AN3099
Application note

IBIS models for signal integrity simulation of SPEAr3xx applications

Introduction

This application note is intended for hardware developers that are using the SPEAr3xx embedded MPU in their target design.

IBIS models are mandatory to run signal integrity simulation in the application PCB. PCB simulation is very important to make sure that the layout of the PCB does not introduce any functional problems or timing marginality in high speed interfaces like DDR2.

The IBIS models provided for SPEAr3xx are organized in a model library containing several models for each I/O pin (or for a functional group of I/Os). Each I/O pin or functional group of I/O has a set of models; each model corresponds to a certain operating mode of the I/O pads.

The operating modes are programmable and are defined by a proper setting of two registers of the SPEAr3xx device (for more details please refer to the miscellaneous registers and DDR memory controller sections of the SPEAr3xx user manual).

This document explains how to extract the correct model from the library sp300_v3.ibs after reading the register setting or knowing the operating mode.

1 PL_CLK & PL_GPIO IBIS model selection

The I/O signals of the PL_CLK & PL_GPIO interface form two main groups.

Please refer to the following table for the I/O signals and the associated model name.

Table 1. Relation between the PL_.... block signals and the used model

PL_.... block signal name	IBIS model group name
PL_CLK[4:1]	BDPROG_2ROWS_xxxxx xxxxx = MODEOP0 MODEOP1 DRV0 DRV1 SL
PL_GPIO[97:95] PL_GPIO[92:85] PL_GPIO[82:73] PL_GPIO[70:69] PL_GPIO[67:58] PL_GPIO[56:53] PL_GPIO[50:43] PL_GPIO[40:33] PL_GPIO[30:25] PL_GPIO[22:19] PL_GPIO[16:11] PL_GPIO[8:5] PL_GPIO[2:1]	BDPROG_2ROWS_xxxxx xxxxx = MODEOP0 MODEOP1 DRV0 DRV1 SL
PL_GPIO[94:93] PL_GPIO[84:83] PL_GPIO[72:71] PL_GPIO[68] PL_GPIO[57] PL_GPIO[52:51] PL_GPIO[42:41] PL_GPIO[32:31] PL_GPIO[24:23] PL_GPIO[18:17] PL_GPIO[10:9] PL_GPIO[4:3] PL_GPIO[0]	BDPROG_STD_xxxxx xxxxx = MODEOP0 MODEOP1 DRV0 DRV1 SL

Each model group name contains a set of models depending upon the value of the 5 bits xxxx for the selected PL_GPIO [….] or PL_CLK [….] signals.

You must search in the user manual for the following register names:

1. PL_GPIO0_PAD_PRG (Address 0x130)
2. PL_GPIO1_PAD_PRG (Address 0x134)
3. PL_GPIO2_PAD_PRG (Address 0x138)
4. PL_GPIO3_PAD_PRG (Address 0x13C)
5. PL_GPIO4_PAD_PRG (Address 0x140)

For all the models the first two bits are the high voltage level:

MODEOP0 = 1

MODEOP1 = 1

The correlation between bits 0-2 of the model group names and the bits of each register is given below:

- Register PL_GPIO0_PAD_PRG (Address 0xFCA80130).
  PL_GPIO [23:20]:
    DRV1 = PL_GPIO0_PAD_PRG [27]
    DRV0 = PL_GPIO0_PAD_PRG [26]
    SL = PL_GPIO0_PAD_PRG [25]
  PL_GPIO [19:16]:
    DRV1 = PL_GPIO0_PAD_PRG [22]
    DRV0 = PL_GPIO0_PAD_PRG [21]
    SL = PL_GPIO0_PAD_PRG [20]
  PL_GPIO [15:12]:
    DRV1 = PL_GPIO0_PAD_PRG [17]
    DRV0 = PL_GPIO0_PAD_PRG [16]
    SL = PL_GPIO0_PAD_PRG [15]
  PL_GPIO [11:8]:
    DRV1 = PL_GPIO0_PAD_PRG [12]
    DRV0 = PL_GPIO0_PAD_PRG [11]
    SL = PL_GPIO0_PAD_PRG [10]
  PL_GPIO [7:4]:
    DRV1 = PL_GPIO0_PAD_PRG [7]
    DRV0 = PL_GPIO0_PAD_PRG [6]
    SL = PL_GPIO0_PAD_PRG [5]
  PL_GPIO [3:0]:
    DRV1 = PL_GPIO0_PAD_PRG [2]
    DRV0 = PL_GPIO0_PAD_PRG [1]
    SL = PL_GPIO0_PAD_PRG [0]
- Register PL_GPIO1_PAD_PRG (Address 0xFCA80134).
  PL_GPIO [47:44]:
    DRV1 = PL_GPIO1_PAD_PRG [27]
    DRV0 = PL_GPIO1_PAD_PRG [26]
    SL = PL_GPIO1_PAD_PRG [25]

PL_GPIO [43:40]:
DRV1 = PL_GPIO1_PAD_PRG [22]
DRV0 = PL_GPIO1_PAD_PRG [21]
SL = PL_GPIO1_PAD_PRG [20]
PL_GPIO [39:36]:
DRV1 = PL_GPIO1_PAD_PRG [17]
DRV0 = PL_GPIO1_PAD_PRG [16]
SL = PL_GPIO1_PAD_PRG [15]
PL_GPIO [35:32]:
DRV1 = PL_GPIO1_PAD_PRG [12]
DRV0 = PL_GPIO1_PAD_PRG [11]
SL = PL_GPIO1_PAD_PRG [10]
PL_GPIO [31:28]:
DRV1 = PL_GPIO1_PAD_PRG [7]
DRV0 = PL_GPIO1_PAD_PRG [6]
SL = PL_GPIO1_PAD_PRG [5]
PL_GPIO [27:24]:
DRV1 = PL_GPIO1_PAD_PRG [2]
DRV0 = PL_GPIO1_PAD_PRG [1]
SL = PL_GPIO1_PAD_PRG [0]

- Register PL_GPIO2_PAD_PRG (Address 0xFCA80138).

PL_GPIO [71:68]:
DRV1 = PL_GPIO2_PAD_PRG[27]
DRV0 = PL_GPIO2_PAD_PRG[26]
SL = PL_GPIO2_PAD_PRG[25]
PL_GPIO [67:64]:
DRV1 = PL_GPIO2_PAD_PRG[22]
DRV0 = PL_GPIO2_PAD_PRG[21]
SL = PL_GPIO2_PAD_PRG[20]
PL_GPIO [63:60]:
DRV1 = PL_GPIO2_PAD_PRG[17]
DRV0 = PL_GPIO2_PAD_PRG[16]
SL = PL_GPIO2_PAD_PRG[15]
PL_GPIO [59:56]:
DRV1 = PL_GPIO2_PAD_PRG[12]
DRV0 = PL_GPIO2_PAD_PRG[11]
SL = PL_GPIO2_PAD_PRG[10]
PL_GPIO [55:52]:
DRV1 = PL_GPIO2_PAD_PRG[7]
DRV0 = PL_GPIO2_PAD_PRG[6]
SL = PL_GPIO2_PAD_PRG[5]
PL_GPIO [51:48]:

DRV1 = PL_GPIO2_PAD_PRG[2]
DRV0 = PL_GPIO2_PAD_PRG[1]
SL = PL_GPIO2_PAD_PRG[0]

- Register PL_GPIO3_PAD_PRG (Address 0xFCA8013C).

PL_GPIO [95:92]:
DRV1 = PL_GPIO3_PAD_PRG[27]
DRV0 = PL_GPIO3_PAD_PRG[26]
SL = PL_GPIO3_PAD_PRG[25]

PL_GPIO [91:88]:
DRV1 = PL_GPIO3_PAD_PRG[22]
DRV0 = PL_GPIO3_PAD_PRG[21]
SL = PL_GPIO3_PAD_PRG[20]

PL_GPIO [87:84]:
DRV1 = PL_GPIO3_PAD_PRG[17]
DRV0 = PL_GPIO3_PAD_PRG[16]
SL = PL_GPIO3_PAD_PRG[15]

PL_GPIO [83:80]:
DRV1 = PL_GPIO3_PAD_PRG[12]
DRV0 = PL_GPIO3_PAD_PRG[11]
SL = PL_GPIO3_PAD_PRG[10]

PL_GPIO [79:76]:
DRV1 = PL_GPIO3_PAD_PRG[7]
DRV0 = PL_GPIO3_PAD_PRG[6]
SL = PL_GPIO3_PAD_PRG[5]

PL_GPIO [75:72]:
DRV1 = PL_GPIO3_PAD_PRG[2]
DRV0 = PL_GPIO3_PAD_PRG[1]
SL = PL_GPIO3_PAD_PRG[0]

- Register PL_GPIO4_PAD_PRG (Address 0xFCA80140).

PL_CLK [4]:
DRV1 = PL_GPIO3_PAD_PRG[22]
DRV0 = PL_GPIO3_PAD_PRG[21]
SL = PL_GPIO3_PAD_PRG[20]

PL_CLK [3]:
DRV1 = PL_GPIO3_PAD_PRG[17]
DRV0 = PL_GPIO3_PAD_PRG[16]
SL = PL_GPIO3_PAD_PRG[15]

PL_CLK [2]:
DRV1 = PL_GPIO3_PAD_PRG[12]
DRV0 = PL_GPIO3_PAD_PRG[11]
SL = PL_GPIO3_PAD_PRG[10]

PL_CLK [1]:

DRV1 = PL_GPIO3_PAD_PRG[7]
DRV0 = PL_GPIO3_PAD_PRG[6]
SL = PL_GPIO3_PAD_PRG[5]

2 DDR IBIS model selection

The I/O signals of the DDR interface form three main groups.

Each group has one set of models associated.

Please refer to the following table for the I/O signals and the associated model name.

Table 2. Relation between the DDR block signals and the used model

DDR block signal name	IBIS model group name
DDR_MEM_CLKP & DDR_MEM_CLKN DDR_MEM_DQS_0 & nDDR_MEM_DQS_0 DDR_MEM_DQS_1 & nDDR_MEM_DQS_1	BDCLK_R_IN_xxxxxxxx xxxxxxxx = DDR PRGA PRGB ZOUT MDZI ODTA ODTB ODEN
DDR_MEM_ADD[14,12,10,8,6,4,2,0] DDR_MEM_DQ[15,13,11,9,6,4,2,0] DDR_MEM_GATE_0 DDR_MEM_DM_1 DDR_MEM_CLKEN DDR_MEM_CS_1 DDR_MEM_ODT_1 DDR_MEM_BA_1 DDR_MEM_CAS	BDRES_GNDE_xxxxxxx xxxxxxxx = ZOUT PROGA PROGB MDZI DDR ODEN ODTA ODTB
DDR_MEM_ADD[13,11,9,7,5,3,1] DDR_MEM_DQ[14,12,10,8,7,5,3,1] DDR_MEM_GATE_1 DDR_MEM_DM_0 DDR_MEM_CS_0 DDR_MEM_ODT_0 DDR_MEM_BA_0 DDR_MEM_BA_2 DDR_MEM_RAS DDR_MEM_WE	BDRES_VDDE_xxxxxxx xxxxxxxx = ZOUT PROGA PROGB MDZI DDR ODEN ODTA ODTB

Each model group name contains a set of models depending upon the value of the 8 bits xxxxxxxx for the selected DDR_MEM_* signal.

Please refer to the SPEAr3xx user manual for the following register name:

- DDR_PAD
- MEMCTL_AHB_SET_02

The correlation between bits 0-8 of the model group names and the bits of each register are listed below:

- Register DDR_PAD (Address 0xFCA800F0).

  DDR = not (DDR_PAD [14]) = not (DDR_EN_PAD). (0 = DDR2 interface)

  The read value DDR_PAD [14] must be inverted.

  PRGA = PROGA = DDR_PAD [2] = PROG_a.

  PRGB = PROGB = DDR_PAD [1] = PROG_b.

  (PROG_a =1 & PROG_b =0 => 333[Mhz])

  ZOUT = DDR_PAD [3] = S_W_mode (0 = Strong drive strength)

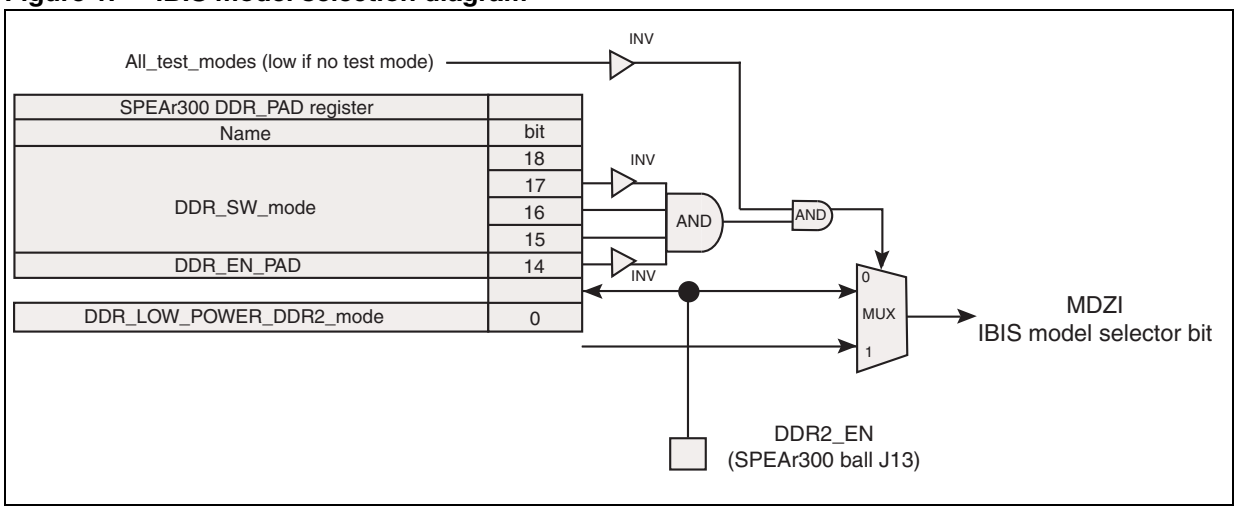
  MDZI = See the schematic in *Figure 1*:

In *Figure 1*:

MDZI = 0 -> pseudo-differential 1.8 V receiver for DDRII mode.

MDZI = 1 -> 1.8V pure digital CMOS receiver for DDRII mode

Figure 1. IBIS model selection diagram



- Register MEMCTL_AHB_SET_02 [1:0] (Address 0xFC60002C)

MEMCTL_AHB_SET_02 [rtt_pad_termination]

Table 3. Correlation between bit 0 and bit 1 of the register MEM_CTL_AHB_SET_02

MEMCTL_AHB_SET_02[1]	MEMCTL_AHB_SET_02[0]	ODTEN	ODTA	ODTB	Description
0	0	0 1	1 0	1 0	disabled
0	1	1	1	1	75 Ohm
1	0	1 1	1 0	0 1	150 Ohm
1	1	Reserved	Reserved	Reserved	Reserved

Note: The signals ODTEN, ODTA, ODTB (Table 3) can be used for the PCB simulation of the following signals: DDR_MEM_DQS_0, nDDR_MEM_DQS_0, DDR_MEM_DQS_1,

nDDR_MEM_DQS_1, DDR_MEM_DQ [15:0]. For all the other signals ODTEN = 0, ODTA = 1, ODTB = 1.

2.1 Example

The setting of the registers DDR_PAD and MEMCTL_AHB_SET_02 is normally done at each system boot-up by the XLOADER part of the boot code.

Let's assume, as an example, that at the end of the boot operation the application reads these two registers and finds the following values:

- Register DDR_PAD (Address 0xFCA800F0) = 0x00003AA5.
- Register MEMCTL_AHB_SET_02 Register (Address 0xFC60002C) = 0x05000202.

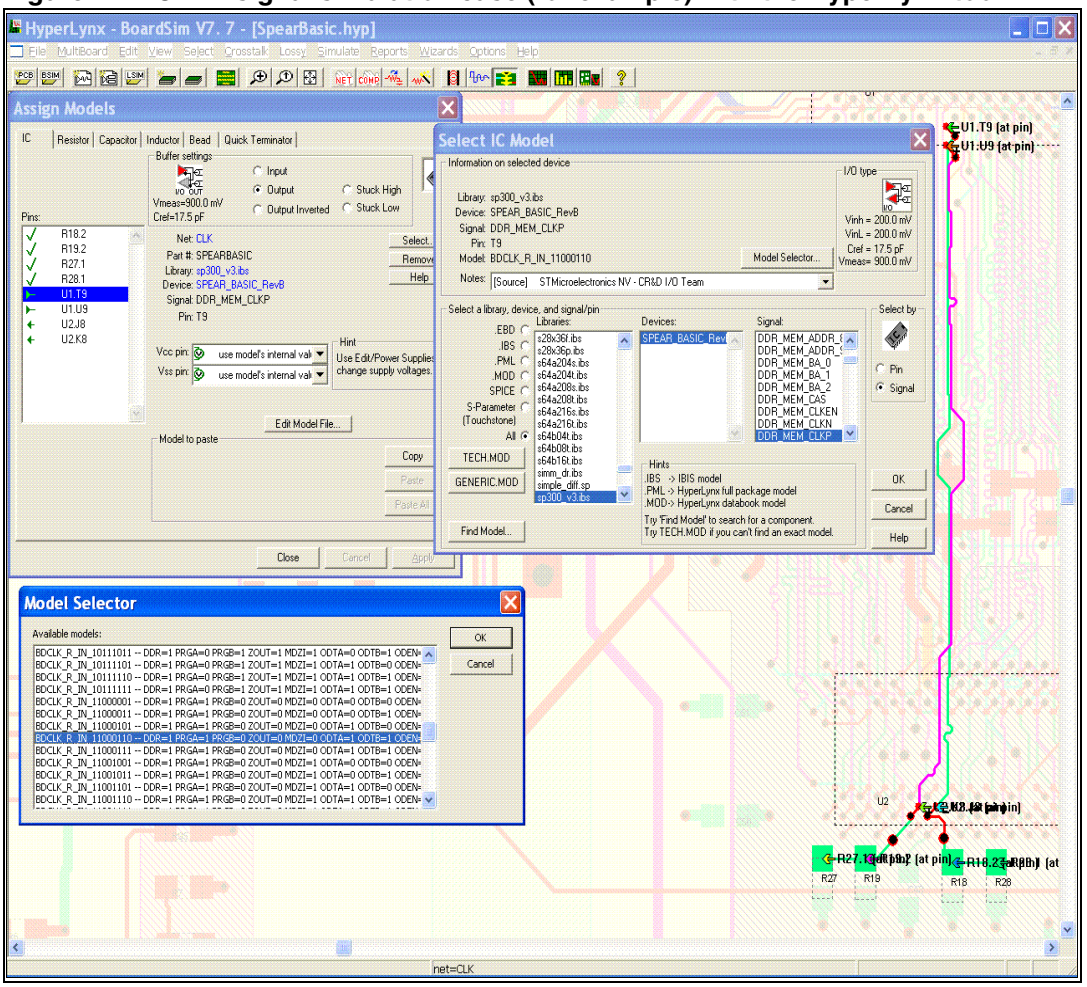
The correct model to be selected from the library with the above values is the following:

Table 4. Relation between the DDR clock signals and the used model

DDR block signal name	IBIS model used for application board
DDR_MEM_CLKP & DDR_MEM_CLKN	BDCLK_R_IN_xxxxxxxx = BDCLK_R_IN_**1**100**0110**
DDR_MEM_DQS_0 & nDDR_MEM_DQS_0 DDR_MEM_DQS_1 & nDDR_MEM_DQS_1	If READ => BDCLK_R_IN_xxxxxxxx = BDCLK_R_IN_**1**100**0**101 If WRITE => BDCLK_R_IN_xxxxxxxx = BDCLK_R_IN_**1**100**0110**
DDR_MEM_DQ[15,13,11,9,6,4,2,0]	If READ => BDRES_GNDE_xxxxxxx = BDRES_GNDE_010**01**110 If WRITE => BDRES_GNDE_xxxxxxx = BDRES_GNDE_010**01011**
DDR_MEM_ADD[14,12,10,8,6,4,2,0] DDR_MEM_GATE_0 DDR_MEM_DM_1 DDR_MEM_CLKEN DDR_MEM_CS_1 DDR_MEM_ODT_1 DDR_MEM_BA_1 DDR_MEM_CAS	BDRES_GNDE_xxxxxxx = BDRES_GNDE_010**01011**
DDR_MEM_DQ[14,12,10,8,7,5,3,1]	If READ => BDRES_VDDE_xxxxxxx = BDRES_VDDE_010**01**110 If WRITE => BDRES_VDDE_xxxxxxx = BDRES_VDDE_010**01011**
DDR_MEM_ADD[13,11,9,7,5,3,1] DDR_MEM_GATE_1 DDR_MEM_DM_0 DDR_MEM_CS_0 DDR_MEM_ODT_0 DDR_MEM_BA_0 DDR_MEM_BA_2 DDR_MEM_RAS DDR_MEM_WE	BDRES_VDDE_xxxxxxx = BDRES_VDDE_010**01011**

*Note: In **Bold** the fixed bx bit values for the STM DDR2 application board.*

Figure 2. CLKP signal simulation case (for example) with the HyperLynx tool



As shown in *Figure 2*, if the simulation tool supports the model selector function, it's pretty easy to select the right model. When you select the pin name of the package (ball name) to be used in the simulation, the tool points automatically to the signal name and shows the model group name, listing in a window all the models contained in the group. As you can see in the model selector windows, all the models with all the available combination of the 7 or 8 bits are presented.

Knowing the operational mode or the content of the two registers explained in the previous paragraph, you can select the right model to be used by the simulation. In this specific example the selected model is BDCLK_R_IN_11000110.

There are simulation tools that do not support the model selector function. In this case, you must probably manually remove from the IBIS model library all the models with the bit settings that are not used in the simulation, only leaving in the library the models with the combination of bits related to the used operating mode.

3 USB IBIS model selection

All the I/O signals of the USB interface are grouped in one main group.

Please refer to the following table for the I/O signals and the associated model name.

Table 5. Relation between the USB block signals and the used model

GMAC block signal name	IBIS model group name
USB_DEVICE_DP USB_DEVICE_DM USB_HOST0_DP USB_HOST0_DM USB_HOST1_DP USB_HOST1_DM	usb_tx

Caution: The USB IBIS model of the library sp300_v3.ibs is for the typical case only.

4 Revision history

Table 6. Document revision history

Date	Revision	Changes
04-Dec-2009	1	Initial release.