td>
<td></td>
</tr>
</tbody>
</table>
Reliability Checking
‘Glitch’ and ‘MetaStability’ Prevention

CharFlo-Memory! locates ‘glitch-free’ ‘metastability-free’ setup/hold time

Reliability Problem!

Legend
Leader in IP Characterization and Circuit Simulation
Even functionally working, ‘too short’ setup/hold time will prolong the access time which degrades chip performance.
Read Margin Check
Sense-Amp Signal vs Internal Noise

- CharFlo-Memory™ can locate
  - Bit and BitB signal
  - Sense control signal
- CharFlo-Memory™ can automate the simulation output and measurement for reliability check

Legend
Leader in IP Characterization and Circuit Simulation
Determine Read-Margin Setting
Checking for Yields and Speed

- Determine Read-Margin setting based upon access time,
  Voltage(Bit-BitB) against noise-margin

<table>
<thead>
<tr>
<th>Read Margin</th>
<th>Access Time</th>
<th>V(Bit-BitB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>4.3212 ns</td>
<td>365.8 mv</td>
</tr>
<tr>
<td>0100</td>
<td>1.8824 ns</td>
<td>95.9 mv</td>
</tr>
<tr>
<td>1101</td>
<td>2.8109 ns</td>
<td>200.6 mv</td>
</tr>
<tr>
<td>0111</td>
<td>1.8824 ns</td>
<td>95.9 mv</td>
</tr>
<tr>
<td>1111</td>
<td>1.6969 ns</td>
<td>68.3 mv</td>
</tr>
</tbody>
</table>

* Noise margin is 100 mv minimum normally, and 200 mv for the safer.

Wrong Read Margin (RM) setting causes Poor Yield!

Legend
Leader in IP Characterization and Circuit Simulation
CCS/ECSM Model Support
Timing, Power and Noise

- SpiceCut can partition memory circuit to receiver, driver, and channel-connected blocks (CCB).
- Based on those partitions, CharFlo-Memory! can characterize CCS Timing, Power and Noise models.
SpiceCut Functions

- **Circuit characterization**
  Extract ‘critical-path’ circuits for
  - Timing / Power simulations
  - Access / Setup / Hold Time / Minimum Clock

- **Circuit verification for**
  - Worst / best wordlines of decoders
  - Exhaustive address pattern simulation
  - Coupling analysis

- **Built-in RC reduction**
Recognize Memory Structure

- Pattern matching with core cell
- Locate all memory banks, words and bits

**Memory Cell**

```
.subckt dual_cell 3 4 5 6 14 16
.alias word = 14
.alias bit = 5
.alias bitb = 4
.alias word2 = 16
.alias bit2 = 6
.alias bit2b = 3

M29  9  8  VDD  7  P  L=.40U  W=.60U
M30  VDD  9  8  7  P  L=.40U  W=.60U
M33  0  8  9  11 N  L=.40U  W=1.37U
M34  0  9  8  11 N  L=.40U  W=1.37U
M37  4  14  9  11 N  L=.40U  W=.85U
M38  5  14  8  11 N  L=.40U  W=.85U
M39  3  16  9  11 N  L=.40U  W=.85U
M40  6  16  8  11 N  L=.40U  W=.85U
.ENDS
```

**SpiceCut Commands**
Extract Critical-Path Circuit
Using SpiceCut for Access Time

Complete ‘multiple-paths’ critical circuit for all PVTs
Extract Critical-Path Circuit
Using SpiceCut for Setup/Hold Time

- SpiceCut provides multiple automatic methods for extracting critical-path circuits of setup/hold time toward the reference nodes, such as
  - Latch node
  - Bit-line control node
  - Word-line / word-line control node
  - Memory-cell internal node

- The critical-path circuit for setup/hold time is a very complete one including ‘multiple-paths’. That is good for simulating all PVTs.
‘Latch Node’ Example
For Address Setup / Hold Time

Latch output N101 can be automatically located, and then critical-path circuits extracted for Address setup/hold time.
‘Inside Memory-Cell’ Example
For Data-In Setup / Hold Time

Internal node N1 / N2 of memory cell can be automatically located, and then critical-path circuits extracted for Data-In setup/hold time.
Critical-Path Circuit for Power
Performance Optimization
Segmenting MOSFET Loadings

- When building critical-path circuits, segment those inactive MOSFETs, resistors and capacitors on the nets with heavy loadings, such as
  - Word-lines,
  - Bit-lines and
  - Nodes along critical-path circuits

- Example: word-line segment

---

Legend: Leader in IP Characterization and Circuit Simulation
# Performance Optimization

Segmenting MOSFET Loadings

45nm 256K SRAM using Subcircuit Spice model, with about 1.7 Million MOSFETs

<table>
<thead>
<tr>
<th>Accuracy Comparison</th>
<th>Before Segment 50,488 MOSFETs</th>
<th>After Segment 10,292 MOSFETs</th>
<th>Accuracy Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rise</td>
<td>Fall</td>
<td>Rise</td>
</tr>
<tr>
<td>Access Time</td>
<td>325.94p</td>
<td>351.53p</td>
<td>324.95p</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speed and Memory Usage Comparison</th>
<th>Before Segment 50,488 MOSFETs</th>
<th>After Segment 10,292 MOSFETs</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU Time</td>
<td>Memory Usage</td>
<td>CPU Time</td>
</tr>
<tr>
<td>Access Time</td>
<td>8,279 sec</td>
<td>3.8 GB</td>
<td>674 sec</td>
</tr>
</tbody>
</table>
Power Gating Solutions
SpiceCut for Building Critical-Path Circuits

- SpiceCut automatically recognizes Virtual_Vdd and Virtual_VSS in memory circuits with Power Gating
- On Virtual_Vdd / Virtual_VSS, SpiceCut executes
  - RC reduction and port merging
  - MOSFET segmenting on loadings
- Based on Power Gating structures, SpiceCut enables
  - Memory structure recognition
  - Path tracing/pattern identification
  - Critical-path circuits generation

Legend
Leader in IP Characterization and Circuit Simulation
Legend and Foundry

- Legend’s tools have been adopted and silicon-proven by major foundries
  * TSMC
  * UMC
  * Dongbu
  * Vanguard
  * HHNEC
  * Jazz
  * Tower
- Direct access the updated SPICE models
  * TSMC
  * UMC
  * IBM
  * Vanguard
  * X-FAB
  * Chartered
  * SMIC
  * Tower
  * Jazz
Legend and Library Vendors

- Support push-button re-characterization for
  - Artisan
  - Virage
  - TSMC
  - Virtual-Silicon
  - Dolphin Technology
  - Faraday (UMC Alliance)
  - VeriSilicon
  - Synopsys

- The instance models using Legend’s tools have been silicon-proven by customers and major foundries.
The Conclusion
Legend’s Memory Characterization

- Production and silicon proven including 45nm/65nm
- Automatically build critical-path circuits and structures from layout-extracted circuit with RC
- Access memory internals and reliability checking
- Optimized critical-path circuits for Power-Gating
- RC reduction and ‘active-net’ extraction for speed
- Support CCS timing, power and noise models
- Foundry and IP vendors’ support
- Excellent price-performance and automation
Appendix

‘ActiveNet’ Method for Efficient Layout Extraction
Active Nets Flow I
Generate Active Nets for Layout Extraction

Memory Instance

- Schematic Spice Netlist
- Programmable MSL Specification
- Layout GDS2 File

Interpret to SpiceCut Controls

SpiceCut™

- Critical-path circuits
- List of Active Nets in Critical-path Circuits

Layout Extraction Tools

StarRC-XT™
Calibre xRC™

Legend
Leader in IP Characterization and Circuit Simulation
Active Nets Flow II
Characterization based on Active Nets

Layout Extraction Tools

Layout-extracted Spice Netlist with RC on Active Nets only

Programmable MSL Specification

MSL™

Interpret to SpiceCut Controls

Build critical-path circuits

SpiceCut™

Customized by compiler designs

Perform memory characterization

MemChar™

Legend
Leader in IP Characterization and Circuit Simulation