

## Introduction

The MachXO™ sysIO™ buffers provide 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 design software.

## sysIO Buffer Overview

The MachXO sysIO interface contains multiple Programmable I/O Cell (PIC) blocks. Each PIC contains two Programmable I/Os (PIOs). Two adjacent PIOs can be joined to provide a differential I/O pair (labeled as “T” and “C”). In the MachXO256, MachXO1200 and MachXO2280 devices, the PIOs are arranged in groups of six (PIOA, PIOB, PIOC, PIOD, PIOE, PIOF) on the top and bottom sides of the device and in groups of four (PIOA, PIOB, PIOC, PIOD) and two (PIOA, PIOB) on the left and right sides of the device. In the MachXO640 device, the PIOs are arranged in groups of six (PIOA, PIOB, PIOC, PIOD, PIOE, PIOF) on the top and bottom sides and in groups of four (PIOA, PIOB, PIOC, PIOD) on the left and right sides of the device.

The larger two devices, MachXO1200 and MachXO2280, support single-ended, differential receiver and differential output sysIO buffers. The two smaller devices, MachXO256 and MachXO640, support single-ended sysIO buffers. For more information on the architecture of the sysIO buffer please refer to the device data sheets.

## Supported sysIO Standards

The MachXO sysIO buffer supports both single-ended and differential standards. Single-ended standards can be further subdivided into LVCMOS, LVTTTL and other standards. The buffers support the LVTTTL, LVCMOS 1.2, 1.5, 1.8, 2.5 and 3.3V standards. In the LVCMOS and LVTTTL modes, the buffer has individually configurable options for drive strength, bus maintenance (weak pull-up, weak pull-down or bus-keeper latch) on I/O buffers. MachXO1200 and MachXO2280 devices also support differential standards like LVDS, RSDS, BLVDS and LVPECL. Table 8-1 lists the sysIO standards supported in the MachXO devices.

**Table 8-1. sysIO Standards Supported**

Standard	V <sub>CCIO</sub> (V)		
	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
LVTTTL	3.135	3.3	3.465
PCI <sup>3</sup>	3.135	3.3	3.465
LVDS <sup>1,2</sup>	2.375	2.5	2.625
LVPECL <sup>1</sup>	3.135	3.3	3.465
BLVDS <sup>1</sup>	2.375	2.5	2.625
RSDS <sup>1</sup>	2.375	2.5	2.625

1. Inputs on chip. Outputs are implemented with the addition of external resistors.
2. MachXO1200 and MachXO2280 devices have dedicated LVDS buffers
3. Input on the top bank of the MachXO1200 and MachXO2280 only.

**sysIO Banking Scheme**

The MachXO family has a non-homogeneous I/O banking structure. The MachXO256 has two I/O banks, the MachXO640 has four I/O banks and the two largest devices, the MachXO1200 and the MachXO2280, have eight I/O banks. The figures below show the banking structures in each of the devices. Each sysIO bank has a  $V_{CCIO}$  supply voltage.

On the MachXO1200 and MachXO2280 devices, the top and bottom banks of the sysIO buffer pair consist of two single-ended output drivers, two single-ended and one differential input buffer. The sysIO buffers on the top side bank also supports PCI buffers. The left and right side sysIO buffer pairs, along with the two single-ended output and input drivers, a differential input and a differential driver on half the I/Os of the bank. On the MachXO640 and MachXO256 devices, all the banks of the sysIO buffer pair consists of two single-ended output drivers (with complementary outputs) and two single-ended and input buffers. All the banks will also support differential output buffers using external resistors. 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.

Figures 1, 2 and 3 show the banking schemes for the MachXO640, MachXO256 and MachXO1200/MachXO2280 devices respectively.

**Figure 8-1. MachXO256 sysIO Banking**

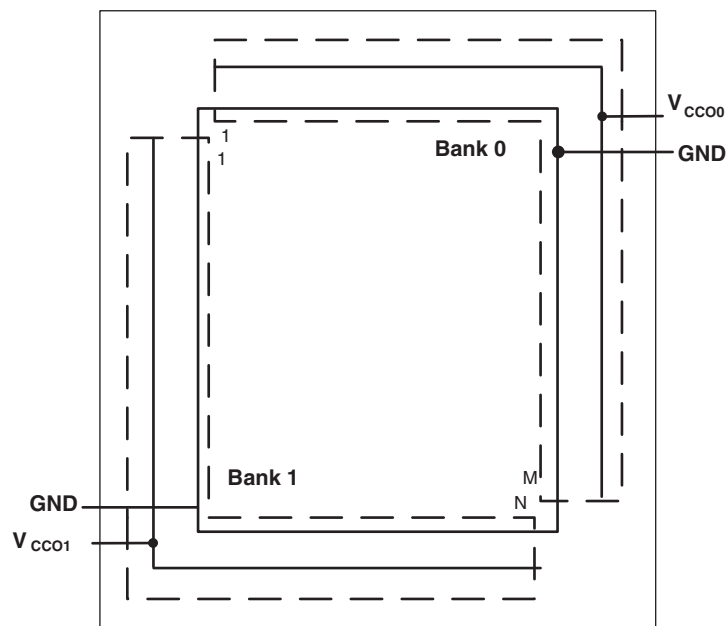


Figure 8-2. MachXO640 sysIO Banking

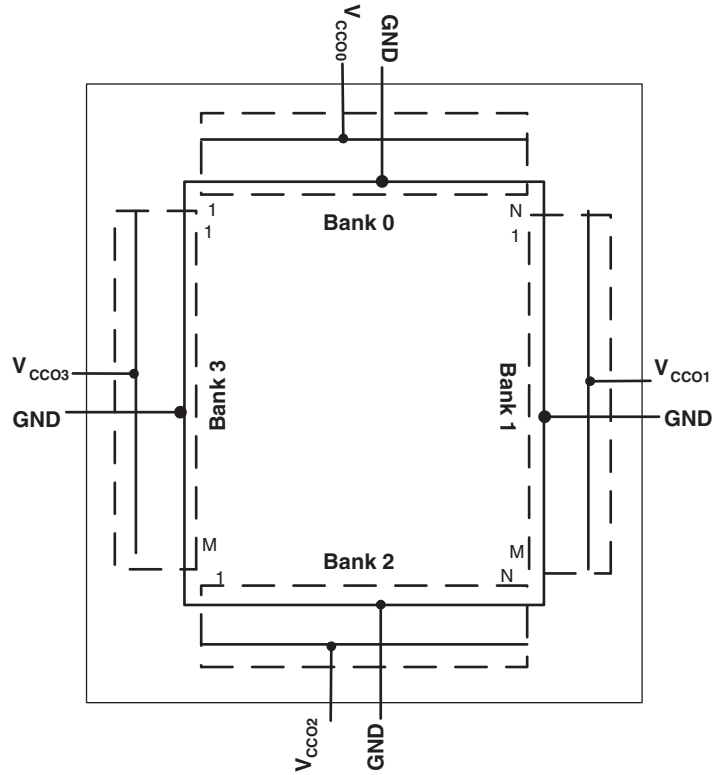
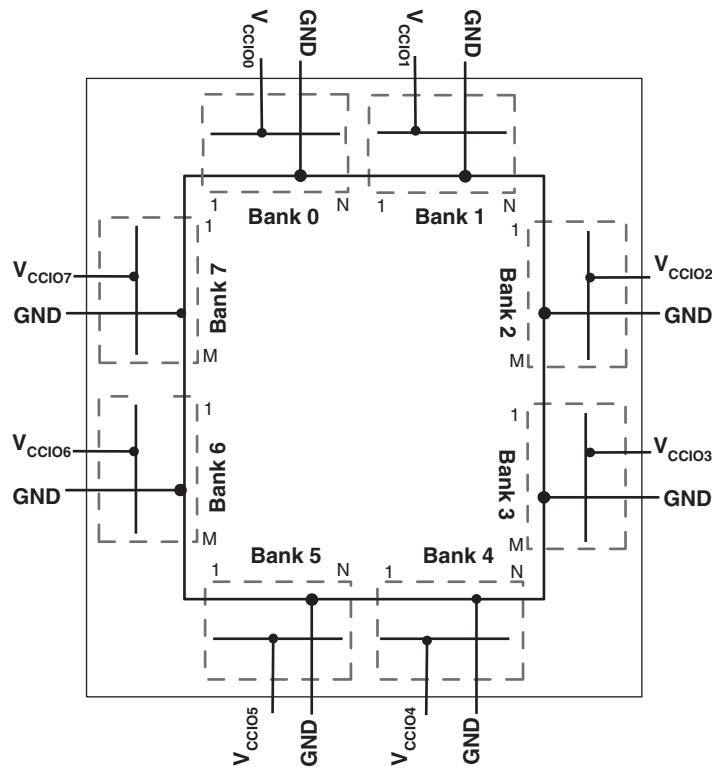


Figure 8-3. MachXO1200 and MachXO2280 sysIO Banking



### **V<sub>CCIO</sub> (1.2V/1.5V/1.8V/2.5V/3.3V)**

Each bank has a separate V<sub>CCIO</sub> supply that powers the single-ended output drivers and the ratioed input buffers such as LVTTTL, LVCMOS and PCI. LVTTTL, LVCMOS3.3, LVCMOS2.5 and LVCMOS1.2 also have fixed threshold options allowing them to be placed in any bank and is independent of bank V<sub>CCIO</sub>. The V<sub>CCIO</sub> 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.

The V<sub>CCIO</sub> of one of the banks is also used to power the JTAG pins (Bank1 for MachXO256, Bank2 for MachXO640 and Bank5 for MachXO1200 and MachXO2280 devices). Therefore, the threshold of the JTAG pins is determined by the V<sub>CCIO</sub> of the JTAG bank.

### **V<sub>CCAUX</sub> (3.3V)**

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

### **Mixed Voltage Support in a Bank**

The MachXO sysIO buffer is connected to three parallel ratioed input buffers. These three parallel buffers are connected to V<sub>CCIO</sub>, V<sub>CCAUX</sub> and to V<sub>CC</sub>, giving support for thresholds that track with V<sub>CCIO</sub>, as well as fixed thresholds for 3.3V (V<sub>CCAUX</sub>) and 1.2V (V<sub>CC</sub>) inputs. This allows the input threshold for ratioed buffers to be assigned on a pin-by-pin basis, rather than being tracked with V<sub>CCIO</sub>. This option is available for all 1.2V, 2.5V and 3.3V ratioed inputs and is independent of the bank V<sub>CCIO</sub> voltage. For example, if the bank V<sub>CCIO</sub> 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<sub>CCIO</sub>. This option only takes effect after configuration. Output standards within a bank are always set by V<sub>CCIO</sub>. Table 8-2 shows the sysIO standards that the user can mix in the same bank.

**Table 8-2. Mixed Voltage Support**

V <sub>CCIO</sub>	Input sysIO Standards					Output sysIO Standards				
	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 in the MachXO640 and MachXO256**

Description	Bank 0	Bank1	Bank2	Bank3
I/O Buffer Type	Single-ended	Single-ended	Single-ended	Single-ended
Output Standards Supported	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>
Inputs	All Single-ended	All Single-ended	All Single-ended	All Single-ended
Clock Inputs	All Single-ended	All Single-ended	All Single-ended	All Single-ended

1. These differential standards are implemented by using complementary LVCMOS driver with external resistor pack.
2. MachXO256 only has 2 banks, Banks 0 and Bank 1.

**Table 8-4. I/O Standards Supported by I/O Pins on Each Side of the MachXO1200 and MachXO2280**

Description	Top	Right	Bottom	Left
I/O Buffer Type	Single-ended	Single-ended and Differential	Single-ended	Single-ended and Differential
Output Standards Supported	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  PCI33 LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  LVDS25 <sup>2</sup> LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12  LVDS25 <sup>2</sup> LVDS25E <sup>1</sup> LVPECL <sup>1</sup> BLVDS <sup>1</sup> RSDS <sup>1</sup>
Inputs	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential
Clock Inputs	All Single-ended, Differential (PLL input)	All Single-ended, Differential (PLL input)	All Single-ended, Differential (PLL input)	All Single-ended, Differential (PLL input)
PCI Support	PCI33 with 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.
2. These are supported on half the I/Os of the bank.

**LVCMOS Buffer Configurations**

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

**Programmable PULLUP/PULLDOWN/BUSKEEPER**

When configured as LVCMOS or LVTTTL, 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

All LVCMOS and LVTTTL single-ended drivers have programmable drive strength. This option can be set for each I/O independently. The table below lists the programmable drive strengths available for each I/O standard. The actual value will vary with 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.

**Table 8-5. Programmable Drive Strength**

Single-ended I/O Standard	Programmable Drive (mA)
LVCMOS12	2, 6
LVCMOS15	4, 8
LVCMOS18	4, 8, 12, 14
LVCMOS25	4, 8, 12, 14
LVCMOS33	4, 8, 12, 14
LVTTTL	4, 8, 12, 16

## Programmable Slew Rate

Each LVCMOS or LVTTTL 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 also has an individual slew rate control. This allows a designer to specify slew rate control on a pin-by-pin basis. This slew rate control affects both the rising edge and the falling edges.

## Open-Drain Control

Each LVCMOS and LVTTTL output buffer can be configured to function as an open-drain output. The user can implement an open-drain output by turning on the OPENDRAIN attribute in the software.

## Programmable PCICLAMP

Each sysIO buffer on the top bank of the MachXO1200 and MachXO2280 devices can be configured to support PCI33. The buffers also have a PCI clamp diode that will be turned on when the sysIO buffer is configured as PCI33.

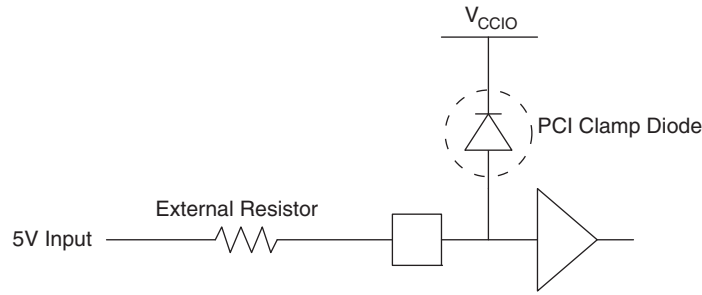
The PCI clamp is mainly used when implementing a 3.3V PCI interface. The PCI Specification revision 2.2 requires the use of clamping diodes for 3.3V operation. For more information on the PCI interface, please refer to the PCI specification, revision 2.2.

The PCI clamp can also be optionally turned on for LVCMOS and LVTTTL sysIO buffers on the top bank of the MachXO1200 and MachXO2280 devices. In this case, the PCI clamp is used to implement a 5V input interface.

## 5V Input Interface Using the PCI Clamp Diode

All the I/Os on the top side of the MachXO1200 and MachXO2280 devices (Banks 0 and 1) have a clamp diode that is used to clamp the voltage at the input to  $V_{CCIO}$ . This clamp diode can be used along with an external resistor to make an input 5V tolerant.

Figure 8-4. 5V Input Interface Example



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

In order to interface to 5V input, it is recommended that the V<sub>CCIO</sub> is set between 2.5V to 3.3V.

Below is an example of how to calculate the value of this external resistor when V<sub>CCIO</sub> is 2.75V.

- Maximum voltage at input pin, V<sub>INMAX</sub> = 3.75V (see the [MachXO Family Data Sheet](#) for more details)
- Bank V<sub>CCIO</sub> = 2.75V
- Maximum voltage drop across clamp diode, V<sub>D</sub> = V<sub>INMAX</sub> - V<sub>CCIO</sub> = 3.75 - 2.75 = 1V
- The current across the clamp diode at V<sub>D</sub> 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<sub>D</sub> is 1V, the clamp diode current is I<sub>D</sub> = 382μA.

Table 8-6. Power Clamp Data from a Typical IBIS Model<sup>1</sup>

Voltage	I (max.)	Units
-1.40	408	μA
-1.30	402	μA
-1.20	395	μA
-1.10	389	μA
-1.00	382	μA
-0.90	380	μA
-0.80	380	μA
-0.70	380	μA
-0.60	380	μA
-0.50	380	μA
-0.40	353	μA
-0.30	51	μA
-0.20	3.8	μA
-0.10	0.23	μA
0.00	0.15	μA

1. Always consult the latest IBIS data available from Lattice. IBIS model information is available by device family on the [Lattice web site](#).

- Assume the maximum output voltage of the driving device is V<sub>EXT</sub> = 5.25V. The value of the external resistor can then be calculated as follows:

$$R_{EXT} = (V_{EXT} - V_{INMAX})/I_D = (5.25V - 3.75V)/382\mu A = 3.93 \text{ K ohm}$$

If the  $V_{CCIO}$  of the bank is increased, it will also increase the value of the external resistor required. Keep in mind that changing the Bank  $V_{CCIO}$  will change the value of the input threshold voltage.

## Software sysIO Attributes

sysIO attributes can be specified in the HDL, using the Preference Editor GUI in ispLEVER®, the Spreadsheet View in Lattice Diamond™ design software, or in the ASCII Preference file (.perf) 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 6 and 7 list the available I/O types.

**Table 8-7. IO\_TYPE Attribute Values for MachXO640 and MachXO256 Devices**

sysIO Signaling Standard	IO_TYPE
DEFAULT	LVC MOS25
RS DS	RS DS
Emulated LVDS 2.5V	LVDS25E <sup>1</sup>
Bus LVDS 2.5V	BLVDS25 <sup>1</sup>
LVPECL 3.3V	LVPECL33 <sup>1</sup>
LVTTL	LVTTL33
3.3V LVC MOS	LVC MOS33
2.5V LVC MOS	LVC MOS25
1.8V LVC MOS	LVC MOS18
1.5V LVC MOS	LVC MOS15
1.2V LVC MOS	LVC MOS12

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

**Table 8-8. IO\_TYPE Attribute Values for MachXO1200 and MachXO2280 Devices**

sysIO Signaling Standard	IO_TYPE
DEFAULT	LVC MOS25
LVDS 2.5V	LVDS25 <sup>2</sup>
RS DS	RS DS
Emulated LVDS 2.5V	LVDS25E <sup>1</sup>
Bus LVDS 2.5V	BLVDS25 <sup>1</sup>
LVPECL 3.3V	LVPECL33 <sup>1</sup>
LVTTL	LVTTL33
3.3V LVC MOS	LVC MOS33
2.5V LVC MOS	LVC MOS25
1.8V LVC MOS	LVC MOS18
1.5V LVC MOS	LVC MOS15
1.2V LVC MOS	LVC MOS12
3.3V PCI	PCI33 <sup>3</sup>

1. These differential standards are implemented by using a complementary LVC MOS driver with external resistor pack.
2. Available on 50% of the I/Os on the right and left side banks.
3. Available on the top side bank.



## OPENDRAIN

LVC MOS and LV TTL 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 LV TTL and LVC MOS output standards. These can be set on each I/O pin individually. The programmable drive available on a pad will depend on the  $V_{CCIO}$ . Table 8-9 shows the drive strength available for different I/O standards and the defaults for each of them.

**Table 8-9. Programmable Drive Strength Values at Various  $V_{CCIO}$  Voltages**

Output Standard	Drive (mA)	Default (mA)
LV TTL	4, 8, 12, 16	8
LVC MOS33	4, 8, 12, 14	8
LVC MOS25	4, 8, 12, 14	12
LVC MOS18	4, 8, 12, 14	8
LVC MOS15	4, 8	8
LVC MOS12	2, 6	6

## PULLMODE

The PULLMODE attribute is available for all the LV TTL and LVC MOS inputs and outputs. This attribute can be enabled for each I/O independently.

Values: UP, DOWN, NONE, KEEPER

Default: UP

**Table 8-10. PULLMODE Values**

PULL Options	PULLMODE Value
Pull up (Default)	UP
Pull Down	DOWN
Bus Keeper	KEEPER
Pull Off	NONE

## PCICLAMP

PCI33 inputs and outputs are available on the top bank of the MachXO1200 and MachXO2280 devices. When an I/O is configured as a PCI33 standard, the PCICLAMP is enabled for that buffer. The PCICLAMP is also available for all LVC MOS33 and LV TTL inputs. This is used to implement a 5V input interface.

Values: ON, OFF

Default: ON (for PCI33 input and output)  
OFF (for LVC MOS33 and LV TTL inputs)

## SLEWRATE

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

Values: FAST, SLOW

Default: FAST

## 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. Users can also assign pins directly using the Preference Editor GUI in ispLEVER or the Spreadsheet View in Diamond. The appendices of this document explain this in greater detail.

## Design Considerations and Usage

This section discusses some rules and considerations for designing with the MachXO sysIO buffer.

### Banking Rules

- If the  $V_{CCIO}$  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  $V_{CCIO}$  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.
- PCI I/O standards with PCI clamps are only available on the top bank (Banks 0 and 1) on the MachXO1200 and MachXO2280 devices.
- PCI I/O standards with PCI clamps are not available on the MachXO640 and MachXO256 devices.
- Only 50% of the I/Os on the left and right banks of the MachXO1200 and MachXO2280 devices support a True LVDS driver. True LVDS receivers are available on all banks for the MachXO1200 and MachXO2280 devices.
- All banks support emulated differential outputs using an external resistor pack and complementary LVCMOS driver.

### Zero Hold Time

The user can achieve a zero hold time for his or her inputs by specifying a zero hold time preference in the software. The software will add additional delays to the input path in order to achieve this zero hold time.

### Fast Output Path

The MachXO devices have a dedicated fast output I/O connection from the adjacent PFUs to the I/O buffers within the PIO. This connection provides faster output delays for faster clock-to-output propagation delays and pin-to-pin propagation delays. The software will automatically use this fast output path to achieve faster  $t_{CO}$  and  $t_{PD}$  requirements. You can fulfill the  $t_{CO}$  and  $t_{PD}$  requirement by assigning these preferences in the Preference Editor in the software.

### Dedicated Pins

#### Global Set Rest (GSR)

The GSR in the MachXO devices is an asynchronous Global Set Reset. This signal can be programmed to come from either the PFU logic or from the dedicated GSR input pad. When the software does not see any logic associated with the GSR, then it will automatically use the dedicated GSR input path. This provides faster timing. When the reset used is a logic reset, the polarity is programmable. When the dedicated GSR input from the GSR pad is used, the polarity has to be active low.

**Tristate All (TSALLPAD)**

All MachXO devices have a dedicated TSALLPAD pin that is used to enable or disable the tristate control to all the output buffers. By default, the pin will function as an I/O unless programmed to be a TSALLPAD. This signal also has programmable global polarity control. By default, the polarity is active high. This global tristate control signal can also be generated using user logic.

When the TSALLPAD is enabled, the software will implement the tristate control using the TSALL software primitive. The polarity control of the TSALLPAD will control the polarity of TSALL.

When TSALLPAD is not used in the design, but is required for test purposes, the TSALL primitive can be instantiated in the HDL and the TSALLPAD is connected to the input of this primitive.

**Differential I/O Implementation**

MachXO devices support a variety of differential standards, as detailed in the following sections.

**LVDS (MachXO1200 and MachXO2280)**

True LVDS (LVDS25) drivers are available on 50% of the I/Os on the left and right sides of the MachXO1200 and MachXO2280 devices. LVDS input support is available on all sides of the MachXO1200 and MachXO2280 devices.

**LVDSSE**

The single-ended sysIO buffer pairs in all the MachXO devices support LVDS output drivers using complementary LVCMOS drivers with external resistors (LVDS25E) on all four sides of the device.

The MachXO1200 and MachXO2280 devices also support LVDSSE inputs on all four sides of the device.

Please refer to the [MachXO Family Data Sheet](#) for a detailed explanation of these LVDS implementations.

**BLVDS**

All single-ended sysIO buffer pairs in all the MachXO devices support Bus-LVDS output driver using complementary LVCMOS drivers with external resistors on all the four sides of the device.

The MachXO1200 and MachXO2280 devices also support BLVDS inputs on all four sides of the device.

Please refer to the [MachXO Family Data Sheet](#) for a detailed explanation of BLVDS implementation.

**RSDS**

All single-ended sysIO buffer pairs in all the MachXO devices support RSDS output driver using complementary LVCMOS drivers with external resistors.

The MachXO1200 and MachXO2280 devices also support RSDS inputs on all four sides of the device.

Please refer to the [MachXO Family Data Sheet](#) for a detailed explanation of RSDS implementation.

**LVPECL**

All single-ended sysIO buffers pairs in all the MachXO devices support LVPECL output driver using complementary LVCMOS drivers with external resistors.

The MachXO1200 and MachXO2280 devices also support LVPECL inputs on all four sides of the device.

Please refer to the [MachXO Family Data Sheet](#) for a detailed explanation of LVPECL 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](http://www.latticesemi.com)

---

---

**Revision History**

Date	Version	Change Summary
—	—	Previous Lattice releases.
July 2007	01.4	Correction in VCCIO (1.2V/1.5V/1.8V/2.5V/3.3V) text section. VCCIO of Bank2 is used to power the JTAG pin for MachXO640, rather than Bank3.
September 2010	01.5	Updated for Lattice Diamond design software support.
		Updated IBIS information.

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

Using these HDL attributes, you can assign the 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. Refer to the Precision RTL Synthesis and Synplify user manuals for a complete list of synthesis attributes. These manuals are available through the ispLEVER or Diamond 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 and Synplify synthesis tools.

#### Syntax

**Table 8-11. VHDL Attribute Syntax for Precision RTL Synthesis and Synplify**

Attribute	Syntax
IO_TYPE	attribute IO_TYPE: string; attribute IO_TYPE of <i>Pinname</i> : signal is “IO_TYPE Value”;
OPENDRAIN	attribute OPENDRAIN: string; attribute OPENDRAIN of <i>Pinname</i> : signal is “OpenDrain Value”;
DRIVE	attribute DRIVE: string; attribute DRIVE of <i>Pinname</i> : signal is “Drive Value”;
PULLMODE	attribute PULLMODE: string; attribute PULLMODE of <i>Pinname</i> : signal is “Pullmode Value”;
PCICLAMP	attribute PCICLAMP: string; attribute PCICLAMP of <i>Pinname</i> : signal is “PCIClamp Value”;
SLEWRATE	attribute PULLMODE: string; attribute PULLMODE of <i>Pinname</i> : signal is “Slewrates Value”;
LOC	attribute LOC: string; attribute LOC of <i>Pinname</i> : 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;

**--\*\*\*OPENDRAIN 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 "16";

**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;

--\*\*\***PCICLAMP 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";

**LOC**

--\*\*\**Attribute Declaration*\*\*\*

ATTRIBUTE LOC : string;

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

ATTRIBUTE LOC OF input\_vector: SIGNAL IS "E3,B3,C3 ";

## Verilog Synplify

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

### Syntax

**Table 8-12. Verilog Synplify Attribute Syntax**

Attribute	Syntax
IO_TYPE	<i>PinType PinName</i> /* synthesis IO_TYPE= <i>IO_Type Value</i> */;
OPENDRAIN	<i>PinType PinName</i> /* synthesis OPENDRAIN = <i>OpenDrain Value</i> */;
DRIVE	<i>PinType PinName</i> /* synthesis DRIVE= <i>Drive Value</i> */;
PULLMODE	<i>PinType PinName</i> /* synthesis PULLMODE= <i>Pullmode Value</i> */;
PCICLAMP	<i>PinType PinName</i> /* synthesis PCICLAMP = <i>PCIClamp Value</i> */;
SLEWRATE	<i>PinType PinName</i> /* synthesis SLEWRATE= <i>Slewrates Value</i> */;
LOC	<i>PinType PinName</i> /* synthesis LOC= <i>pin_locations</i> */;

### Examples

#### //IO\_TYPE, PULLMODE, SLEWRATE and DRIVE assignment

```
output portB /*synthesis IO_TYPE="LVCMOS33" PULLMODE ="UP" SLEWRATE ="FAST"
DRIVE ="14"*/;
output portC /*synthesis IO_TYPE="LVDS25" */;
```

#### //OPENDRAIN

```
output portA /*synthesis OPENDRAIN ="ON"*/;
```

#### //PCICLAMP

```
output portA /*synthesis IO_TYPE="PCI33" PCICLAMP ="PCICLAMP"*/;
```

#### //IO 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 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 8-13. 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 PULLMODE Pullmode Value
PCICLAMP	// pragma attribute PinName PCICLAMP PCIClamp Value
SLEWRATE	// pragma attribute PinName SLEWRATE Slewrate 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

***LOC***
// pragma attribute portB loc E3

```



## Appendix B. sysIO Attributes Using the ispLEVER Preference Editor or Diamond Spreadsheet View

You can assign the sysIO buffer attributes using the Pre-map Preference Editor GUI available in the ispLEVER or the Spreadsheet View GUI in Diamond. The Pin Attribute sheet in ispLEVER and Port Assignments sheet in Diamond list all the ports in your design and all the available sysIO attributes as preferences. When you click on each of these cells you get a list of all the valid I/O preferences for that port. Each column takes precedence over the next. Therefore, when you choose a particular IO\_TYPE, 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-click on a cell to list all the available pin locations. The Preference Editor and the Spreadsheet View will also do a DRC check to search for any 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 and the Spreadsheet View are written into the preference file (.prf).

Figure 8-5 shows the Pin Attribute sheet and the Cell Attribute sheet view of the Preference Editor. For further information on how to use the Preference Editor, please refer to the ispLEVER Help documentation. You can get to this in the Help menu option of the software.

Figure 8-5. Pin Attributes Tab

Type	Signal/Group Name	Gro...	Pin Lo...	Bank	IO Type	Drive	Slewrate	Pullmode	PCIclamp	SCHMITT_TRIGGER	OPENDRAIN	Output Load
1	Output Port	q(6)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
2	Output Port	q(5)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
3	Output Port	q(7)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
4	Output Port	q(3)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
5	Output Port	q(0)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
6	Output Port	be	N/A		LVDS25E				N/A			N/A
7	Output Port	q(2)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
8	Output Port	q(1)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
9	Output Port	q(4)	N/A		LVC MOS15	8	SLOW	UP	N/A		OFF	
10	Input Port	al(2)	N/A		LVC MOS33				N/A			N/A
11	Input Port	al(3)	N/A		LVC MOS33				N/A			N/A
12	Input Port	al(1)	N/A		LVC MOS33				N/A			N/A
13	Input Port	al(0)	N/A		LVC MOS33				N/A	ON		N/A
14	Input Port	ah(1)	N/A		LVC MOS25				N/A	ON		N/A
15	Input Port	ah(5)	N/A		LVC MOS25				N/A			N/A
16	Input Port	al(7)	N/A		LVC MOS33				N/A			N/A
17	Input Port	es	N/A		LVC MOS12			KEEPER	N/A	OFF		N/A
18	Input Port	al(6)	N/A		LVC MOS33				N/A			N/A
19	Input Port	rst	N/A		LVC MOS25				N/A			N/A
20	Input Port	ah(7)	N/A		LVC MOS25				N/A			N/A
21	Input Port	ah(6)	N/A		LVC MOS18				N/A			N/A
22	Input Port	ah(2)	N/A		LVC MOS25				N/A			N/A

Figure 8-6 shows the Port Assignment sheet in the Diamond Spreadsheet View. For further information on how to use the Spreadsheet View, please refer to the Reference Guide available in Diamond.

Figure 8-6. Port Assignment Tab

Type	Name	Group by	Pin	Bank	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	OPENDRAIN	Outload (pF)	MaxSkew	Clock Load Only
Output Port	FLASH_BYTE	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_CE	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_OE	N/A			LVC MOS25	KEEPER	12	FAST	OFF	0.000		N/A
Output Port	FLASH_WE	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_RSTn	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A
Output Port	FLASH_ADDR...	N/A			LVC MOS25	UP	12	FAST	OFF	0.000		N/A

Architecture: MachXO Device: LCMXO256C Package: TQFP100

## 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 separate sections: a schematic section for those 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 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, OPENDRAIN, DRIVE, PULLMODE, PCICLAMP and SLEWRATE.

#### 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=LVCMOS33;
```

### 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;
```