A Mechanized Refinement Framework for Analysis of Custom Memories

Sandip Ray
University of Texas at Austin

Jayanta Bhadra
Freescale Semiconductor Inc.

Presentation for FMCAD 2007

Memory Verification Problem

Specification:

- A state machine that atomically reads and writes data at addressed locations.

Verification:

- Given a network of transistors, prove that it correctly implements such a state machine.

Memory verification is a crucial component in the validation of a microprocessor or SoC.

- Memories account for over 50% of the transistor count of a microprocessor.

Traditional Memory Verification Approach

Abstract the transistor network into a "switch level model" (Bryant,1984; Bryant et al., 1987)

- Represent the network as a set of nodes connected by transistor switches
  - Each node has state 0, 1, or X
  - Each switch has state open, closed, or indeterminate
- State transitions are specified by switch equations
  - Typically constructed by partitioning the network into channel connected subcomponents.

Compare the switch level model with high level specification

Switch level analyzers such as ANAMOS accurately capture many aspects of transistor circuits.

Deficiencies of Switch level Models

Switch level analyzers ignore many analog effects.

Strength assignment in ANAMOS produces significant mismatch with detailed analog simulation if transistors have closely matching but different strengths.

The deficiencies have been addressed by designing more and more sophisticated analyzers (Agarwal, 1990)

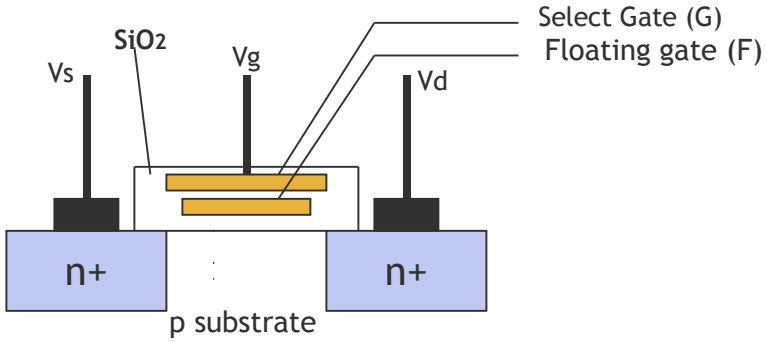
Fundamental Problem:

Crisply approximating analog behaviors with equations in a discrete algebra

The problem is exacerbated by the advent of FLASH memories.

FLASH Memories

FLASH contains both traditional (MOS) and floating gate (FG) transistors.



The capacitive coupling between G, F, and substrate is used to regulate the threshold voltage by controlling the charge stored.

- Low threshold = Logic 1
- High threshold = Logic 0

The capacitive coupling breaks the view of a transistor as an on/off switch as taken by switch-level analyzers.

Our Observation

Custom memories (SRAM and FLASH) are not ad hoc transistor networks.

- Memory cores operate over a limited range of input stimuli: "legal patterns"
- Legal patterns should work and all others assumed to fail.

The behavior of the individual bit cells on each legal pattern sequence is validated by extensive analog simulation across process corners and operating conditions.

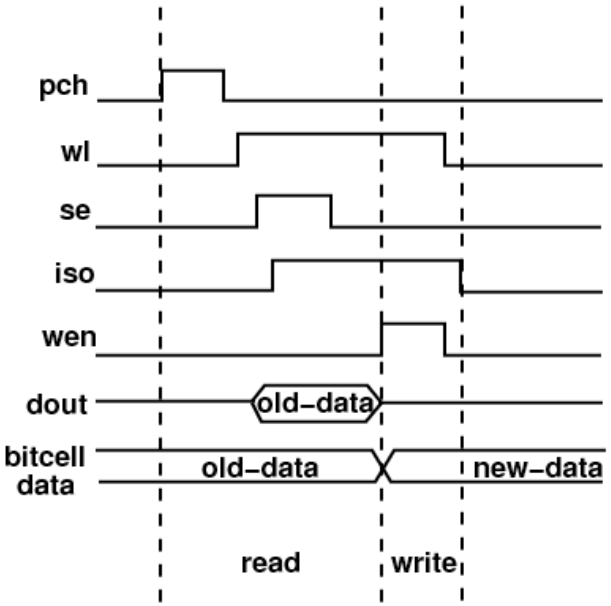
We can naturally model the behavior of a bit cell as a state machine.

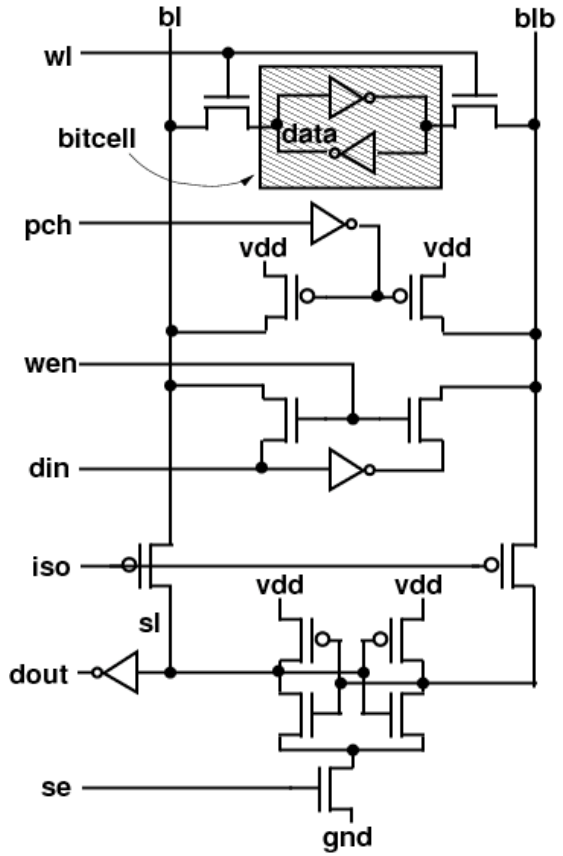
Operating constraints can be modeled by guarded transitions.

The memory array can then be formalized as an interacting composition of state machines.

The approach is agnostic to the type of memory (SRAM or FLASH)

Memory Bit Cells as State Machine





Under the timing constraints for operation the bit cell behavior can be modeled as a state machine.

FLASH operations are a bit more complex but still can be modeled as state machines.

Our Work

A library of formal state machine models for components of embedded memory (bit cells, sense amplifiers, etc.)

- Developed in the logic of the ACL2 theorem prover
- Operating constraints modeled using ACL2's encapsulation feature
- Closely correspond to the models used in SPICE simulations

Modeled as interactive composition of these state machines:

- An SRAM memory array
- A NOR Flash array configuration

Proved that each is a refinement of the corresponding high-level specification (up to finite stuttering)

Complexity of invariant definition can be managed by taking advantage of compositionality.

Concluding Observations

To our knowledge this work is the first platform to formally verify FLASH memories.

Since we focus on modeling behaviors:

We somewhat circumvent the problem of abstracting arbitrary analog operations with switch-level equations

But the approach **cannot** be applied to arbitrary transistor networks.

The work is in its early stages.

More automation necessary both in extracting behavioral models from memory circuits and in the verification.

We plan to extend the approach and apply it on industrial memory implementations.