AN4803
Application note

High-speed SI simulations using IBIS and board-level simulations using HyperLynx SI on STM32 32-bit ARM® Cortex® MCUs

Introduction

This application note serves as a guide on how to use the IBIS (I/O Buffer Information Specification) models of STMicroelectronics STM32 32-bit ARM® Cortex® MCUs and it is also a guide in how to use the external peripherals to perform board-level simulations with the HyperLynx® SI (Signal Integrity) software to address SI issues.

In order to use a concrete case, this application uses STM32F7xx Series as an example due to its complexity. All the information and conclusions can be extrapolated to all the STM32 32-bit ARM® Cortex® MCUs.

The STM32F7xx Series is based on ARM® Cortex®-M7 with FPU (floating point unit) processor. It is the latest generation of ARM® processors for embedded systems. It was developed to provide a low-cost platform that meets the needs of a MCU implementation. They have a reduced pin count and perform a low-power consumption while delivering an outstanding computational performance and a low-interrupt latency.

The STM32F7xx has a frequency of up to 216 MHz and a system speed of up to 100 MHz when interfacing with a high-speed interface such as SDRAM (Synchronous Dynamic Random-Access Memory).

As the signal speed increases it creates SI and EMC (Electro Magnetic Compliance) issues. It could be detected via test equipments as a signal degradation: overshooting, undershooting, ringing, crosstalk or timing delay. The signal degradation could be caused by a board design failure on certification (CE/FCC) or by timing violation issues between the IC (Integrated Circuit) drivers and the receiver. The accent should be put on getting the designs right the first time, avoiding costly over design, and saving recurrent layouts and prototypes. Therefore, performing a SI simulation is very important before doing any prototype.

List of tables

Table 1. I/Os in/output buffer for "io8p_arsudq_ft" selector 11
Table 2. Document revision history 24

Contents

1 SI fundamentals and STM32 signals 5

1.1 Signal integrity fundamentals 5

1.1.1 Signal integrity 5

1.1.2 Transmission line 5

1.1.3 Transmission line model 5

1.1.4 Characteristic impedance 7

1.2 IBIS model 7

1.2.1 IC modeling 7

1.2.2 Basic structure of an IBIS file 8

2 STM32 IBIS model selection/selector 10

2.1 GPIO structure 10

2.2 Model selector 10

2.3 Example of model selector on STM32F7xx MCU 10

3 Application example with HyperLynx simulator 12

3.1 HyperLynx simulation with SDRAM 12

3.1.1 SDRAM signals 12

3.1.2 SDRAM simulation 13

3.2 HyperLynx simulation with Quad-SPI 18

3.2.1 Quad-SPI signals 18

3.2.2 Quad-SPI simulation 19

4 References 22

5 Terminology 23

6 Revision history 24

List of figures

Figure 1. Transmission line at high frequency . . . 5
Figure 2. Transmission line with IC modeling . . . 7
Figure 3. IBIS editor . . . 8
Figure 4. IBIS data . . . 9
Figure 5. SDRAM schematic . . . 12
Figure 6. Discovery PCB . . . 13
Figure 7. Discovery PCB . . . 14
Figure 8. Signal selection . . . 14
Figure 9. Assign IBIS model . . . 15
Figure 10. Free-form schematic . . . 15
Figure 11. Waveform with IO speed of 0x00 . . . 16
Figure 12. Waveform with IO speed of 0x10 . . . 17
Figure 13. Waveform with IO speed of 0x11 . . . 17
Figure 14. Quad-SPI schematic NOR memory interface . . . 18
Figure 15. Quad-SPI schematic STM32 interface . . . 18
Figure 16. Signal selection . . . 19
Figure 17. Free-form schematic QSPI_CLK . . . 19
Figure 18. Waveform with R44=0 Ohm . . . 20
Figure 19. Waveform with R44=33 Ohm . . . 20
Figure 20. Terminator wizard menu . . . 21
Figure 21. Waveform with R44=40.6 Ohm . . . 21

1 SI fundamentals and STM32 signals

1.1 Signal integrity fundamentals

When the board traces carry signals containing high frequencies, special attention should 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, then the greater the need 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.

1.1.1 Signal integrity

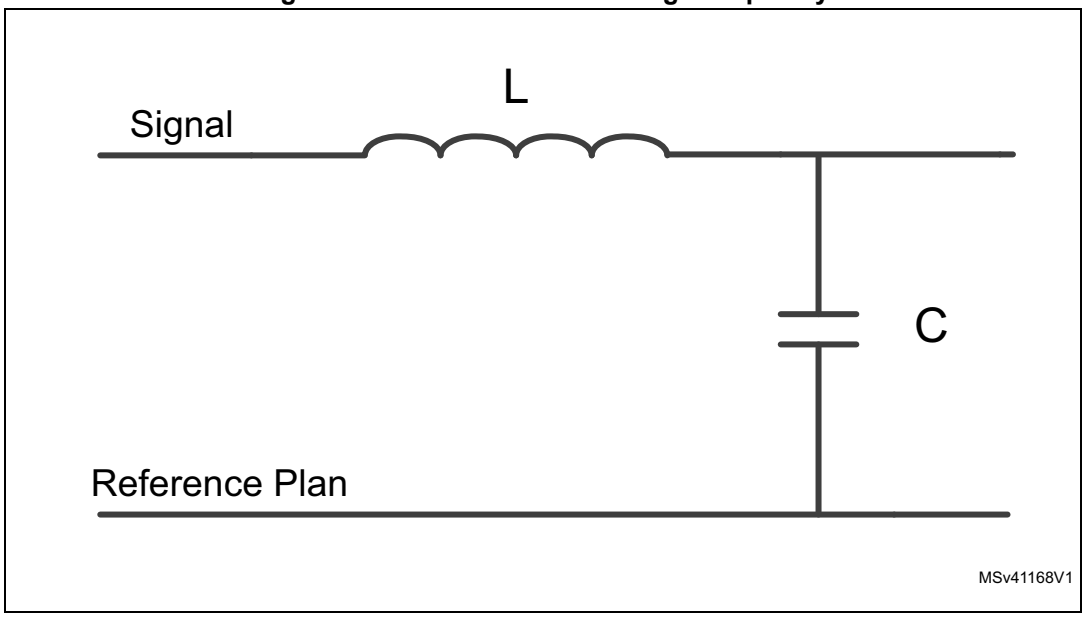
“Signal integrity” refers to the process of understanding and controlling behaviors of an ideal digital signal. It has recently became a critical element for any new digital PCB design as the clock speeds have increased by more than hundreds of megahertz. At high speeds, we can find signal and clock distortion, rise and fall time-edge distortion, setup timing violations and propagation delay times.

1.1.2 Transmission line

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

1.1.3 Transmission line model

Figure 1. Transmission line at high frequency



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

Equation1:

$$v = \frac{c}{\sqrt{\varepsilon_r}}$$

Equation 2:

$$TD = \frac{\sqrt{\varepsilon_r}}{c}$$

Where:

- v: propagation velocity, in meters/second
- c: speed of light in a vacuum (3×10^8 m/s)
- ε_r: 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:

Equation 3:

$$TD = \sqrt{LC}$$

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

1.1.4 Characteristic impedance

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

Equation 4:

$$Z_o = \sqrt{\frac{L}{C}}$$

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.

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

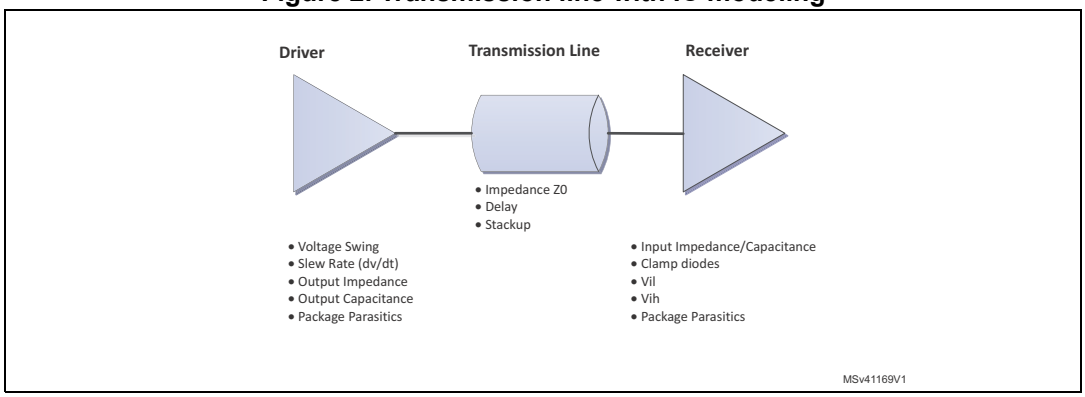
The potential problems that can be analyzed by means of the simulations include among others:

- 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

1.2.1 IC modeling

The *Figure 2* below shows an example of two ICs modeling:

Figure 2. Transmission line with IC modeling



1.2.2 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* below, 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

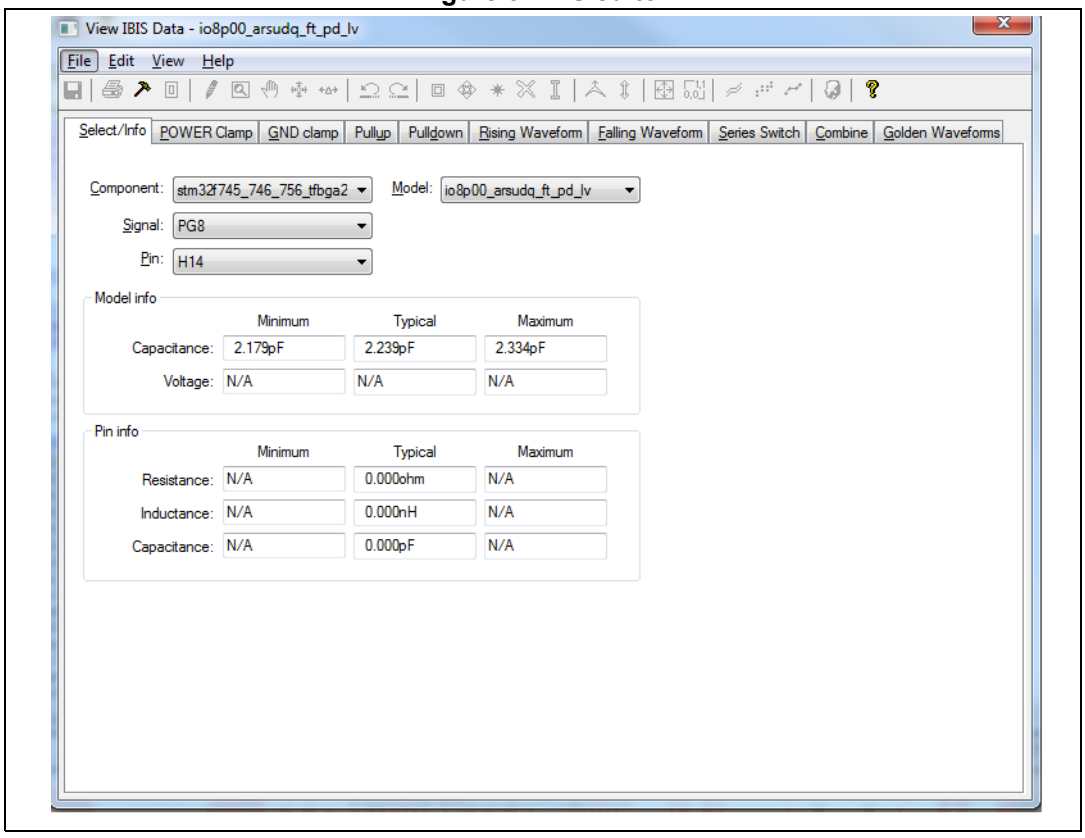
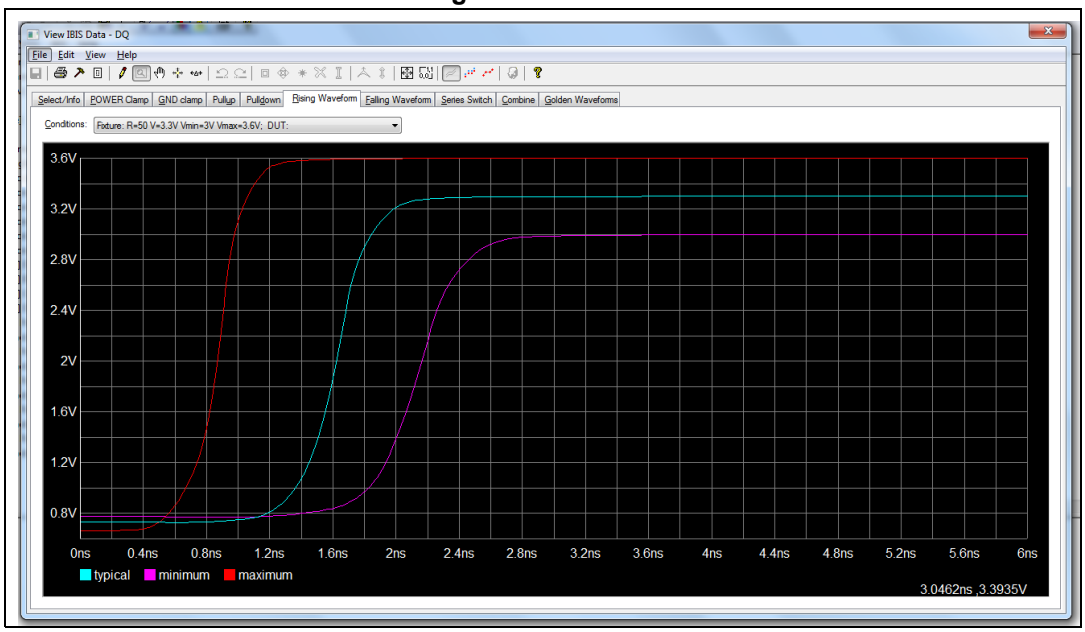


Figure 4. IBIS data



2 STM32 IBIS model selection/selector

This section presents the IBIS model selector of the available GPIO (General-Purpose Input/Output) pin in STM32 MCUs.

2.1 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.

2.2 Model selector

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

- Two operating voltage ranges:
  - V33(3.3V): refer to 2.7 V to 3.6 V external voltage range VDDx
  - V18 (1.8V or lv): refer to 1.7 V (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 STM32xxx datasheet on the section I/O port characteristics and also to STM32xx Reference Manual on the section General Purpose I/O (GPIO) for software configuration and selection.

2.3 Example of model selector on STM32F7xx MCU

In the example below we keep the same selected IO/Pin as in *Section 1.2.2*. The pin is H14 port PG8. This pin belongs to the family "io8p_arsudq_ft" of IO buffer.

In *Table 1* below, the pin H14 with selected GPIO configurations is highlighted in different colors as per table footnote legend.

Table 1. I/Os in/output buffer for "io8p_arsudq_ft" selector

IO Model Name Selection (io8p_ar3wsudq_ft)	IO Parameters		
	Voltage Range	Buffer Speed	Pull-Up/ Pull-Down
io8p00[1]_ar3wsudq_ft_pd[2]_lv [3] "SPEED00 1P8V, PD=40kOhm"	1.7Vto 2.7V[3]	Low Speed[1]	Pull Down 40K[2]
io8p00_ar3wsudq_ft_pu_lv "SPEED00 1P8V, PU=40kOhm"			Pull Up 40K
io8p00_ar3wsudq_ft_lv "SPEED00 1P8V"			Disabled
io8p01_ar3wsudq_ft_pd_lv "SPEED01 1P8V, PD=40kOhm"	1.7V to 2.7V	Medium Speed	Pull Down 40K
io8p01_ar3wsudq_ft_pu_lv "SPEED01 1P8V, PU=40kOhm"			Pull Up 40K
io8p01_ar3wsudq_ft_lv "SPEED01 1P8V"			Disabled
io8p10_ar3wsudq_ft_pd_lv "SPEED10 1P8V, PD=40kOhm"	1.7V to 2.7V	Fast Speed	Pull Down 40K
io8p10_ar3wsudq_ft_pu_lv "SPEED10 1P8V, PU=40kOhm"			Pull Up 40K
io8p10_ar3wsudq_ft_lv "SPEED10 1P8V"			Disabled
io8p11_ar3wsudq_ft_pd_lv "SPEED11 1P8V, PD=40kOhm"	1.7V to 2.7V	High Speed	Pull Down 40K
io8p11_ar3wsudq_ft_pu_lv "SPEED11 1P8V, PU=40kOhm"			Pull Up 40K
io8p11_ar3wsudq_ft_lv "SPEED11 1P8V"			Disabled
io8p00_ar3wsudq_ft_pd "SPEED00, PD=40kOhm"	2.7V to 3.6V	Low Speed	Pull Down 40K
io8p00_ar3wsudq_ft_pu "SPEED00, PU=40kOhm"			Pull Up 40K
io8p00_ar3wsudq_ft "SPEED00"			Disabled
io8p01_ar3wsudq_ft_pd "SPEED01, PD=40kOhm"	2.7V to 3.6V	Medium Speed	Pull Down 40K
io8p01_ar3wsudq_ft_pu "SPEED01, PU=40kOhm"			Pull Up 40K
io8p01_ar3wsudq_ft "SPEED01"			Disabled
io8p10_ar3wsudq_ft_pd "SPEED10, PD=40kOhm"	2.7Vto 3.6V	Fast Speed	Pull Down 40K
io8p10_ar3wsudq_ft_pu "SPEED10, PU=40kOhm"			Pull Up 40K
io8p10_ar3wsudq_ft "SPEED10"			Disabled
io8p11_ar3wsudq_ft_pd "SPEED11, PD=40kOhm"	2.7V to 3.6V	High Speed	Pull Down 40K
io8p11_ar3wsudq_ft_pu "SPEED11, PU=40kOhm"			Pull Up 40K
io8p11_ar3wsudq_ft "SPEED11"			Disabled

1. Green color highlights low buffer speed configuration (00) for pin H14.
2. Blue color highlights pull-down configuration for pin H14.
3. Orange color highlights low voltage range (1.8 V) for pin H14.

3 Application example with HyperLynx simulator

3.1 HyperLynx simulation with SDRAM

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

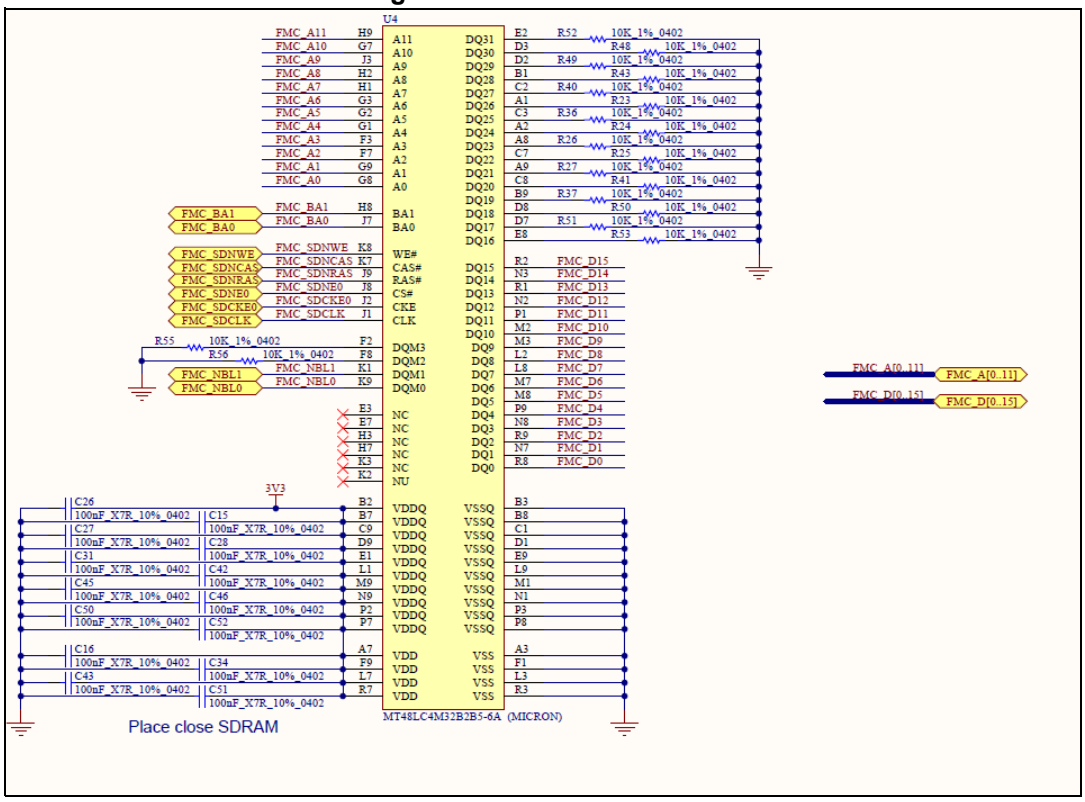
3.1.1 SDRAM signals

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 could 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 *Figure 5* below:

Figure 5. SDRAM schematic



3.1.2 SDRAM simulation

The following sequence describes the steps for design and simulation:

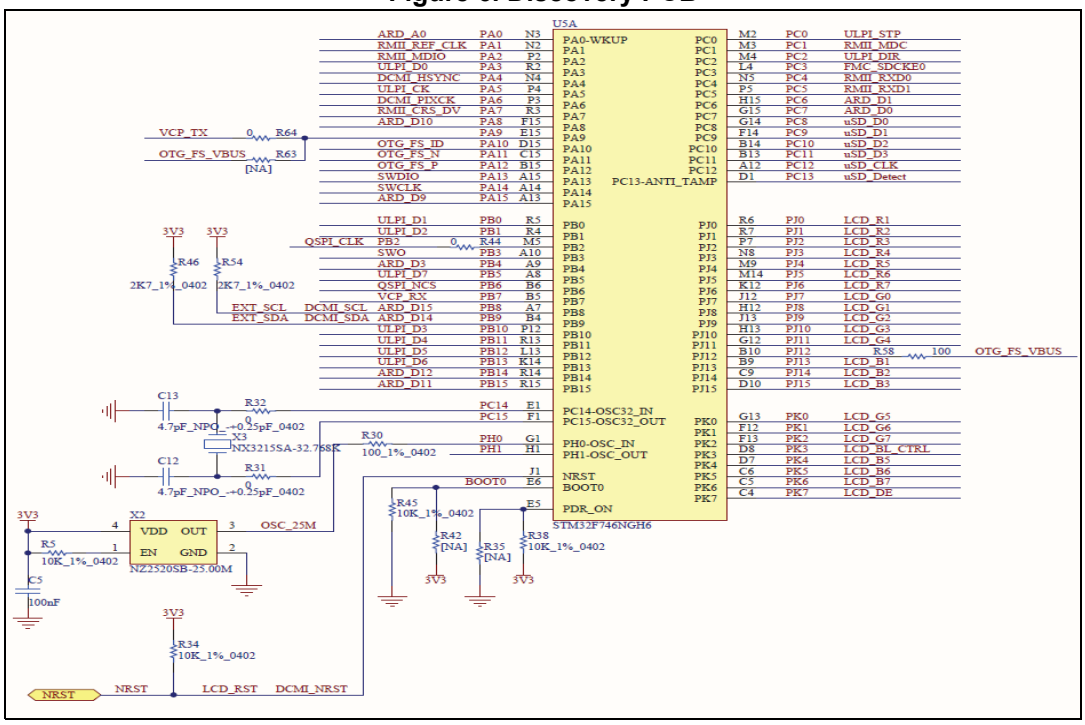
- **Step 1:** schematic design

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. Discovery PCB



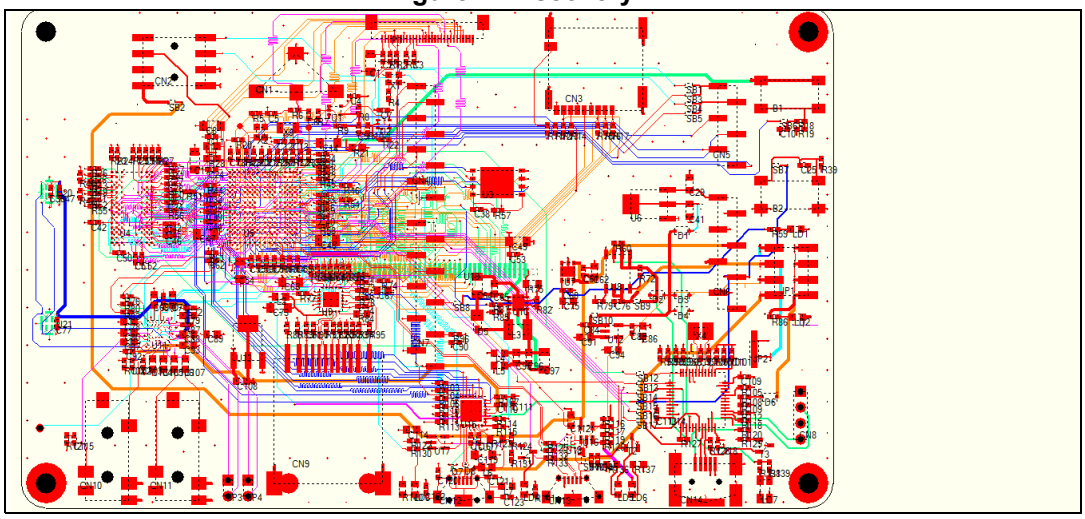
- **Step 3:** translate PCB board file to simulation file

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

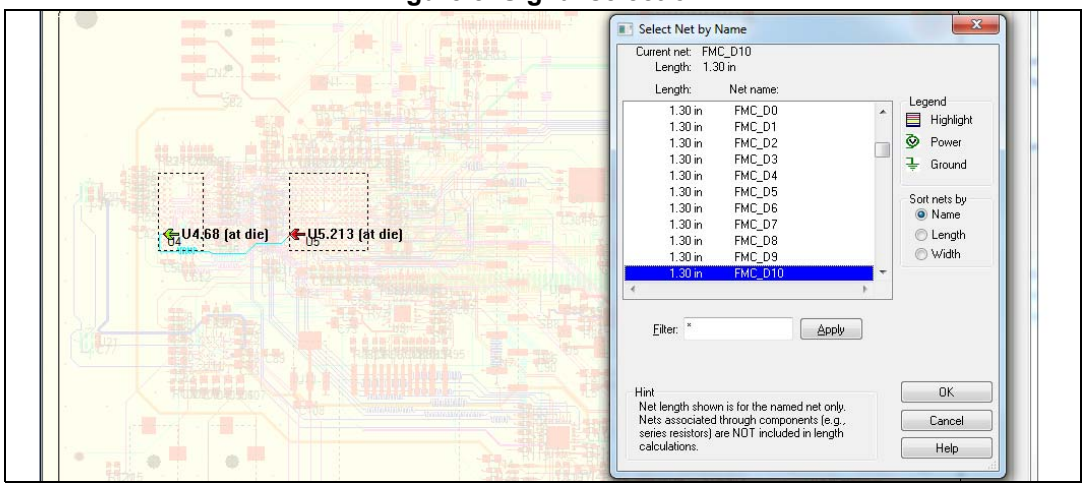
- **Step 4:** select the signal to simulate.
  - **Step 4.1:** open the **MB1191B-V14.hpy** file.

Figure 7. Discovery PCB



  - **Step 4.2:** select the signal to simulate (e.g., SDRAM FMC_D10). Go to Select in upper menu and choose Net by Name for SI Analysis (see *Figure 8*).

Figure 8. Signal selection

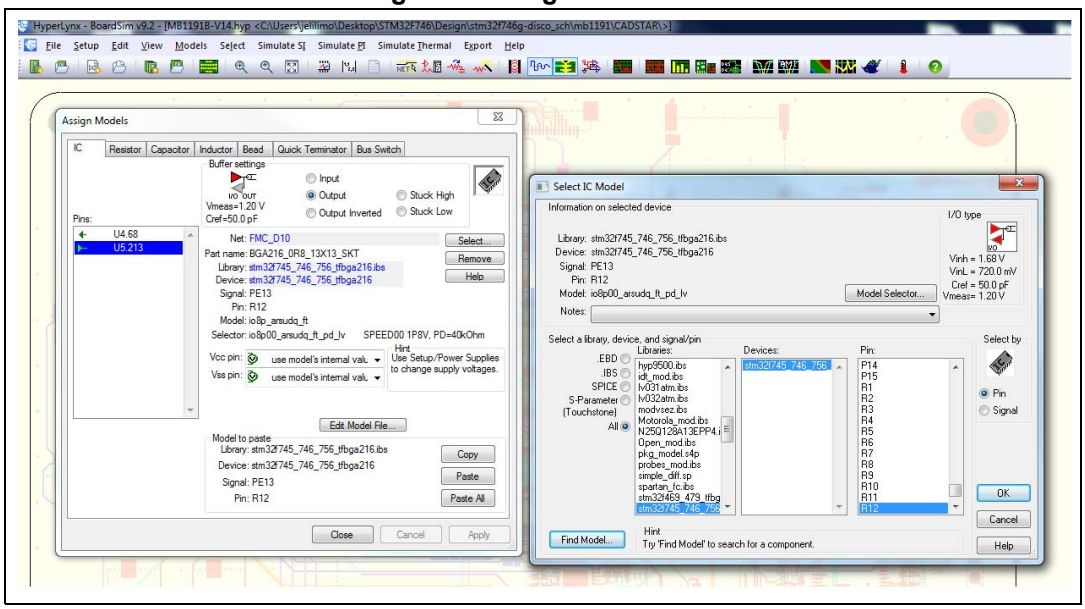


- **Step 5:** assign IBIS model for STM32F746 and MT48LC4M32B2B5.

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.

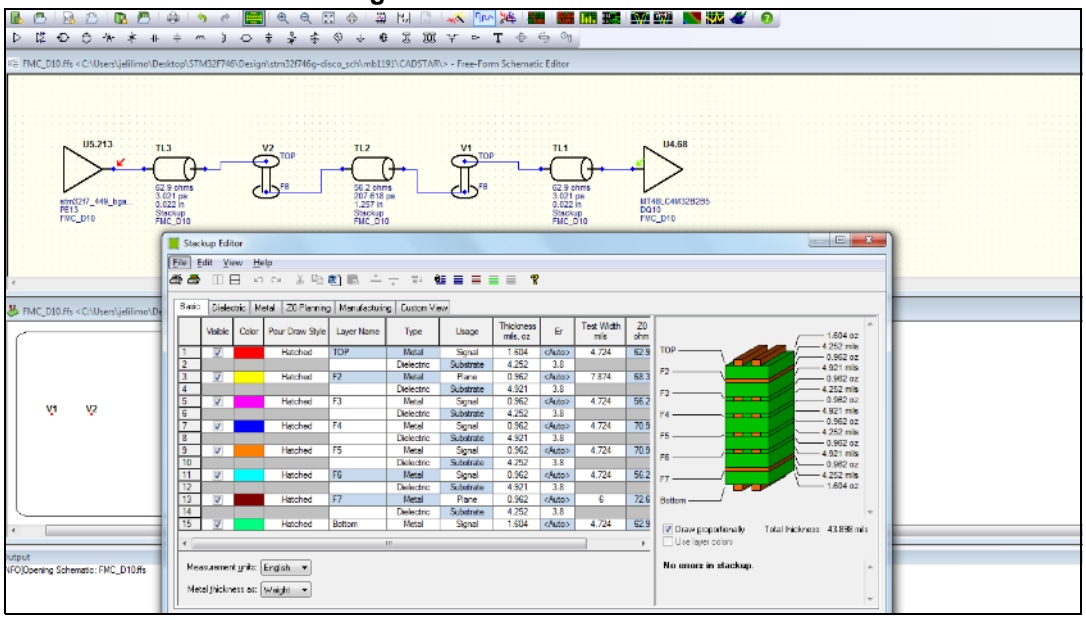
After downloading the model for each IC and add it to the HyperLynx lib path. Assign the IBIS model for each signal vs IC *Figure 9*:

Figure 9. Assign IBIS model



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

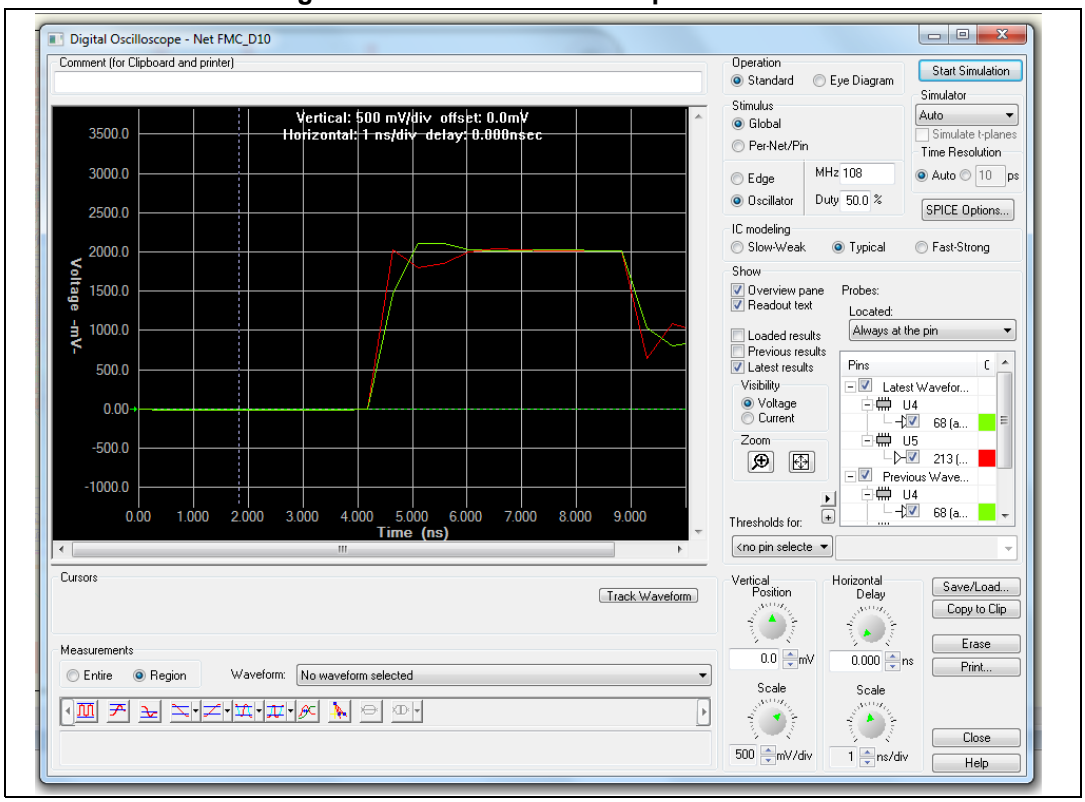
Figure 10. Free-form schematic



- **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 IO 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, we can see 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 IO frequency with this setting is 8MHz and rise time of 100ns. This is can be explained by output signal transitions under the loading conditions C_{ref} and R_{ref} for IO buffer model at lower speed 0x00.

In order to improve the shape of the waveform at the output of STM32F476, we need to change the IO speed to handle more signal frequency content to 0x10 (IO max freq. of 100MHz) and 0x11 (IO max freq. of 180 MHz) (see *Figure 12* below):

Figure 12. Waveform with IO speed of 0x10

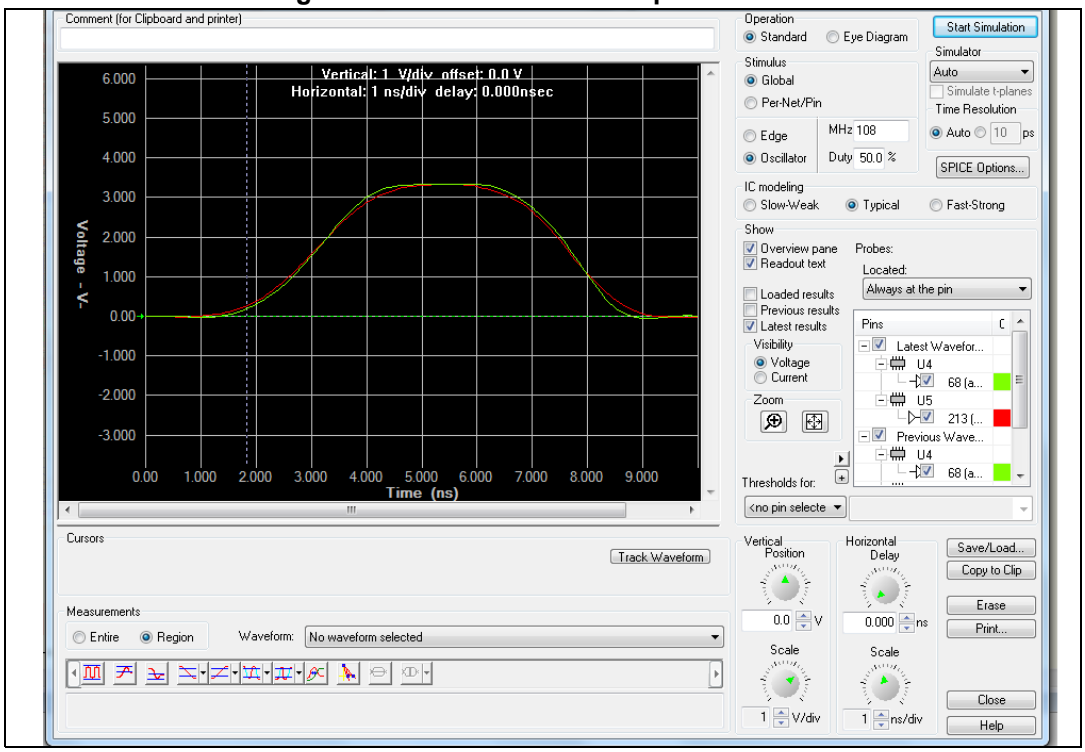
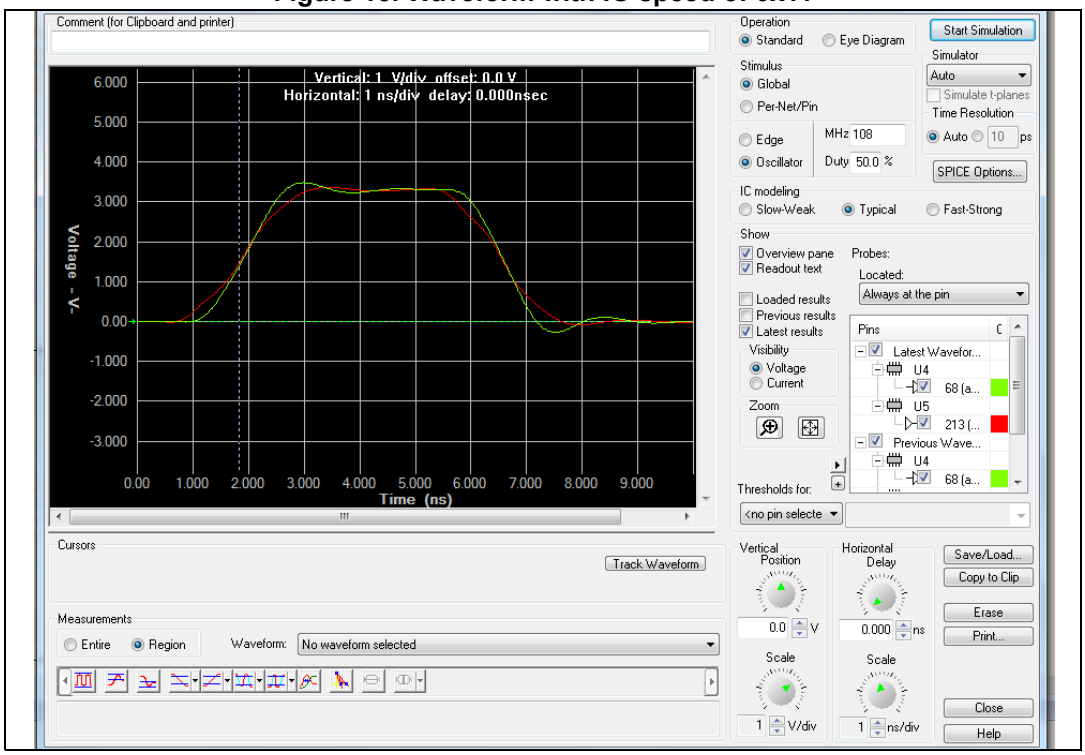


Figure 13. Waveform with IO speed of 0x11



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

3.2 HyperLynx simulation with Quad-SPI

3.2.1 Quad-SPI signals

Figure 14. Quad-SPI schematic NOR memory interface

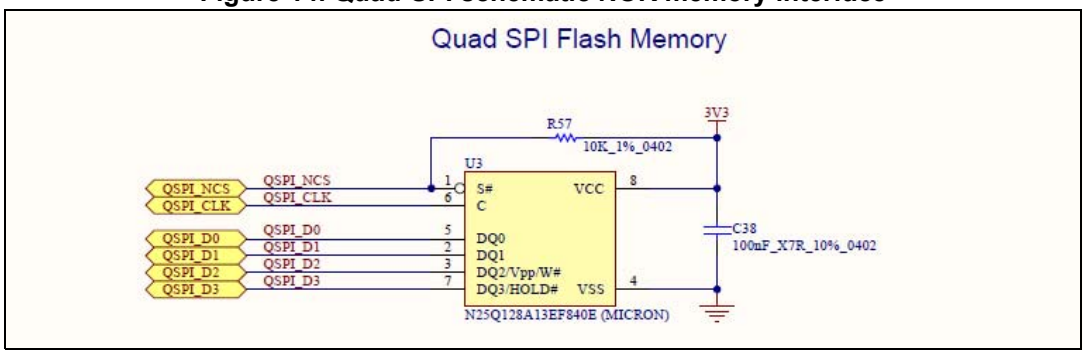
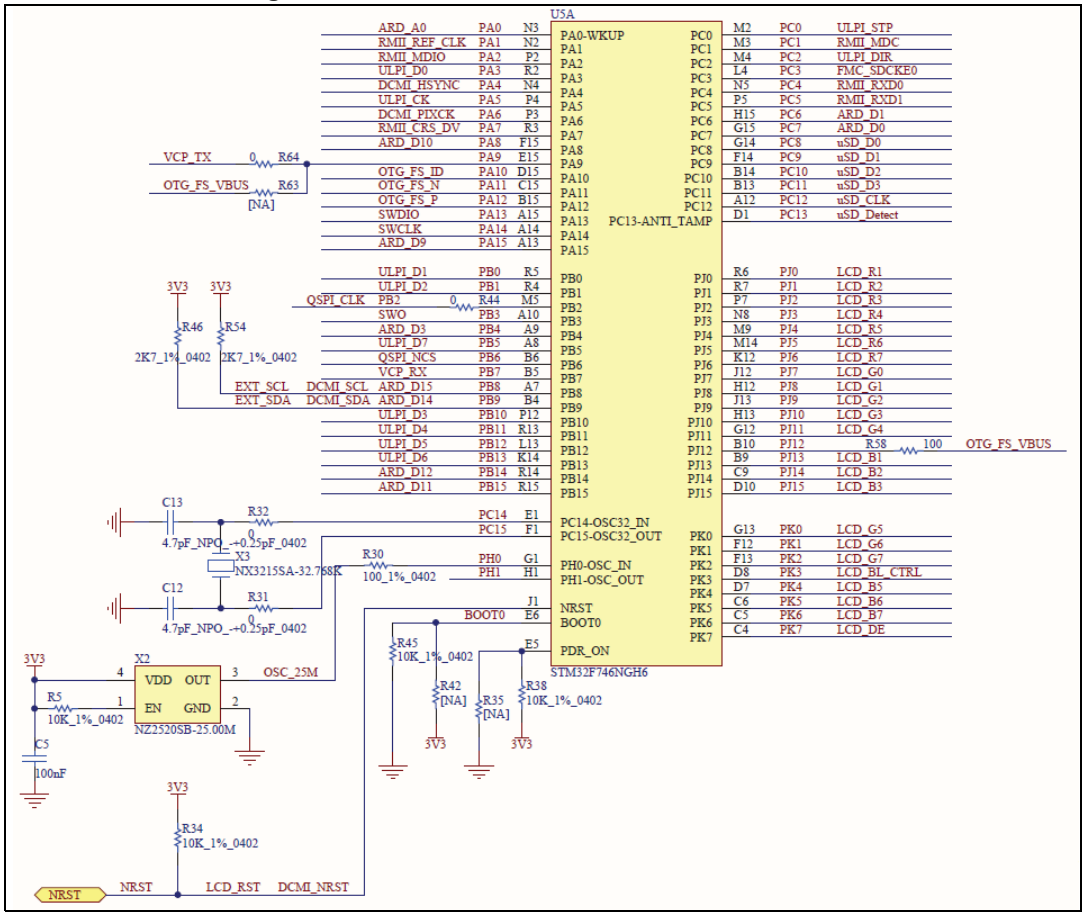


Figure 15. Quad-SPI schematic STM32 interface



3.2.2 Quad-SPI simulation

The following sequence describes the steps for design and simulation of clock signal for Quad-SPI 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).

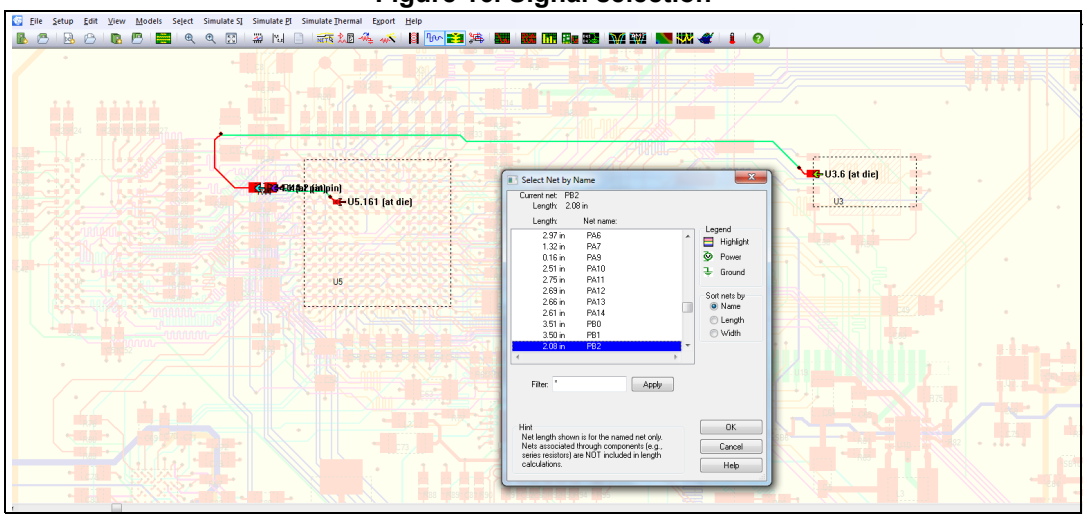
- **Step 2:** open PCB board file to simulation Quad-SPI

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

- **Step 3:** select the signal to simulate

Select the clock signal we want to simulate (e.g., QSPI_CLK/ PB2). Go to Select in upper menu and choose Net by Name for SI Analysis (see *Figure 16*).

Figure 16. Signal selection



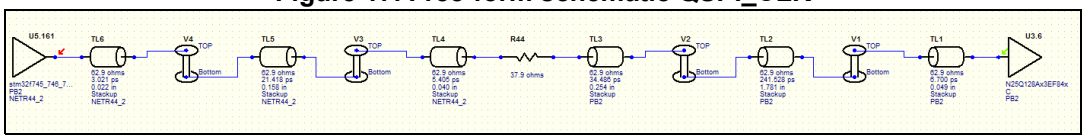
- **Step 4:** assign IBIS model for STM32F746 and N25Q128A13EF840E

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 16*.

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

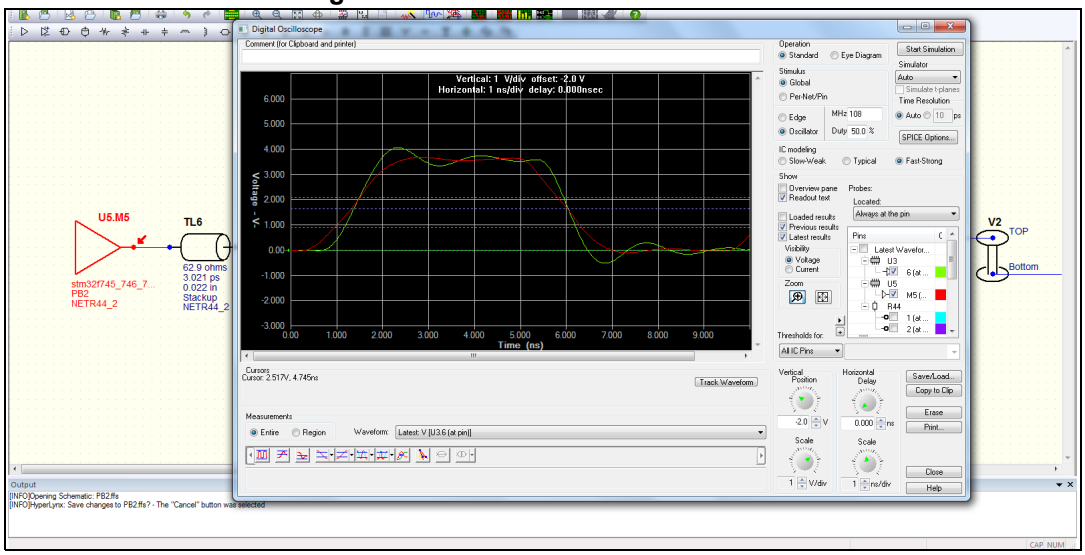
Figure 17. Free-form schematic QSPI_CLK



- **Step 6:** configure and start the simulation.

Set the frequency to 108 MHz and the Duty to 50% (see *Figure 18*).

Figure 18. Waveform with R44=0 Ohm

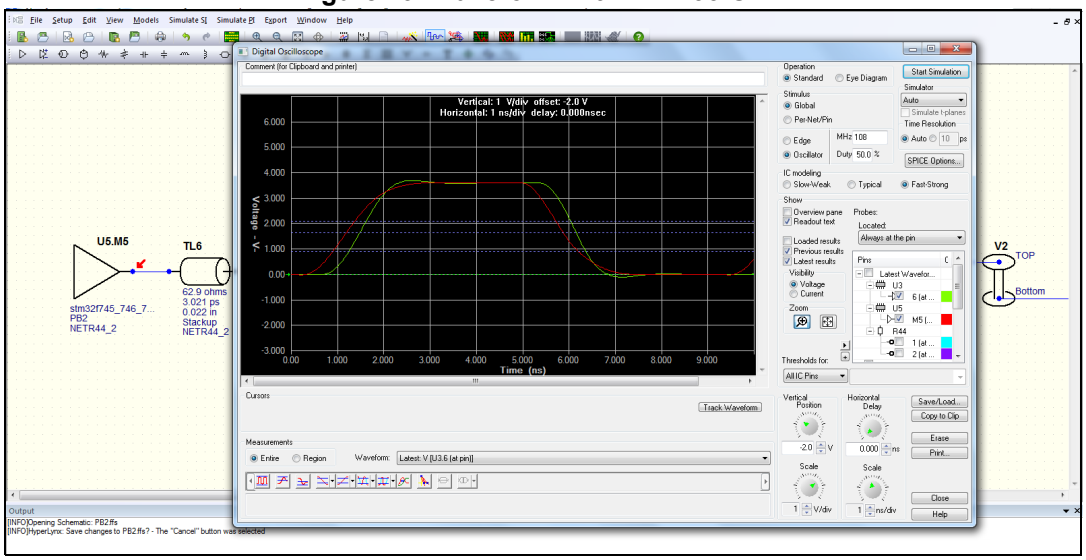


- **Step 7:** compare and analyze the results by changing R44 serial resistor

In the previous steps, the Series source termination resistor was 0 Ohm, 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.

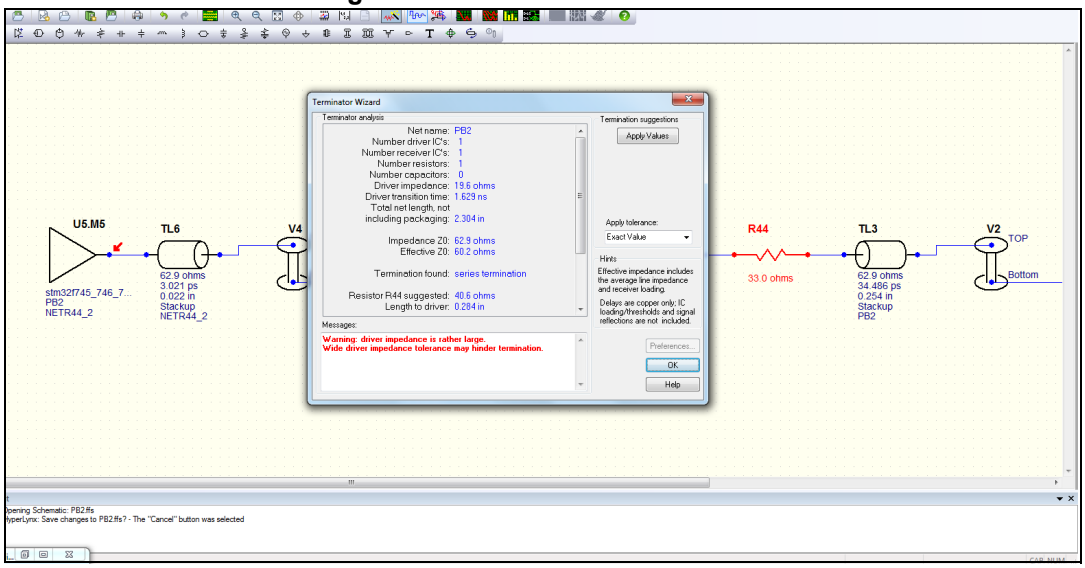
Double click on the R44 and change its value to 33 Ohm, see *Figure 19* below.

Figure 19. Waveform with R44=33 Ohm



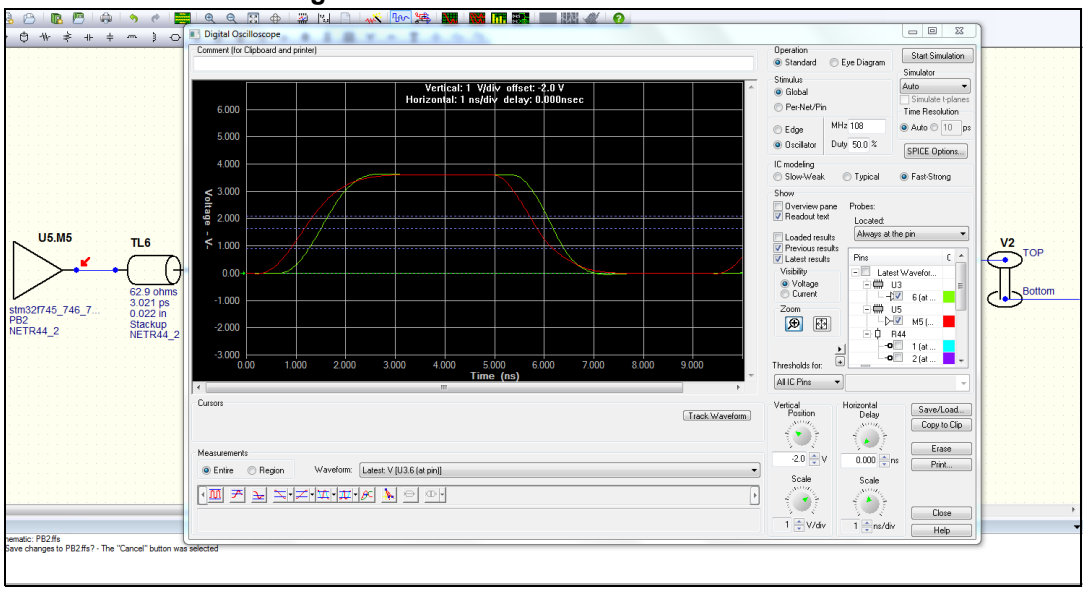
We can see a better improvement of shape of clock output from STM32. Or we can run the Terminator Wizard to analyzes the selected net and suggest the optimum termination values for R44.

Figure 20. Terminator wizard menu



When we select Apply values, the serial resistor R44 will take this value in this schematic, which is 40.6 Ohm. See the shape of the wave with simulation in *Figure 21* below:

Figure 21. Waveform with R44=40.6 Ohm



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

4 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

5 Terminology

SI: Signal Integrity, denotes the correct timing and quality of the signal.

EMC: Electro Magnetic 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.

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.

FR4: 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.

SDRAM: Synchronous Dynamic Random Access Memory.

Quad-SPI (QSPI): is a specialized communication interface targeting single, dual or quad SPI flash memories.

6 Revision history

Table 2. Document revision history

Date	Revision	Changes
01-Apr-2016	1	Initial release.