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

Characterization
Re-Characterization

CharFlo-Custom!

Build Critical-Path Circuit by

Global-SpiceCut

User-defined Stimulus & Measure

***Data Sheet or Spec**

Compiled Memory

Characterization
Re-Characterization

CharFlo-Memory!

Build Critical-Path Circuit by

SpiceCut
Turbo-SpiceCut

Build Compiler Model by

Compiler Model Generator

Standard/IO Cell Library

Characterization
Re-Characterization

CharFlo-Cell!

Verification & QA
Defect Repairing

Model Diagnoser

Leader in IP Characterization and Circuit Simulation

Legend
Design Technology

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™ :**

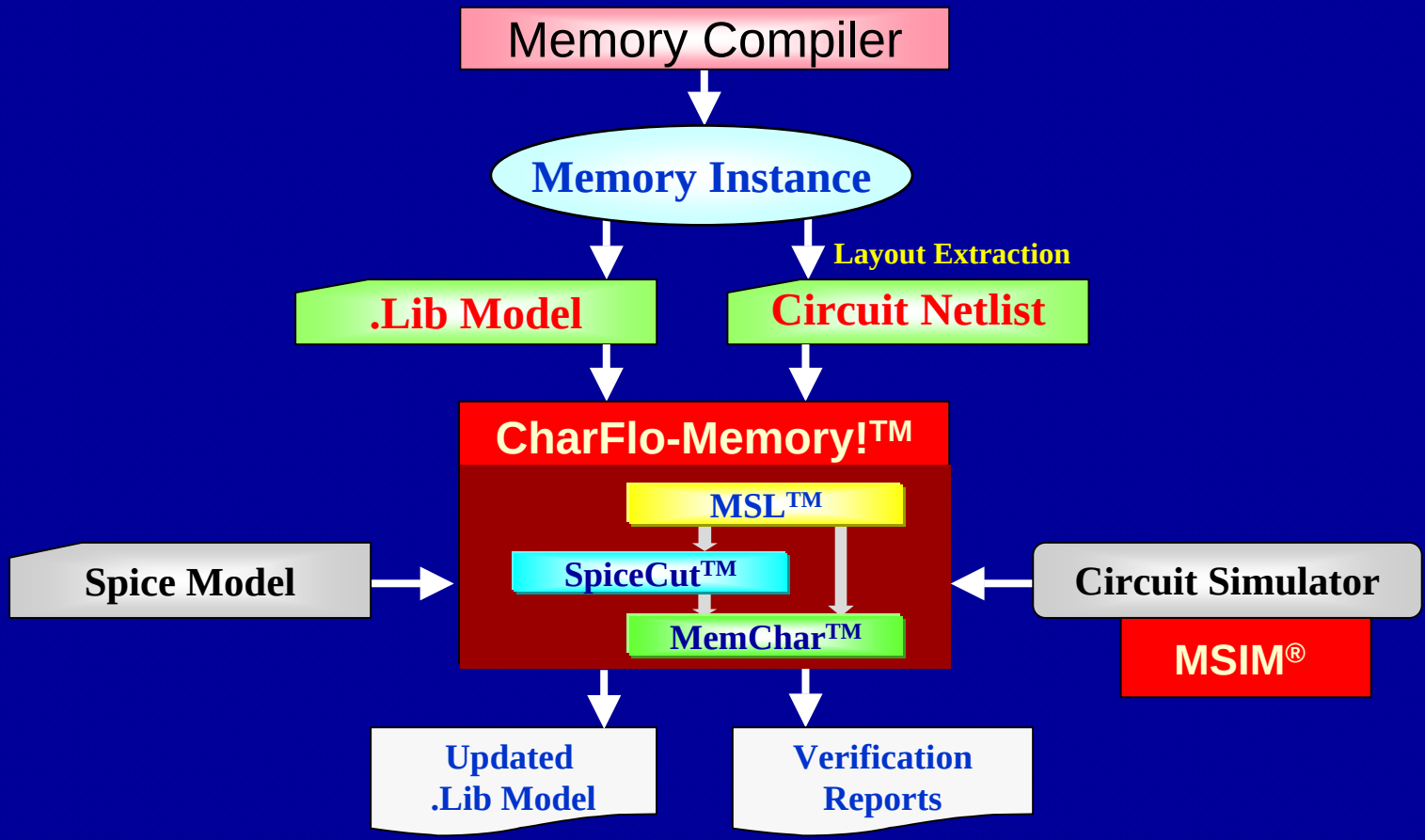
  Memory characterization tool with optimization.

- **MSIM™ :**

  High-accuracy circuit simulator

CharFlo-Memory!™

Push-button Characterization Flow



Leader in IP Characterization and Circuit Simulation

Legend
Design Technology

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

Vendors' Memory Compilers	180nm	130nm	90nm	65nm	45/40nm
Artisan*	YES	YES	YES	YES	YES
Virage	YES	YES	YES	YES	YES
TSMC*	*N/A*	YES	YES	YES	YES
Faraday*	YES	YES	YES	YES	YES
Virtual Silicon*	YES	YES	*N/A*	*N/A*	*N/A*
Synopsys (Avanti)	YES	*N/A*	*N/A*	*N/A*	*N/A*
Dolphin Technology*	*N/A*	YES	YES	YES	YES
VeriSilicon*	YES	YES	*N/A*	*N/A*	*N/A*

**** Legend's customer and/or partner***

N/A means indicated compilers not available

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

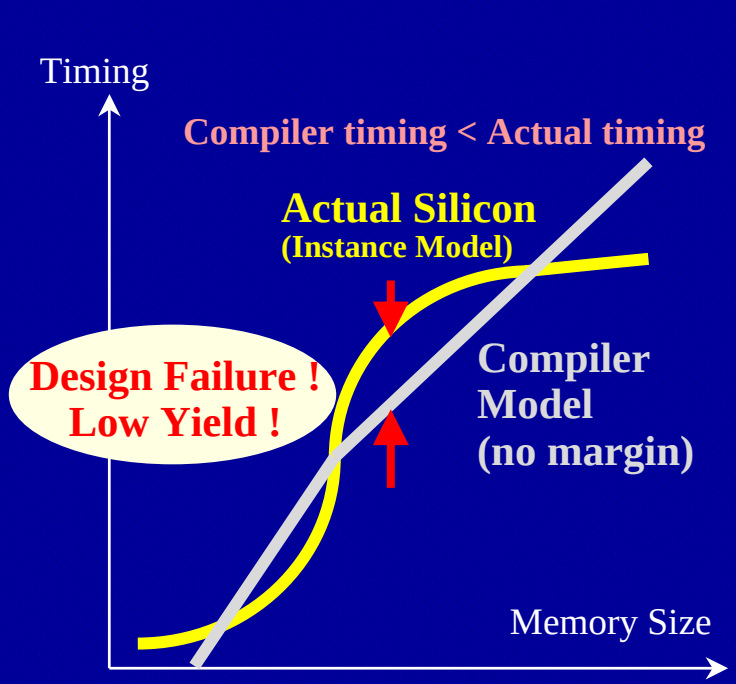
CharFlo-Memory!™

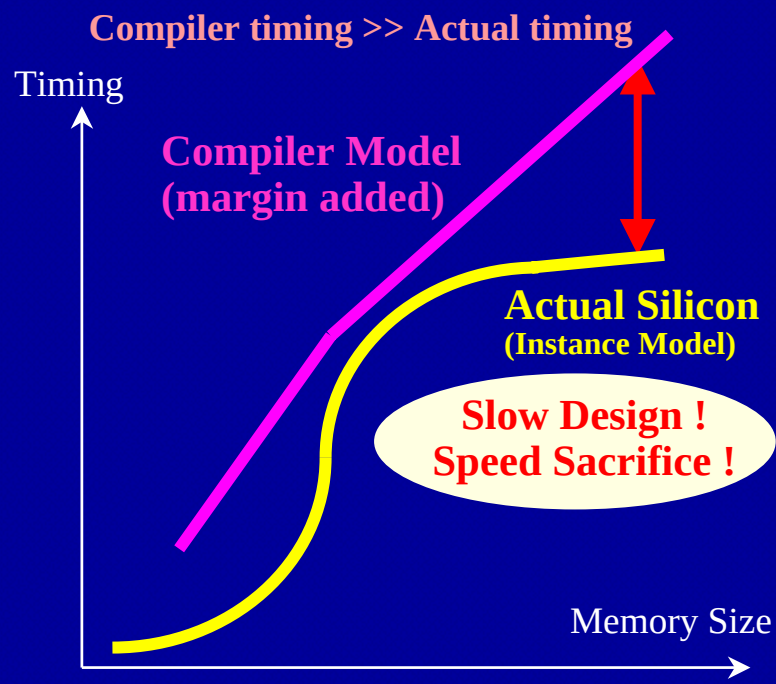
A Complete Solution for Characterization

- 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

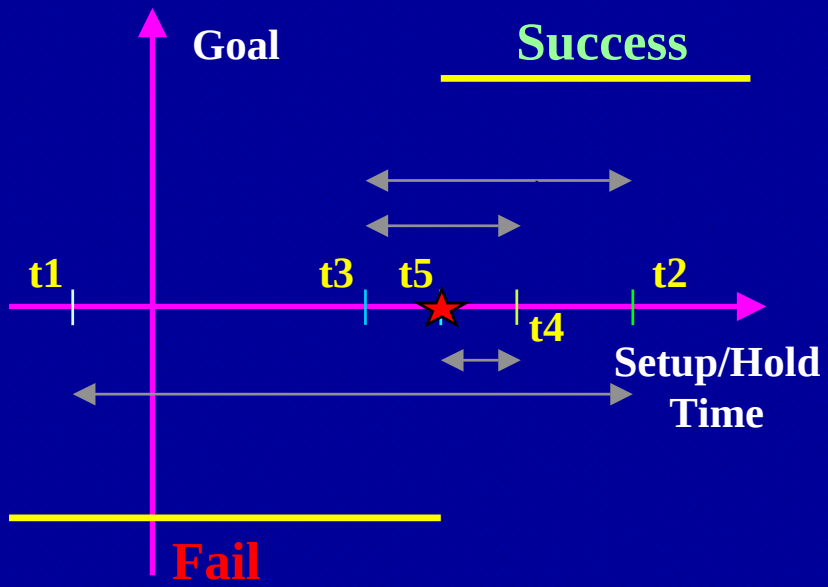




CharFlo-Memory!™ can produce actual instance model!

Bi-Section Optimization

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

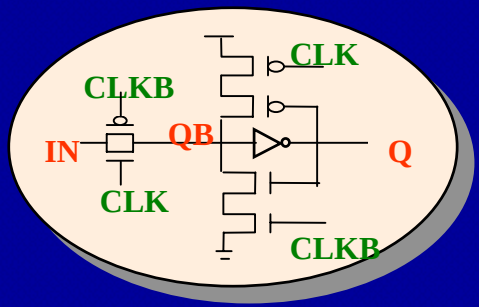


t1 Start
t2 Start
t3 = (t1 + t2) / 2
t4 = (t2 + t3) / 2
t5 = (t3 + t4) / 2

Time	Goal	BiSect Error
t1	Fail	
t2	Success	Norm(t2-t1)
t3	Fail	Norm(t3-t2)
t4	Success	Norm(t4-t3)
t5	Success	Norm(t5-t4)

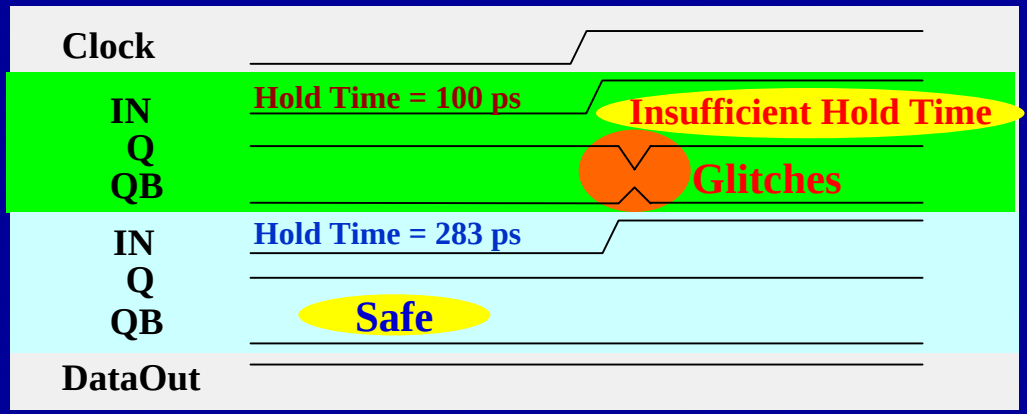
Reliability Checking

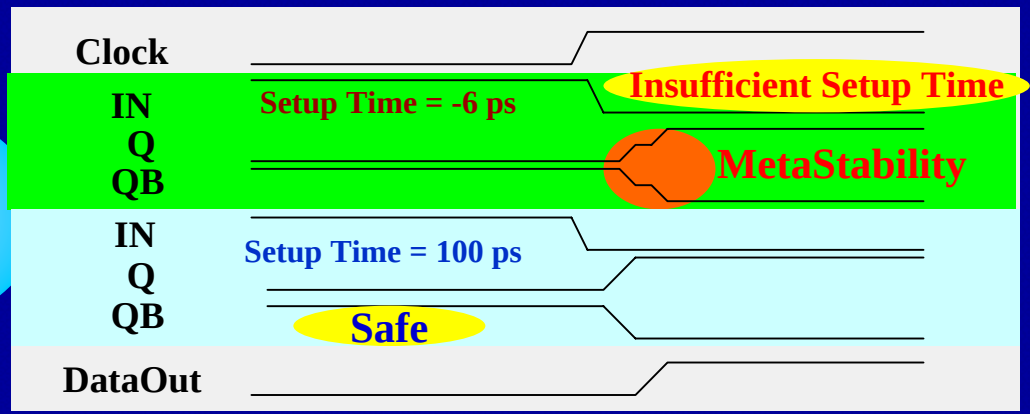
‘Glitch’ and ‘MetaStability’ Prevention



Reliability Problem !

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

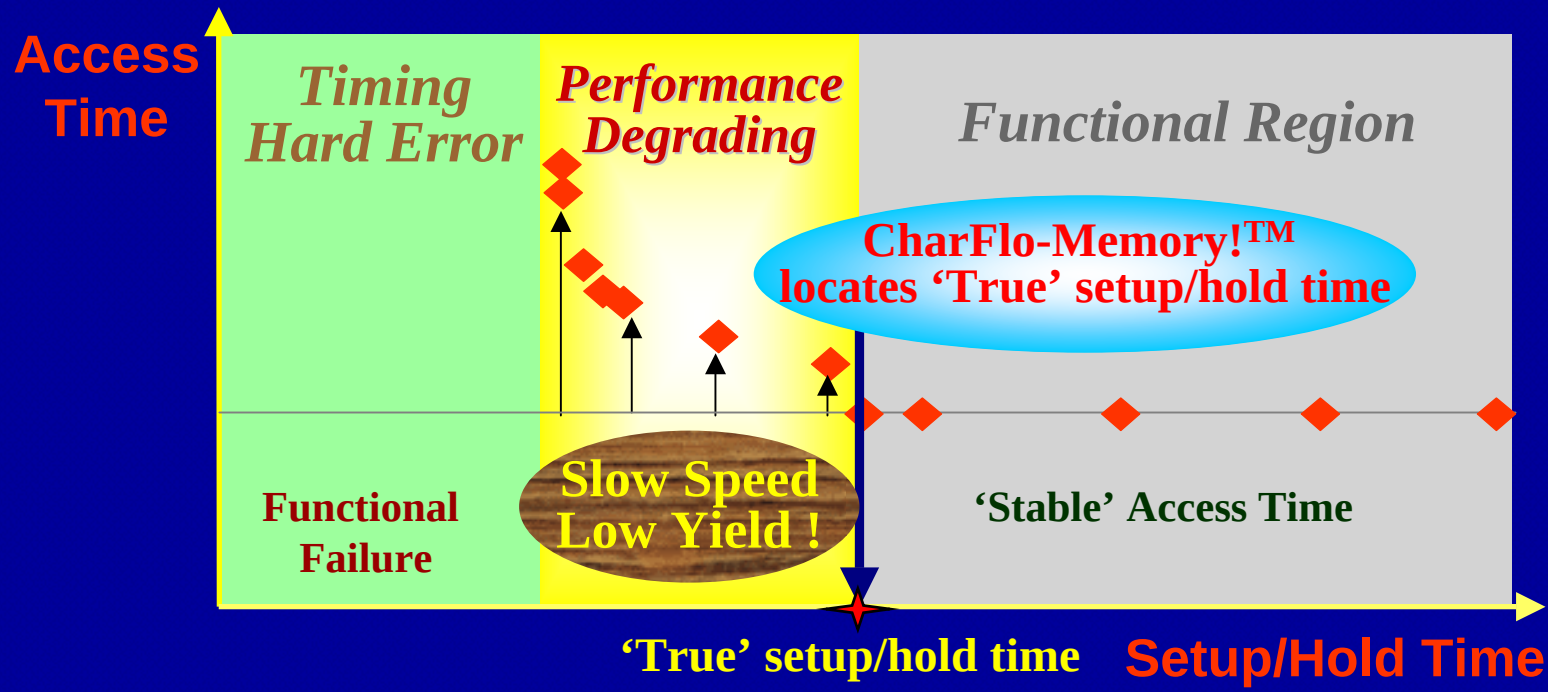




Reliability Checking

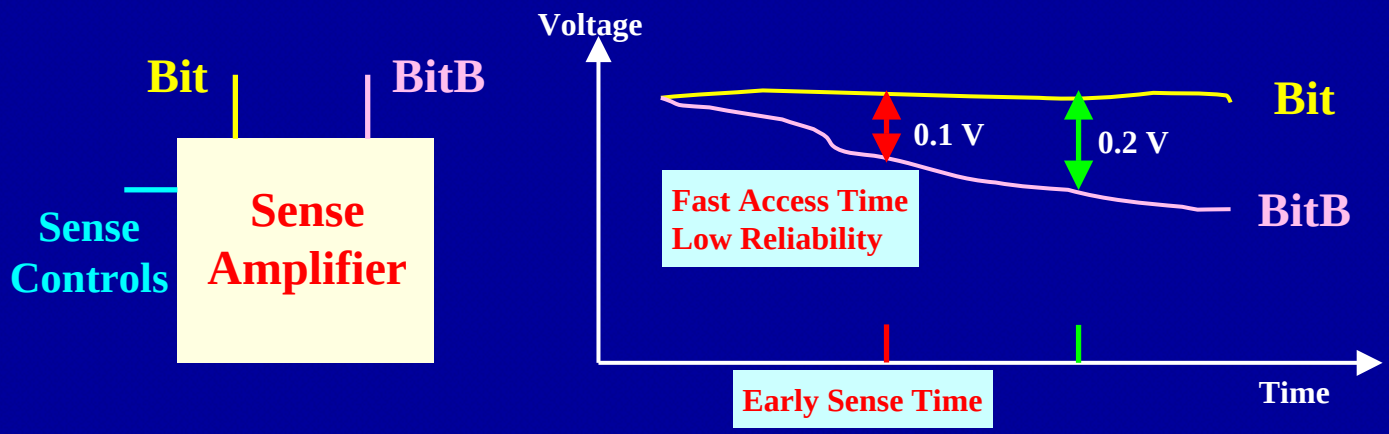
'Performance Degrading' Prevention

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

Any change on voltage supply, .temp, process corners will impact Bit, BitB and Sense Controls.

Determine Read-Margin Setting

Checking for Yields and Speed

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

Read Margin	Access Time	V(Bit-BitB)	*SRAM 8Kx24*
1000	4.3212 ns	365.8 mv	Slow
0100	1.8824 ns	95.9 mv	
1101	2.8109 ns	200.6 mv	
0111	1.8824 ns	95.9 mv	
1111	1.6969 ns	68.3 mv	Danger

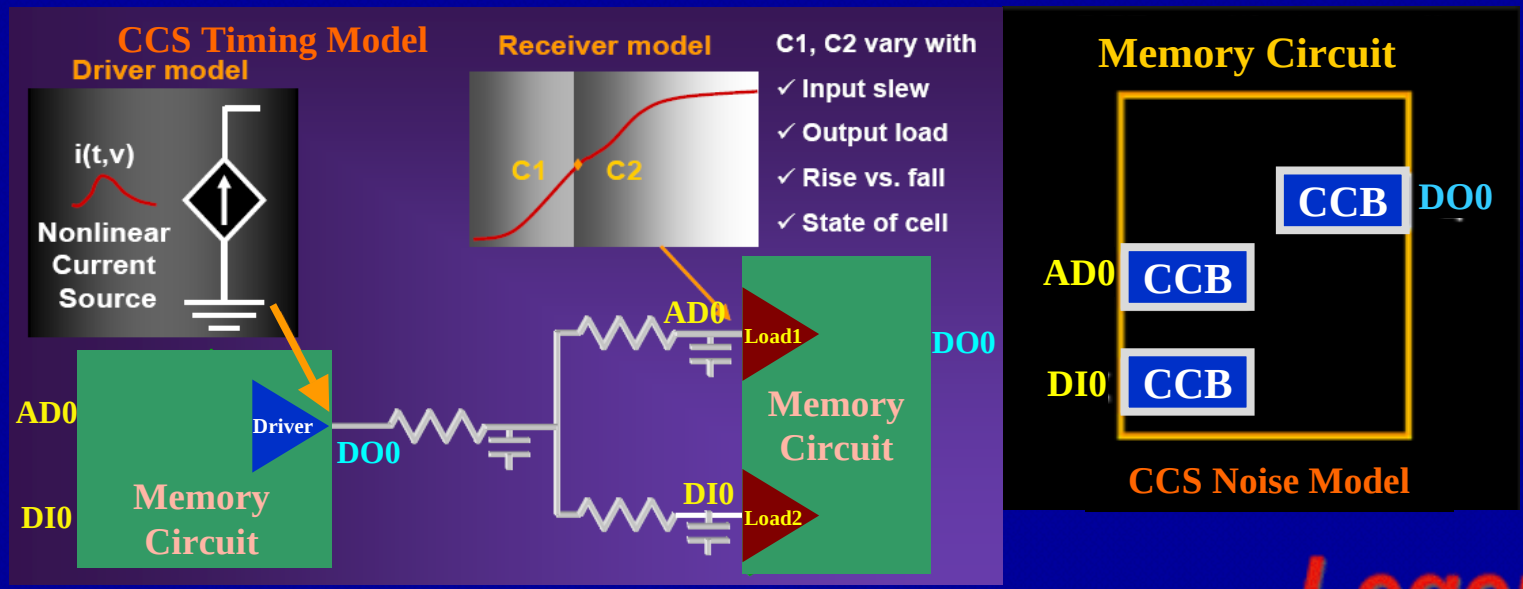
** Noise margin is 100 mv minimum normally , and 200 mv for the safer.*

Wrong Read Margin (RM) setting causes Poor Yield!

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**

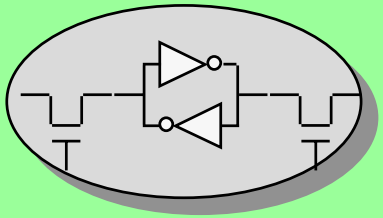
  Extract ‘critical-path’ circuits for

  - 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**

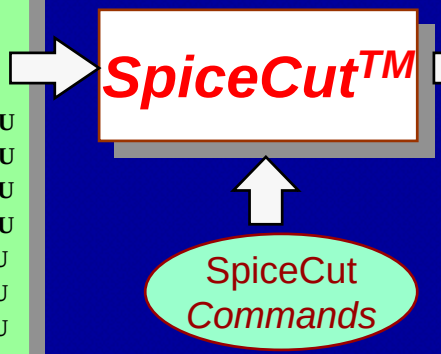
Recognize Memory Structure

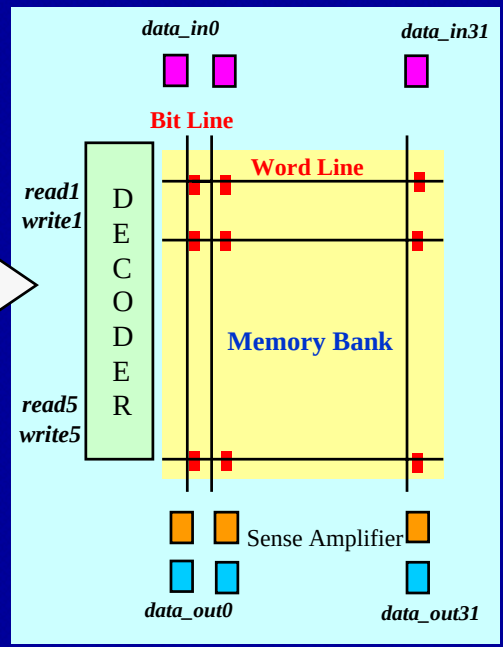
Memory Cell



```
.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
```

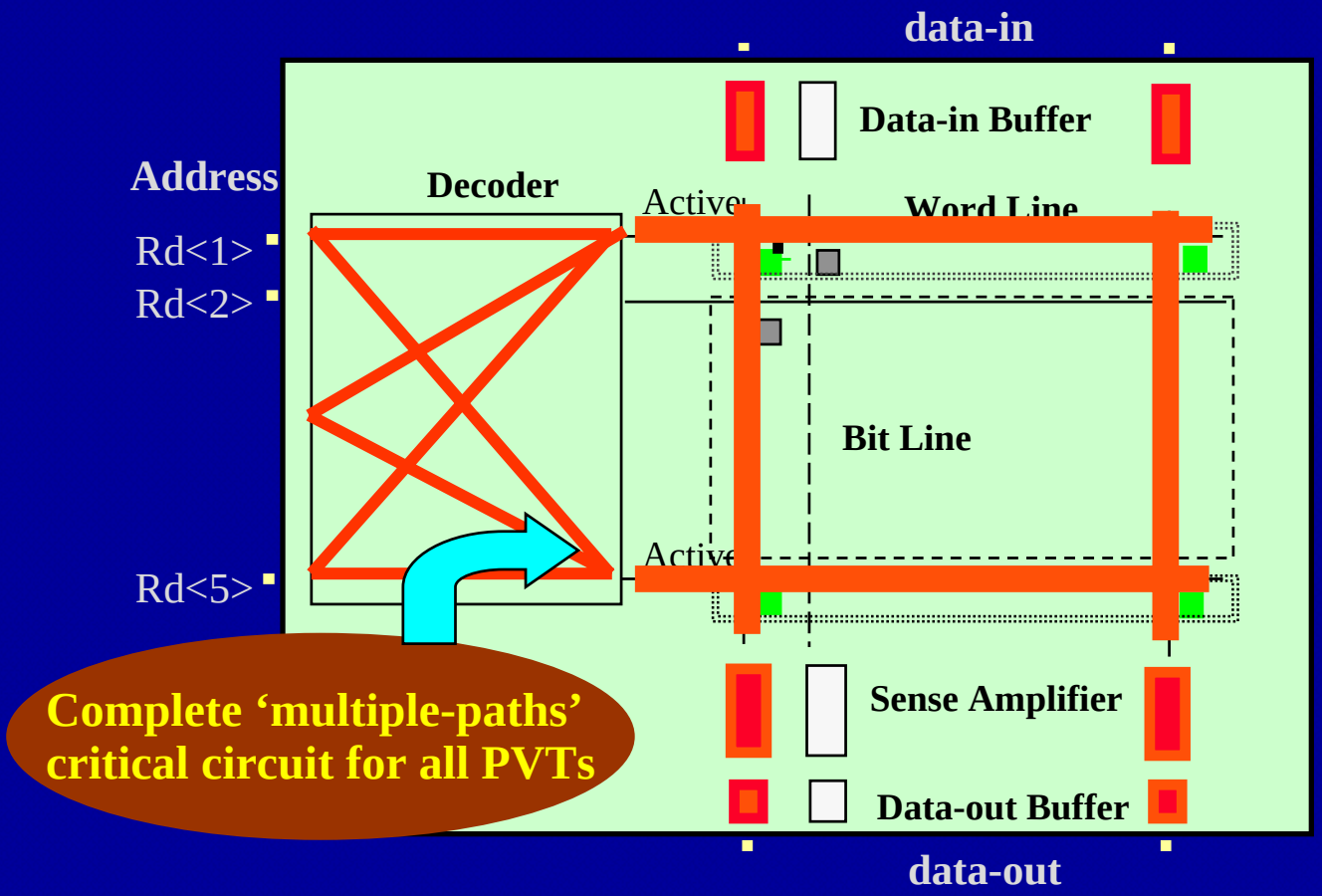
- Pattern matching with core cell
- Locate all **memory banks**, **words** and **bits**





Extract Critical-Path Circuit

Using SpiceCut for Access Time



Extract Critical-Path Circuit

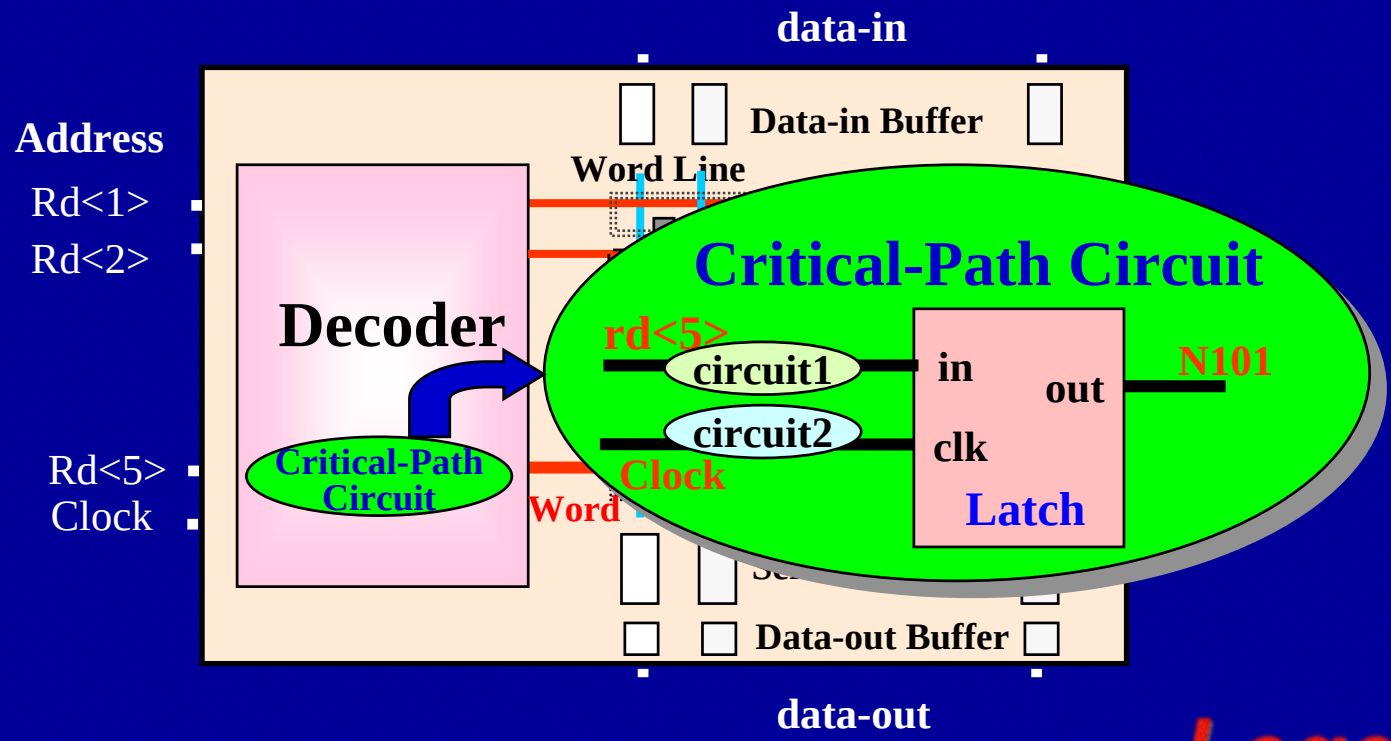
Using SpiceCut for Setup/Hold Time

- 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

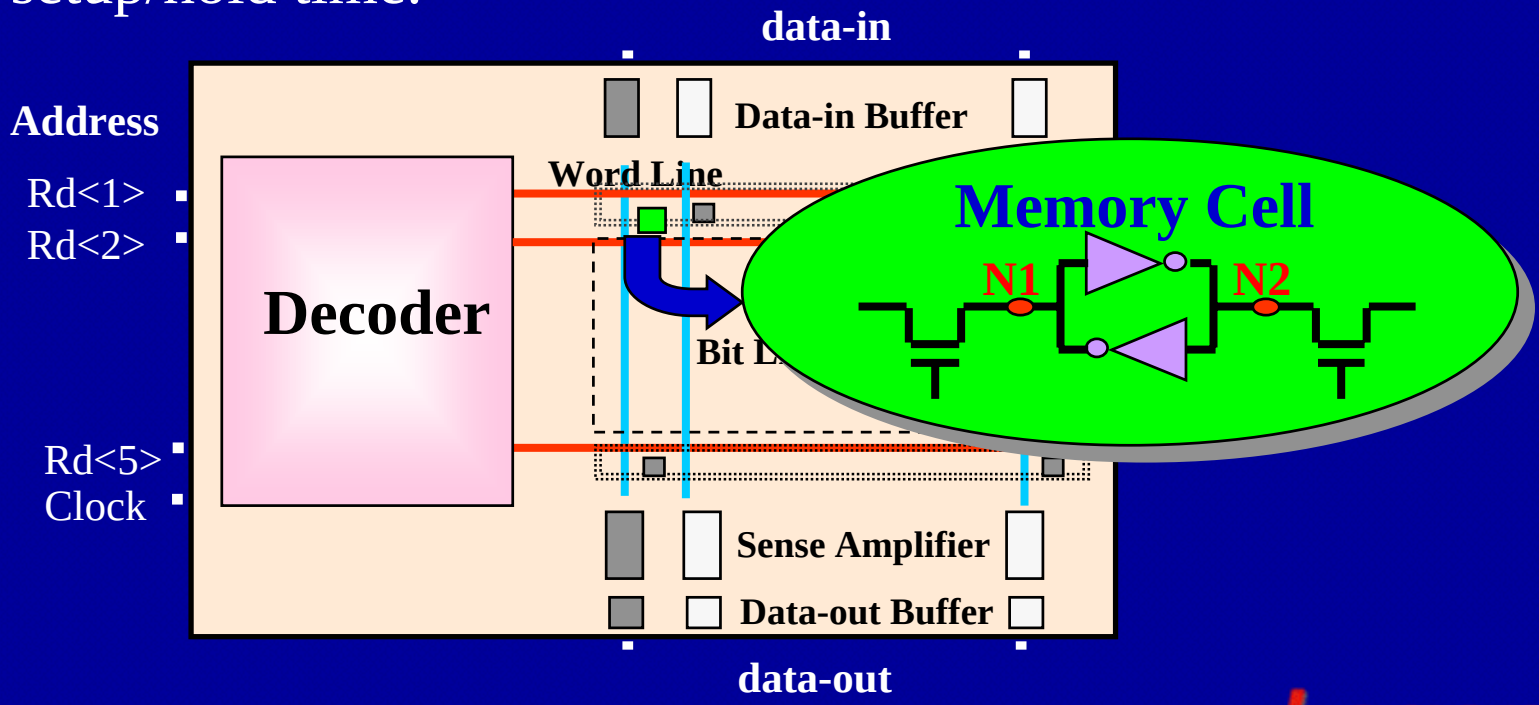
Latch output **N101** can be automatically located, and then critical-path circuits extracted for Address setup/hold time.



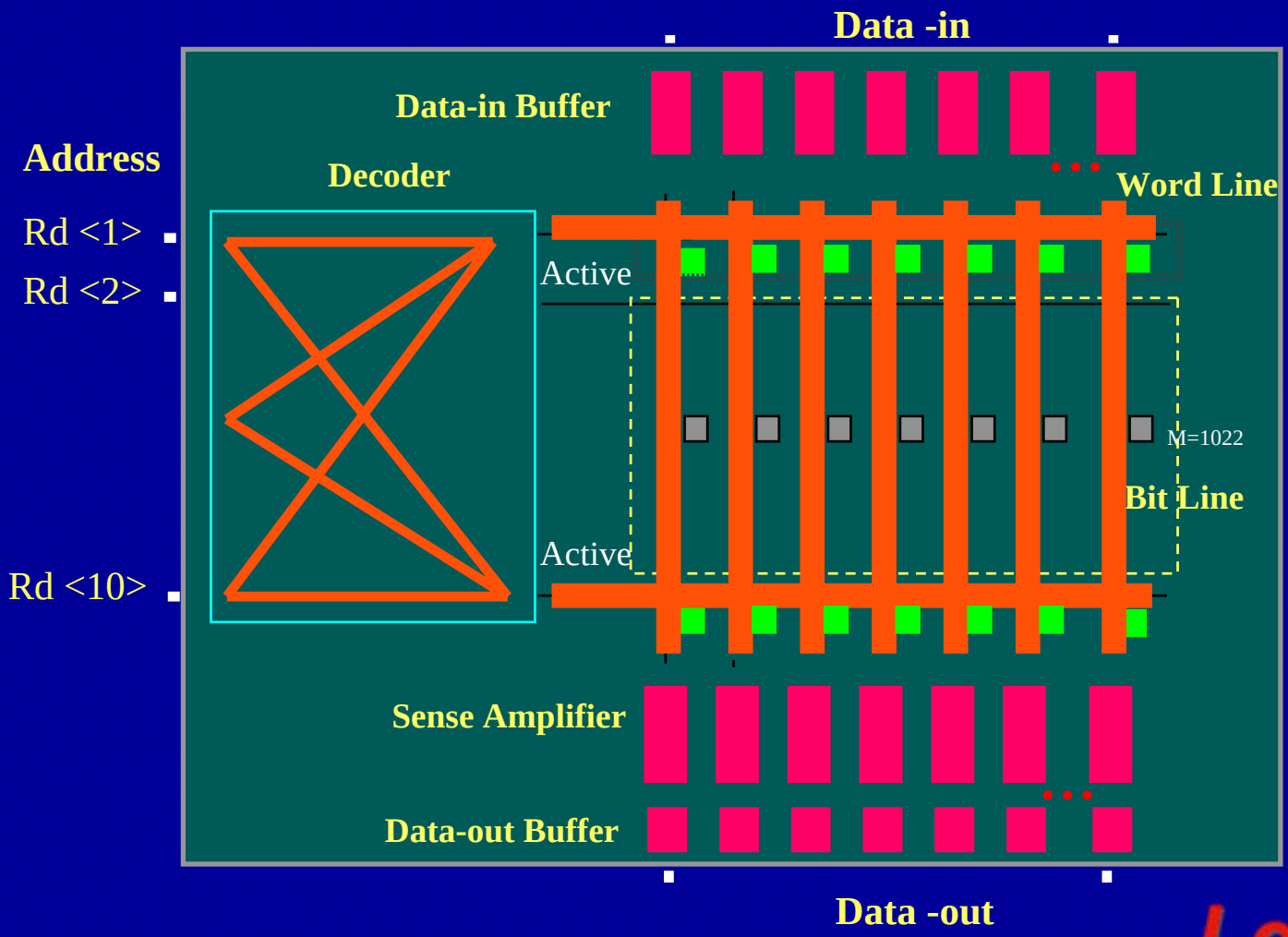
‘Inside Memory-Cell’ Example

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.



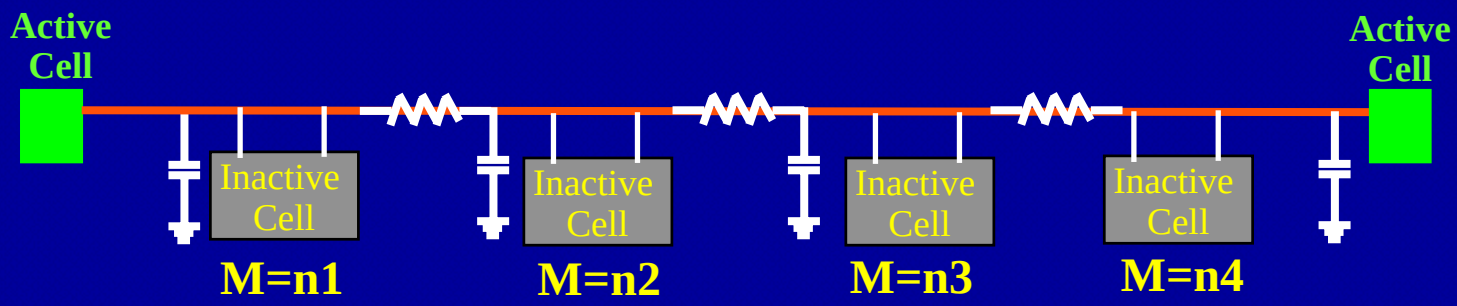
Critical-Path Circuit for Power



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



Performance Optimization

Segmenting MOSFET Loadings

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

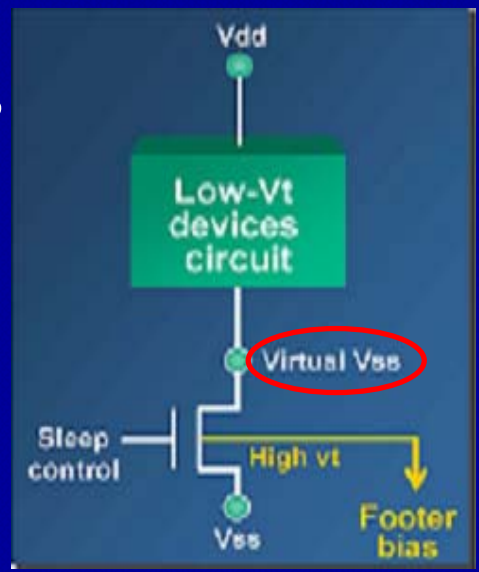
Accuracy Comparison	Before Segment *50,488 MOSFETs*		After Segment *10,292 MOSFETs*		Accuracy Differences	
	Rise	Fall	Rise	Fall	Rise	Fall
Access Time	325.94p	351.53p	324.95p	351.98p	**0.30%**	**0.13%**

Speed and Memory Usage Comparison	Before Segment *50,488 MOSFETs*		After Segment *10,292 MOSFETs*		Improvement	
	CPU Time	Memory Usage	CPU Time	Memory Usage	CPU Time	Memory Usage
Access Time	8,279 sec	3.8 GB	674 sec	1.6GB	**12.28X**	**2.38X**

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 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	Dolphin Technology
Virage	Faraday (UMC Alliance)
TSMC	VeriSilicon
Virtual-Silicon	Synopsys

- The instance models using Legend's tools have been silicon-proven by customers and major foundries.

The Conclusion

Legend's Memory Characterization

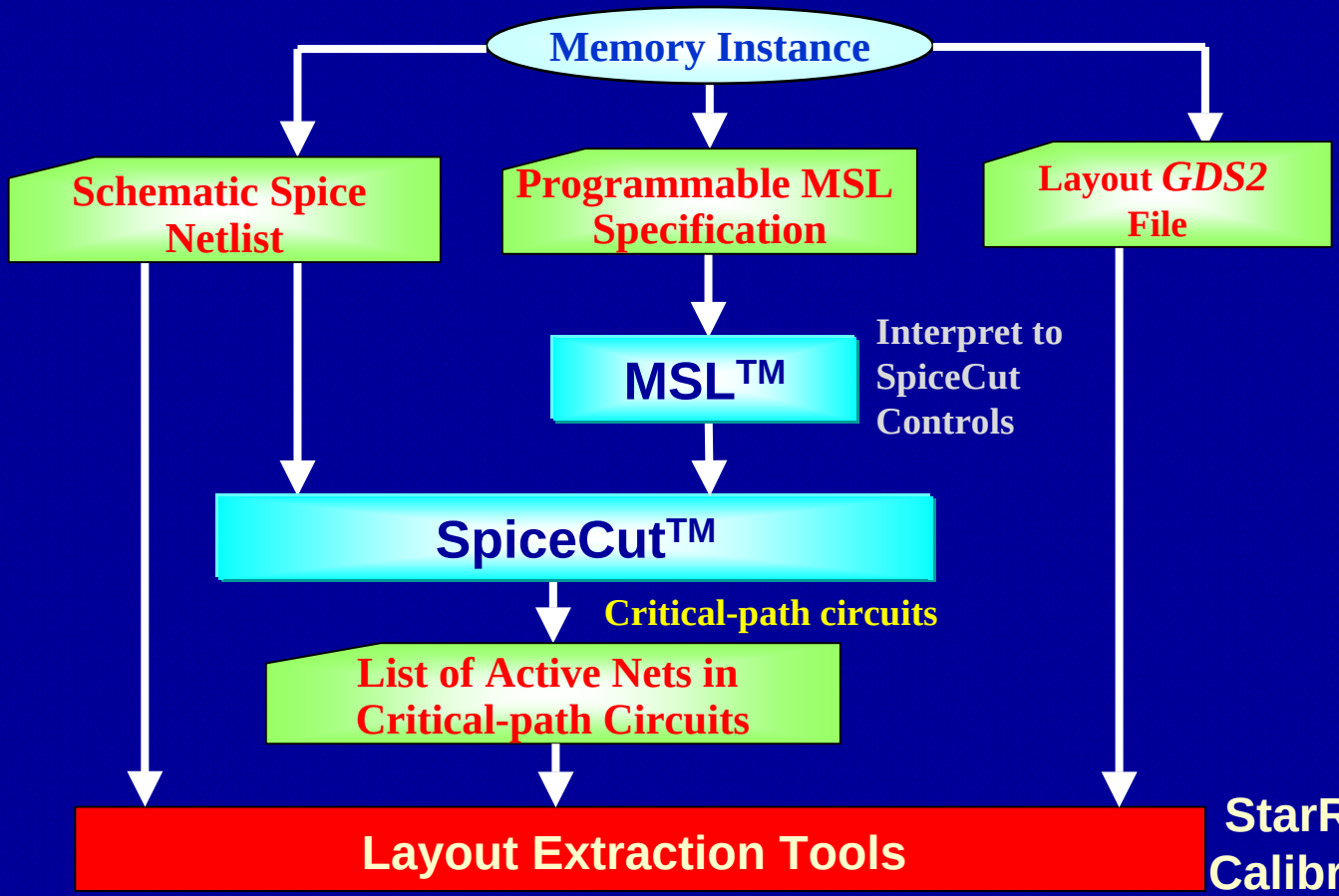
- Production and silicon proven including 45nm/65nm
- Automatically build critical-path circuits and structures from layout-extracted circuit with RC
- Access memory internals and reliability checking
- Optimized critical-path circuits for Power-Gating
- RC reduction and 'active-net' extraction for speed
- 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



Active Nets Flow II

Characterization based on Active Nets