IBIS Quality Control Through Scripting

Justin Butterfield

DAC IBIS Summit

San Francisco, CA, USA

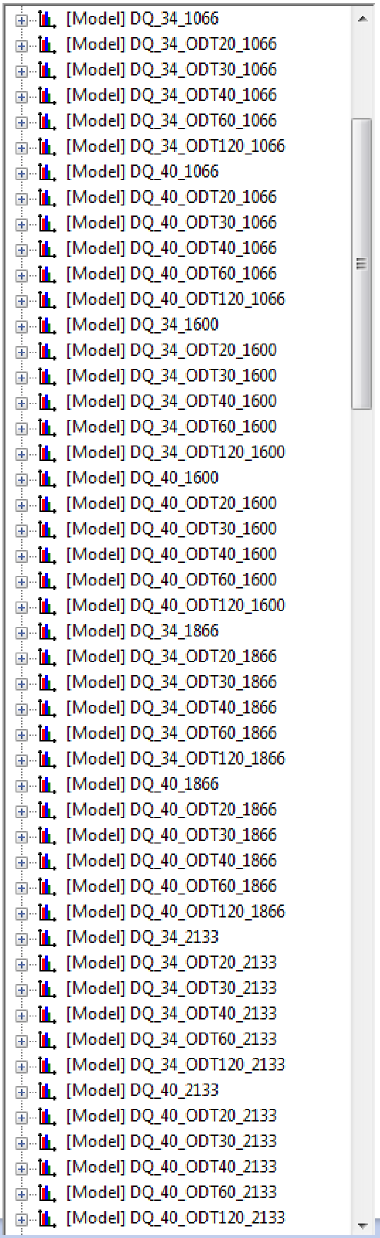
June 5, 2012

Outline

- Motivations
- Scripts to automate IBIS generation
  - Building an IBIS template
  - Importing table data into the template
  - Formatting package RLC data
- Script for checking IBIS parameters
- Scripts for EBDs
  - EBD generation
  - EBD connectivity report

Why Use Scripts for Quality Control

- IBIS files can be very large and complex
- Micron 2Gb DDR3 SDRAM IBIS model statistics:
  - 154 [Model]s, 20 [Submodel]s, 40+ parameters per [Model]
- Large number of [Model]s due to combinations of features
  - Including: Drive strength, ODT, Speed Grade (Vinl/Vinh levels & Overshoot)
  - Most [Model]s only vary a few parameters
- Prone to copy and paste errors and typos
  - Incorrect Vinl/Vinh, Overshoot, and C_comp values
  - Incorrect table data and Submodel mapping
- ibischk does not check accuracy!



IBIS Generation Flow

- Build an IBIS template
- Generate table data (I-V, V-t, Ramp, etc.)
- Import table data into IBIS template
- Format and add package models
- Check IBIS file
- Generate EBD files
- Check EBD files

Build IBIS Template: Concept

- IBIS Template = IBIS file without table data

```
[Model]         INPUT_1600
Model_type      Input
|
Vinl = 575.000mV
Vinh = 925.000mV
|                           typ                min                max
C_comp                     0.600pF            0.500pF            0.700pF
|
[Model Spec]
Vinl                       0.5750V            0.5375V            0.6125V
Vinh                       0.9250V            0.8875V            0.9625V
|
[Voltage Range]            1.5000V            1.4250V            1.5750V
[Temperature Range]       50.0              110.0                0.0
|
|***************************************************************************
|  (table data here)
|***************************************************************************
[Model]         INPUT_1866
...
```

Build IBIS Template: Requirements

- Define base models
  - E.g. DQ, DQS, CLK
  - These can match the [Model Selector] names
- Distinguish between IBIS parameters that are
  - Standard – fixed for the interface specification
    - E.g. [Voltage Range], Vinl/Vinh, Vref/Cref/Rref, Overshoot
  - Variable – vary from device to device
    - E.g. C_comp, [Temperature Range]
- Script builds the template ensuring consistent results

Build IBIS Template: Input Examples

- Standard parameters
  - Read in as a predefined technology file
  - Tables of parameters for combinations of [Models] to be created
- Variable parameters
  - Input text file specified by the user
  - Directs the script to a subset of the technology file

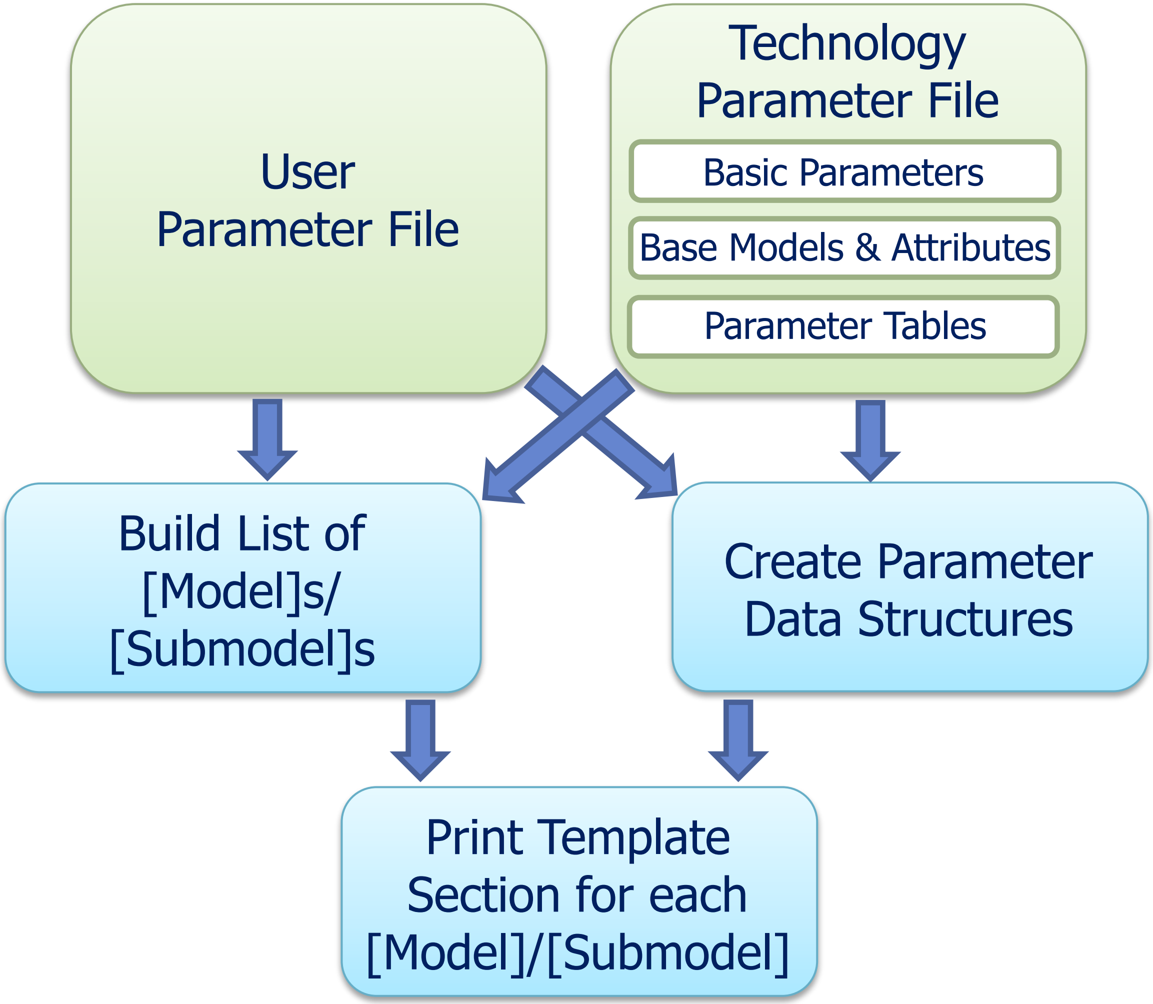
Base_Model	DQ	DQ	DQ	DQ	DQ	DQ
Speed_Grade	800	1066	1333	1600	1866	2133
Vil_ac	-0.175	-0.175	-0.15	-0.15	-0.135	-0.135
Vih_ac	0.175	0.175	0.15	0.15	0.135	0.135
Vil_dc	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1
Vih_dc	0.1	0.1	0.1	0.1	0.1	0.1
D_overshoot_ampl_h	0.4	0.4	0.4	0.4	0.4	0.4
D_overshoot_ampl_l	0.4	0.4	0.4	0.4	0.4	0.4
D_overshoot_area_h	0.25n	0.19n	0.15n	0.13n	0.11n	0.10n
D_overshoot_area_l	0.25n	0.19n	0.15n	0.13n	0.11n	0.10n

Speed_Grades	1066	1600	1866	2133

C_comp	typ	min	max
DQ	1.200	1.125	1.275
DQ_ODT20	1.078	0.966	1.223
DQ_ODT30	1.118	1.019	1.240
DQ_ODT40	1.134	1.044	1.238
DQ_ODT60	1.161	1.074	1.259
DQ_ODT120	1.174	1.097	1.255

Micron

Build IBIS Template: Process Flow



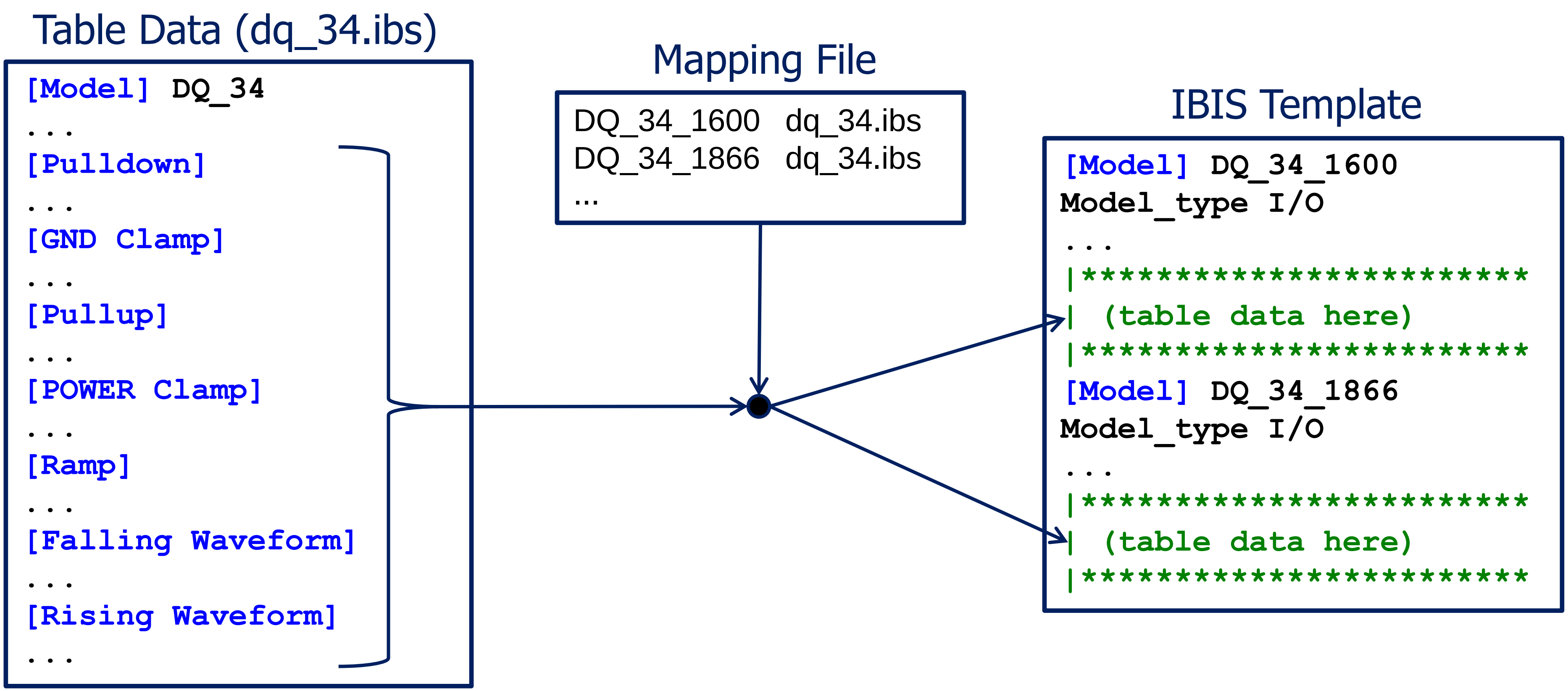
Build IBIS Template: Details

- Build list of [Model]s
  - Loop over base models and features requiring separate [Model]s
  - Preserve the attributes of each [Model]
- Data structures created for each parameter
  - Data{Base_Model}{Speed_Grade}{Parameter_Name}
- Iterate for each [Model] to print the template section
  - Look up parameter values in the data structures
  - Subroutines to print each set of parameters to the template
  - Different cases of print routines for each Model_type

Micron®

Import Table Data

- Read in the table data from IBIS files with a single [Model]
- Use a mapping file to assign each [Model] to its table data
- Script pastes the table data into the IBIS template



Micron

Format Package RLC Data

- Create [Package] and [Pin] list RLC data from a spreadsheet
  - Align columns and calculate typ/min/max [Package] values
- Calculate Z_0 and time delay then check for outliers

[Pin]	signal_name	model_name	R_pin	L_pin	C_pin
B3	DQ0	DQ	0.307	1.337E-09	3.680E-13
C2	DQ2	DQ	0.404	1.710E-09	4.585E-13
C7	DQ1	DQ	0.288	1.258E-09	3.676E-13
C8	DQ3	DQ	0.406	1.598E-09	4.789E-13

Signal	Zo	Time Delay
DQ0	60.28	2.218E-11
DQ2	61.07	2.800E-11
DQ1	58.51	2.151E-11
DQ3	57.76	2.766E-11

```
[Package]
|        typ         min          max
R_pkg    307m        208m         421m
L_pkg    1.64nH      1.059nH      2.514nH
C_pkg    0.35pF      0.261pF      0.479pF
|
[Pin]    signal_name model_name   R_pin    L_pin     C_pin
B3       DQ0         DQ           307m     1.337nH   0.368pF
C2       DQ2         DQ           404m     1.710nH   0.459pF
C7       DQ1         DQ           288m     1.258nH   0.368pF
C8       DQ3         DQ           406m     1.598nH   0.479pF
```

IBIS Parameter Check

- Script to search the IBIS file for a parameter
  - E.g. C_comp, Vinl/Vinh, Vref/Cref/Rref, Overshoot, etc.
- Creates a report detailing the parameter's values in each [Model]
- Results can be filtered by [Model] names

DQ_34_1066	C_comp	1.200pF	1.125pF	1.275pF
DQ_34_ODT20_1066	C_comp	1.078pF	0.966pF	1.223pF
DQ_34_ODT30_1066	C_comp	1.118pF	1.019pF	1.240pF
DQ_34_ODT40_1066	C_comp	1.134pF	1.044pF	1.238pF
DQ_34_ODT60_1066	C_comp	1.161pF	1.074pF	1.259pF
DQ_34_ODT120_1066	C_comp	1.174pF	1.097pF	1.255pF

Micron

 |

Build EBD Model

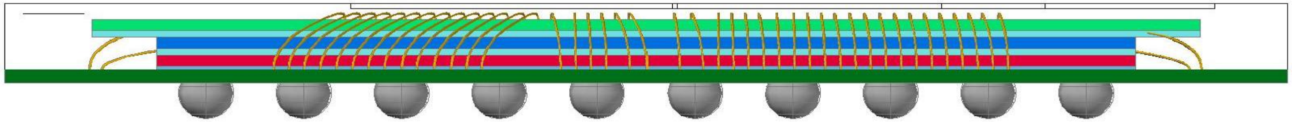
- Generate EBD model from spreadsheet data
  - Useful for fixed topologies such as stacked die packages
  - Input RLC & Len values and Pin & Node connections

Pin	Signal	Sub R	Sub L	Sub C	Ref.Pin	Ref.Pin
J2	CS0#	0.086	1.067E-09	5.879E-13	U1.119	
J3	BA0	0.110	1.533E-09	7.118E-13	U1.117	U2.117

Substrate RLC

Wire Bond RLC

```
[Path Description] CS0#
Pin J2
  Len=1 R=0.086 L=1.067e-09 C=5.879e-13 /
Fork
  Len=0 R=0.054 L=2.944e-10 C=4.220e-14 /
  Node U1.119
Endfork
|
[Path Description] BA0
Pin J3
  Len=1 R=0.110 L=1.533e-09 C=7.118e-13 /
Fork
  Len=0 R=0.054 L=2.950e-10 C=4.230e-14 /
  Node U1.117
Endfork
Fork
  Len=0 R=0.060 L=3.254e-10 C=4.666e-14 /
  Node U2.117
Endfork
```



EBD Connectivity Report

- Report showing EBD Pin to Node
  - Script prints out an entry for each Pin to Node connection
- Helpful for checking:
  - EBD Path names and IBIS Signals match
  - Correct number of node connections exist on each Path

EBD			IBIS			
Path Name	**Pin**	**Node**	**File Name**	**Component**	**Signal**	**Model**
DQ31	B5	U0.6	g69a_bd.ibs	MT42L64M32D1G69A	DQ31	DQ
DQ31	B5	U1.6	g69a_bd.ibs	MT42L64M32D1G69A	DQ31	DQ
DQ29	B6	U0.9	g69a_bd.ibs	MT42L64M32D1G69A	DQ29	DQ
DQ29	B6	U1.9	g69a_bd.ibs	MT42L64M32D1G69A	DQ29	DQ
DQ26	B7	U0.13	g69a_bd.ibs	MT42L64M32D1G69A	DQ26	DQ
DQ26	B7	U1.13	g69a_bd.ibs	MT42L64M32D1G69A	DQ26	DQ
DQ25	C9	U0.15	g69a_bd.ibs	MT42L64M32D1G69A	DQ25	DQ
DQ25	C9	U1.15	g69a_bd.ibs	MT42L64M32D1G69A	DQ25	DQ
DQ30	D7	U0.7	g69a_bd.ibs	MT42L64M32D1G69A	DQ30	DQ
DQ30	D7	U1.7	g69a_bd.ibs	MT42L64M32D1G69A	DQ30	DQ

Conclusions

- Cons
  - Must invest time to write and validate scripts
  - Risk of propagating errors
  - Possibility of incorrect usage by inexperienced users
- Pros
  - Improved model consistency and quality
  - Efficient checks on large models
  - Save time creating and validating models

Micron®

Focused on Memory | Engineered for Innovation