Using Differential I/O (LVDS, SubLVDS, LVPECL) in iCE65™ mobileFPGA™s

SiliconBlue®

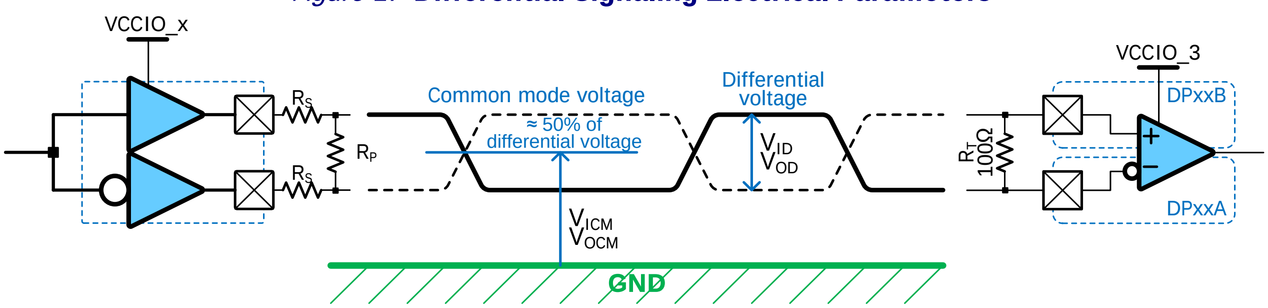
October 1, 2010 (2.1) **Application Note AN008**

Introduction

Differential I/O standards are popular in a variety of consumer applications, especially those that require high-speed data transfers such as graphic display drivers and camera interfaces. In these systems, multiple signals are typically time-division-multiplexed onto a smaller number of high-speed, differential serial channels.

Differential signals require two Programmable I/O (PIO) pins, working as a pair or a channel, as shown in Figure 1. One side of the pair represents the true polarity of the signal while the other side of the pair represents the opposite polarity. The resulting logic value is the difference between the two sides of the signal pair.

Figure 1: **Differential Signaling Electrical Parameters**



The key electrical parameters are the common mode voltage and the differential voltage. For iCE65 applications, the common mode voltage is essentially half the I/O Bank supply voltage. The differential voltage depends on the values of the external compensation resistors, discussed in "LVDS and SubLVDS Termination" on page 3.

Figure 2: Representative Electrical Characteristics of Various Differential I/O Standards

	LVDS FPD-Link	subLVDS FlatLink3G	LVPECL		Units
iC65 Support	Inputs and Outputs	Inputs and Outputs	Input Only		
Output Frequency	350	210	TBD	Max	MHz, Mbps
Input Frequency	TBD	TBD	TBD	Max	MHz, Mbps
VCCIO	2.5	1.8	2.5	Nom/Typ	V
V_{OD}	250	100	600	Min	mV
	350	150		Nom/Typ	
	450	200		Max	
V_{OS}, V_{CM}	1.125	0.8		Min	V
	1.25	0.9		Nom	
	1.375	1.0		Max	
V_{IDTH}	250	70	400	Min	mV
	450	200	1900	Max	
V_{ICM}		0.6	1.0	Min	V
		1.2	V_{CC}-0.3	Max	

Differential signaling provides many advantages. In the examples discussed here, all the differential I/O standards have **reduced voltage swing**, which allows **faster switching** speeds and **potentially higher bandwidth**. Reduced voltage swings also mean **less dynamic power** consumption and **reduced electromagnetic interference** (EMI).

Differential switching provides **improved noise immunity** and **reduces duty-cycle distortion** caused the differences in rise- and fall-time by the output driver.

The higher potential switching speeds of differential I/O allows data to be multiplexed onto **a reduced number of wires** at a much higher data rate per line. The reduced number of wires reduces system cost and in some cases simplifies the system design. The internal phase-locked loop (PLL) available in iCE65P FPGAs provides convenient on-chip clock multiplication or division to support such applications.

Differential Outputs

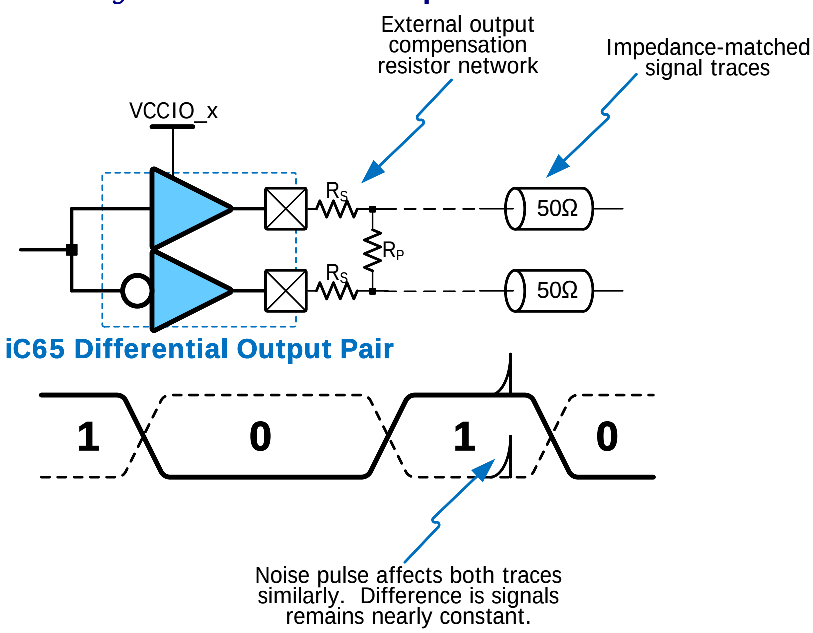
For some differential I/O standards, such as LVDS, the output driver is actually a current source. On iCE65/iCE65P FPGAs, however, differential outputs are constructed using a pair of single-ended PIO pins as shown in Figure 3, and an external resistor network consisting of three resistors. Because differential outputs are built from two single-ended LVCMOS outputs, differential outputs are available in any I/O bank.

The two FPGA outputs must be part of the same I/O tile as indicated in the iCE65 or iCE65P data sheet. The pair choice also depends on the chosen device package as not all I/O tile pairs are bonded out in all packages. Consult the package pinout sections of the iCE65L or iCE65P device data sheets for additional information.

PIO pairs in I/O Banks 0, 1, and 2 (and I/O Bank 3 on the iCE65L01) are preferred as they provide cleaner switching edges for most applications. The PIO pins in I/O Bank 3 on iCE65L04/P04, or iCE65L08 FPGAs have higher driver current and are better suited for applications that exceed 330 Mbps.

Each differential I/O pair requires a three-resistor termination network to adjust output characteristics to match those for the specific differential I/O standard. The output characteristics depend on the values of the parallel resistor (R_P) and series resistors (R_S). These resistors should be surface mounted as close as possible to the FPGA output pins.

Figure 3: **Differential Output Pair**



Differential Inputs

Differential inputs are only supported in I/O Bank3 and are not supported on iCE65L01 FPGAs. The number of differential input pairs is shown in Table 1.

Table 1: **Maximum Differential Input Pairs Available on iCE65/iCE65P FPGAs**

	iCE65L01	iCE65L02	iCE65L04/ iCE65P04	iCE65L08
Maximum Differential Input Pairs	0	16	20	25

Differential inputs require specific PIO pin pairs as listed in the iCE65L and iCE65P data sheets. Each differential input pair consists of one pin labeled DPxxA and another labeled DPxxB, where "xx" represents the differential pair number. Both pins must be in the same differential pair.

Connect the positive or true polarity side of the differential pair to the DPxxA input and the negative or complementary side of the pair to the DPxxB input. If it is easier to route the differential pair, the input pins can be swapped, which produces an inverted input value. The inverted input value can subsequently be inverted by logic within the FPGA.

An input termination resistor must be connected between the DPxxA and DPxxB pins to generate the differential signal. The resistor must be twice the trace impedance, as described in the following section.

Typically, the resulting signal pair is routed on the printed circuit board (PCB) using controlled impedance and delay matching.

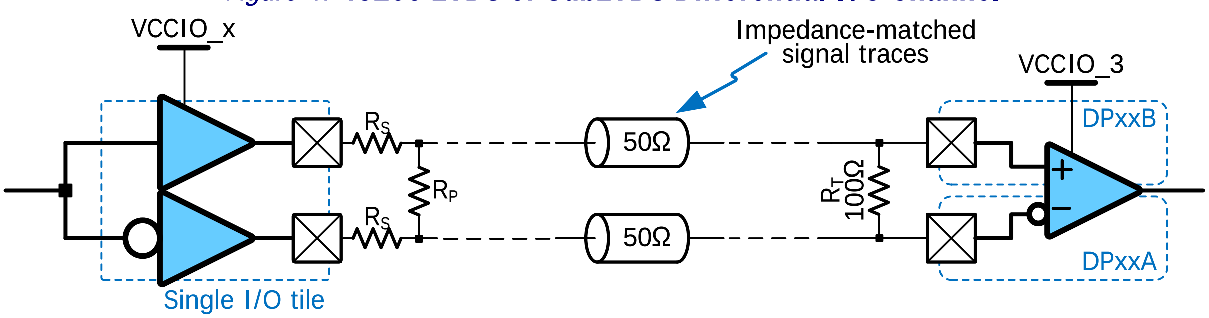
LVDS and SubLVDS Termination

LVDS and SubLVDS inputs require external compensation and termination resistors for proper operation, as shown in Figure 4. A termination resistor, R_T, between the positive and negative inputs at the receiver forms a current loop. The current across this resistor generates the voltage detected by the receiver's differential input comparator.

Similarly, iCE65 LVDS and SubVLDS outputs require an external resistor network, consisting of two series resistors, R_S, and a parallel resistor, R_P. This resistor network adjusts the FPGA's output driver to provide the necessary current and voltage characteristic s required by the specification.

The signals are routed with matched trace impedance, Z_O, on the printed circuit board, typically with 50Ω impedance.

Figure 4: **iCE65 LVDS or SubLVDS Differential I/O Channel**



The resistor values for the compensation network are described below. These equations are also provided in the Differential I/O spreadsheet. The variables are defined and described in Table 2.

Input Termination Resistor (RT)

$$R_T = 2 \times Z_0 \quad \text{[Equation 1]}$$

Output Parallel Resistor (RP)

$$R_P = 2 \times \left(\frac{Z_0 \times VCCIO}{VCCIO - (2 \times V_{OD})} \right) \quad \text{[Equation 2]}$$

Output Series Resistor (RS)

$$R_S = \left(\frac{Z_0 \times \frac{R_P}{2}}{\frac{R_P}{2} - Z_0} \right) - R_{OUTPUT} \qquad \text{[Equation 3]}$$

Table 2: **Compensation Resistor Equation Variables**

Variable	Description
Z_0	Characteristic impedance of the printed circuit board trace; data sheet values assume 50 Ω traces
VCCIO	I/O Bank supply voltage, nominal, volts
V_{OD}	Differential output voltage swing, nominal, volts
R_{OUTPUT}	iCE65/iCE65P output source resistance, 30 Ω
R_P	LVDS/SubLVDS output source compensation parallel resistor
R_S	LVDS/SubLVDS output source compensation series resistor
R_T	LVDS/SubLVDS input termination resistor

Using the Companion iCE65 Differential I/O Calculator Spreadsheet

The iCE65 data sheet recommends specific values for LVDS and SubLVDS differential outputs but also assume that the differential signals are routed with 50 Ω characteristic impedance (Z_0). This may not be possible in all applications. Likewise, the system may require some other slightly different I/O standard.

This application note includes a companion spreadsheet, shown in Figure 5, to calculate resistor values for non-standard conditions. The values in Figure 5 are the default conditions for the LVDS I/O standard. For other standards, simply modify the VCCIO voltage, the differential output voltage, V_{OD}, and the characteristic impedance of the printed circuit board traces, Z_0.

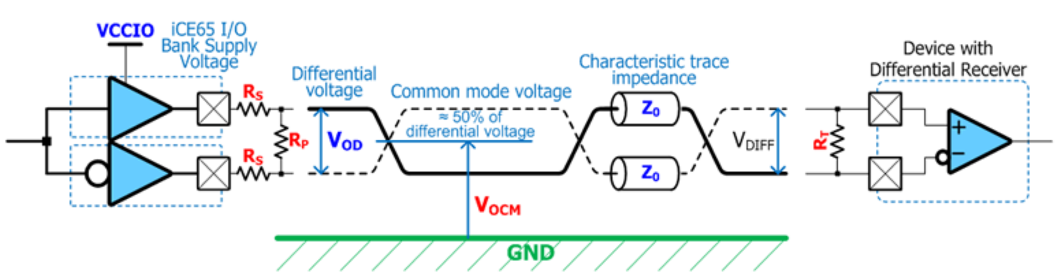
- iCE65 Differential I/O Spreadsheet
  www.siliconbluetech.com/media/downloads/AN008_iCE65_Differential_IO_Spreadsheet.xls

Figure 5: **iCE65 Differential I/O Calculator Spreadsheet**

iCE65™ Differential Output Calculator (Version 2.0, 24-MAY-2010)

	I/O Bank Supply	Diff. Output Voltage	Trace Impedance	Common Mode Output Voltage	Source Series Resistor	Source Parallel Resistor	Termination Resistor	Resistor Network Current	Termination Current	Actual Differential Voltage
Symbol	VCCIO	V_{OD}	Z_0	V_{OCM}	R_P	R_P	R_T	I_{RES}	I_{RT}	V_{DIFF}
Value	2.5	0.35	50	1.25	150	140	100	6.0	3.5	0.349
Units	volts	volts	ohm	volts	ohm	ohm	ohm	mA	mA	volts

Companion to Application Note **AN008: Using Differential I/O (LVDS, SubLVDS) in iCE65 FPGAs**
http://www.siliconbluetech.com/media/downloads/SiliconBlue_AN008.pdf



1.) The R_S and R_P resistors should be surface mounted as close to the iCE65 output balls/pins as possible.
2.) The termination resistor, R_T, should be as close to the recieving device's differential inputs as possible.
3.) The actual differential voltage, V_{DIFF}, may vary slightly from differential output voltage due to rounding of resistor values.

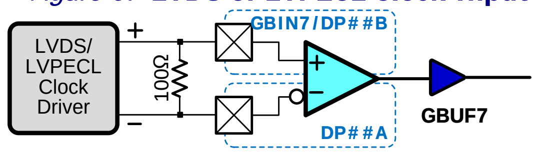
The spreadsheet automatically calculates the common mode output voltage, V_{OCM}, which is half of the VCCIO supply voltage. The spreadsheet also automatically calculates the values for the resistor network.

Finally, the spreadsheet also calculates the current draw through the resistor network and for the termination resistor. The spreadsheet rounds the resistor values to the nearest 10 Ω. Consequently, the spreadsheet also calculates the actual differential voltage based on the specified resistor values.

Differential Clock Input

iCE65 FPGA have eight global buffers for distributing clocks or other high fanout signals. Global buffer GBUF7, shown in Figure 6, is specifically designed to accept a differential clock input on the associated GBIN7/DPxxB, DPxxA differential input pair, which is part of I/O Bank 3. Connect an external 100 Ω termination resistor across the input pair. When VCCIO_3 is 2.5V, this global buffer input accepts either LVDS or LVPECL clock inputs.

Figure 6: **LVDS or LVPECL Clock Input**



The differential global buffer input is not available for iCE65P devices packaged in the CB132 package. This restriction is an artifact of the pin compatibility between the CB132 and CB284 package. Similarly, there are pinout differences for devices in the CB196 package.

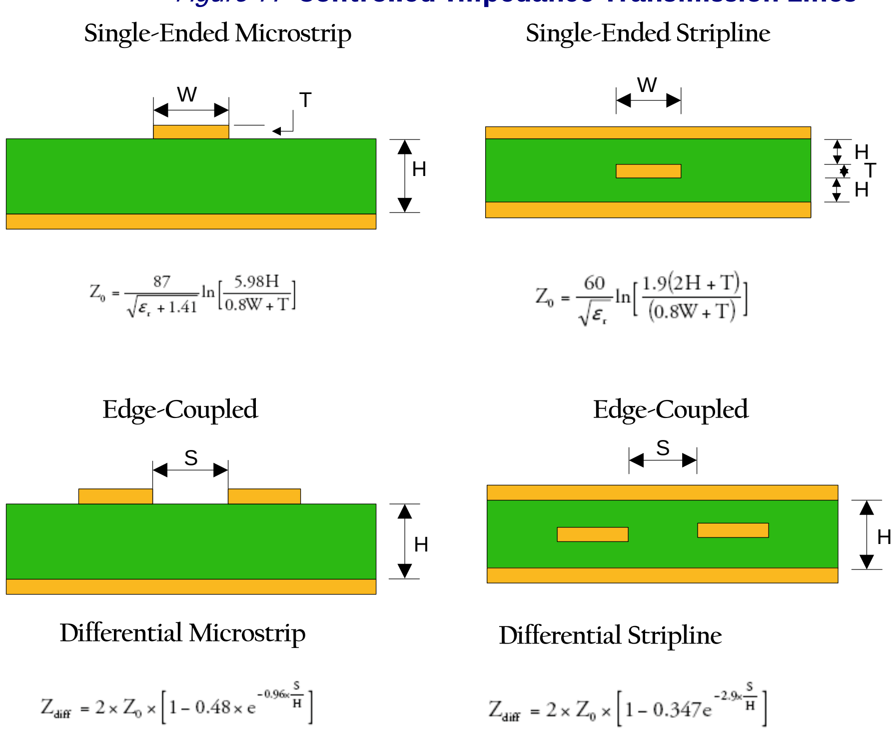
iCE65 Differential Signaling Board Layout Requirements

Figure 7 depicts several transmission line structures commonly used in printed circuit boards (PCBs). Each structure consists of a signal line and a return path with uniform cross-section along its length. The structure's dimensions along with the dielectric material properties determine the characteristic impedance of the transmission line. Figure 7 shows examples of edge-coupled microstrips, and edge-coupled or broad-side-coupled striplines.

To maintain constant differential impedance along the length, maintain uniform trace width and spacing, including good symmetry between the two lines.

For differential outputs, place the surface-mounted R_P and R_S resistors as close to the package balls as possible. Similarly, place the 100 Ω termination resistor, R_T, as close as possible to the differential input pair.

Figure 7: **Controlled Impedance Transmission Lines**



Typical PC board trace impedance is Z_0 = 50 Ohms. For a single-ended microstrip, the trace impedance is calculated by using the following equation:

$$Z_0 = \frac{87}{\sqrt{\varepsilon_r + 1.41}} \ln\left[\frac{5.98H}{0.8W + T}\right]$$

Equation 1: ***Single-ended microstrip trace impedance***

$$Z_0 = \frac{60}{\sqrt{\varepsilon_r}} \ln\left[\frac{1.9(2H + T)}{(0.8W + T)}\right]$$

Equation 5: ***Single-ended stripline trace impedance***

For an LVDS pair, differential impedance can be determined by using the following equations for differential microstrip and differential stripline:

$$Z_{diff} = 2 \times Z_0 \times \left[1 - 0.48 \times e^{-0.96 \times \frac{S}{H}}\right]$$

Equation 6: ***Differential impedance for differential microstrip***

$$Z_{diff} = 2 \times Z_0 \times \left[1 - 0.347 e^{-2.9 \times \frac{S}{H}}\right]$$

Equation 7: ***Differential impedance for differential stripline***

Because the coupling of two traces can lower the effective impedance, use 60 Ω design rules to achieve a differential impedance of approximately 100 Ohms.

Layout recommendations to minimize reflection

Skew delay is introduced if the trace lengths between the signals in the differential pair are not similar. With differing trace lengths, the signals on each side of the differential pair arrive at slightly different times and reflect off the receiver termination, creating a common-mode noise source on the transmission line.

Common-mode noise degrades the receiver's eye diagram, reduces signal integrity, and creates crosstalk between neighboring signals on the board. To minimize reflections due to unmatched trace lengths, consider the following guidelines:

- Match the length of each signal within the differential signal pair to within 20 mils.
- Minimize turns and vias or feed-throughs. Route differential pairs as straight as possible from point-to-point. Do not use 90 degree turns when routing differential pairs. Instead, use 45 degree bevels or rounded curves.
- Minimize vias on or near differential trace lines as these may create additional impedance discontinuities that will increase reflections at the receiver. If vias are required, place them as close to the receiver as possible.
- Use controlled impedance PCB traces. That is, control trace spacing, width, and thickness using Stripline or Microstrip layout techniques

EMI and Noise Cancellation

Differential signaling offers tremendous advantages over single ended signaling because it is less susceptible to noise. In differential signaling, two current carrying conductors are routed together, with the current in one conductor always equal in magnitude but opposite in direction to the other conductor. The result is that both electromagnetic fields cancel each other.

Common-mode noise rejection is another advantage of differential signaling. The receiver ignores any noise that couples equally on both sides of the differential signal, as shown in Figure 9.

Figure 8: **Single-ended Input Retains Signal Noise**

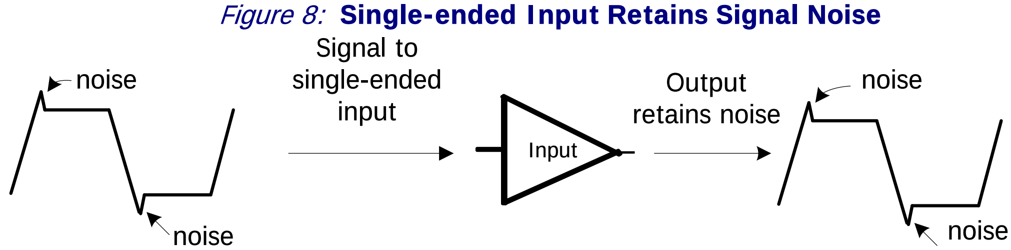
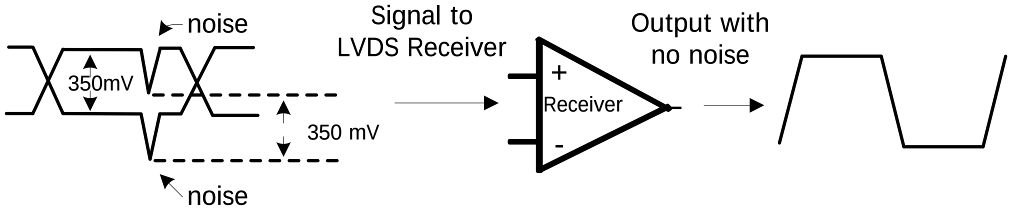


Figure 9: **Differential Input Eliminates Common-Mode Noise**



Reducing EMI Noise

Any skew between differential signals can generate EMI noise on the board. The following are some guidelines to reduce EMI emission onboard:

Match edge rates and signal skew between differential signals as closely as possible. Different signal rise and fall times and skew between signal pairs create common-mode noise, which generates EMI noise.

Maintain spacing between differential signal pairs that is less than twice the PCB trace width.

Defining Differential I/O

Single-ended I/O connections are automatically inferred from the top-level VHDL or Verilog circuit description during logic synthesis.

However, differential I/O connections must be specifically instantiated using design primitives from the SiliconBlue technology library. Table 3 lists the specific I/O primitive required by differential I/O type.

- SiliconBlue Technology Library
  www.siliconbluetech.com/media/icecube-development-software/SBT_ICE_Technology_Library.PDF

Table 3: **Differential I/O Types and Required SiliconBlue I/O Primitive**

Differential I/O Type	SiliconBlue Design Primitive	PIN_TYPE	IO_STANDARD
Differential clock input	SB_GB_IO	See "PIN_TYPE Parameter" on page 7.	SB_LVDS_INPUT
Differential input	SB_IO		SB_LVDS_INPUT
Differential output	SB_IO		SB_LVCMOS

The SB_IO and SB_GB_IO primitives also require specific parameter settings. Differential inputs always require that IO_STANDARD be set to SB_LVDS_INPUT. Differential outputs typically require IO_STANDARD be set to SB_LVCMOS, although other values are allowed for I/O Bank 3 in all iCE65 FPGA except the iCE65L01.

SB_IO Primitive

Table 4 lists the signal ports for the SB_IO primitive, which describes one of the Programmable I/O (PIO) pins on an iCE65 FPGA. The table also shows the signal direction for each port, relative to the PIO pin (the SB_IO primitive). These same signals also appear on the SB_GB_IO primitive, which describes a global buffer input.

Table 4: **Port Names, Signal Direction, and Description for SB_IO (SB_GB_IO) Primitive**

Port Name	Direction	Description
PACKAGE_PIN	I/O	Connection to top-level input, output, or bidirectional signal port.
LATCH_INPUT_VALUE	Input	iCEgate latch input. When High, hold the last pad value. Used for power reduction in some PIN_TYPE modes. There is one control input per I/O Bank. 0 = Input data flows freely 1 = Last data value on pad held constant to save power
CLOCK_ENABLE	Input	Clock enable input, shared connection to all flip-flops within the SB_IO primitive. If this port is left unconnected, automatically tied High. 0 = Flip-flops hold their current value 1 = Flip-flops accept new data on the active clock edge
INPUT_CLOCK	Input	Clock for all input flip-flops. If this port is left connected, automatically tied Low.

Port Name	Direction	Description
OUTPUT_CLOCK	Input	Clock for all output flip-flops. If this port is left connected, automatically tied Low.
OUTPUT_ENABLE	Input	Enables the output buffer. 0 = Output disabled, pad is high-impedance (Hi-Z) 1 = Output enabled, actively driving
D_OUT_0	Input	Data output. For DDR output modes, this is the value clocked out on the rising edge of the OUTPUT_CLOCK.
D_OUT_1	Input	Data output used in DDR output modes. This is the value clocked out on the falling edge of the OUTPUT_CLOCK.
D_IN_0	Output	Data input. For DDR input modes, this is the value clocked into the device on the rising edge of the INPUT_CLOCK.
D_IN_1	Output	For DDR input modes, this is the value clocked into the device on the falling edge of the INPUT_CLOCK.

SB_GB_IO Primitive

Global buffer inputs provide a direct connection from a PIO pin to an associated global buffer. This connection can be instantiated using an SB_GB_IO primitive. An SB_GB_IO primitive has all the ports for an SB_IO primitive, shown in Table 4, plus the additional connection shown in Table 5. Global Buffer Input 7 (GBIN7) is the only one that supports differential clock inputs. See "Differential Clock Input" on page 4 for more information.

Table 5: **Additional Port Names on SB_GB_IO Primitive**

Port Name	Direction	Description
GLOBAL_BUFFER_OUTPUT	Output	Output from associated Global Buffer. This output may or may not be controlled by the iCEgate latch, as described below. **iCE65 FPGAs:** Output not affected by iCEgate latch. **iCE65P FPGAs:** Output controlled by iCEgate latch if connected.

PIN_TYPE Parameter

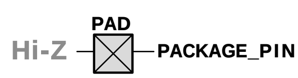
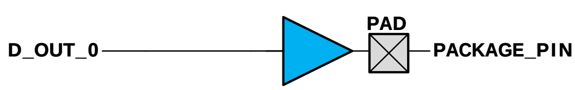
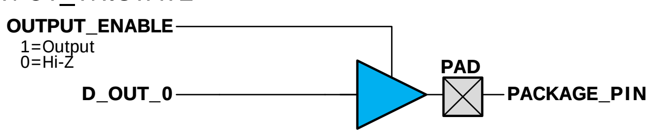
The PIN_TYPE parameter defines the structure and the functionality of any instantiated SiliconBlue SB_IO primitive. PIN_TYPE is a six-bit binary value. The upper four bits, PIN_TYPE[5:2], define the output structure while the lower two bits, PIN_TYPE[1:0] define the input structure. Both fields are required, but operate independently.

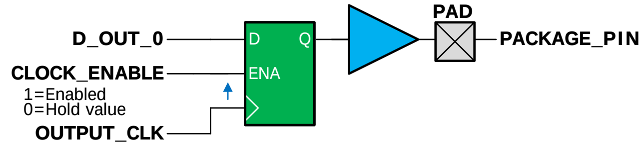
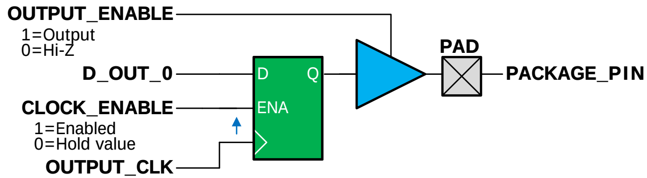
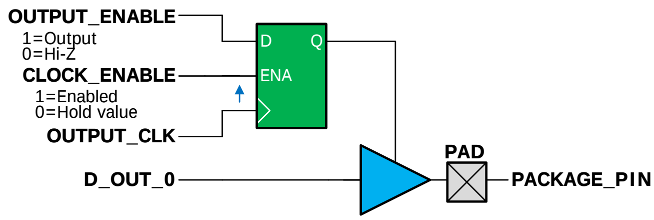
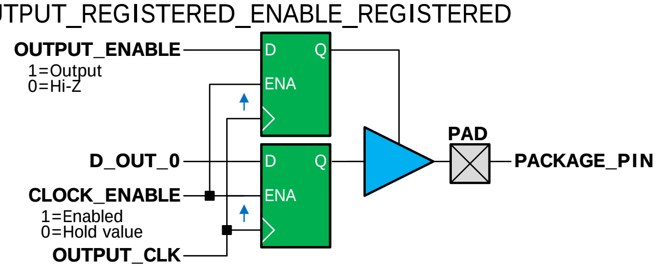
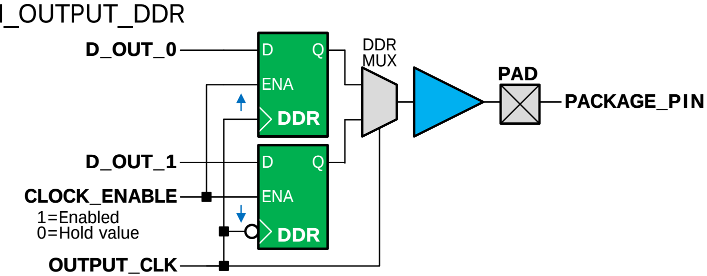
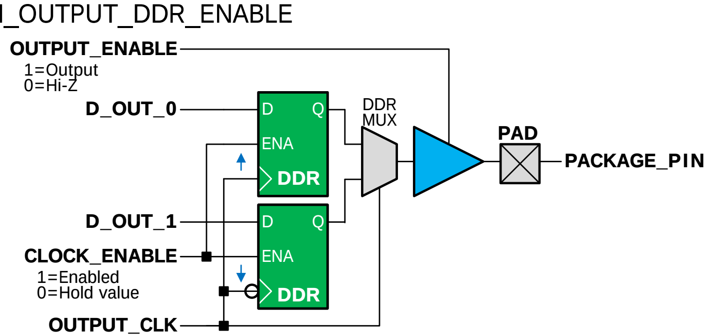
Global buffer inputs, defined using the SB_GB_IO primitive, are also full-featured PIO pins. However, if only the GLOBAL_BUFFER_OUTPUT is connected on the SB_GB_IO primitive, then the PIN_TYPE parameter has no real effect.

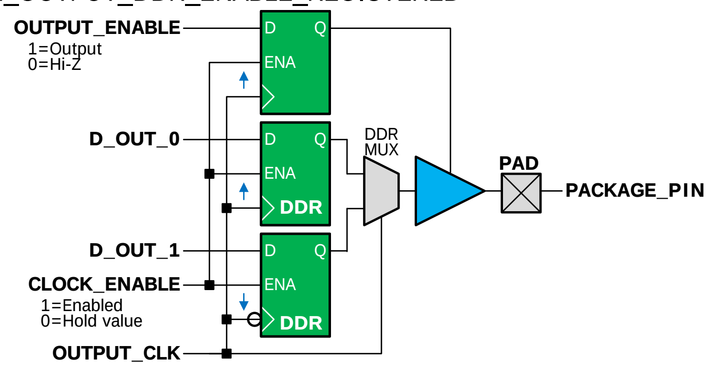
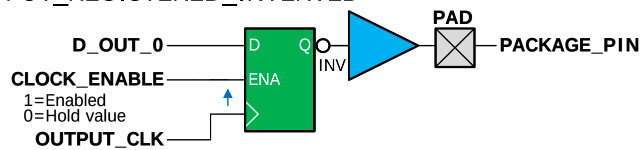
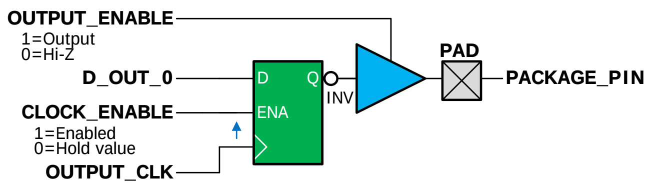
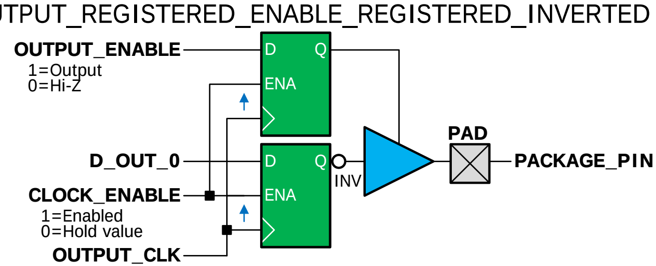
Output Field (PIN_TYPE[5:2])

Table 6 shows the various PIN_TYPE definitions for the output side an SB_IO or SB_GB_IO primitive.

Table 6: **Output Structures and PIN_TYPE Values**

Style	Mnemonic (Diagram)	PIN_TYPE[5:2]			
None (output disabled)	PIN_NO_OUTPUT 	0	0	0	0
Non-registered Output	PIN_OUTPUT 	0	1	1	0
	PIN_OUTPUT_TRISTATE 	1	0	1	0

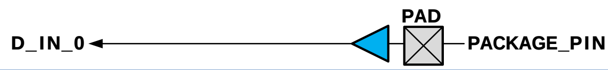
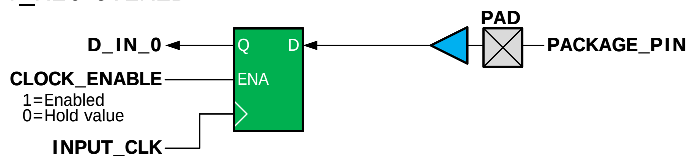
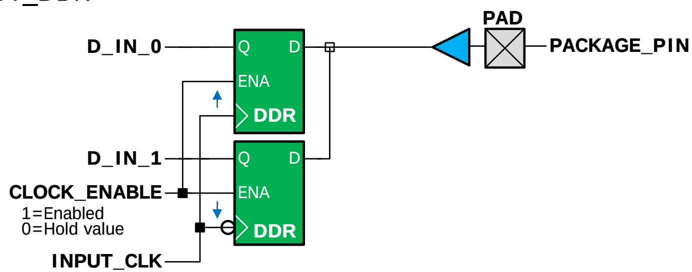
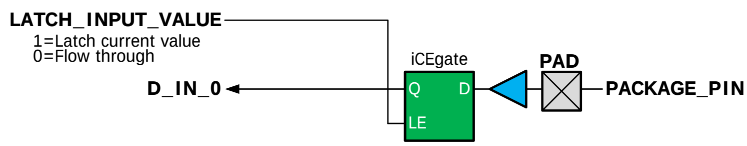
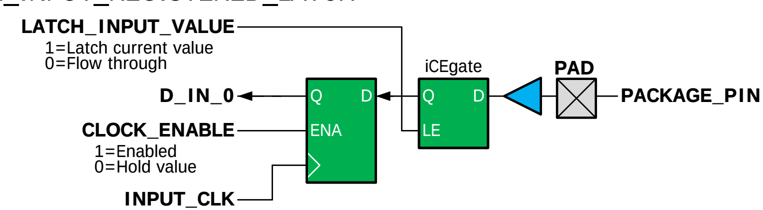
<table>
<tr><th>Style</th><th>Mnemonic (Diagram)</th><th colspan="4">PIN_TYPE[5:2]</th></tr>
<tr><td rowspan="4">Registered Outputs</td><td>PIN_OUTPUT_REGISTERED
</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>PIN_OUTPUT_REGISTERED_ENABLE
</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>PIN_OUTPUT_ENABLE_REGISTERED
</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>PIN_OUTPUT_REGISTERED_ENABLE_REGISTERED
</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td rowspan="2">Double Data Rage (DDR) Output</td><td>PIN_OUTPUT_DDR
</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>PIN_OUTPUT_DDR_ENABLE
</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Style	Mnemonic (Diagram)	PIN_TYPE[5:2]			
	PIN_OUTPUT_DDR_ENABLE_REGISTERED  	1	1	0	0
Registered Output, Inverted	PIN_OUTPUT_REGISTERED_INVERTED  	0	1	1	1
	PIN_OUTPUT_REGISTERED_ENABLE_INVERTED  	1	0	1	1
	PIN_OUTPUT_REGISTERED_ENABLE_REGISTERED_INVERTED  	1	1	1	1

Input Field (PIN_TYPE[1:0])

Table 6 shows the various PIN_TYPE definitions for the input side of an SB_IO or SB_GB_IO primitive.

Table 7: **Input Structures and PIN_TYPE Values**

Style	Mnemonic (Diagram)	PIN_TYPE[1:0]	
Direct	PIN_INPUT 	0	1
Registered	PIN_INPUT_REGISTERED 	0	0
Double Data Rage (DDR) Output	PIN_INPUT_DDR 	0	0
iCEgate Low-Power Latch	PIN_INPUT_LATCH 	1	1
	PIN_INPUT_REGISTERED_LATCH 	1	0

IO_STANDARD Parameter

Differential inputs or outputs require specific settings for the IO_STANDARD parameter, as summarized in Table 8. Regardless of actual electrical requirements (LVDS, SubLVDS), the IO_STANDARD parameter on differential inputs must be set to `SB_LVDS_INPUT`. Differential outputs are recommended for I/O Banks 0, 1, 2 on any iCE65/iCE65P FPGA or additionally for I/O Bank 3 of iCE65L01 FPGAs. Consequently, set IO_STANDARD to `SB_LVCMOS` for differential outputs. Characterization for differential outputs in I/O Bank 3 on non-iCE65L01 FPGA is ongoing.

Table 8: **IO_STANDARD Settings for Differential I/O**

	Differential Inputs (Not available on iCE65L01)	**Differential Outputs** (I/O Banks 0, 1, 2) (I/O Bank 3 on iCE65L01)	**Differential Outputs** (I/O Bank 3, except iCE65L01)
IO_STANDARD Setting	SB_LVDS_INPUT	SB_LVCMOS	Undergoing characterization

NEG_TRIGGER Parameter

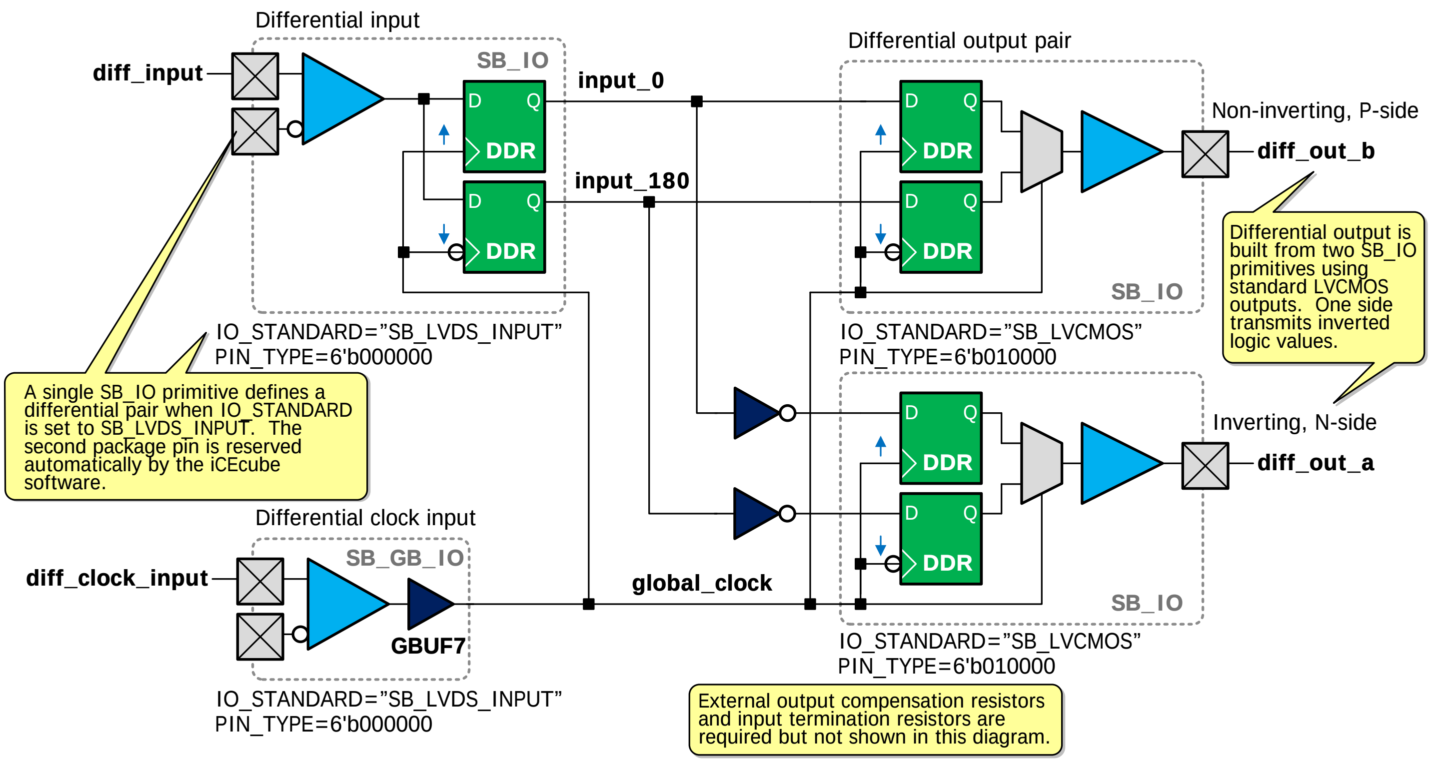
The optional NEG_TRIGGER parameter, when set to '1', inverts the clock polarity within the PIO pin.

HDL Implementation Example

Most applications that use differential I/O do so because they have high bandwidth requirements. Consequently, most of these applications use also Double Data Rate (DDR) flip-flops to double the effective data rate at the PIO pin.

The design example shown in Figure 10 uses the DDR flip-flops embedded in every PIO pin.

Figure 10: **Differential I/O Design Example**



VHDL Component Declaration

For VHDL implementations, the SB_IO and SB_GB_IO primitives must be declared as components before they are first used. Verilog does not require this declaration.

Under development.

Differential Clock Input

The following Verilog and VHDL code snippets demonstrate how to instantiate a differential clock input using an SB_GB_IO primitive. The differential clock input is always connected to Global Buffer GBIN7 and always in I/O Bank 3 (except on the iCE65L01 FPGA because it does not support differential inputs). In this example, shown in Figure 10, the PIO pin only connects an external differential clock signal to the FPGA's GBUF7 global buffer. The only ports connected on the primitive are the PACKAGE_PIN and the GLOBAL_BUFFER_OUTPUT ports. Consequently, the PIN_TYPE setting for this SB_GB_IO primitive does not actually matter.

Set the IO_STANDARD parameter to "`SB_LVDS_INPUT`". Doing so also causes the iCEcube software to reserve a second pin for the other side of the differential input pair.

An external 100 Ω termination resistor must be connected between the two inputs. The resistor must be physically placed as close as possible to the two package balls.

For VHDL implementations, the SB_GB_IO component must be declared.

The PIN_TYPE for this primitive may be different, depending on whether only the global buffer input is used, or if the data input paths are used, or both. The example shown here is for a dedicated differential clock input.

Verilog

```
// Differential clock input example: Verilog
// IMPORTANT:  The PIN_TYPE is different for a regular LVDS input used as a clock.
//             This example is specifically for the differential clock input.
defparam differential_clock_input.PIN_TYPE = 6'b000000 ; // {NO_OUTPUT, PIN_INPUT_REGISTERED}
defparam differential_clock_input.IO_STANDARD = "SB_LVDS_INPUT" ;
SB_GB_IO differential_clock_input (
   .PACKAGE_PIN(diff_clock_input),
   .LATCH_INPUT_VALUE ( ),
   .CLOCK_ENABLE ( ),
   .INPUT_CLK ( ),
   .OUTPUT_CLK ( ),
   .OUTPUT_ENABLE ( ),
   .D_OUT_0 ( ),
   .D_OUT_1 ( ),
   .D_IN_0 ( ),
   .D_IN_1 ( ),
   .GLOBAL_BUFFER_OUTPUT(global_clock) // Global buffer output
);
```

VHDL

Under development.

Differential Input

The following Verilog and VHDL code snippets demonstrate how to instantiate a differential input using an SB_ IO primitive. Differential inputs are always implemented in I/O Bank 3 (except on the iCE65L01 FPGA because it does not support differential inputs). In this example, shown in Figure 10, the differential input also connects to Double Data Rate (DDR) input flip-flops. There is no output connected. Consequently, set the PIN_TYPE parameter to the binary value "`000000`", which defines no output (see Table 6) and DDR input (see Table 7).

Set the IO_STANDARD parameter to "`SB_LVDS_INPUT`". Doing so also causes the iCEcube software to reserve a second pin for the other side of the differential input pair.

An external 100 Ω termination resistor must be connected between the two inputs. The resistor must be physically placed as close as possible to the two package balls.

For VHDL implementations, the SB_ IO component must be declared.

Verilog

```
// Differential input, DDR data
defparam differential_input.PIN_TYPE = 6'b000000 ; // {NO_OUTPUT, PIN_INPUT_DDR}
defparam differential_input.IO_STANDARD = "SB_LVDS_INPUT" ;
SB_IO differential_input (
   .PACKAGE_PIN(diff_input),
   .LATCH_INPUT_VALUE ( ),
   .CLOCK_ENABLE ( ),
   .INPUT_CLK (global_clock),
   .OUTPUT_CLK ( ),
   .OUTPUT_ENABLE ( ),
   .D_OUT_0 ( ),
   .D_OUT_1 ( ),
   .D_IN_0 (input_0),
   .D_IN_1 (input_180)
);
```

VHDL

Under development.

Differential Output Pair

The following Verilog and VHDL code snippets demonstrate how to instantiate a differential output pair using an SB_ IO primitive. Differential outputs are specified as two separate single-ended outputs. One output provides the non-inverted or P-side of the pair while the other output provides the inverted or N-side of the pair. In this example, shown in Figure 10, the differential output also connects to Double Data Rate (DDR) input flip-flops for higher output performance. There is no input connected. Consequently, set the PIN_TYPE parameter to the binary value "`010000`", which defines DDR output (see Table 6) and a benign input (see Table 7).

Differential outputs can be placed in any I/O bank, although the pair must be part of an I/O tile, as shown in the iCE65 or iCE65P data sheet. Because of better slew rate control, SiliconBlue recommends placing differential outputs in I/O Banks 0, 1, or 2 (or I/O Bank 3 on the iCE65L01 FPGA). Consequently, set the IO_STANDARD parameter to "`SB_LVCMOS`".

An external compensation resistor network must be connected between the two inputs. The resistors must be physically placed as close as possible to the two package balls.

For VHDL implementations, the SB_ IO component must be declared.

Verilog

```
// Differential output pair, DDR data
// Non-inverting, P-side of pair
defparam differential_output_b.PIN_TYPE = 6'b010000 ; // {PIN_OUTPUT_DDR, PIN_INPUT_REGISTER }
defparam differential_output_b.IO_STANDARD = "SB_LVCMOS" ;
SB_IO differential_output_b (
   .PACKAGE_PIN(diff_output_b),
   .LATCH_INPUT_VALUE ( ),
   .CLOCK_ENABLE ( ),
   .INPUT_CLK ( ),
   .OUTPUT_CLK (global_clock),
   .OUTPUT_ENABLE ( ),
   .D_OUT_0 (input_0),      // Non-inverted
   .D_OUT_1 (input_180),    // Non-inverted
   .D_IN_0 ( ),
   .D_IN_1 ( )
);

// Inverting, N-side of pair
defparam differential_output_a.PIN_TYPE = 6'b010000 ; // {PIN_OUTPUT_DDR, PIN_INPUT_REGISTER }
defparam differential_output_a.IO_STANDARD = "SB_LVCMOS" ;
SB_IO differential_output_a (
   .PACKAGE_PIN(diff_output_a),
   .LATCH_INPUT_VALUE ( ),
   .CLOCK_ENABLE ( ),
   .INPUT_CLK ( ),
   .OUTPUT_CLK (global_clock),
   .OUTPUT_ENABLE ( ),
   .D_OUT_0 (~input_0),     // Inverted
   .D_OUT_1 (~input_180),   // Inverted
   .D_IN_0 ( ),
   .D_IN_1 ( )
);
```

VHDL

Under development.

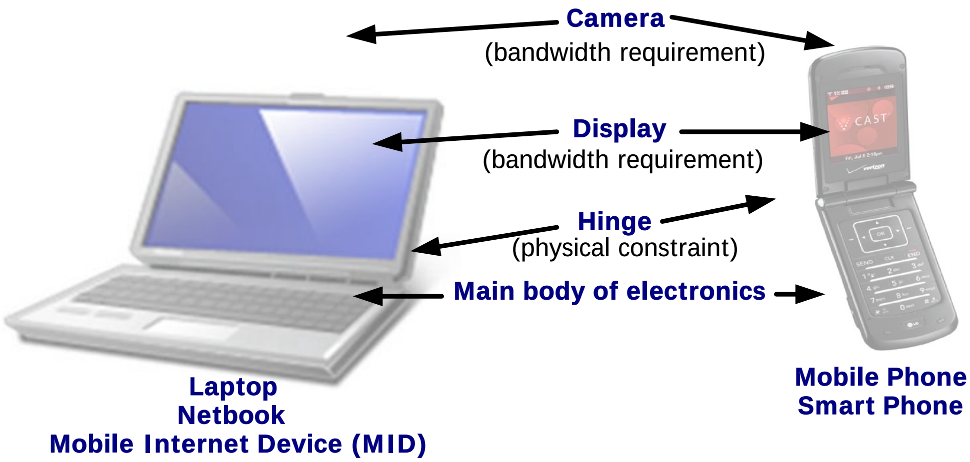
Applications

Applications that benefit most from differential I/O are those with high bandwidth communication requirements such as graphic displays, cameras and imagers, or chip-to-chip interfaces.

While driving such displays using single-ended LVCMOS I/O is possible, portable or hand-held applications place additional physical constraints on a design, as shown in Figure 11. Typically, the high bandwidth device—the graphic display or camera—is separate from the main body that holds the majority of the system electronics. The main body and the high-bandwidth device are mechanically connected by some sort of hinge mechanism. Sending a wide, LVCMOS signal cable bundle across the hinge to the display is simply impractical. Likewise, a custom, wide, flex-cable is prohibitively expensive.

The higher bandwidth possible with differential signaling allows the same data to be transported over fewer electrical connections. Few connections results in a smaller, lower-cost flexible cable. Likewise, the smaller voltage swing results in lower electromagnetic interference (EMI).

Figure 11: **Portable Devices Benefit from Differential I/O**



SiliconBlue iCE65/iCE65P FPGAs offer a broad range of possible solutions for handheld applications, primarily in bridging and format conversion applications.

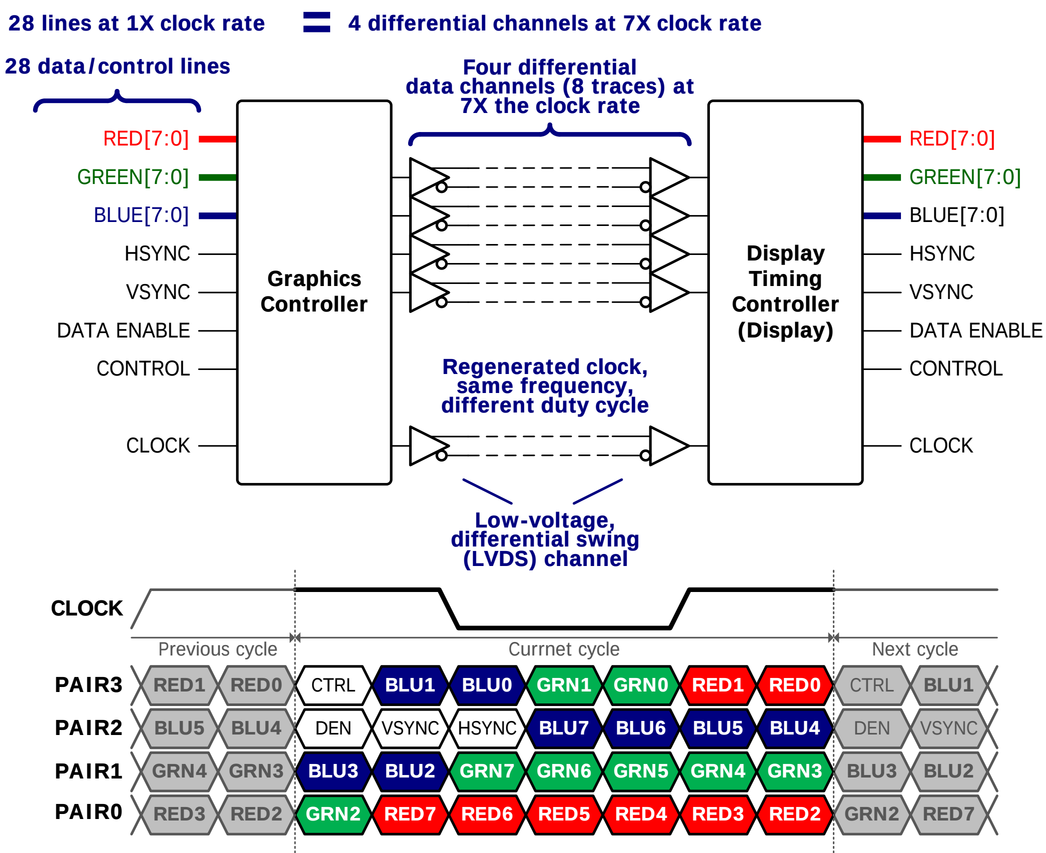
- Covert RGB data to high-speed differential data and back.
- Connect a processor without an integrated differential interface to a differential display or camera.
- Convert from one high-speed differential interface to another.
- Scale, rotate, and rebroadcast one stream onto another display format or another differential I/O format.

Graphic Displays

Graphic displays demand high data rates, especially high-resolution displays that support a broad color range. In portable or handheld applications, the challenge is to provide a high-bandwidth communications path between the graphics controller and the display that fits in the physical constraint of the hinge. There are a variety of standard interfaces that leverage differential switching. Perhaps the most widely-used example is Flat Panel Display Link (FDP-Link), shown in Figure 12. The example shows a 24-bit per pixel design, although there are other implementations that use fewer colors and differential pairs. A standard 24-bit RGB interface requires up to 28 single-ended signals. Instead of sending a cable bundle with 24 wires across the hinge, FPD-Link serializes the 28 data/control lines onto four differential I/O pairs, plus a clock differential pair resulting in 64% fewer wires. FPD-Link uses the Low-Voltage Differential Swing (LVDS) I/O standard.

Dividing 28 lines by four differential pairs also means that the data rate across the interface is seven times higher than the output clock rate.

Figure 12: **Example Differential I/O Solution: Flat Panel Display Interface**



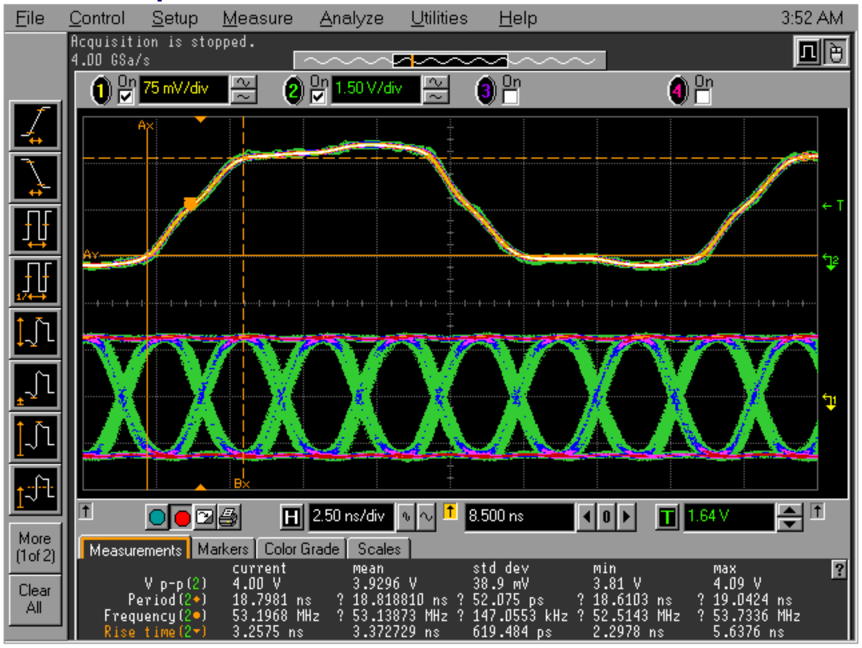
At the receiving end, the differential data is de-serialized and converted back into a wider bundle of single-ended signals.

The integrated PLL in iCE65P-family FPGAs simplifies this style of interface, although display drivers do not require the PLL. An iCE65 FPGA can be at either end of the serial interface, either to allow a processor without an FPD-Link interface to communicate with an FPD-Link display, or to allow a processor with only an FPD-Link display interface to communicate to an RGB display or a display that uses a different format.

Figure 13 shows an oscilloscope output from an example FPD-Link transmitter interface build on an iCE65P04 FPGA using the iCEman65P Evaluation Kit. The top-trace is the clock, here operating at 53 MHz. The iCE65P FPGA's PLL multiplies the clock by 3.5 times to generate an internal 185.5 MHz clock. Data is clocked onto the LVDS outputs using Double Data Rate (DDR) flip-flops, so the resulting data rate is 371 million bits per second, which consequently exceeds the maximum output data rate of 350 million bits per second.

- AN014: iCE65™ mobileFPGA™ as an LVDS, FPD-Link Display Driver
  www.siliconbluetech.com/media/downloads/SiliconBlue_AN014_LVDS_Display.pdf

Figure 13: **Example FPD-Link Transmitter on iCEman65P Evaluation Board**



Cameras and Imagers

Differential interfaces are also popular for high-resolution and high-speed cameras and imagers.

Summary

This application note provides an overview of iCE LVDS technology, focusing on its advantages, implementation, application, and its various electrical and timing characteristics. It also includes detailed recommendations for instantiating LVDS transmitter and receivers in your design and calculating the required external terminations to guarantee optimum performance.

References

- IEEE 1596.3 Standards,
  ieeexplore.ieee.org/iel1/3807/11142/00503756.pdf
- National Semiconductor, "LVDS Owner's Manual", National Semiconductor, 2008,
  www.national.com/analog/interface/lvds_owners_manual
- Jimmy Ma, "A Closer Look at LVDS Technology", Pericom, 2001
  www.pericom.com/pdf/applications/AN041.pdf
- National Semiconductor, "An Introduction to FPD-Link",
  www.national.com/an/AN/AN-1032.pdf
- National Semiconductor, "LVDS Display Interface (LDI) TFT Data Mapping for Interoperability with FPD-Link",
  www.national.com/an/AN/AN-1127.pdf

Disclaimer

Revision History

Version	Date	Description
2.1	1-OCT-2010	Updated with recommended maximum output frequency for LVDS and SubLVDS applications.
2.0	24-MAY-2010	Major update of previous version.

SiliconBlue®

SiliconBlue Technologies Corporation

3255 Scott Blvd.
Building 7, Suite 101
Santa Clara, California 95054
United States of America

Tel: +1 408-727-6101
Fax: +1 408-727-6085

www.SiliconBlueTech.com