LatticeECP/EC and LatticeXP sysIO Usage Guide

September 2012 Technical Note TN1056

Introduction

The LatticeECP™, LatticeEC™ and LatticeXP™ sysIO™ buffers give the designer the ability to easily interface with other devices using advanced system I/O standards. This technical note describes the sysIO standards available and how they can be implemented using Lattice's design software.

sysIO Buffer Overview

The LatticeECP/EC and LatticeXP sysIO interfaces contain multiple Programmable I/O Cells (PIC) blocks. In the case of the LatticeEC and LatticeECP devices, each PIC contains two Programmable I/Os (PIO), PIOA and PIOB, connected to their respective sysIO buffers. In the LatticeXP device, each PIC also contains two PIOs, PIOA and PIOB, but every fourth PIC will have only PIOA. Two adjacent PIOs can be joined to provide a differential I/O pair (labeled as "T" and "C").

Each Programmable I/O (PIO) includes a sysIO Buffer and I/O Logic (IOLOGIC). The LatticeECP/EC and LatticeXP sysIO buffers support a variety of single-ended and differential signaling standards. The sysIO buffer also supports the DQS strobe signal that is required for interfacing with the DDR memory. One of every 16 PIOs in the LatticeECP/EC and one of every 14 PIOs in the case of the LatticeXP contains a delay element to facilitate the generation of DQS signals. The DQS signal from the bus is used to strobe the DDR data from the memory into input register blocks. For more information on the architecture of the sysIO buffer, please refer to the device data sheets.

The IOLOGIC includes input, output and tristate registers that implement both single data rate (SDR) and double data rate (DDR) applications along with the necessary clock and data selection logic. Programmable delay lines and dedicated logic within the IOLOGIC are used to provide the required shift to incoming clock and data signals and the delay required by DQS inputs in DDR memory. The DDR implementation in the IOLOGIC and the DDR memory interface support are discussed in more details in Lattice technical note number TN1050, *LatticeECP/EC DDR Usage Guide.*

Supported sysIO Standards

The LatticeECP/EC and LatticeXP sysIO buffer supports both single-ended and differential standards. Single-ended standards can be further subdivided into LVCMOS, LVTTL, PCI and other standards. The buffers support the LVTTL, LVCMOS 1.2, 1.5, 1.8, 2.5 and 3.3V standards. In the LVCMOS and LVTTL modes, the buffer has individually configurable options for drive strength, bus maintenance (weak pull-up, weak pull-down, or a bus-keeper latch). Other single-ended standards supported include SSTL and HSTL. Differential standards supported include LVDS, RSDS, BLVDS, LVPECL, differential SSTL and differential HSTL. Table 8-1 lists the sysIO standards supported in the Lattice EC/ECP and LatticeXP devices.

Table 8-1. Supported sysIO Standards

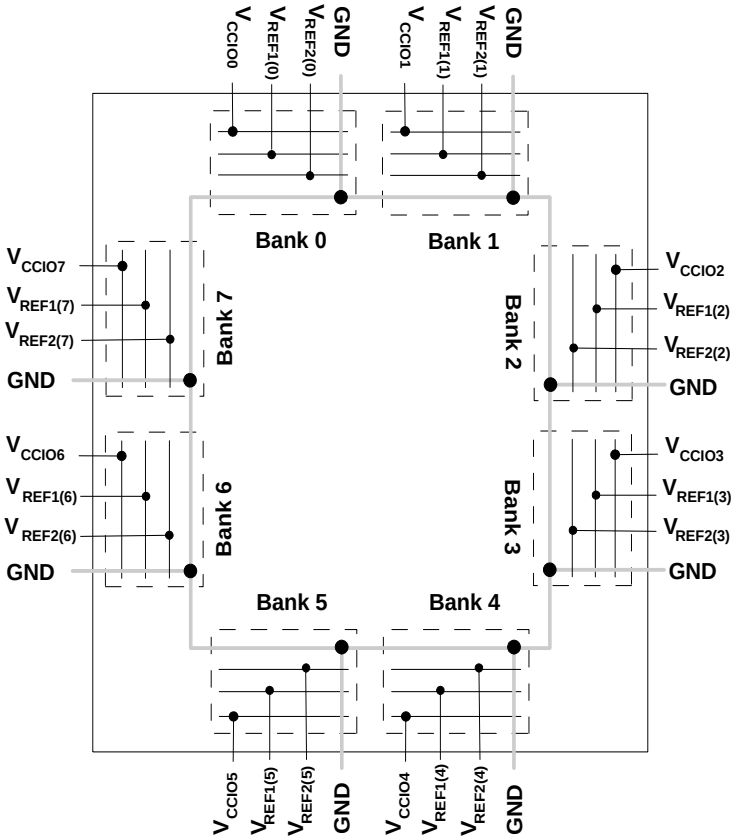
Standard	V_{CCIO}			V_{REF} (V)		
	Min.	Typ.	Max.	Min.	Typ.	Max.
LVCMOS 3.3	3.135	3.3	3.465	—	—	—
LVCMOS 2.5	2.375	2.5	2.625	—	—	—
LVCMOS 1.8	1.71	1.8	1.89	—	—	—
LVCMOS 1.5	1.425	1.5	1.575	—	—	—
LVCMOS 1.2	1.14	1.2	1.26	—	—	—
LVTTL	3.135	3.3	3.465	—	—	—
PCI	3.135	3.3	3.465	—	—	—
SSTL18 Class I	1.71	2.5	1.89	0.833	0.9	0.969
SSTL2 Class I, II	2.375	2.5	2.625	1.15	1.25	1.35
SSTL3 Class I, II	3.135	3.3	3.465	1.3	1.5	1.7
HSTL15 Class I	1.425	1.5	1.575	0.68	0.75	0.9
HSTL15 Class III	1.425	1.5	1.575	—	0.9	—
HSTL 18 Class I, II	1.71	1.8	1.89	—	0.9	—
HSTL 18 Class III	1.71	1.8	1.89	—	1.08	—
LVDS	2.375	2.5	2.625	—	—	—
LVPECL[1]	3.135	3.3	3.465	—	—	—
BLVDS[1]	2.375	2.5	2.625	—	—	—
RSDS[1]	2.375	2.5	2.625	—	—	—

1. Inputs on chip. Outputs are implemented with the addition of external resistors.

sysIO Banking Scheme

LatticeECP/EC and LatticeXP devices have eight programmable sysIO banks, two per side. Each sysIO bank has a V_{CCIO} supply voltage and two reference voltages, V_{REF1} and V_{REF2}. On the top and bottom banks, the sysIO buffer pair consists of two single-ended output drivers and two sets of single-ended input buffers (both ratioed and referenced). The left and right side sysIO buffer pair along with the two single-ended output and input drivers will also have a differential driver. The referenced input buffer can also be configured as a differential input. The two pads in the pair are described as “true” and “comp”, where the true pad is associated with the positive side of the differential input buffer and the comp (complementary) pad is associated with the negative side of the differential input buffer. Figure 8-1 shows the eight banks and their associated supplies.

Figure 8-1. sysIO Banking



V_{CCIO} (1.2V/1.5V/1.8V/2.5V/3.3V)

Each bank has a separate V_{CCIO} supply that powers the single-ended output drivers and the ratioed input buffers such as LVTTL, LVCMOS, and PCI. LVTTL, LVCMOS3.3, LVCMOS2.5 and LVCMOS1.2 also have fixed threshold options allowing them to be placed in any bank. The VCCIO voltage applied to the bank determines the ratioed input standards that can be supported in that bank. It is also used to power the differential output drivers.

V_{CCAUX} (3.3V)

In addition to the bank V_{CCIO} supplies, devices have a V_{CC} core logic power supply, and a V_{CCAUX} auxiliary supply that powers the differential and referenced input buffers. V_{CCAUX} is required because V_{CC} does not have enough headroom to satisfy the common-mode range requirements of these drivers and input buffers.

V_{CCJ} (1.2V/1.5V/1.8V/2.5V/3.3V)

The JTAG pins have a separate V_{CCJ} power supply that is independent of the bank V_{CCIO} supplies. V_{CCJ} determines the electrical characteristics of the LVCMOS JTAG pins, both the output high level and the input threshold.

Input Reference Voltage (V_{REF1}, V_{REF2})

Each bank can support up to two separate V_{REF} input voltages, V_{REF1} and V_{REF2}, that are used to set the threshold for the referenced input buffers. The location of these V_{REF} pins is pre-determined within the bank. These pins can be used as regular I/Os if the bank does not require a V_{REF} voltage.

V_{REF1} for DDR Memory Interface

When interfacing to DDR memory, the V_{REF1} input must be used as the reference voltage for the DQS and DQ input from the memory. A voltage divider between V_{REF1} and GND is used to generate an on-chip reference volt-

age that is used by the DQS transition detector circuit. This voltage divider is only present on V_{REF1} it is not available on $V_{REF2.}$ For more information on the DQS transition detect logic and its implementation please refer to Lattice technical note number TN1050, *LatticeECP/EC DDR Usage Guide.*

Mixed Voltage Support in a Bank

The LatticeECP/EC and LatticeXP sysIO buffer is connected to three parallel ratioed input buffers. These three parallel buffers are connected to $V_{CCIO,}$ V_{CCAUX} and to V_{CC} giving support for thresholds that track with V_{CCIO} as well as fixed thresholds for 3.3V (V_{CCAUX}) and 1.2V (V_{CC}) inputs. This allows the input threshold for ratioed buffers to be assigned on a pin-by-pin basis, rather than tracking it with $V_{CCIO.}$ This option is available for all 1.2V, 2.5V and 3.3V ratioed inputs and is independent of the bank V_{CCIO} voltage. For example, if the bank V_{CCIO} is 1.8V, it is possible to have 1.2V and 3.3V ratioed input buffers with fixed thresholds, as well as 2.5V ratioed inputs with tracking thresholds.

Prior to device configuration, the ratioed input thresholds always track the bank $V_{CCIO,}$ this option only takes effect after configuration. Output standards within a bank are always set by $V_{CCIO.}$ Table 8-2 shows the sysIO standards that the user can mix in the same bank.

Table 8-2. Mixed Voltage Support

	Input sysIO Standards					Output sysIO Standards				
V_{CCIO}	1.2V	1.5V	1.8V	2.5V	3.3V	1.2V	1.5V	1.8V	2.5V	3.3V
1.2V	Yes			Yes	Yes	Yes				
1.5V	Yes	Yes		Yes	Yes		Yes			
1.8V	Yes		Yes	Yes	Yes			Yes		
2.5V	Yes			Yes	Yes				Yes	
3.3V	Yes			Yes	Yes					Yes

sysIO Standards Supported in Each Bank

Table 8-3. I/O Standards Supported by Various Banks

Description	Top Side Banks 0-1	Right Side Banks 2-3	Bottom Side Banks 4-5	Left Side Banks 6-7
Types of I/O Buffers	Single-ended	Single-ended and Differential	Single-ended	Single-ended and Differential
Output standards supported	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I SSTL25 Class I, II SSTL33 Class I, II HSTL15 Class I, III HSTL18_I, II, III SSTL18D Class I, SSTL25D Class I, II SSTL33D Class I, II HSTL15D Class I, III, HSTL18D Class I, III PCI33 LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I SSTL25 Class I, II SSTL33 Class I, II HSTL15 Class I, III HSTL18 Class I, II, III SSTL18D Class I, SSTL25D Class I, II SSTL33D Class I, II HSTL15D Class I, III HSTL18D Class I, III PCI33 LVDS LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I SSTL2 Class I, II SSTL3 Class I, II HSTL15 Class I, III HSTL18 Class I, II, III SSTL18D Class I, SSTL25D Class I, II, SSTL33D Class I, II HSTL15D Class I, III HSTL18D Class I, III PCI33 LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I SSTL2 Class I, II SSTL3 Class I, II HSTL15 Class I, III HSTL18 Class I, II, III SSTL18D Class I, SSTL25D Class I, II, SSTL33D_I, II HSTL15D Class I, III HSTL18D Class I, III PCI33 LVDS LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]
Inputs	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential
Clock Inputs	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential
PCI Support	PCI33 with clamp	PCI33 no clamp	PCI33 with clamp	PCI no clamp
LVDS Output Buffers		LVDS (3.5mA) Buffers		LVDS (3.5mA) Buffers

1. These differential standards are implemented by using complementary LVCMOS driver with external resistor pack.

LVCMOS Buffer Configurations

All LVCMOS buffers have programmable pull, programmable drive and programmable slew configurations that can be set in the software.

Programmable Pull-up/Pull-Down/Buskeeper

When configured as LVCMOS or LVTTL, each sysIO buffer has a weak pull-up, a weak pull-down resistor and a weak buskeeper (bus hold latch) available. Each I/O can independently be configured to have one of these features or none of them.

Programmable Drive

Each LVCMOS or LVTTL output buffer pin has a programmable drive strength option. This option can be set for each I/O independently. The drive strength setting available are 2mA, 4mA, 6mA, 8mA, 12mA, 16mA and 20mA. Actual options available vary by the I/O voltage. The user must consider the maximum allowable current per bank and the package thermal limit current when selecting the drive strength.

The programmable drive feature also allows the user to match to the impedance of the transmission line.

Table 8-4 shows the drive current setting required to match 50¾ transmission line with 50¾ and 200¾ terminations.

Table 8-4. Impedance Matching Using Programmable Drive Strength

50¾ Transmission Line Termination (¾)	I/O Standard	Drive Strength (mA)
200	LVCMOS18	8
	LVCMOS33	12
50	LVCMOS18	16
	LVCMOS33	20

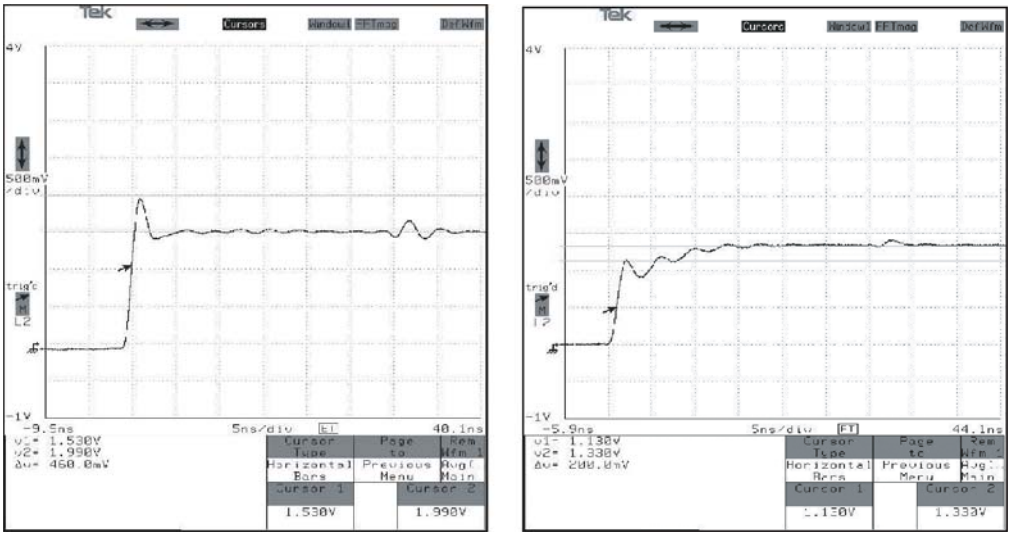
The actual impedance matching may vary on the transmission line design and the load. To find the best matching, it is recommended to drive the transmission line with different combinations of I/O standards and drive strengths that best match the line impedance. Lattice provides IBIS buffer models for the users to further analyze the impedance matching.

The figure below shows how this impedance matching is done for a 50¾ transmission line with 200¾ termination using LVCMOS18 I/O buffers programmed to drive 16mA, 12mA, 8mA and 4mA. From this experiment it is empirical that the best matching is achieved with the 8mA drive setting.

Figure 8-2. Impedance Matching for a 50¾ Transmission Line with 200¾ Termination

LVCMOS18 16mA **LVCMOS18 12mA**

Figure 7-2. Impedance Matching for a 50¾ Transmission Line with 200¾ Termination (Cont.)



LVCMOS18 8mA LVCMOS18 4mA

Programmable Slew Rate

Each LVCMOS or LVTTL output buffer pin also has a programmable output slew rate control that can be configured for either low noise or high-speed performance. Each I/O pin has an individual slew rate control. This allows slew rate control to be specified on pin-by-pin basis. This slew rate control affects both the rising edges and the falling edges.

Open Drain Control

All LVCMOS and LVTTL output buffers can be configured to function as open drain outputs. The user can implement an open drain output by turning on the OPENDRAIN attribute in the software.

The software implements open drain in the LatticeECP/EC and LatticeXP devices by connecting the data and tristate input of the output buffer. Software will implement open drain using this method for simple output buffers. If the user wants to assign open drain functionality to a bidirectional I/O, a similar implementation is required in the HDL design. This can be accomplished by combining the equations for the output enable with the output data. The function of an open drain output is to drive a high Z when the data to the output buffer is driven high and drive a low when the data to the output buffer is driven low.

Differential SSTL and HSTL Support

The single-ended driver associated with the complementary ‘C’ pad can optionally be driven by the complement of the data that drives the single-ended driver associated with the true pad. This allows a pair of single-ended drivers to be used to drive complementary outputs with the lowest possible skew between the signals. This is used for driving complementary SSTL and HSTL signals (as required by the differential SSTL and HSTL clock inputs on synchronous DRAM and synchronous SRAM devices respectively). This capability is also used in conjunction with off-chip resistors to emulate LVPECL and BLVDS output drivers.

PCI Support with Programmable PCICLAMP

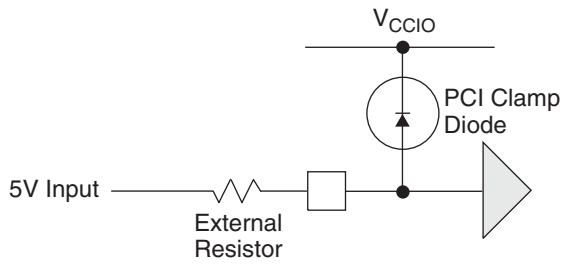
Each sysIO buffer can be configured to support PCI33. The buffers on the top and bottom of the device have an optional PCI clamp diode that may optionally be specified in the ispLEVER® design tool.

The programmable PCICLAMP can be turned ON or OFF. This option is available on each I/O independently on the top and bottom banks.

5V Interface with PCI Clamp Diode

All the I/Os on the top and bottom sides of the device (Banks 0, 1, 4, and 5) have a clamp diode that is used to clamp the voltage at the input to V_{CCIO}. This is especially used for PCI I/O standards. This clamp diode can be used along with an external resistor to make an input 5V tolerant.

Figure 8-3. 5V Tolerant Input Buffer



The value of this external resistor will depend on the PCI clamp diode characteristics. You can find the voltage vs. current data across this diode in the device IBIS model.

In order to interface to 5V input, it is recommended to set the V_{CCIO} between 2.5V to 3.3V.

Below is an example for calculating the value of this external resistor when V_{CCIO} is 2.75V.

- Maximum voltage at input pin, V_{INMAX} = 3.75V (see device data sheet for more details)
- Bank V_{CCIO} = 2.75V
- Maximum voltage drop across clamp diode, $V_D = V_{INMAX} - V_{CCIO} = 3.75 - 2.75 = 1V$
- The current across the clamp diode at V_D can be found in the power clamp data of the IBIS file. Below is the power clamp portion of the IBIS file for a LVCMOS3.3 input model with PCI Clamp turned on. When V_D is 1V, the clamp diode current is I_D = 27.4mA.

Table 8-5. Power Clamp Data from IBIS Model

Voltage	I (Max.)	Units
-1.40	72.5	mA
-1.30	61.2	mA
-1.20	49.9	mA
-1.10	38.6	mA
-1.00	27.4	mA
-0.90	16.9	mA
-0.80	9.52	mA
-0.70	5.35	mA
-0.60	2.31	mA
-0.50	550.8	µA
-0.40	58.0	µA
-0.30	3.61	µA
-0.20	0.07917	µA
-0.10	0.0009129	µA
0.00	0.0001432	µA

- Assume the maximum output voltage of the driving device is V_{EXT} = 5.25V. The value of the external resistor can then be calculated as follows:

$$R_{EXT} = (V_{EXT} - V_{INMAX})/I_D = (5.25V - 3.75V)/27.4 = 54.8 \text{ ohm}$$

If the V_{CCIO} of the bank is increased, it will also increase the value of the external resistor required. Changing the bank V_{CCIO} will also change the value of the input threshold voltage.

Programmable Input Delay

Each input can optionally be delayed before it is passed to the core logic or input registers. The primary use for the input delay is to achieve zero hold time for the input registers when using a direct drive primary clock. To arrive at zero hold time, the input delay will delay the data by at least as much as the primary clock injection delay. This option can be turned ON or OFF for each I/O independently in the software using the FIXEDDELAY attribute. This attribute is described in more detail in the Software sysIO Attributes section. Appendix A shows how this feature can be enabled in the software using HDL attributes.

Software sysIO Attributes

sysIO attributes can be specified in the HDL, using the Preference Editor GUI or in the ASCII Preference file (.prf) file directly. Appendices A, B and C list examples of how these can be assigned using each of the methods mentioned above. This section describes in detail each of these attributes.

IO_TYPE

This is used to set the sysIO standard for an I/O. The V_{CCIO} required to set these I/O standards are embedded in the attribute names itself. There is no separate attribute to set the V_{CCIO} requirements. Table 8-6 lists the available I/O types.

Table 8-6. I/O_TYPE Attribute Values

sysIO Signaling Standard	IO_TYPE
DEFAULT (for LatticeECP/EC)	LVCMOS12
DEFAULT (for LatticeXP)	LVCMOS25
LVDS 2.5V	LVDS25
RSDS	RSDS
Emulated LVDS 2.5V	LVDS25E[1]
Bus LVDS 2.5V	BLVDS25[1]
LVPECL 3.3V	LVPECL33[1]
HSTL18 Class I, II and III	HSTL18_I, HSTL18_II, HSTL18_III
Differential HSTL 18 Class I, II and III	HSTL18D_I HSTL18D_II HSTL18D_III
HSTL 15 Class I and III	HSTL15_I HSTL15_III
Differential HSTL 15 Class I and III	HSTL15D_I HSTL15D_III
SSTL 33 Class I and II	SSTL33_I, SSTL33_II
Differential SSTL 33 Class I and II	SSTL33D_I SSTL3D_II
SSTL 25 Class I and II	SSTL25_I SSTL25_II
Differential SSTL 25 Class I and II	SSTL25D_I SSTL25D_II
SSTL 18 Class I	SSTL18_I
Differential SSTL 18 Class I	SSTL18D_I
LVTTL	LVTTL33
3.3V LVCMOS	LVCMOS33
2.5V LVCMOS	LVCMOS25
1.8V LVCMOS	LVCMOS18
1.5V LVCMOS	LVCMOS15
1.2V LVCMOS	LVCMOS12
3.3V PCI	PCI33

1. These differential standards are implemented by using complementary LVCMOS driver with external resistor pack.

OPENDRAIN

LVCMOS and LVTTL I/O standards can be set to Open Drain configuration by using the OPENDRAIN attribute.

Values: ON, OFF
Default: OFF

DRIVE

The drive strength attribute is available for LVTTL and LVCMOS output standards. These can be set or each I/O pin individually.

Values: NA, 2, 4, 8, 12, 16, 20
LatticeECP/EC Default: 6
LatticeXP Default: 8

The programmable drive available on a pad will depend on the $V_{CCIO.}$ Table 8-7 shows the drive strength available for different $V_{CCIO.}$

Table 8-7. Programmable Drive Strength Values at Various V_{CCIO} Voltages

Drive	V_{CCIO}				
	1.2 V	1.5 V	1.8 V	2.5 V	3.3 V
2	X				
4		X	X	X	X
6	X				
8		X	X	X	X
12			X	X	X
16			X	X	X
20				X	X

PULLMODE

The PULLMODE attribute is available for all the LVTLL and LVCMOS inputs and outputs. This attribute can be enabled for each I/O independently.

Values: UP, DOWN, NONE, KEEPER
Default: UP

PCICLAMP

PCI33 inputs and outputs on the top and bottom of the device have an optional PCI clamp that is enabled via the PCICLAMP attribute. The PCICLAMP is also available for all LVCMOS33 and LVTTL inputs and outputs.

Values: ON, OFF
Default: OFF

SLEWRATE

The SLEWRATE attribute is available for all LVTTL and LVCMOS output drivers. Each I/O pin has an individual slew rate control. This allows the designer to specify the slew rate control on a pin-by-pin basis.

Values: FAST, SLOW
Default: FAST

FIXEDDELAY

The FIXEDDELAY attribute is available to each input pin. When enabled, this attribute is used to achieve zero hold time for the input registers when using global clock.

Values: TRUE, FALSE
Default: FALSE

DIN/DOUT

This attribute can be used when I/O registers need to be assigned. Using DIN will assert an input register and using the DOUT attribute will assert an output register in the design. By default the software will try to assign the I/O registers if applicable. The user can turn this OFF by using the synthesis attribute or using the preference editor of the software. These attributes can only be applied on registers.

LOC

This attribute can be used to make pin assignments to the I/O ports in the design. This attribute is only used when the pin assignments are made in HDL source. Pins assignments can be made directly using the GUI in the Preference Editor of the software. The appendices explain this in more detail.

Design Considerations and Usage

This section discusses some of design rules and considerations that need to be taken into account when designing with the LatticeECP/ECP and LatticeXP sysIO buffer.

Banking Rules

- If V_{CCIO} or V_{CCJ} for any bank is set to 3.3V, it is recommended that it be connected to the same power supply as $V_{CCAUX,}$ thus minimizing leakage.
- If V_{CCIO} or V_{CCJ} for any bank is set to 1.2V, it is recommended that it be connected to the same power supply as $V_{CC,}$ thus minimizing leakage.
- When implementing DDR memory interfaces, the V_{REF1} of the bank is used to provide reference to the interface pins and cannot be used to power any other referenced inputs.
- Only the top and bottom banks (Banks 0, 1, 4, and 5) will support PCI clamps. The left and right side (Banks 2, 3, 6 and 7) do not support PCI Clamp, but will support True LVDS output.

Differential I/O Rules

- All the banks can support LVDS input buffers. Only the banks on the right and left side (Banks 2, 3, 6 and 7) will support True Differential output buffers. The banks on the top and bottom will support the LVDS input buffers but will not support True LVDS outputs. The user can use emulated LVDS output buffers on these banks.
- All banks support emulated differential buffers using external resistor pack and complementary LVCMOS drivers.
- In LatticeXP devices, not all PIOs have LVDS capability. Only four out of every seven I/Os can provide LVDS buffer capability. In LatticeECP/EC devices, there are no restrictions on the number of I/Os that can support LVDS. In both cases LVDS can only be assigned to the TRUE pad. Refer to the device data sheets to see the pin listing for all the LVDS pairs.

Assigning V_{REF}/ V_{REF} Groups for Referenced Inputs

Each bank has two dedicated V_{REF} input pins, V_{REF1} and $V_{REF2.}$ Buffers can be grouped to a particular V_{REF} rail, V_{REF1} or $V_{REF2.}$ This grouping is done by assigning a PGROUP VREF preference along with the LOCATE PGROUP preference.

Preference Syntax

```
PGROUP <pgrp_name> [(VREF <vref_name>)+] (COMP <comp_name>)+;
LOCATE PGROUP <pgrp_name> BANK <bank_num>;
LOCATE VREF <vref_name> SITE <site_name>;
```

Example of VREF Groups

```
PGROUP "vref_pg1" VREF "ref1" COMP "ah(0)" COMP "ah(1)" COMP "ah(2)" COMP "ah(3)"
COMP "ah(4)" COMP "ah(5)" COMP "ah(6)" COMP "ah(7)";

PGROUP "vref_pg2" VREF "ref2" COMP "al(0)" COMP "al(1)" COMP "al(2)" COMP "al(3)"
COMP "al(4)" COMP "al(5)" COMP "al(6)" COMP "al(7)";

LOCATE VREF "ref1" SITE PR29C;
LOCATE VREF "ref2" SITE PR48B;
```

or

```
LOCATE PGROUP " vref_pg1" BANK 2;
LOCATE PGROUP " vref_pg2" BANK 2;
```

The second example show V_{REF} groups, “vref_pg1” assigned to V_{REF} “ref1” and “vref_pg2” assigned to “ref2”. V_{REF} must then be locked to either V_{REF1} or V_{REF2} using LOCATE preference. Or, the user can simply designate to which bank V_{REF} group should be located. The software will then assign these to either V_{REF1} or V_{REF2} of the bank.

If the PGROUP VREF is not used, the software will automatically group all pins that need the same V_{REF} reference voltage. This preference is most useful when there is more than one bus using the same reference voltage and the user wants to associate each of these buses to different V_{REF} resources.

Differential I/O Implementation

The LatticeECP/EC and LatticeXP devices support a variety of differential standards as detailed in the following section.

LVDS

True LVDS (LVDS25) drivers are available on the left and right side of the devices. LVDS input support is provided on all sides of the device. All four sides support LVDS using complementary LVCMOS drivers with external resistors (LVDS25E).

Please refer to the LatticeECP/EC and LatticeXP data sheets for a more detailed explanation of these LVDS implementations.

BLVDS

All single-ended sysIO buffer pairs in the LatticeECP family support the Bus-LVDS standard using complementary LVCMOS drivers with external resistors.

Please refer to the LatticeECP/EC and LatticeXP data sheets to learn more about BLVDS implementation.

RSDS

All single-ended sysIO buffers pairs in the LatticeECP family support the RSDS standard using complementary LVCMOS drivers with external resistors. This mode uses LVDS25E with an alternative resistor pack.

Please refer to the LatticeECP/EC and LatticeXP data sheets for a detailed explanation of RSDS implementation.

LVPECL

All the sysIO buffers will support LVPECL inputs. LVPECL outputs are supported using a complementary LVCMOS driver with external resistors.

Please refer to the LatticeECP/EC and LatticeXP data sheets for further information on LVPECL implementation.

Differential SSTL and HSTL

All single-ended sysIO buffers pairs in the LatticeECP family support differential SSTL and HSTL. Please refer to the LatticeECP/EC and LatticeXP data sheets for a detailed explanation of Differential HSTL and SSTL implementation.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
—	—	Previous Lattice releases.
September 2012	03.4	Updated document with new corporate logo.

Appendix A. HDL Attributes for Synplify® and Precision® RTL Synthesis

Using these HDL attributes, you can assign sysIO attributes directly in your source. You will need to use the attribute definition and syntax for the synthesis vendor you are planning to use. Below are a list of all the sysIO attributes syntax and examples for Precision RTL Synthesis and Synplify. This section only lists the sysIO buffer attributes for these devices. You can refer to the Precision RTL Synthesis and Synplify user manuals for a complete list of synthesis attributes. These manuals are available through ispLEVER Software Help.

VHDL Synplify/Precision RTL Synthesis

This section lists syntax and examples for all the sysIO attributes in VHDL when using Precision RTL Synthesis or Synplicity synthesis tools.

Syntax

Table 8-8. VHDL Attribute Syntax for Synplify and Precision RTL Synthesis

Attribute	Syntax
IO_TYPE	attribute IO_TYPE: string; attribute IO_TYPE of Pinname: signal is “IO_TYPE Value”;
OPENDRAIN	attribute OPENDRAIN: string; attribute OPENDRAIN of Pinname: signal is “OpenDrain Value”;
DRIVE	attribute DRIVE: string; attribute DRIVE of Pinname: signal is “Drive Value”;
PULLMODE	attribute PULLMODE: string; attribute PULLMODE of Pinname: signal is “Pullmode Value”;
PCICLAMP	attribute PCICLAMP: string; attribute PCICLAMP of Pinname: signal is “PCIClamp Value”;
SLEWRATE	attribute PULLMODE: string; attribute PULLMODE of Pinname: signal is “Slewrate Value”;
FIXEDDELAY	attribute FIXEDDELAY: string; attribute FIXEDDELAY of Pinname: signal is “Fixeddelay Value”;
DIN	attribute DIN: string; attribute DIN of Pinname: signal is “ ”;
DOUT	attribute DOUT: string; attribute DOUT of Pinname: signal is “ ”;
LOC	attribute LOC: string; attribute LOC of Pinname: signal is “pin_locations”;

Examples

IO_TYPE

--*Attribute Declaration*****

ATTRIBUTE IO_TYPE: string;

--*IO_TYPE assignment for I/O Pin*****

ATTRIBUTE IO_TYPE OF portA: SIGNAL IS “PCI33”;

ATTRIBUTE IO_TYPE OF portB: SIGNAL IS “LVCMOS33”;

ATTRIBUTE IO_TYPE OF portC: SIGNAL IS “LVDS25”;

OPENDRAIN

--*Attribute Declaration***

ATTRIBUTE OPENDRAIN: string;

--*DRIVE assignment for I/O Pin***

ATTRIBUTE OPENDRAIN OF portB: SIGNAL IS "ON";

DRIVE
--*Attribute Declaration***

ATTRIBUTE DRIVE: string;

--*DRIVE assignment for I/O Pin***

ATTRIBUTE DRIVE OF portB: SIGNAL IS “20”;

PULLMODE

--*Attribute Declaration***

ATTRIBUTE PULLMODE : string;

--*PULLMODE assignment for I/O Pin***

ATTRIBUTE PULLMODE OF portA: SIGNAL IS "DOWN";

ATTRIBUTE PULLMODE OF portB: SIGNAL IS "UP";

PCICLAMP

--***Attribute Declaration***

ATTRIBUTE PCICLAMP: string;

--***PULLMODE assignment for I/O Pin***

ATTRIBUTE PCICLAMP OF portA: SIGNAL IS "ON";

SLEWRATE
--*Attribute Declaration***

ATTRIBUTE SLEWRATE : string;

--* SLEWRATE assignment for I/O Pin***

ATTRIBUTE SLEWRATE OF portB: SIGNAL IS “FAST”;

FIXEDDELAY
--*Attribute Declaration***

ATTRIBUTE FIXEDDELAY: string;

--* SLEWRATE assignment for I/O Pin***

ATTRIBUTE FIXEDDELAY OF portB: SIGNAL IS “TRUE”;

DIN/DOUT
--*Attribute Declaration****

ATTRIBUTE din : string;

ATTRIBUTE dout : string;

--* din/dout assignment for I/O Pin****

ATTRIBUTE din OF input_vector: SIGNAL IS “ “;

ATTRIBUTE dout OF output_vector: SIGNAL IS “ “;

LOC
--*Attribute Declaration****

ATTRIBUTE LOC : string;

--* LOC assignment for I/O Pin****

ATTRIBUTE LOC OF input_vector: SIGNAL IS “E3,B3,C3 “;

Verilog for Synplify

This section lists syntax and examples for all the sysIO Attributes in Verilog using the Synplify synthesis tool.

Syntax

Table 8-9. Verilog Synplify Attribute Syntax

Attribute	Syntax
IO_TYPE	PinType PinName /* synthesis IO_TYPE=”IO_Type Value”*/;
OPENDRAIN	PinType PinName /* synthesis OPENDRAIN =”OpenDrain Value”*/;
DRIVE	PinType PinName /* synthesis DRIVE=”Drive Value”*/;
PULLMODE	PinType PinName /* synthesis PULLMODE=”Pullmode Value”*/;
PCICLAMP	PinType PinName /* synthesis PCICLAMP =” PCIClamp Value”*/;
SLEWRATE	PinType PinName /* synthesis SLEWRATE=”Slewrate Value”*/;
FIXEDDELAY	PinType PinName /* synthesis FIXEDDELAY=”Fixeddelay Value”*/;
DIN	PinType PinName /* synthesis DIN=” “*/;
DOUT	PinType PinName /* synthesis DOUT=” “*/;
LOC	PinType PinName /* synthesis LOC=”pin_locations “*/;

Examples

//IO_TYPE, PULLMODE, SLEWRATE and DRIVE assignment

output portB /*synthesis IO_TYPE="LVCMOS33" PULLMODE =”UP” SLEWRATE =”FAST”

DRIVE =”20”*/;

output portC /*synthesis IO_TYPE="LVDS25" */;

//OPENDRAIN

output portA /*synthesis OPENDRAIN =”ON”*/;

//PCICLAMP

output portA /*synthesis IO_TYPE="PCI33" PULLMODE =”PCICLAMP”*/;

// Fixeddelay

input load /* synthesis FIXEDDELAY="TRUE" */;

// Place the flip-flops near the load input

input load /* synthesis din=”” */;

// Place the flip-flops near the outload output

output outload /* synthesis dout=”” */;

//I/O pin location

input [3:0] DATA0 /* synthesis loc=”E3,B1,F3”*/;

//Register pin location

reg data_in_ch1_buf_reg3 /* synthesis loc=”R40C47” */;

//Vectored internal bus

reg [3:0] data_in_ch1_reg /*synthesis loc =”R40C47,R40C46,R40C45,R40C44” */;

Verilog for Precision RTL Synthesis

This section lists syntax and examples for all the sysIO Attributes in Verilog using the Precision RTL Synthesis synthesis tool.

Syntax

Table 8-10. Verilog Precision RTL Synthesis Attribute Syntax

ATTRIBUTE	SYNTAX
IO_TYPE	//pragma attribute PinName IO_TYPE IO_TYPE Value
OPENDRAIN	//pragma attribute PinName OPENDRAIN OpenDrain Value
DRIVE	//pragma attribute PinName DRIVE Drive Value
PULLMODE	//pragma attribute PinName IO_TYPE Pullmode Value
PCICLAMP	//pragma attribute PinName PCICLAMP PCIClamp Value
SLEWRATE	//pragma attribute PinName IO_TYPE Slewrate Value
FIXEDDELAY	//pragma attribute PinName IO_TYPE Fixeddelay Value
LOC	//pragma attribute PinName LOC pin_location

Example

//**IO_TYPE ***

//pragma attribute portA IO_TYPE PCI33

//pragma attribute portB IO_TYPE LVCMOS33

//pragma attribute portC IO_TYPE SSTL25_II

//* Opendrain ***

//pragma attribute portB OPENDRAIN ON

//pragma attribute portD OPENDRAIN OFF

//* Drive ***

//pragma attribute portB DRIVE 20

//pragma attribute portD DRIVE 8

//* Pullmode***

//pragma attribute portB PULLMODE UP

//* PCIClamp***

//pragma attribute portB PCICLAMP ON

//* Slewrate ***

//pragma attribute portB SLEWRATE FAST

//pragma attribute portD SLEWRATE SLOW

// ***Fixeddelay***

// pragma attribute load FIXEDDELAY TRUE

//*LOC*****

//pragma attribute portB loc E3

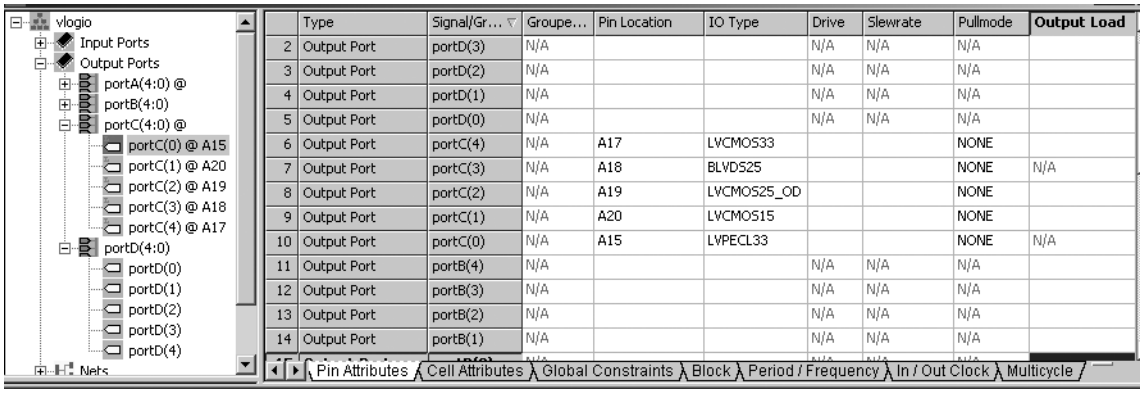
Appendix B. sysIO Attributes Using Preference Editor User Interface

You can also assign the sysIO buffer attributes using the Pre Map Preference Editor GUI available in the ispLEVER tools. The Pin Attribute Sheet list all the ports in your design and all the available sysIO attributes as preferences. Clicking on each of these cells will produce a list of all the valid I/O preference for that port. Each column takes precedence over the next. Hence, when a particular IO_TYPE is chosen, the DRIVE, PULLMODE and SLEWRATE columns will only list the valid combinations for that IO_TYPE. The user can lock the pin locations using the pin location column of the Pin Attribute sheet. Right-clicking on a cell will list all the available pin locations. The Preference Editor will also conduct a DRC check to look for incorrect pin assignments.

You can enter the DIN/ DOUT preferences using the Cell Attributes Sheet of the Preference Editor. All the preferences assigned using the Preference Editor are written into the logical preference file (.lpf).

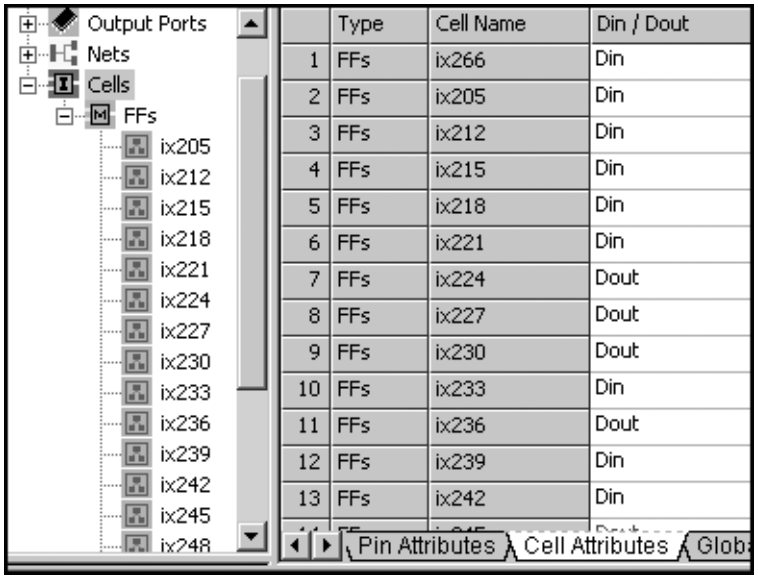
Figure 8-4 and Figure 8-5 show the Pin Attribute Sheet and the Cell Attribute Sheet views of the Preference Editor. For further information on how to use the Preference Editor, refer to the ispLEVER Help documentation located in the Help menu option of the software.

Figure 8-4. Pin Attributes Tab



	Type	Signal/Gr...	Groupe...	Pin Location	IO Type	Drive	Slewrate	Pullmode	Output Load
2	Output Port	portD(3)	N/A			N/A	N/A	N/A	
3	Output Port	portD(2)	N/A			N/A	N/A	N/A	
4	Output Port	portD(1)	N/A			N/A	N/A	N/A	
5	Output Port	portD(0)	N/A			N/A	N/A	N/A	
6	Output Port	portC(4)	N/A	A17	LVCMOS33			NONE	
7	Output Port	portC(3)	N/A	A18	BLVDS25			NONE	N/A
8	Output Port	portC(2)	N/A	A19	LVCMOS25_OD			NONE	
9	Output Port	portC(1)	N/A	A20	LVCMOS15			NONE	
10	Output Port	portC(0)	N/A	A15	LVPECL33			NONE	N/A
11	Output Port	portB(4)	N/A			N/A	N/A	N/A	
12	Output Port	portB(3)	N/A			N/A	N/A	N/A	
13	Output Port	portB(2)	N/A			N/A	N/A	N/A	
14	Output Port	portB(1)	N/A			N/A	N/A	N/A	

Figure 8-5. Cell Attributes Tab



	Type	Cell Name	Din / Dout
1	FFs	ix266	Din
2	FFs	ix205	Din
3	FFs	ix212	Din
4	FFs	ix215	Din
5	FFs	ix218	Din
6	FFs	ix221	Din
7	FFs	ix224	Dout
8	FFs	ix227	Dout
9	FFs	ix230	Dout
10	FFs	ix233	Din
11	FFs	ix236	Dout
12	FFs	ix239	Din
13	FFs	ix242	Din

Appendix C. sysIO Attributes Using Preference File (ASCII File)

You can also enter the sysIO attributes directly in the preference (.prf) file as sysIO buffer preferences. The PRF file is an ASCII file containing two sections: a schematic section for preferences created by the Mapper or translator, and a user section for preferences entered by the user. You can write user preferences directly into this file. The synthesis attributes appear between the schematic start and schematic end of the file. You can enter the sysIO buffer preferences after the schematic end line using the preference file syntax. Below are a list of sysIO buffer preference syntax and examples.

IOBUF

This preference is used to assign the attribute IO_TYPE, PULLMODE, SLEWRATE and DRIVE.

Syntax
IOBUF [ALLPORTS | PORT <port_name> | GROUP <group_name>] (keyword=<value>)+;

where:

<port_name> = These are not the actual top-level port names, but should be the signal name attached to the port. PIOs in the physical design (.ncd) file are named using this convention. Any multiple listings or wildcarding should be done using GROUPs

Keyword = IO_TYPE, OPENDRAIN, DRIVE, PULLMODE, PCICLAMP, SLEWRATE.

Example
IOBUF PORT "port1" IO_TYPE=LVTTL33 OPENDRAIN=ON DRIVE=8 PULLMODE=UP

PCICLAMP =OFF SLEWRATE=FAST;

DEFINE GROUP "bank1" "in*" "out_[0-31]";

IOBUF GROUP "bank1" IO_TYPE=SSTL18_II;

LOCATE

When this preference is applied to a specified component it places the component at a specified site and locks the component to the site. If applied to a specified macro instance it places the macro's reference component at a specified site, places all of the macro's pre-placed components (that is, all components that were placed in the macro's library file) in sites relative to the reference component, and locks all of these placed components at their sites. This can also be applied to a specified PGROUP.

Syntax
LOCATE [COMP <comp_name> | MACRO <macro_name>] SITE <site_name>;

LOCATE PGROUP <pgroup_name> [SITE <site_name>; | REGION <region_name>;]

LOCATE PGROUP <pgroup_name> RANGE <site_1> [<site_2> | <count>] [<direction>] | RANGE <chip_side> [<direction>];

LOCATE BUS < bus_name> ROW|COL <number>;

<bus_name> := string

<number> := integer

Note: If the comp_name, macro_name, or site_name begins with anything other than an alpha character (for example, "11C7"), you must enclose the name in quotes. Wildcard expressions are allowed in <comp_name>.

Example
This command places the port Clk0 on the site A4:

LOCATE COMP “Clk0” SITE “A4”;

This command places the component PFU1 on the site named R1C7:

LOCATE COMP “PFU1” SITE “R1C7”;

This command places bus1 on ROW 3 and bus2 on COL4

LOCATE BUS “bus1” ROW 3;

LOCATE BUS “bus2” COL 4;

USE DIN CELL

This preference specifies the given register to be used as an input Flip Flop.

Syntax
USE DIN CELL <cell_name>;

where:

<cell_name> := string

Example

USE DIN CELL “din0”;

USE DOUT CELL

Specifies the given register to be used as an output Flip Flop.

Syntax
USE DOUT CELL <cell_name>;

where:

<cell_name> := string

Examples
USE DOUT CELL “dout1”;

PGROUP VREF

This preference is used to group all the components that need to be associated to one VREF pin within a bank.

Syntax
PGROUP <pgrp_name> [(VREF <vref_name>)+] (COMP <comp_name>)+;

LOCATE PGROUP <pgrp_name> BANK <bank_num>;

LOCATE VREF <vref_name> SITE <site_name>;

Example
PGROUP “vref_pg1” VREF “ref1” COMP “ah(0)” COMP “ah(1)” COMP “ah(2)” COMP “ah(3)” COMP “ah(4)” COMP “ah(5)” COMP “ah(6)” COMP “ah(7)”;

PGROUP “vref_pg2” VREF “ref2” COMP “al(0)” COMP “al(1)” COMP “al(2)” COMP “al(3)” COMP “al(4)” COMP “al(5)” COMP “al(6)” COMP “al(7)”;

LOCATE VREF “ref1” SITE PR29C;

LOCATE VREF “ref2” SITE PR48B;

or

LOCATE PGROUP “ vref_pg1” BANK 2;

LOCATE PGROUP “ vref_pg2” BANK 2;