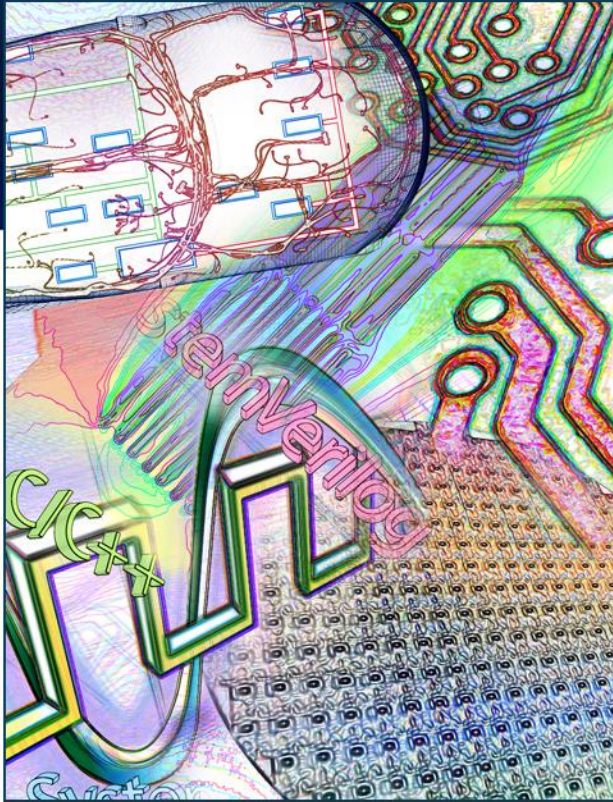


# IBIS Extensions for Turn-around Cycle Simulations



**IBIS Summit, DesignCon,  
February 3, 2017  
Santa Clara, CA**

Arpad Muranyi, Mentor Graphics Corporation  
Randy Wolff, Micron Technology



# Introduction

---

- Using bit time as the unit of measure, the amount of time spent on turning the direction of the bus around is becoming significant at higher bus speeds
- System designers have a strong desire to minimize the amount of wait time between read and write cycles
- Consequently, there is a need to simulate turn around cycles
- There is also a desire to be able to run such simulations with IBIS models to reduce the simulation times and get things done in our life time
- Unfortunately, the current IBIS specification does not support such simulations
- Fortunately, it is very easy to extend the IBIS specification to add support for such simulations

# What is missing in IBIS?

- IBIS assumes that the enable control is not changing during the simulations, i.e. the buffer is either driving or receiving
- IBIS makes the same assumptions for on-die terminations (ODT), they are either on or off during the entire simulation
  - ODT is usually modeled with the [\*\*\* Clamp] I-V curve(s) of the [Model],
  - or the [\*\*\* Clamp] I-V curve(s) of the [Submodel] keyword
- Some simulators provide access for the user to the enable control but be careful about using it
  - the waveforms may not be the same when the buffer transitions between the '1' and '0' states or the 'High-Z' and the '1' or '0' states
  - normal IBIS models only contain the waveforms for transitions between the '1' and '0' states

# How can we solve these shortcomings?

- The solution is surprisingly simple, we just need to generate additional waveforms for the IBIS model
- To describe how the buffer transitions between driving and receiving, we need the following waveforms:
  - '1' to 'High-Z'
  - 'High-Z' to '1'
  - '0' to 'High-Z'
  - 'High-Z' to '0'
- For the ODT, we need to generate similar waveforms to capture how the ODT turns on and off
  - while the buffer is not driving, turn the ODT on or off and record the waveforms

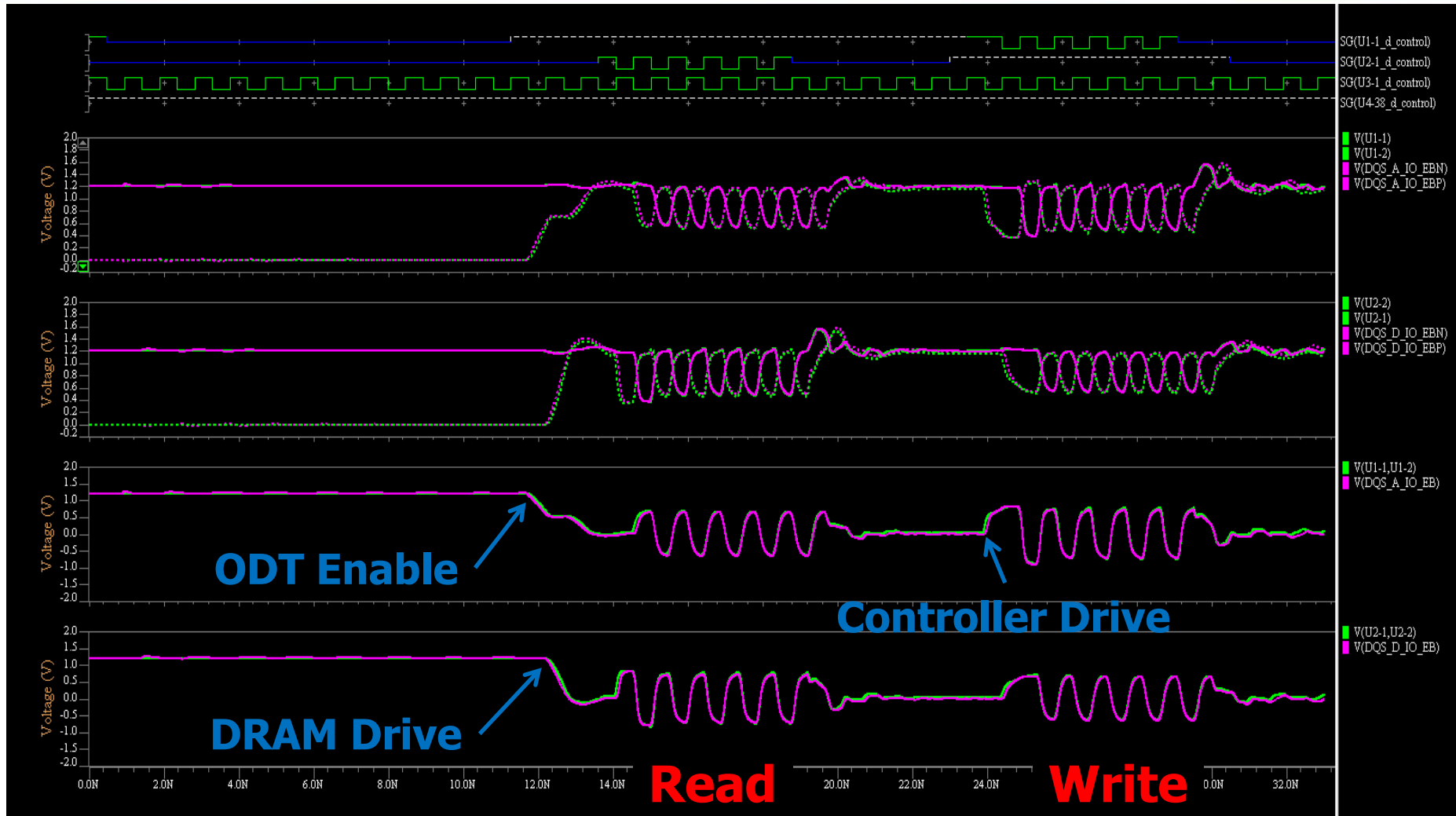
# Need new keywords (or subparameters)

- The simulator needs to know which waveform table(s) to use for which logic transition
- For “turning around” the buffer model
  - We could keep using the existing [Rising Waveform] and [Falling Waveform] keywords with new subparameters, or
  - We could invent new “waveform keywords” to be used for the new waveform tables
    - [1 to Z]
    - [Z to 1]
    - [0 to Z]
    - [Z to 0]

# New keywords / subparameters needed

- For the ODT model
  - The existing Bus\_hold Submodel\_type makes use of I-V and V-t tables the same way as normal [Model]s do
  - This capability would work well to turn ODT on/off smoothly
  - The only catch is that Bus\_hold is triggered by a waveform crossing event on the pad, instead of a digital control signal generated by the EDA tool
  - We could invent a new Submodel\_type which is essentially the same as Bus\_hold, except that it is controlled by a digital stimulus in the EDA tool

# Correlation with SPICE simulation



**SPICE = 77 min.**  
**IBIS = 30 sec.**

# Additional challenges

---

- Some bus transitions involve switching between different levels of ODT strength
- Supporting such simulations would also be desirable
- These simulations would need models which can have multiple ODT models with the ability to switch between them smoothly



---

# **Thank you**

# **Questions?**

**More information can be found in the DesignCon 2017 paper “SI Analysis of DDR Bus during Read/Write operation transitions”**