life.augmented

AN4803

Application note

High-speed SI simulations using IBIS and board-level simulations using HyperLynx® SI on STM32 MCUs and MPUs

Introduction

The purpose of this application note is to guide the end-user on the IBIS (I/O buffer information specification) models of STMicroelectronics STM32 32-bit Arm® Cortex® MCUs and MPUs. It also gives guidance on the use of external peripherals to perform board-level simulations with the HyperLynx® SI (signal integrity) software, to address SI issues.

As a concrete example, this application uses STM32F7 Series and STM32MP15 Series devices. All of the information and conclusions can be extrapolated to any STM32 32-bit Arm® Cortex® MCU or MPU.

As the signal speed increases, SI and EMC (electromagnetic compliance) become issues. These issues are detectable by test equipment, as a signal degradation: overshoot, undershoot, ringing, crosstalk, or timing delay.

One possible cause of signal degradation is a poor board design that fails CE/FCC certification. Another potential cause is timing violations between the IC (integrated circuit) drivers and the receiver. Emphasis must be put on getting the design right the first time to avoid costly redesign, and multiple successive layouts and prototypes. Therefore, performing an SI simulation is very important before designing a prototype.

1 General information

This application note information and conclusions can be extrapolated to all the STM32 32-bit Arm® Cortex® MCUs and MPUs.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Terms and acronyms

This section defines the terms and acronyms used in this document.

Table 1. Acronyms used in this document

Term	Definition
DDR	Double data rate. Define an interface where data are sampled on both clock edges.
DDR3L	DDR memory 3rd generation using low voltage.
EMC	Electromagnetic compatibility, refers to the ability of an electrical device to work satisfactorily in its electromagnetic environment without adversely influencing the surrounding devices, or being influenced by them.
FR4	It is an abbreviation for flame resistant 4, is a type of material used for making a printed circuit board (PCB). It describes the board itself with no copper covering.
IBIS	Input/output buffer information specification is a behavioral-modeling specification. It is a standard for describing the analog behavior of the buffers of a digital device using plain ASCII text formatted data.
IO	Input and output words.
LPDDR2	Low-power DDR memory 2nd generation.
LPDDR3	Low-power DDR memory 3rd generation.
ODT	On-die termination.
PCB	Printed circuit board.
QUADSPI	It is is a specialized communication interface targeting single, dual or quad SPI flash memories.
SI	Signal integrity, denotes the correct timing and quality of the signal.
SDRAM	Synchronous dynamic random access memory.

3 SI fundamentals and STM32 signals

3.1 Signal integrity fundamentals

When the board traces carry signals that contain high frequencies, a particular attention must be given to the design traces that match the impedance of the driver and the receiver devices.

The longer the trace, or the greater the frequencies involved, this results in a greater requirement to control the trace impedance. The PCB (printed circuit board) manufacturer controls the impedance by varying the dimensions and the spacing of a particular trace or laminate. Any impedance mismatch can be extremely difficult to analyze once a PCB is loaded with any components.

3.1.1 Signal integrity

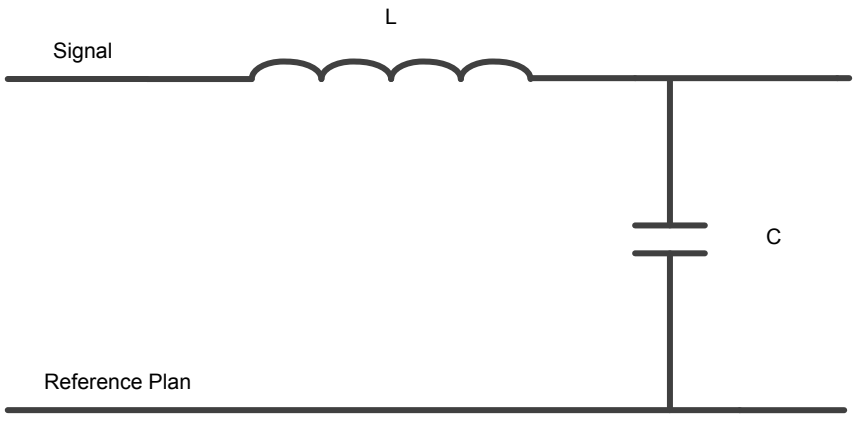
Signal integrity refers to the process of understanding and controlling behaviors of an ideal digital signal. It is a critical element for any new digital PCB design as the clock speeds have increased by more than hundreds of megahertz. At high speeds, some issues such as, signal and clock distortions, rise and fall time-edge distortion, setup timing violations and propagation delay times may appear.

3.1.2 Transmission line

A transmission line is defined as the conductive connection between a driver and a receiver. At low frequencies a wire or a PCB trace is considered to be an ideal circuit (resistive), but at high frequencies, AC (alternated current) circuit characteristics are dominated with inductances and capacitances.

3.1.3 Transmission line model

Figure 1. Transmission line at high frequency



The signals on a transmission line travel at a speed that depends on the surrounding medium. The propagation delay is the inverse of propagation velocity.

$$V = \frac{C}{\sqrt{\epsilon_r}} \quad (1)$$

$$TD = \frac{\sqrt{\epsilon_r}}{C} \quad (2)$$

- v: propagation velocity, in meters/second
- c: speed of light in a vacuum (3 × 108 m/s)
- εr: dielectric constant
- TD: time delay for a signal to propagate down a transmission line of length x
- The propagation delay can also be determined from the equivalent circuit model of the transmission line:

$$TD = \sqrt{LC} \quad (3)$$

Where:

- TD: is the time delay for a signal to propagate down a transmission line of length x
- L: is the total series inductance for the length of the line
- C: is the total shunt capacitance for the length of the line.

The propagation delay is about 3.5 ps/mm in air where the dielectric constant is 1.0. In FR-4 PCBs, the propagation delay is about 7 to 7.5 ps/mm and the dielectric constant is 3.9 to 4.5.

The PCB traces act as transmission lines when the line delay is equal to or greater than 1/6 the rise (or fall) time. The critical length equals 1/6 of the transition electrical length, and the transition electrical length equals to the rise (or fall) times x1/(propagation delay).

Example 1 : For a 2 nanosecond rise time the critical length is 47.6 mm.

3.1.4 Characteristic impedance

The characteristic impedance (Zo) of the transmission line is defined by:

$$Z_o = \sqrt{\frac{L}{C}} \quad (4)$$

Where:

- L: is in henries per unit length
- C: is in farads per unit length.

At very high frequency or with very lossy lines, the resistive loss become significant.

3.2 IBIS model

The IBIS is a behavioral model that describes the electrical characteristics of the digital inputs and outputs of a device through V/I (voltage versus current) and V/T (voltage versus time) data without disclosing any proprietary information.

The IBIS models are used for signal integrity analysis on system boards. These models allow system designers to simulate and therefore foresee fundamental signal integrity concerns in the transmission line that connects different devices.

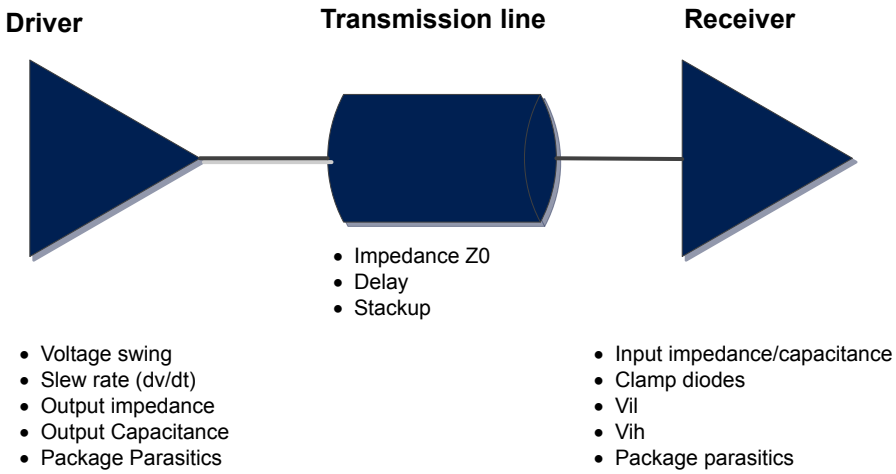
The potential problems that can be analyzed by means of the simulations include among others:

- The degree of energy reflected back to the driver from the wave that reaches the receiver due to mismatched impedance in the line
- Crosstalk
- Ground and power bounce
- Overshoot or undershoot
- Line termination analysis

3.2.1 IC modeling

The figure below shows an example of two ICs modeling:

Figure 2. Transmission iine with iC modeling



3.2.2 Basic structure of an IBIS file

- Header
  - File name, date, version, source, notes, copyright, etc.
- Component model data
  - Default package data (L_pkg, R_pkg, C_pkg)
  - Complete pin list (pin name, signal name, buffer name, and optional L_pin, R_pin, C_pin)
  - Differential pin pairs, on-die terminators, buffer selector, etc.
- IO model data
  - All buffer models for the component must be defined in the file
  - Each flavor of a programmable buffer is separate model

As shown in Figure 3 and Figure 4, the HyperLynx visual IBIS editor is used to open the STM32F746 and the SDRAM (MT48LC4M32B2B5-6A) and to view their characteristics such as the rising and the falling waveforms.

Figure 3. IBIS editor

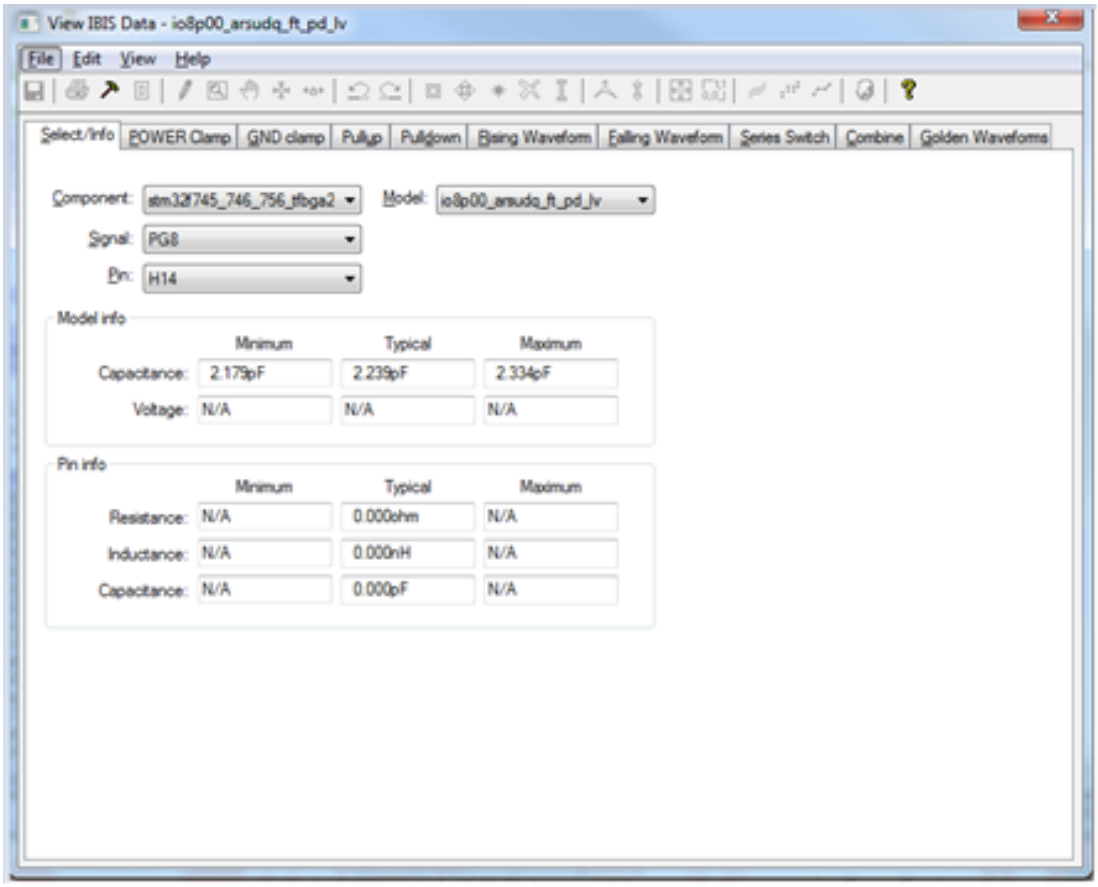
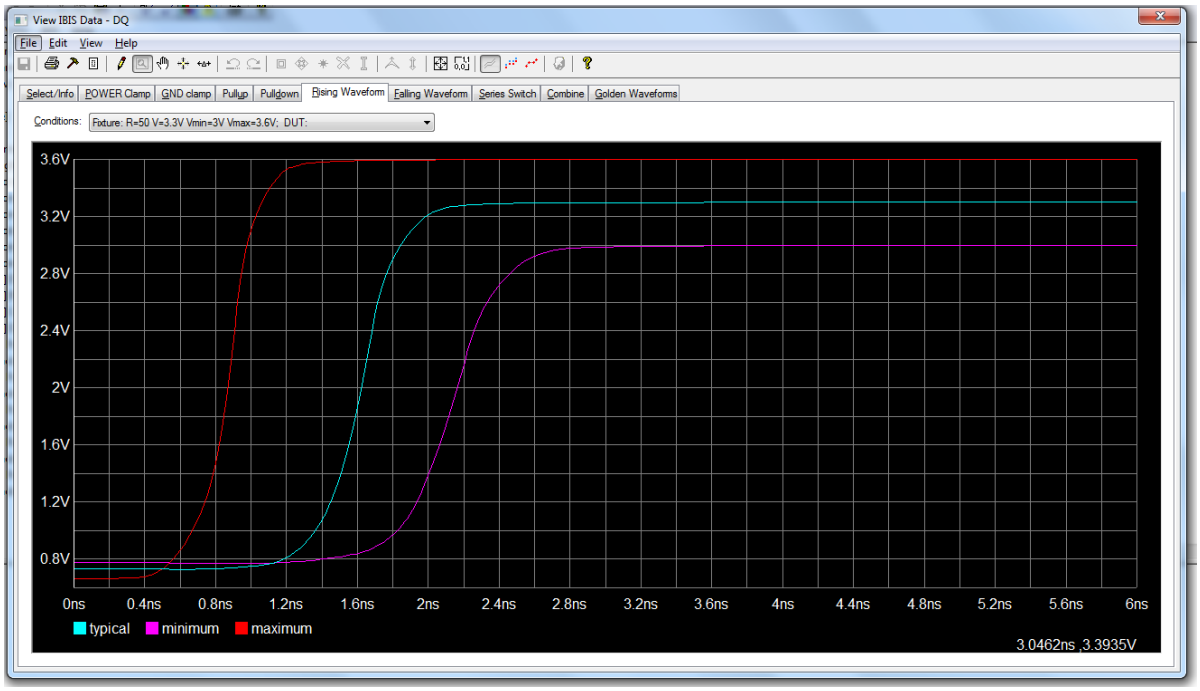


Figure 4. IBIS data



4 STM32 MCUs and MPUs IBIS model selection/selector

This section presents the IBIS model selector of the available GPIO (general-purpose input/output) pin in STM32 MCUs and MPUs.

4.1 GPIO structure

The GPIO includes below features:

- Output driver
- Input buffer
- Pull-up and pull-down
- Electrostatic discharge (ESD) protection
- Input hysteresis
- Level shifter
- Control logic.

4.2 Model selector

The GPIO pins can be selected following below parameters depending on the application requirements:

- Two operating voltage ranges:
  - V33 (3.3 V): refer to 2.7 V to 3.6 V external voltage range VDDx
  - V18 (1.8 V or lv): refer to 1.7 (see *Note* below) to 2.7 V external voltage range VDDx
- Four or less output buffer speed control depending on the required frequency:
  - 00 (low speed)
  - 01 (medium speed)
  - 10 (fast speed)
  - 11 (high speed)
- Controllable internal pull-up and pull-down resistor (enabled/disabled): PD/PU
- Specific IO pins are used to cover special functions: USB and I2C. The same IO is also available as GPIO pin.

Note: For more details, refer to the specific STM32 device datasheet on the section I/O port characteristics and also to the corresponding STM32 reference manual on the section General purpose I/O (GPIO) for software configuration and selection.

4.3 Example of model selector on STM32F746xx MCU

The example below keeps the same selected IO/Pin as in Section 3.2.2 . The pin is H14 port PG8. This pin belongs to the family "io8p_arsudq_ft" of IO buffer.

In Table 2, the pin H14 with the selected GPIO configurations is highlighted in different colors as per the table footnote legend.

Table 2. I/Os in/output buffer for "io8p_arsudq_ft" selector

IO model name selection (io8p_ar3wsudq_ft)	I/O parameters		
	Voltage range	Buffer speed	Pull up/Pull down
io8p00(1)_ar3wsudq_ft_pd(3)_lv(2) "SPEED00 1P8V, PD=40kOhm"	1.7 V to 2.7 V(2)	Low speed(1)	Pull down 40 K(3)
io8p00_ar3wsudq_ft_pu_lv "SPEED00 1P8V, PU=40kOhm"			Pull up 40 K
io8p00_ar3wsudq_ft_lv "SPEED00 1P8V"			Disabled
io8p01_ar3wsudq_ft_pd_lv "SPEED01 1P8V, PD=40kOhm"	1.7 V to 2.7 V	Medium speed	Pull down 40 K
io8p01_ar3wsudq_ft_pu_lv "SPEED01 1P8V, PU=40kOhm"			Pull up 40 K
io8p01_ar3wsudq_ft_lv "SPEED01 1P8V"			Disabled
io8p10_ar3wsudq_ft_pd_lv "SPEED10 1P8V, PD=40kOhm"	1.7 V to 2.7 V	Fast speed	Pull down 40 K
io8p10_ar3wsudq_ft_pu_lv "SPEED10 1P8V, PU=40kOhm"			Pull up 40 K
io8p10_ar3wsudq_ft_lv "SPEED10 1P8V"			Disabled
io8p11_ar3wsudq_ft_pd_lv "SPEED11 1P8V, PD=40kOhm"	1.7 V to 2.7 V	High speed	Pull down 40 K
io8p11_ar3wsudq_ft_pu_lv "SPEED11 1P8V, PU=40kOhm"			Pull up 40 K
io8p11_ar3wsudq_ft_lv "SPEED11 1P8V"			Disabled
io8p00_ar3wsudq_ft_pd "SPEED00, PD=40kOhm"	2.7 V to 3.6 V	Low speed	Pull down 40 K
io8p00_ar3wsudq_ft_pu "SPEED00, PU=40kOhm"			Pull up 40 K
io8p00_ar3wsudq_ft "SPEED00"			Disabled
io8p01_ar3wsudq_ft_pd "SPEED01, PD=40kOhm"	2.7 V to 3.6 V	Medium speed	Pull down 40 K
io8p01_ar3wsudq_ft_pu "SPEED01, PU=40kOhm"			Pull up 40 K
io8p01_ar3wsudq_ft "SPEED01"			Disabled
io8p10_ar3wsudq_ft_pd "SPEED10, PD=40kOhm"	2.7 V to 3.6 V	Fast speed	Pull down 40 K
io8p10_ar3wsudq_ft_pu "SPEED10, PU=40kOhm"			Pull up 40 K
io8p10_ar3wsudq_ft "SPEED10"			Disabled
io8p11_ar3wsudq_ft_pd "SPEED11, PD=40kOhm"	2.7 V to 3.6 V	High speed	Pull down 40 K
io8p11_ar3wsudq_ft_pu "SPEED11, PU=40kOhm"			Pull up 40 K
io8p11_ar3wsudq_ft "SPEED11"			Disabled

1. *Onthis IBIS model name selection, the fragment "XX" on "io8p**XX**_ar3wsudq_ft_pu_lv" defines the speed. 00 = low speed, 01 = medium speed, 10 = fast speed, 11 = high speed. Refer to column "Buffer speed" in this table.*
2. *Onthis IBIS model name selection, the fragment "ZZ" on "io8p00_ar3wsudq_ft_pd_**ZZ**" defines the voltage range. The value "lv" = 1.7 to 2.7 V, [empty] = 2.7 to 3.6 V.*
3. *Onthis IBIS model name selection, the fragment "YY" on "io8p00_ar3wsudq_ft_**YY**_lv" defines the pull-up or pull-down. The value "pu" = pull down, "pd" = pull down, [empty] = disabled.*

5 Application example with HyperLynx simulator

5.1 HyperLynx simulation with SDRAM

This design example shows how to perform a simulation with HyperLynx on MCU Discovery board with STM32F746 (32F746GDISCOVERY). The SDRAM data bus are the critical signals on this PCB board to be analyzed.

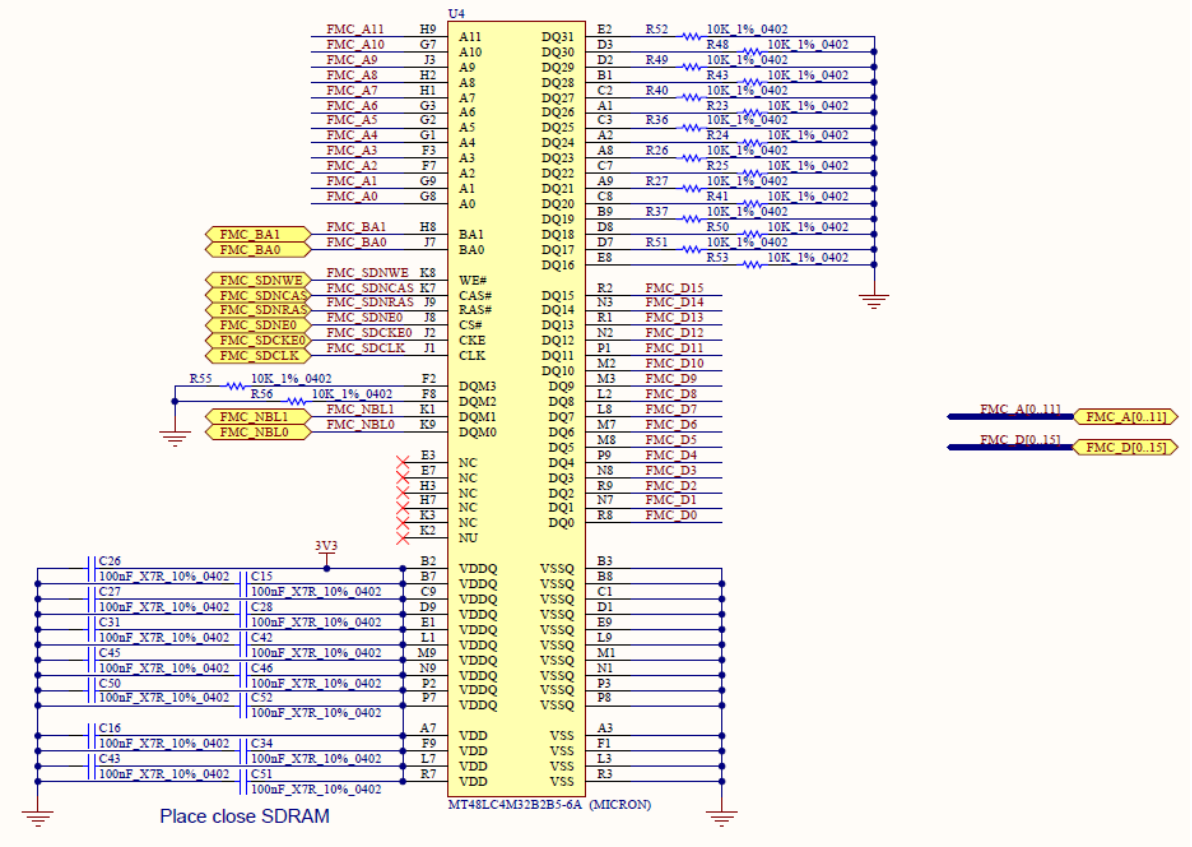
5.1.1 SDRAM signals

The FMC controller, and in particular the SDRAM memory controller, has many signals, most of them have similar functionalities and work together. The controller I/O signals can be split in four groups as follows:

- Address group: consists of row and column address and bank address
- Command group: includes the row address strobe (NRAS), the column address strobe (NCAS), and write enable (SDWE)
- Control group: includes chip select bank1 and bank2 (SDNE0/1), clock enable bank1 and bank2 (SDCKE0/1), and output byte mask for write access (DQM)
- Data group/lane contains x8/x16/x32 signals and the data mask (DQM)

In this Discovery board the memory used is an SDRAM with x16 bus widths and have two data group lanes from Micron (part number: MT48LC4M32B2B5-6A) as shown in the figure below.

Figure 5. SDRAM schematic



5.1.2 SDRAM simulation

The following sequence describes the steps for design and simulation

.

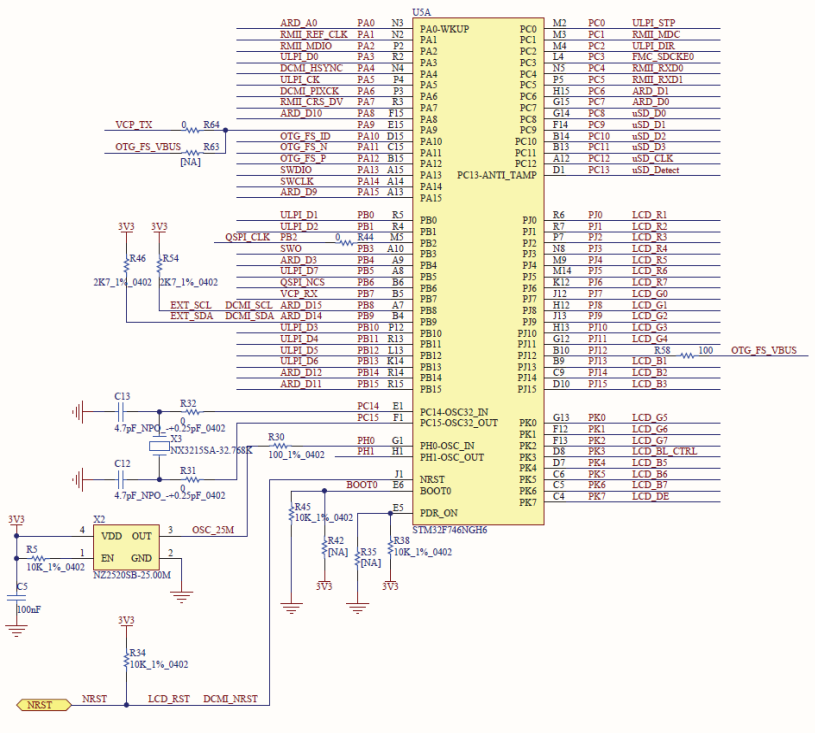
Step 1. Schematics design

The schematic shown in Figure 5 is the connected signal between the SDRAM and the STM32F746 (FMC_xx).

Step 2. PCB design

Use the gerber viewer Gerbv 2.6.1 to see the PCB design. Figure 6 shows the PCB design of the CPU board with STM32F746 and SDRAM chips are placed close to each other, where SDRAM is on the left side.

Figure 6. 32F746GDISCOVERY schematic



Step 3. Translate PCB board file to simulation file

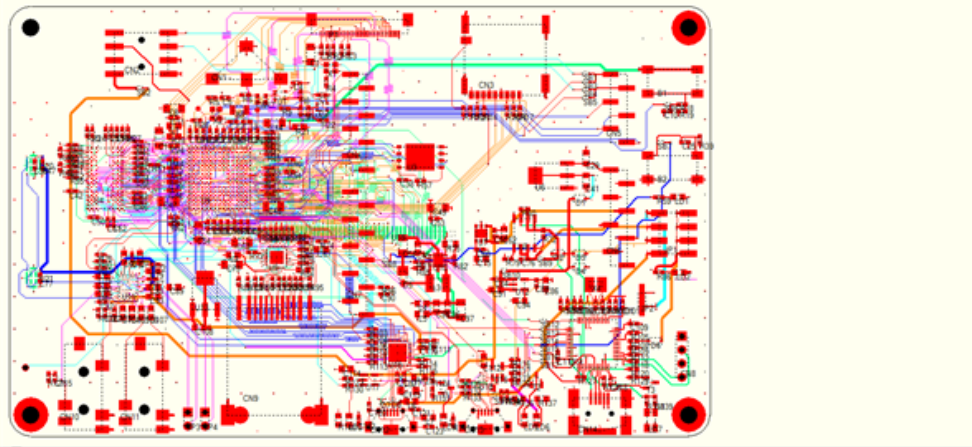
Using the HyperLynx simulation tool from Mentor Graphics® to do PCB board simulation. Run HyperLynx and open the MB1191B-V14.paf file, and then translate it to MB1191B- V14.hpy file for simulation as shown in Figure 7.

Note: This Discovery board layout was designed with Zuken CADStar, so to do simulation with HyperLynx, use PAF file with the same file name.

Step 4. Select the signal to simulate.

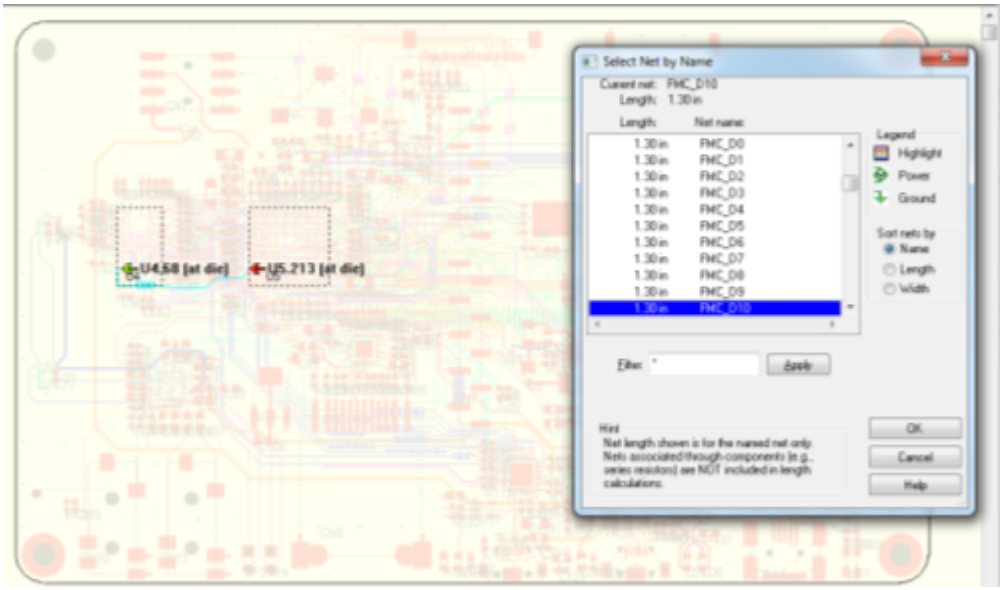
Step 4a. Open the MB1191B-V14.hpy file.

Figure 7. 32F746GDISCOVERY PCB



Step 4b. Select the signal to simulate (for example SDRAM FMC_D10). Go to Select in upper menu and choose Net by Name for SI Analysis (see Figure 8).

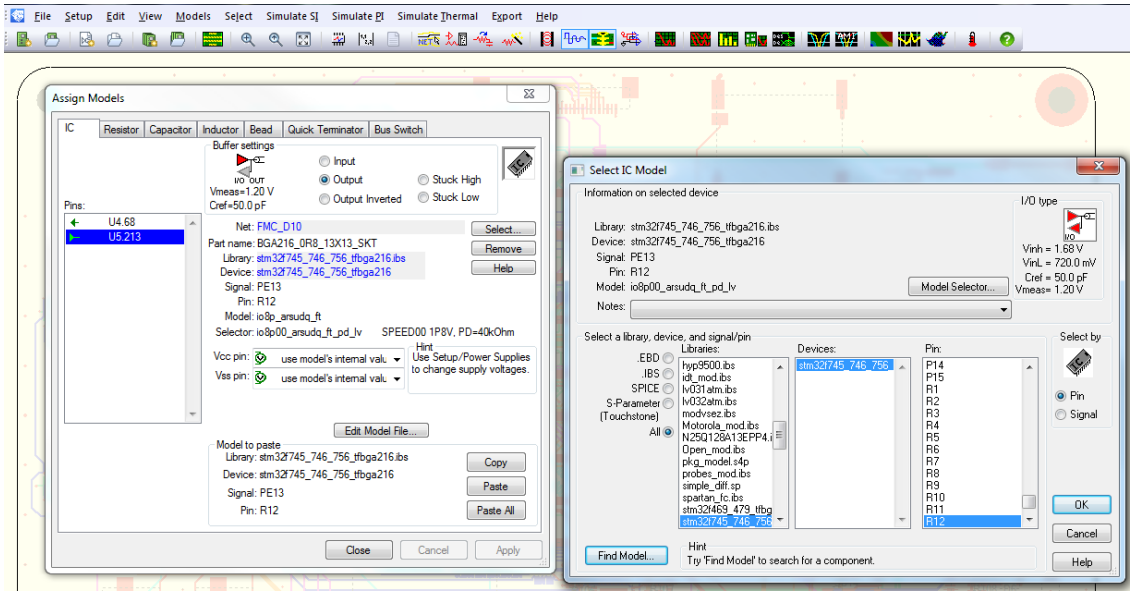
Figure 8. Signal selection



Step 5. Assign IBIS model for STM32F746 and MT48LC4M32B2B5.

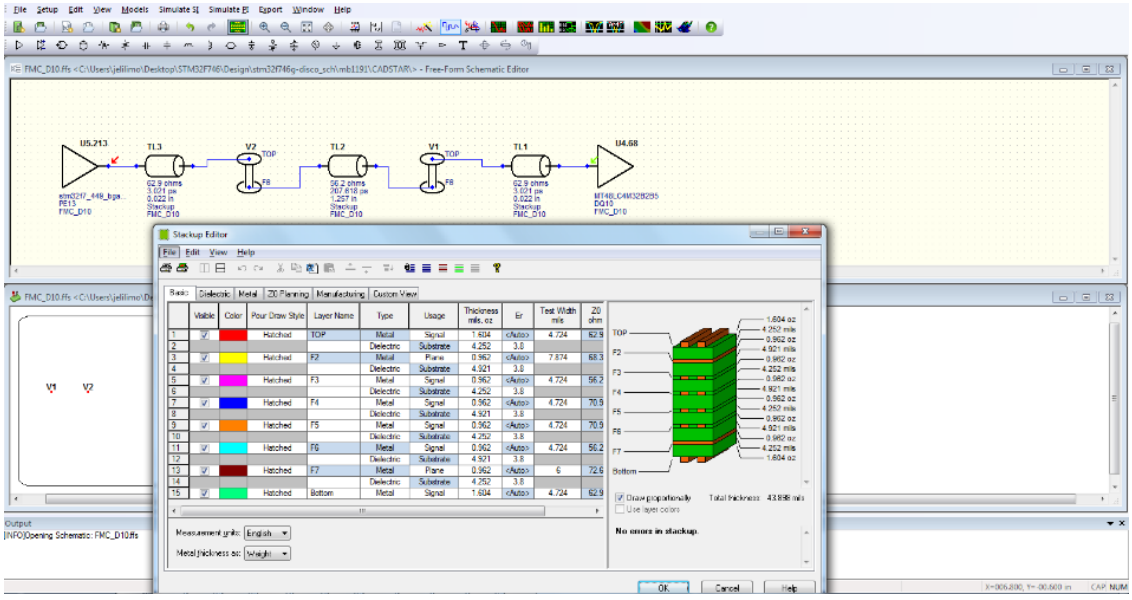
The IBIS model is usually available on the manufacturer's website. The IBIS Model file associated with STM32F746 can be downloaded from the STMicroelectronics web site at www.st.com and for MT48LC4M32B2B5 can be downloaded from Micron website. After downloading the model for each IC and add it to the HyperLynx lib path. Assign the IBIS model for each signal vs IC Figure 7.

Figure 9. Assign IBIS model



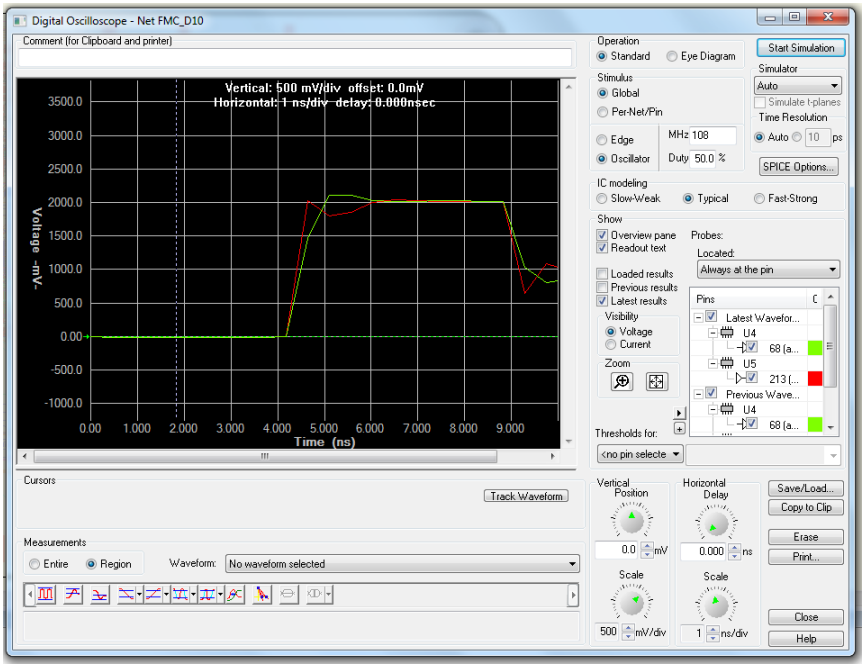
Step 6. Export the selected signal to the free-form schematic and configure the stack- up information.

Figure 10. Free-form schematic



Step 7. Configure and start the simulation.
Set the frequency to 108 MHz and the Duty to 50% (see Figure 11).

Figure 11. Waveform with I/O speed of 0x00



Step 8. Compare and analyze the results by changing IO speed selection for STM32F746 (in red FMC_SDRAM coming out of STM32F7 and in green waveform at SDRAM input).

In the previous steps, the IO speed was set to 0x00. The data signal in red coming out of the STM32F746 is already distorted: square shape with reduced swing and straight slope due to IO speed limitation. The maximum I/O frequency with this setting is 8 MHz and rise time of 100 ns. This can be explained by output signal transitions under the loading conditions Cref and Rref for IO buffer model at lower speed 0x00.

In order to improve the shape of the waveform at the output of STM32F476, the IO speed must be changed to handle more signal frequency content to 0x10 (IO maximum frequency of 100 MHz) and 0x11 (IO maximum frequency of 180 MHz) (see Figure 12).

Figure 12. Waveform with I/O speed of 0x10

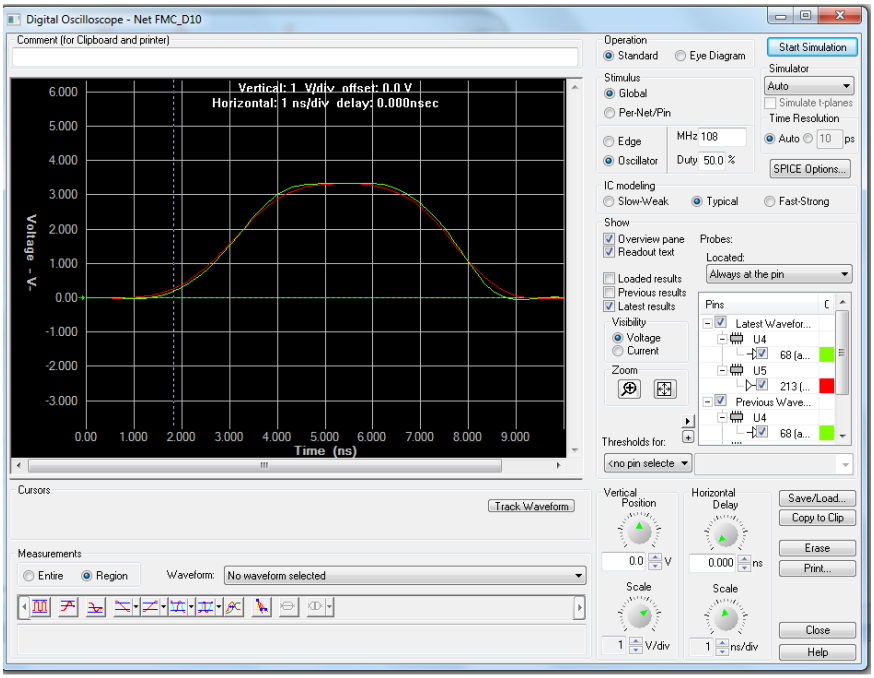
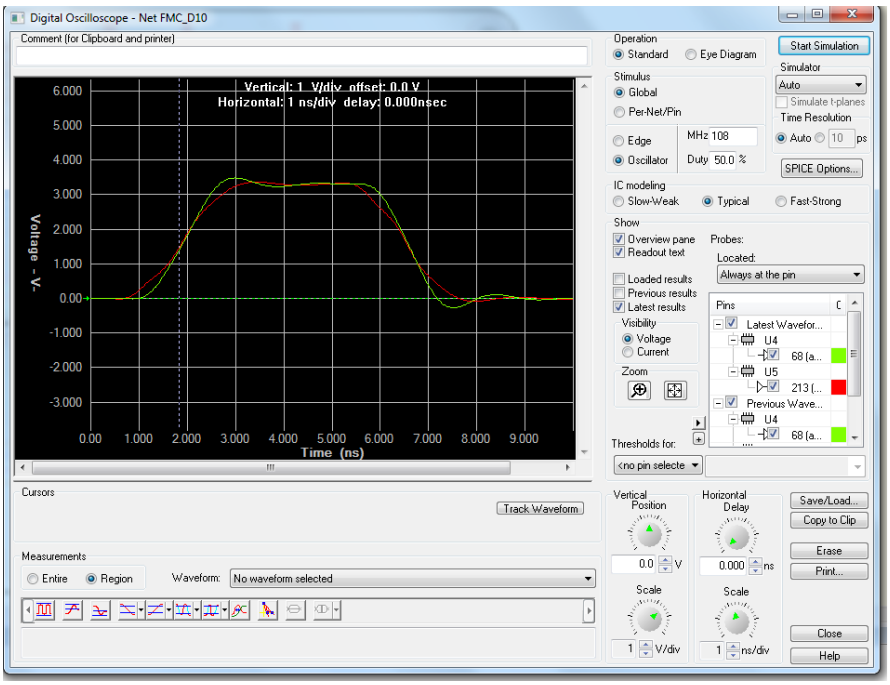


Figure 13. **Waveform with I/O speed of 0x11**



Use the right configuration of IOs speed to match frequency content of target signal for a good SI without any distortion.

5.2 HyperLynx simulation with QUADSPI

5.2.1 QUADSPI signals

Figure 14. **QUADSPI schematic NOR memory interface**

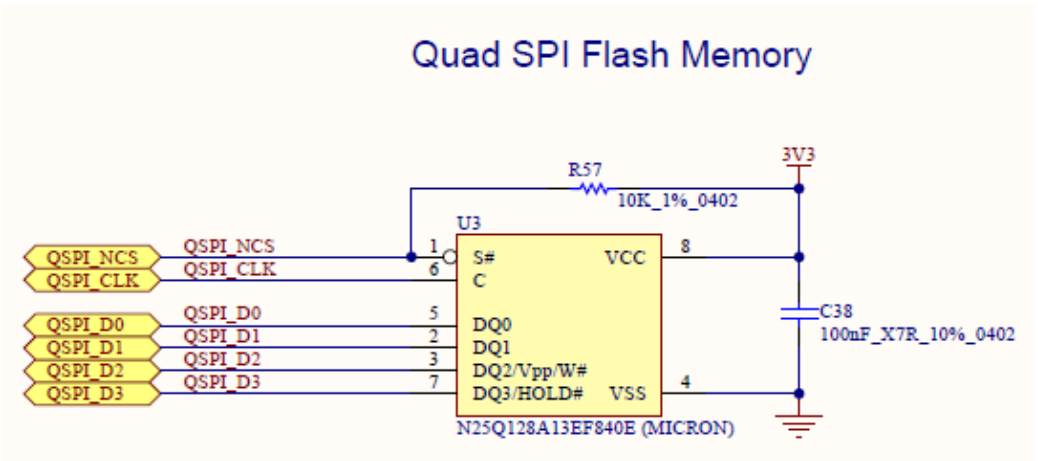


Figure 15. **QUADSPI schematic STM32 interface (part 1)**

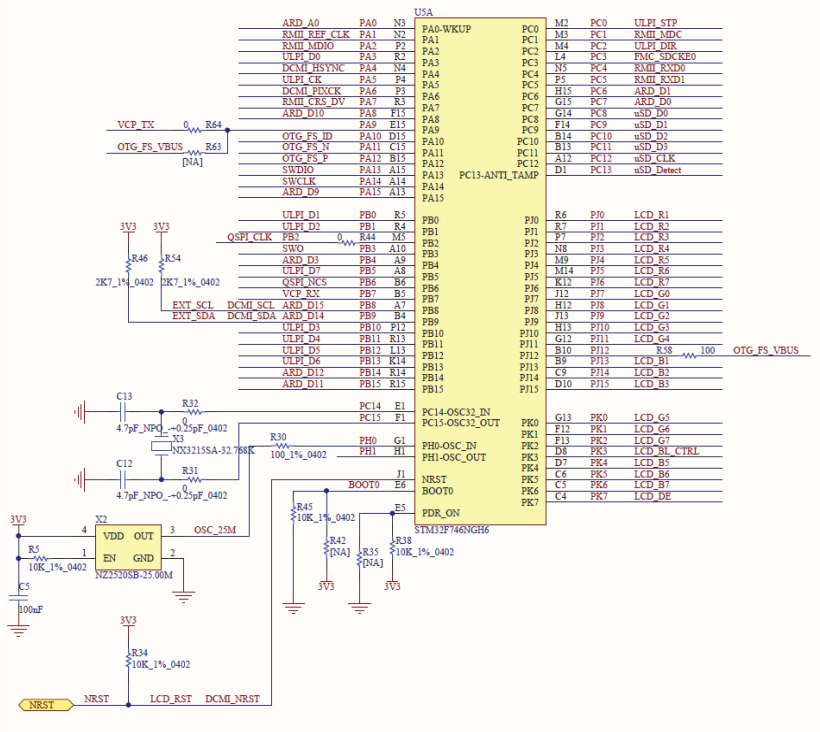
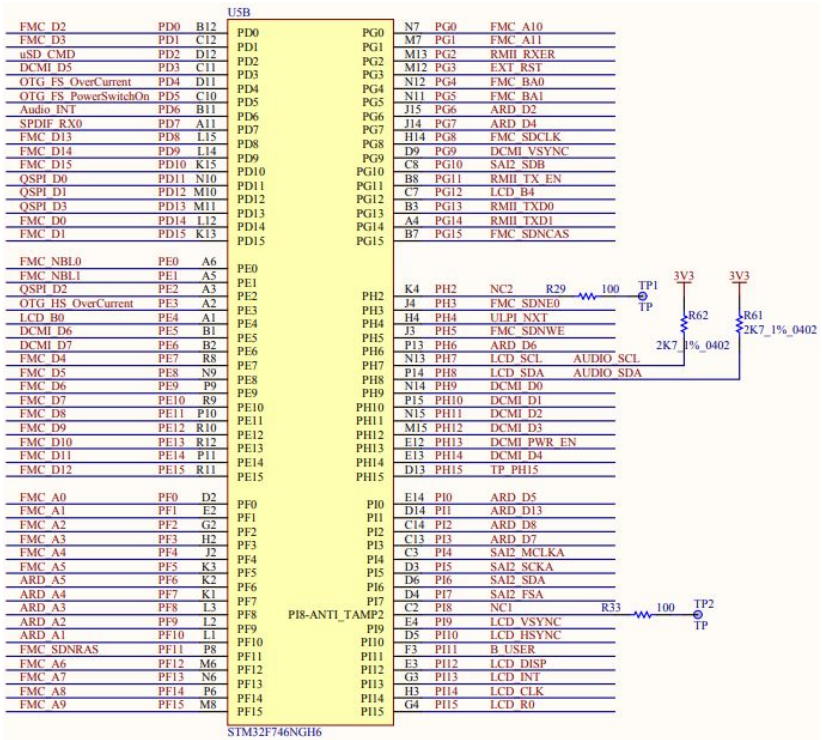


Figure 16. **QUADSPI schematic STM32 interface (part 2))**



5.2.2 QUADSPI simulation

The following sequence describes the steps for design and simulation of clock signal for QUADSPI interface:

Step 1. Schematic design.

The schematic shown in Figure 14 and Figure 15 is the connected signal between the serial NOR flash memory and the STM32F746 (QSPI_xx).

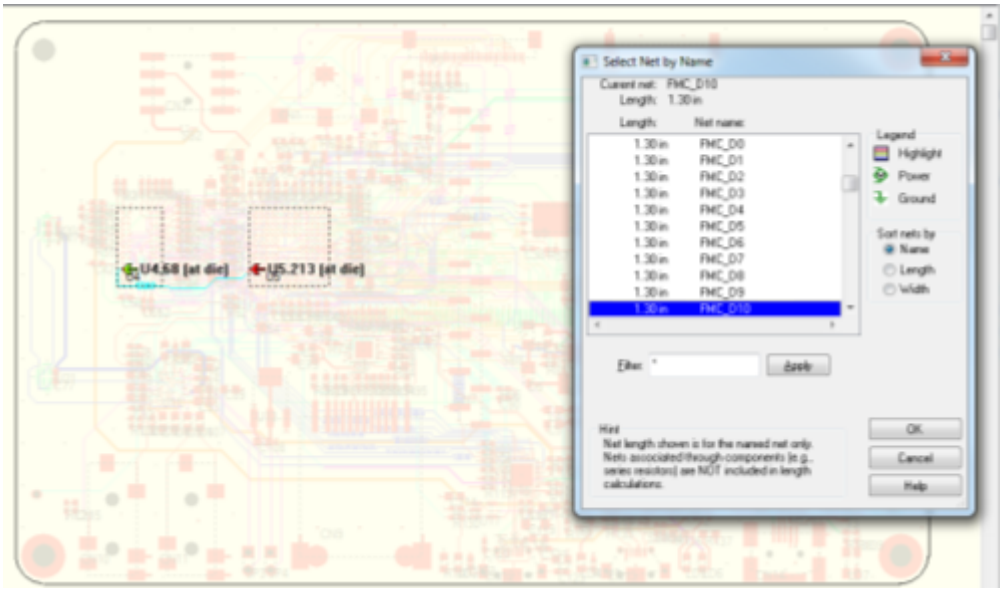
Step 2. Open the PCB board file to simulation QUADSPI.

Run HyperLynx and open the MB1191B-V14.hpy file for simulation.

Step 3. Select the signal to simulate.

Select the clock signal to simulate (for example QSPI_CLK/ PB2). Go to Select in upper menu and choose Net by Name for SI Analysis (see Figure 8).

Figure 17. Signal selection



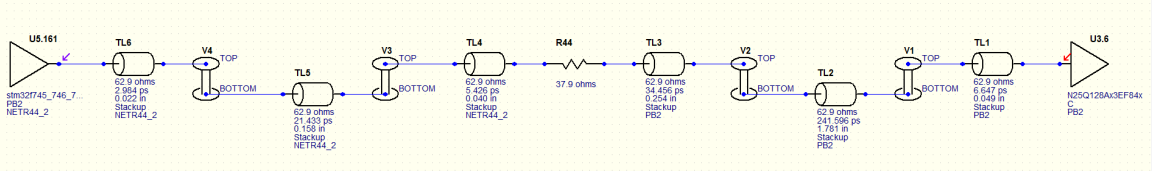
Step 4. Assign IBIS model for STM32F746 and N25Q128A13EF840E.

The IBIS model is usually available on the manufacturer's website. The IBIS model file associated with STM32F746 can be downloaded from the STMicroelectronics web site at www.st.com and for N25Q128A13EF840E can be downloaded from Micron web site.

After downloading the model for each IC and add it to the HyperLynx lib path. Assign the IBIS model for each signal vs IC Figure 8. Signal selection.

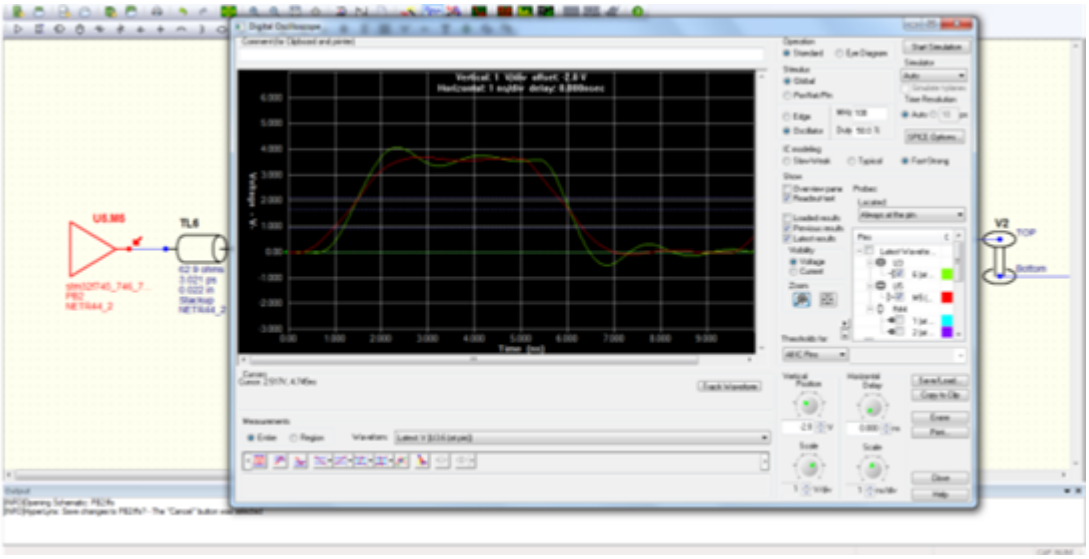
Step 5. Export the selected signal to the free-form schematic and configure the stack- up information.

Figure 18. Free-form schematic QSPI_CLK



Step 6. Configure and start the simulation.
Set the frequency to 108 MHz and the Duty to 50% (see figure below)

Figure 19. Waveform with R44 = 0Ω

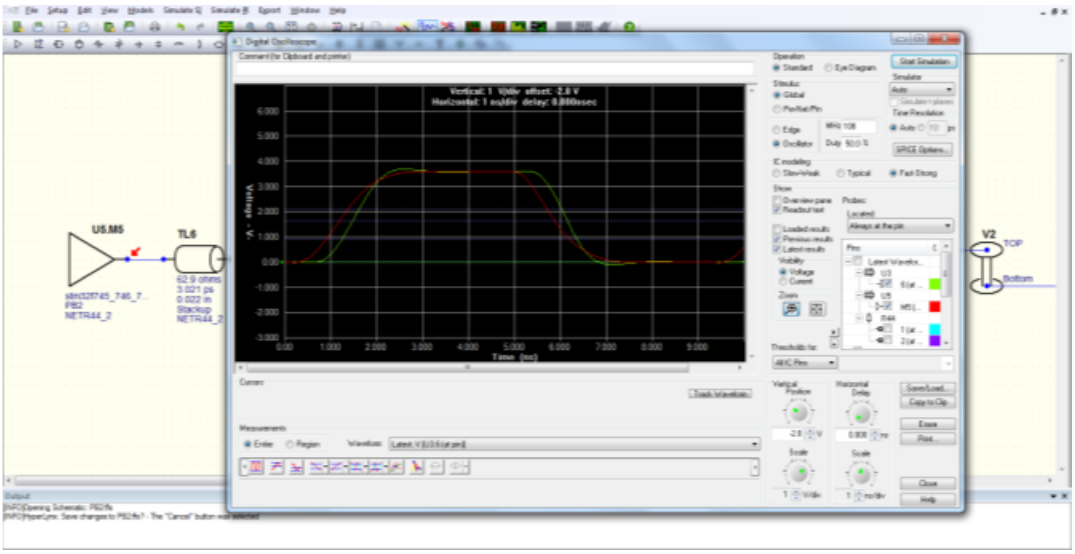


Step 7. Compare and analyze the results by changing R44 serial resistor.

In the previous steps, the series source termination resistor was 0Ω, the green waveform (at input of QSPI memory) is showing an overshooting and undershooting due to mismatching of the characteristic impedance. This type of termination requires that the sum of the buffer impedance and the value of the resistor be equal to the characteristic impedance of the line.

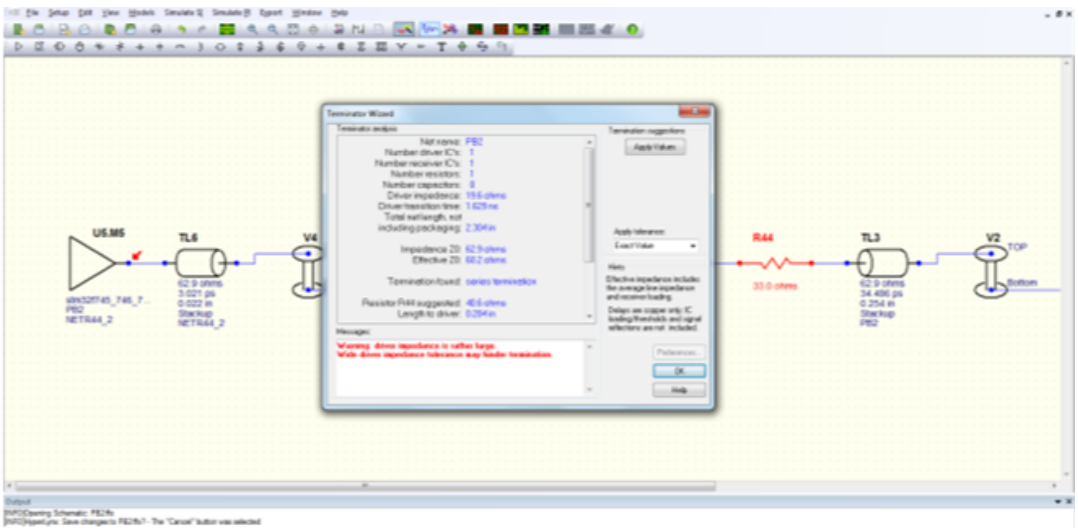
Double click on the R44 and change its value to 33Ω, see figure Figure 20.

Figure 20. Waveform with R44 = 33Ω



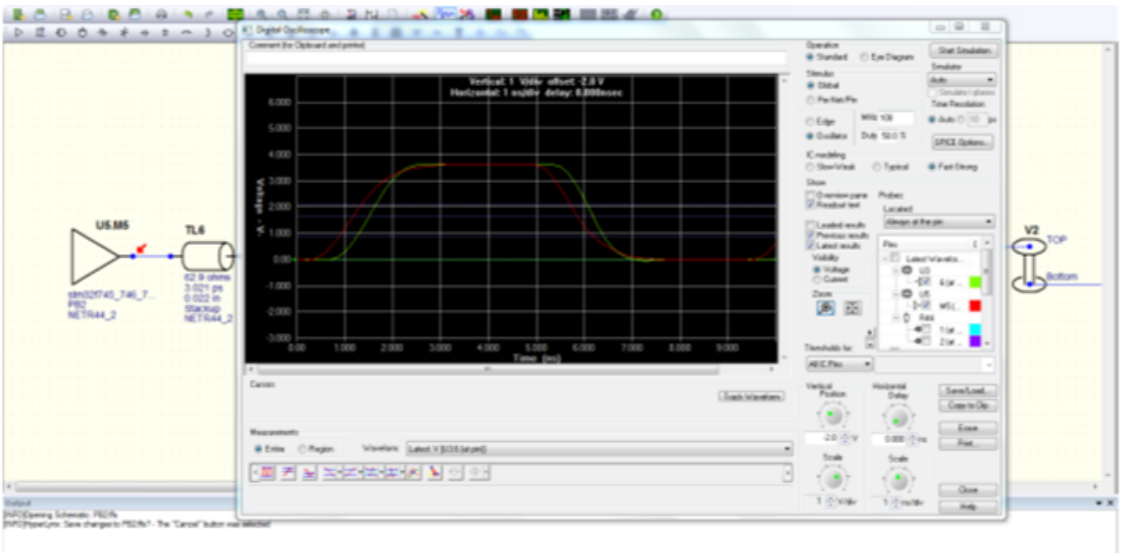
An improvement of shape of clock output from STM32 can be observed. Else, the Terminator Wizard can be run to analyze the selected net and suggest the optimum termination values for R44.

Figure 21. Terminator Wizard menu



When "Apply values" is selected, the serial resistor R44 takes the value in this schematic, which is 40.6Ω. See the shape of the wave with simulation in Figure 22.

Figure 22. **Waveform with R44 = 40.6Ω**



The Termination Wizard analyzes the selected net, presents a list of trace statistics and makes suggestions for the optimum value of R. It takes into account the capacitive loading of the receiver ICs, total line length, and driver impedance.

5.3 HyperLynx simulation with DDR3L

5.3.1 DDR3L signals

This example shows how to perform a simulation on STM32MP15 Series and DDR3L with DDRx batch simulation of Hyperlynx. The DDR SDRAM has:

- dedicated signals
- differential pair for clock
- address/command, controls
- reference voltage
- calibration resistor
- data lines in which four bytes are each composed with:
  - eight data
  - a data mask
  - differential pair for data strobe

Figure 23. **DDR3L schematic STM32MP15 interface**

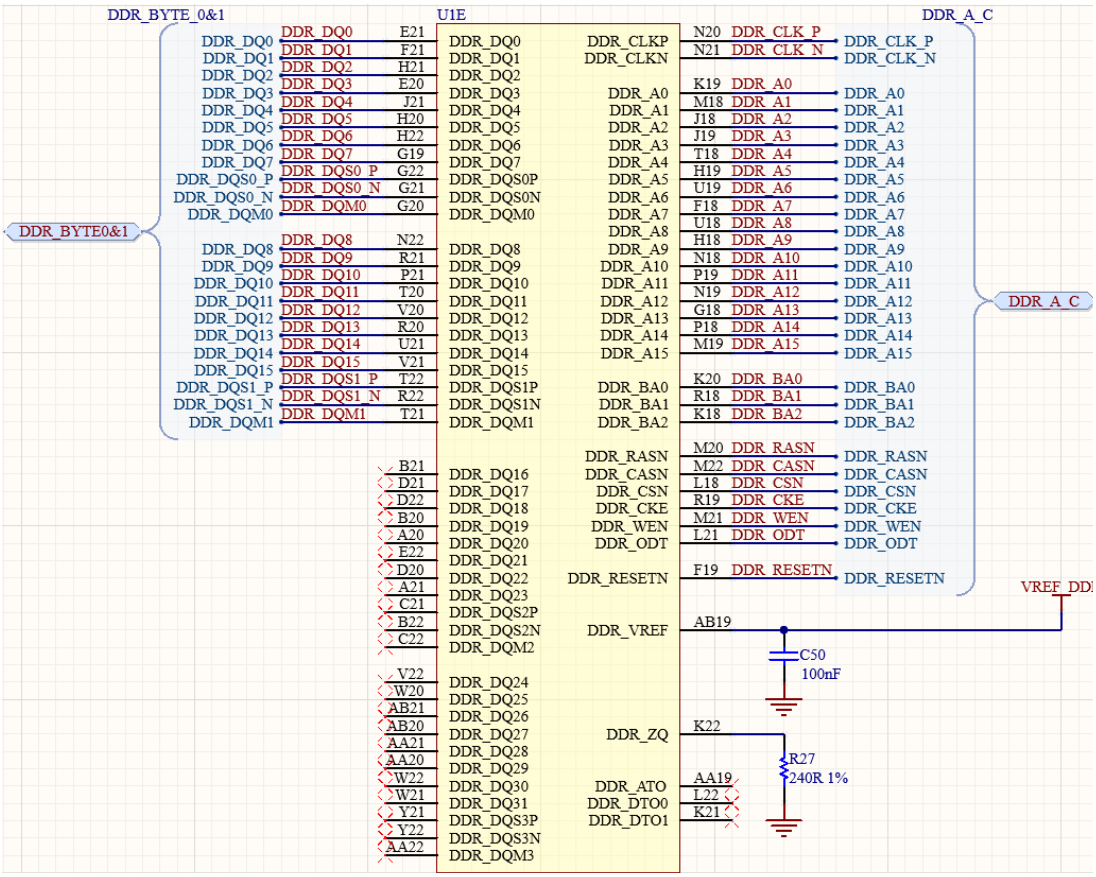
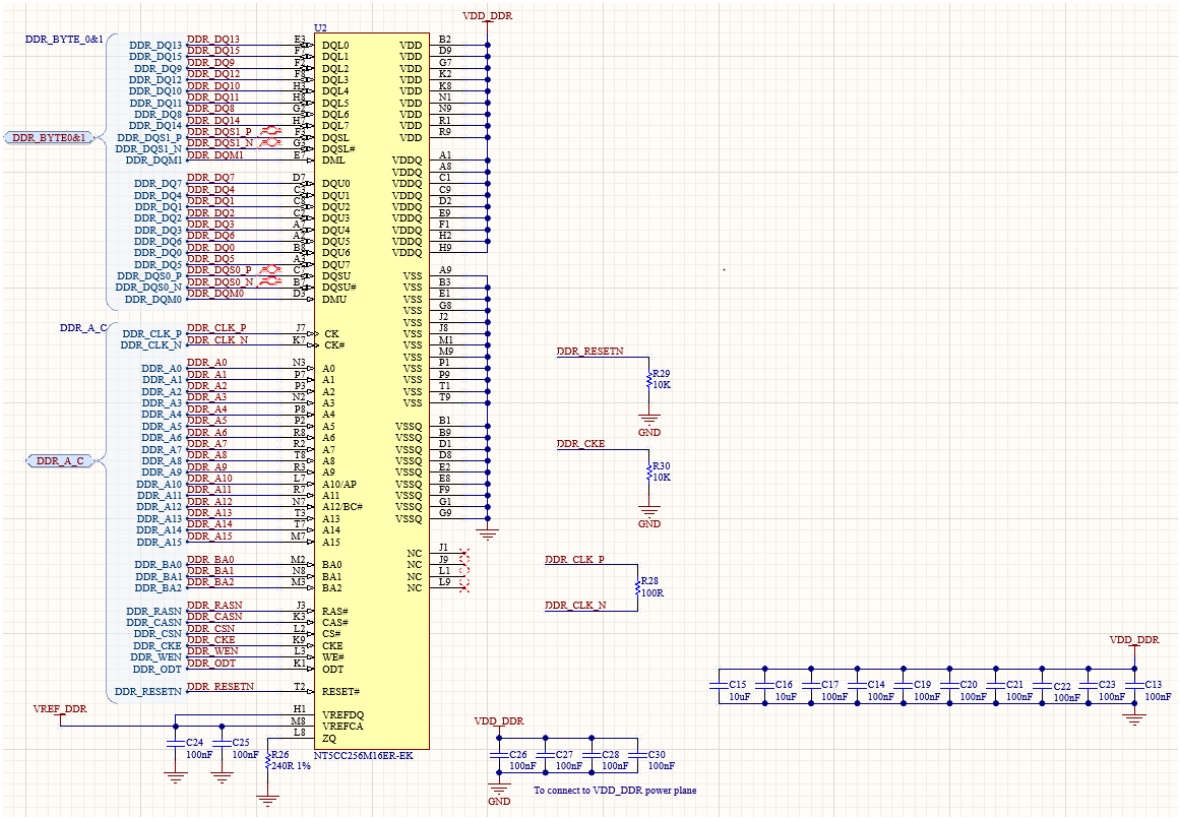


Figure 24. **DDR3L schematic memory interface**

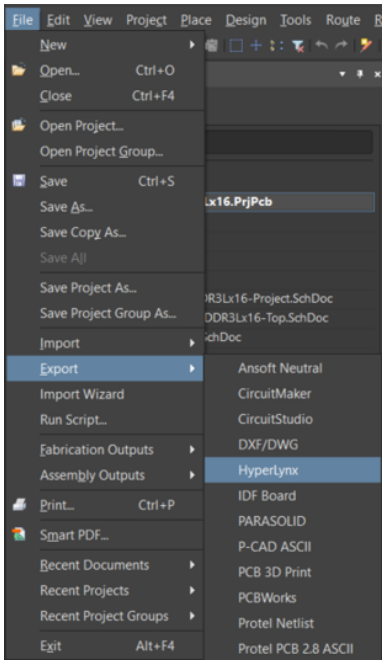


5.3.2 DDR3L simulation

When your layout is finished follow these steps:

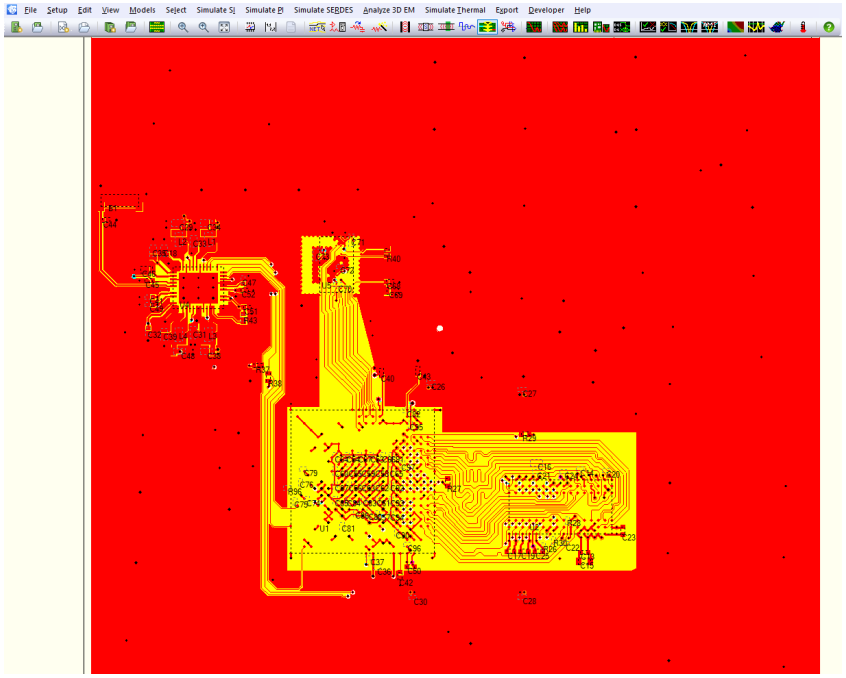
Step 1. Export your PCB into Hyperlynx file through Altium.

Figure 25. Altium export menu



Step 2. With Hyperlynx, open .hyp file created.

Figure 26. STM32MP15XXAA PCB

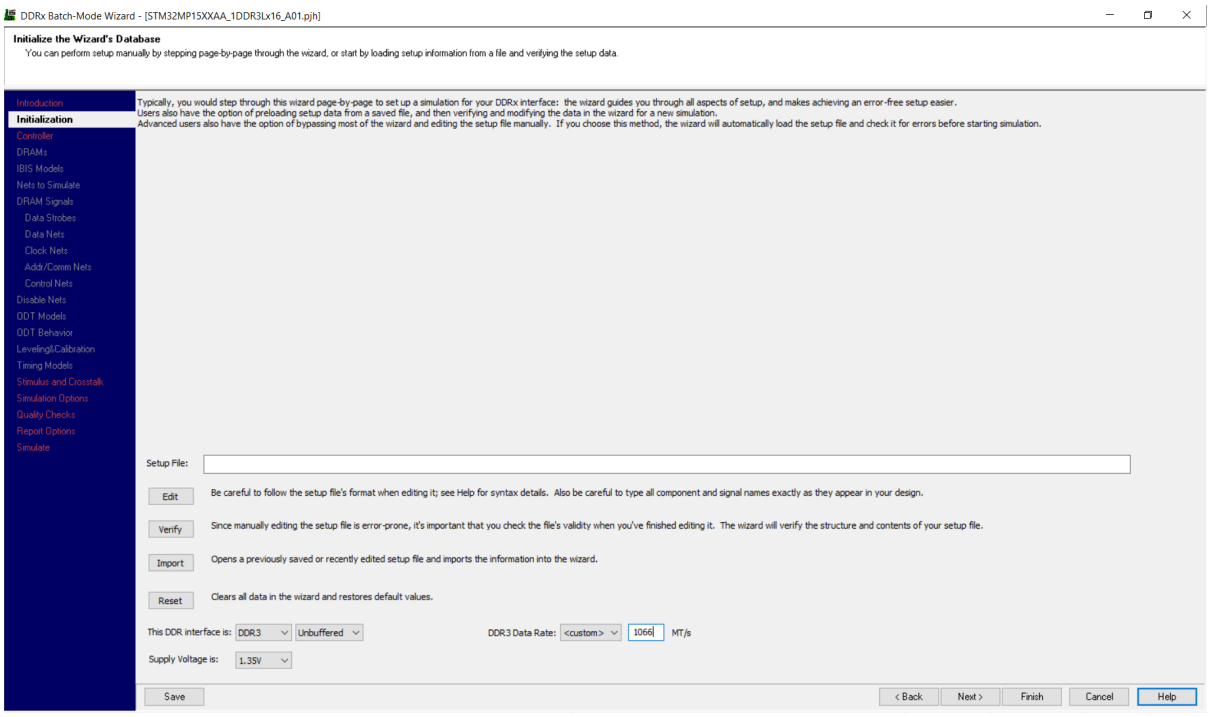


Step 3. Fill the stack up, the power supplies.

Do not forget to set as signal the layers used to route data and address/command tracks, as plane the ground layer and the layer which has the VDD_DDR power plane.

Step 4. Assign IBIS model by reference designator.

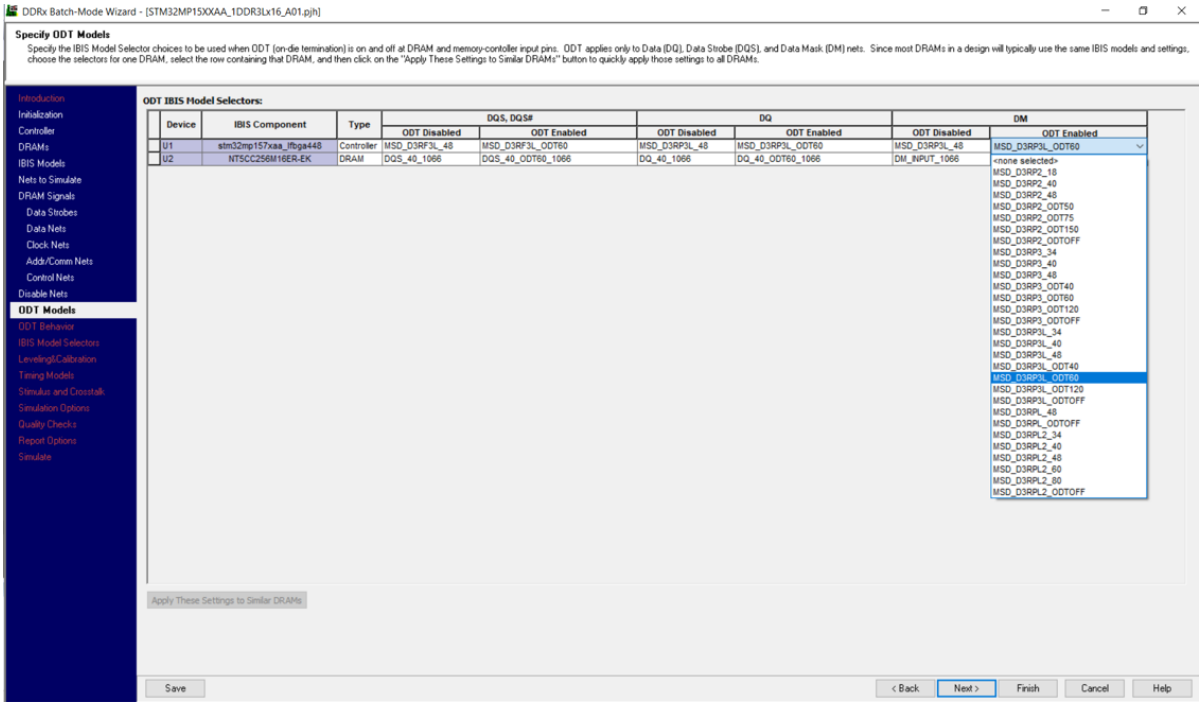
Figure 27. DDRx batch simulation



Step 5. Continue the set up of your simulation, assign the controller, the DRAM and the nets.

Step 6. Select the models of driver impedance and ODT for data, data mask, and differential pair for data strobe

Figure 28. DDRx batch simulation: ODT for data lines



Rule of data naming for output driver impedance in the column ‘ODT Disable’:

```
MSD_D3RP3_xx for DDR3 single-ended signal, xx is the value of driver impedance
MSD_D3RP3L_ xx for DDR3L single-ended signal, xx is the value of driver impedance
MSD_D3RPL2_ xx for LPDDR2 single-ended signal, xx is the value of driver impedance
MSD_D3RF3_xx for DDR3 differential signal, xx is the value of driver impedance
MSD_D3RF3L_ xx for DDR3L differential signal, xx is the value of driver impedance
MSD_D3RFL2_ xx for LPDDR2 differential signal, xx is the value of driver impedance
```

Rule of data naming for ODT receiver in the column ‘ODT Enable’:

```
MSD_D3RP3_ODTyy for DDR3 single-ended signal, yy is the value of ODT receiver
MSD_D3RP3L_ODTyy for DDR3L single-ended signal, yy is the value of ODT receiver
MSD_D3RPL2_ODTyy for LPDDR2 single-ended signal, yy is the value of ODT receiver
MSD_D3RF3_ODTyy for DDR3 differential signal, yy is the value of ODT receiver
MSD_D3RF3L_ODTyy for DDR3L differential signal, yy is the value of ODT receiver
MSD_D3RFL2_ODTyy for LPDDR2 differential signal, yy is the value of ODT receiver
```

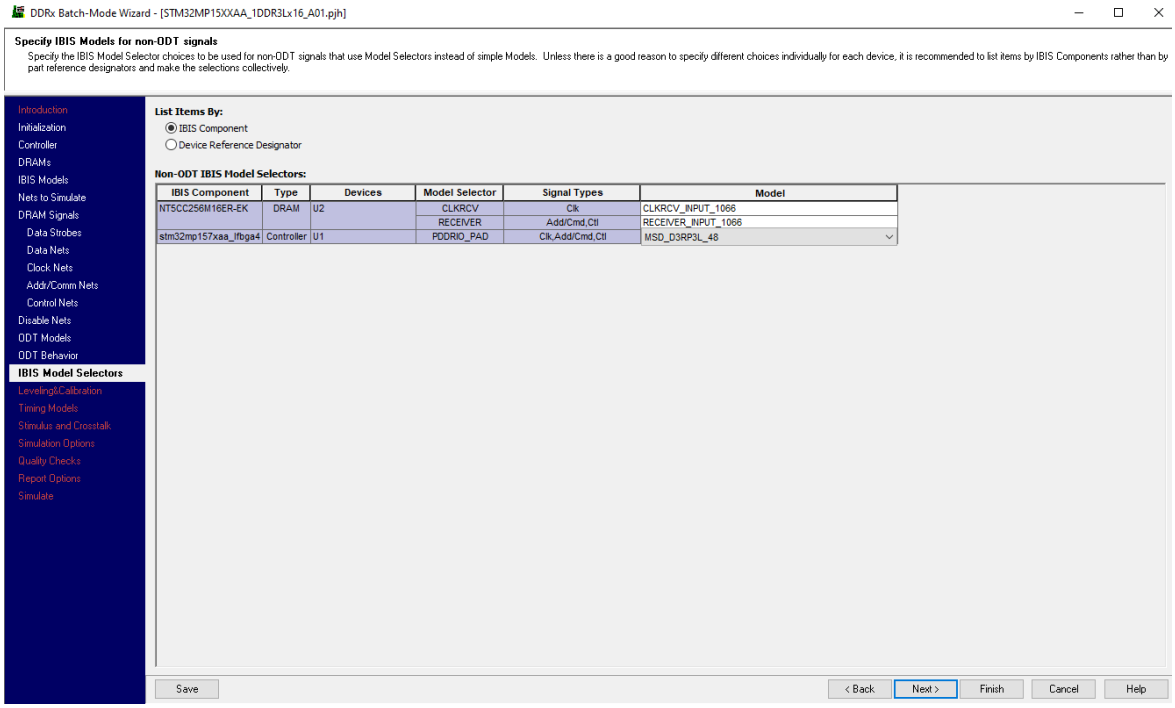
For LPDDR3, use LPDDR2 models.

Step 7. For the controller, select the models of driver impedance for address/command, controls, and differential pair for clock.

Even if clock is a differential signal, for simulation it is set as a single-ended signal. For the DRAM, select the input with the right data rate.

Step 7a. For the DRAM, select the input with the right data rate.

Figure 29. DDRx batch simulation: non ODT signal



Rule of address/command, control, and clock naming for driver impedance in the column ‘Model’:

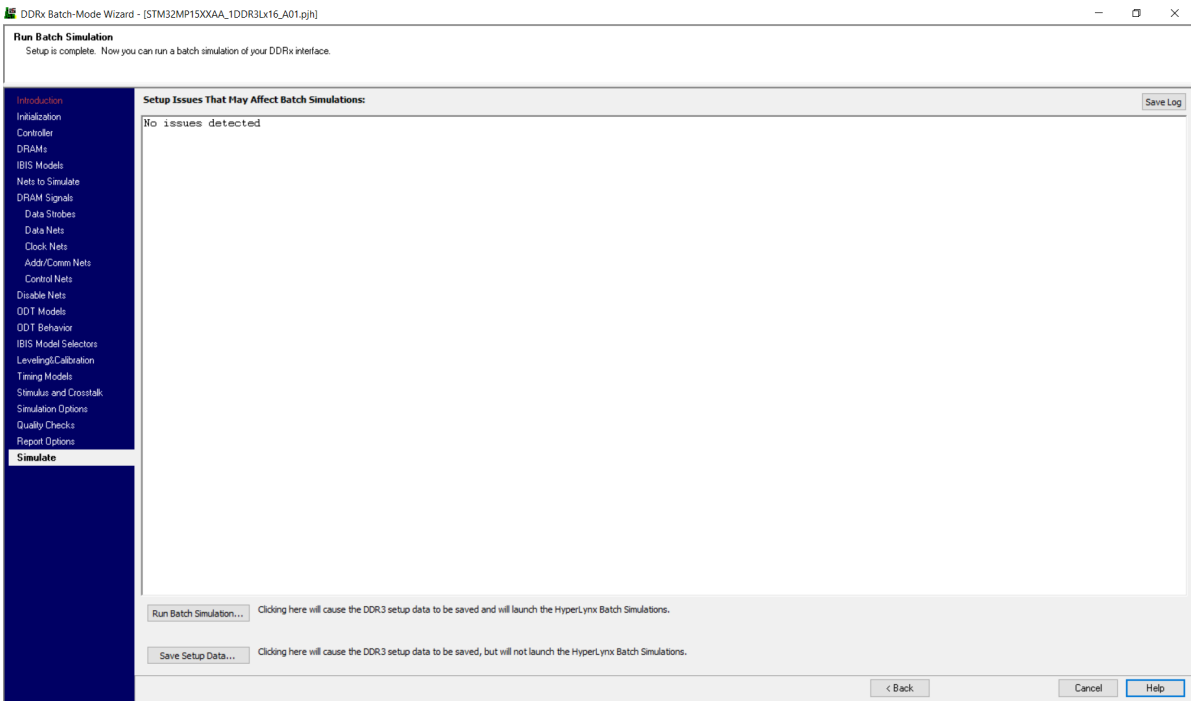
```
MSD_D3RP3_xx for DDR3 single-ended signal, xx is the value of driver impedance
MSD_D3RP3L_ xx for DDR3L single-ended signal, xx is the value of driver impedance
MSD_D3RPL2_ xx for LPDDR2 single-ended signal, xx is the value of driver impedance
```

For LPDDR3, use LPDDR2 models.

Full explanation of model selector can be found in the `STM32MP15 IBIS` file.

Step 8. Continue the setup until no issue is detected than click ‘’run Batch Simulation’’.
If the simulation is failed, find the best settings in accordance with your design.

Figure 30. DDRx batch simulation: simulate



6 References

- HyperLynx® LineSim User Guide Software Version 9.1, Mentor Graphics, March 2014
- HyperLynx® BoardSim User Guide Software Version 9.2, Mentor Graphics, December 2014
- High-Speed Digital System Design, Hall, Stephen, Hall Garrett, and McCall, James, John Wiley and Sons, Inc., 2000

Revision history

Table 3. **Document revision history**

Date	Version	Changes
1-Avr-2016	1	Initial release.
4-Sep-2019	2	Updated all the document to include MPUs on its scope.
10-Aug-2022	3	Updated: • Section Introduction • Figure 15 Added: • Section 5.3.1 • Section 5.3.2

Contents

List of tables

List of figures