Using Digitally Controlled Impedance (DCI)

Introduction

As FPGAs get bigger and system clock speeds get faster, PCB board design and manufacturing has become more difficult. With ever faster edge rates, maintaining signal integrity becomes a critical issue. Designers must make sure that most PC board traces are terminated properly to avoid reflections or ringing.

To terminate a trace, resistors are traditionally added to make the output and/or input match the impedance of the receiver or driver to the impedance of the trace. However, due to the increase in the device I/O counts, adding resistors close to the device pins increases the board area and component count and might even be physically impossible. To address these issues and to achieve better signal integrity, Xilinx developed a new I/O technology for the Virtex-II device family, Digitally Controlled Impedance (DCI).

DCI adjusts the output impedance or input termination to accurately match the characteristic impedance of the transmission line. DCI actively adjusts the impedance of the I/O to equal an external reference resistance. This compensates for changes in I/O impedance due to process variation. It also continuously adjusts the impedance of the I/O to compensate for variations of temperature and supply voltage fluctuations.

In the case of controlled impedance drivers, DCI controls the driver impedance to match two reference resistors, or optionally, to match half the value of these reference resistors. DCI eliminates the need for external termination resistors.

DCI provides the termination for transmitters or receivers. This eliminates the need for termination resistors on the board, reduces board routing difficulties and component count, and improves signal integrity by eliminating stub reflection. Stub reflection occurs when termination resistors are located too far from the end of the transmission line. With DCI, the termination resistors are as close as possible to the output driver or the input buffer, thus, eliminating stub reflections completely.

Xilinx DCI

DCI uses two multi-purpose reference pins in each bank to control the impedance of the driver or the parallel termination value for all of the I/Os of that bank. The N reference pin (VRN) must be pulled up to V_{CCO} by a reference resistor, and the P reference pin (VRP) must be pulled down to ground by another reference resistor. The value of each reference resistor should be equal to the characteristic impedance of the PC board traces, or should be twice that value (configuration option).

When a DCI I/O standard is used on a particular bank, the two multi-purpose reference pins cannot be used as regular I/Os. however, if DCI I/O standards are not used in the bank, these pins are available as regular I/O pins. Check the Virtex-II pinout for detailed pin descriptions.

DCI adjusts the impedance of the I/O by selectively turning transistors in the I/Os on or off. The impedance is adjusted to match the external reference resistors. The impedance adjustment process has two phases. The first phase, which compensates for process variations, is done during the device startup sequence. The second phase, which maintains the impedance in response to temperature and supply voltage changes, begins immediately after the first phase and continues indefinitely, even while the part is operating. By default, the DONE pin does not go High until the impedance adjustment process has completed.

For controlled impedance output drivers, the impedance can be adjusted either to match the reference resistors or half the resistance of the reference resistors. For on-chip termination, the termination is always adjusted to match the reference resistors.

DCI can configure output drivers to be the following types:

1. Controlled Impedance Driver (Source Termination)
2. Controlled Impedance Driver with Half Impedance (Source Termination)

It can also configure inputs to have he following types of on-chip terminations:

1. Termination to V_{CCO} (Single Termination)
2. Termination to $V_{CCO}/2$ (Split Termination, Thevenin equivalent)

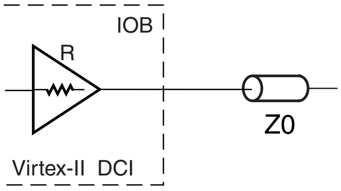
For bidirectional operation, both ends of the line can be DCI-terminated permanently:

1. Termination to V_{CCO} (Single Termination)
2. Termination to $V_{CCO}/2$ (Split Termination, Thevenin equivalent)

Alternatively, bidirectional point-to-point lines can use controlled-impedance drivers (with 3-state buffers) on both ends.

Controlled Impedance Driver (Source Termination)

Some I/O standards, such as LVTTL, LVCMOS, etc., must have a drive impedance that matches the characteristic impedance of the driven line. DCI can provide a controlled impedance output drivers that eliminate reflections without an external source termination. The impedance is set by the external reference resistors, whose resistance should be equal to the trace impedance. Figure 2-96 illustrates a controlled impedance driver inside Virtex-II device. The DCI I/O standards that support Controlled Impedance Driver are: LVDCI_15, LVDCI_18, LVDCI_25, and LVDCI_33.



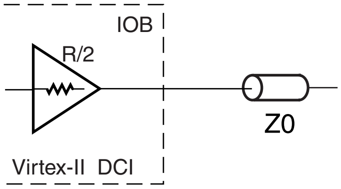
UG002_c2_047_120400

Figure 2-96: **Controlled Impedance Driver**

Controlled Impedance Driver With Half Impedance (Source Termination)

DCI can also provide drivers with one half of the impedance of the reference resistors. The DCI I/O standards that support controlled impedance driver with half impedance are: LVDCI_DV2_15, LVDCI_DV2_18, LVDCI_DV2_25, and LVDCI_DV2_33

Figure 2-97 illustrates a controlled driver with half impedance inside a Virtex-II device.

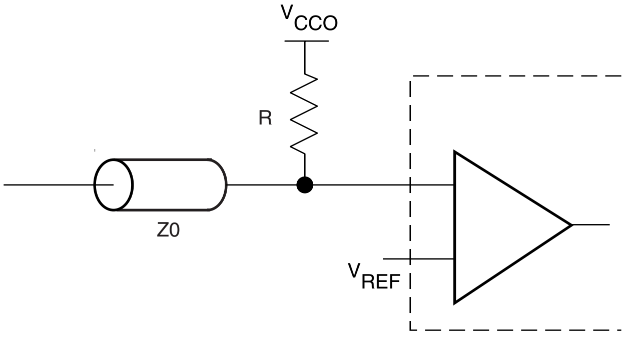


UG002_c2_052_120400

Figure 2-97: **Controlled Impedance Driver With Half Impedance**

Termination to V_{CCO} (Single Termination)

Some I/O standards, such as HSTL Class III, IV, etc., require an input termination to V_{CCO}. See Figure 2-98.

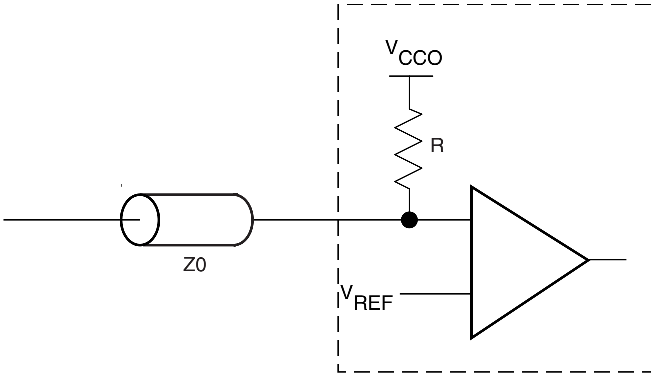


ug002_c2_053_110600

Figure 2-98: **Single Termination Without DCI**

DCI can provide this termination to V_{CCO} using single termination. The termination resistance is set by the reference resistors. For GTL and HSTL standards, they should be controlled by 50-ohm reference resistors. The DCI I/O standards that support single termination are: GTL_DCI, GTLP_DCI, HSTL_III_DCI, and HSTL_IV_DCI.

Figure 2-99 illustrates single termination inside a Virtex-II device.



ug002_c2_054_110600

Figure 2-99: **Single Termination Using DCI**

Termination to V_{CCO}/2 (Split Termination)

Some I/O standards, such as HSTL Class I, II, SSTL3_I, etc., require an input termination voltage of V_{CCO}/2. See Figure 2-100.

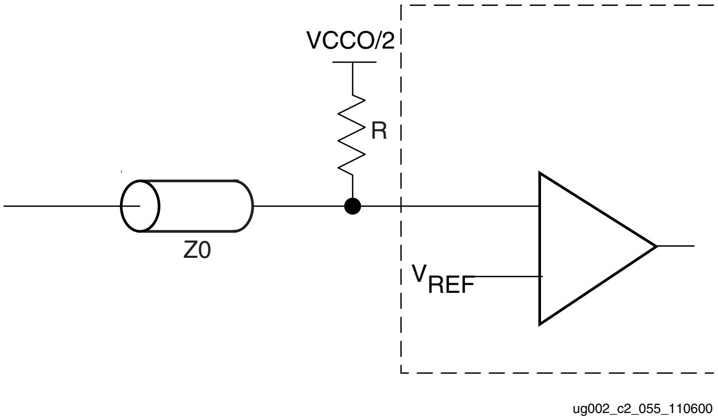


Figure 2-100: **Split Termination Without DCI**

This is equivalent to having a split termination composed of two resistors. One terminates to V_{CCO}, the other to ground. The resistor values are 2R. DCI provides termination to V_{CCO}/2 using split termination. The termination resistance is set by the external reference resistors, i.e., the resistors to V_{CC} and ground are each twice the reference resistor value. If users are planning to use HSTL or SSTL standards, the reference resistors should be 50-ohms. The DCI I/O standards that support split termination are: HSTL_I_DCI, HSTL_II_DCI, SSTL2_I_DCI, SSTL2_II_DCI, SSTL3_I_DCI, and SSTL3_II_DCI.

Figure 2-101 illustrates split termination inside a Virtex-II device.

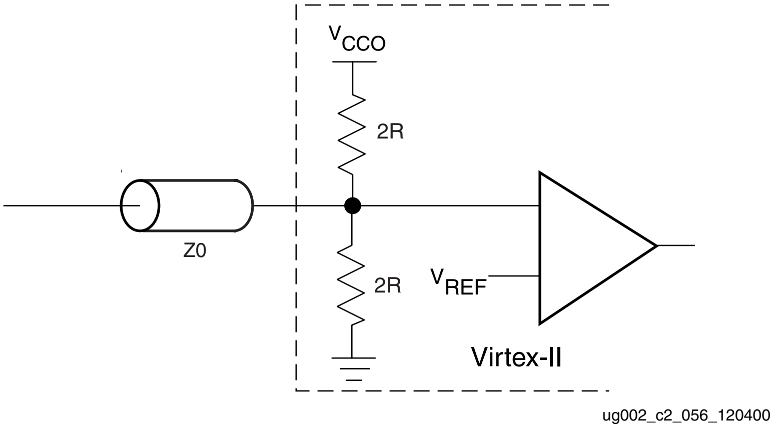


Figure 2-101: **Split Termination Using DCI**

Driver With Single Termination

Some I/O standards, such as HSTL Class IV, require an output termination to V_{CCO}. Figure 2-102 illustrates the output termination to V_{CCO}.

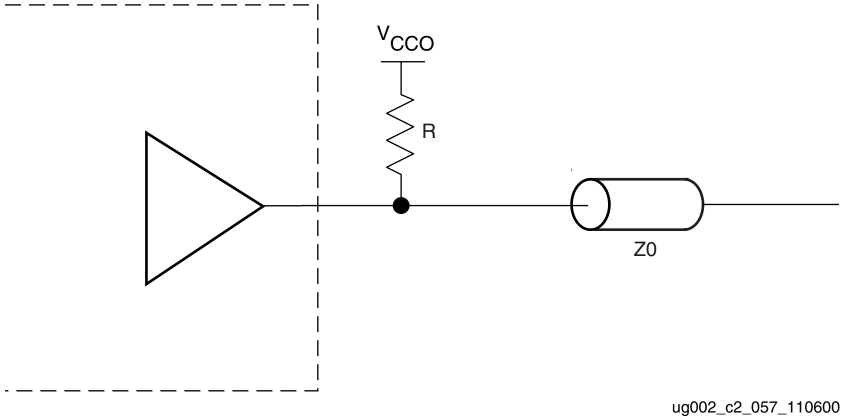


Figure 2-102: **Driver With Single Termination Without DCI**

DCI can provide this termination to V_{CCO} using single termination. In this case, DCI only controls the impedance of the termination, but not the driver. If users are planning to use GTL or HSTL standards, the external reference resistors should be 50-ohms. The DCI I/O standards that support a driver with single termination are: GTL_DCI, GTLP_DCI, and HSTL_IV_DCI.

Figure 2-103 illustrates a driver with single termination inside a Virtex-II device

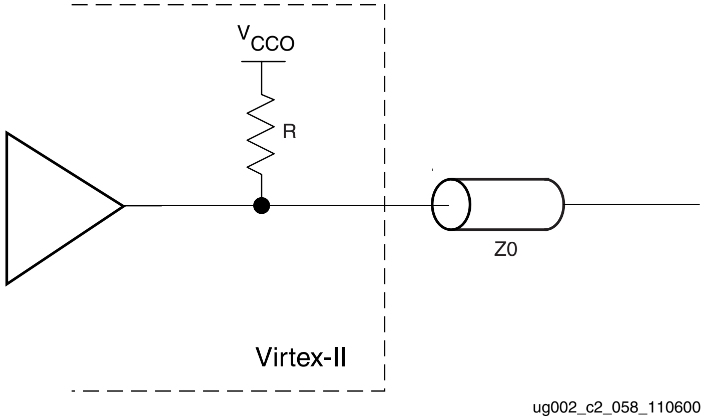
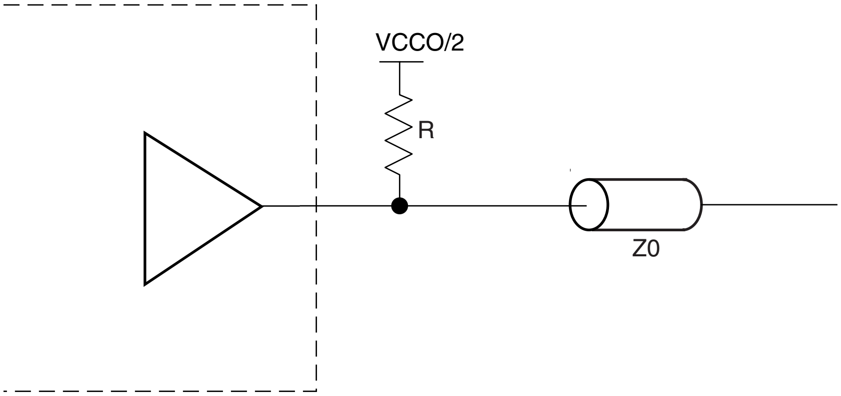


Figure 2-103: **Driver With Single Termination Using DCI**

Driver With Split Termination

Some I/O standards, such as HSTL Class II, require an output termination to $V_{CCO}/2$. See Figure 2-104.

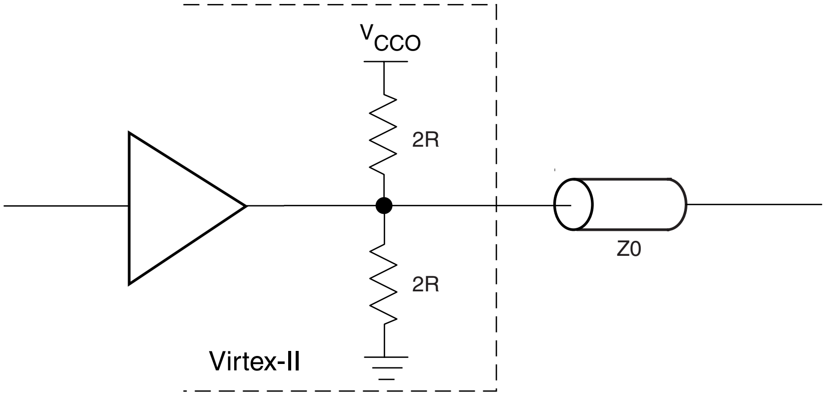


ug002_c2_059_110600

Figure 2-104: **Driver With Split Terminating**

DCI can provide this termination to $V_{CCO}/2$ using split termination. It only controls the impedance of the termination, but not the driver. For HSTL or SSTL standards, the external reference resistors should be 50-ohms. The DCI I/O standards that support a Driver with split termination are: HSTL_II_DCI, SSTL2_II_DCI, and SSTL3_II_DCI.

Figure 2-105 illustrates a driver with split termination inside a Virtex-II device.



ug002_c2_060_110600

Figure 2-105: **Driver With Split Termination Using DCI**

Software Support

This section lists the valid DCI I/O buffer library components and describes how to use DCI in the Xilinx software.

DCI I/O Buffer Library Components

The DCI input buffer library components, including global clock buffer, are the following:

- IBUFG_GTLP_DCI
- IBUFG_GTL_DCI
- IBUFG_HSTL_I_DCI
- IBUFG_HSTL_II_DCI
- IBUFG_HSTL_III_DCI
- IBUFG_HSTL_IV_DCI
- IBUFG_LVDCI_15
- IBUFG_LVDCI_18
- IBUFG_LVDCI_25
- IBUFG_LVDCI_33
- IBUFG_LVDCI_DV2_15
- IBUFG_LVDCI_DV2_18
- IBUFG_LVDCI_DV2_25
- IBUFG_LVDCI_DV2_33
- IBUFG_SSTL2_I_DCI
- IBUFG_SSTL2_II_DCI
- IBUFG_SSTL3_I_DCI
- IBUFG_SSTL3_II_DCI
- IBUF_GTLP_DCI
- IBUF_GTL_DCI
- IBUF_HSTL_I_DCI
- IBUF_HSTL_II_DCI
- IBUF_HSTL_III_DCI
- IBUF_HSTL_IV_DCI
- IBUF_LVDCI_15
- IBUF_LVDCI_18
- IBUF_LVDCI_25
- IBUF_LVDCI_33
- IBUF_LVDCI_DV2_15
- IBUF_LVDCI_DV2_18
- IBUF_LVDCI_DV2_25
- IBUF_LVDCI_DV2_33
- IBUF_SSTL2_I_DCI
- IBUF_SSTL2_II_DCI
- IBUF_SSTL3_I_DCI
- IBUF_SSTL3_II_DCI

The following are DCI output buffer library components:

- OBUF_GTLP_DCI
- OBUF_GTL_DCI
- OBUF_HSTL_I_DCI
- OBUF_HSTL_II_DCI
- OBUF_HSTL_III_DCI
- OBUF_HSTL_IV_DCI
- OBUF_LVDCI_15
- OBUF_LVDCI_18
- OBUF_LVDCI_25
- OBUF_LVDCI_33
- OBUF_LVDCI_DV2_15
- OBUF_LVDCI_DV2_18
- OBUF_LVDCI_DV2_25
- OBUF_LVDCI_DV2_33
- OBUF_SSTL2_I_DCI
- OBUF_SSTL2_II_DCI
- OBUF_SSTL3_I_DCI
- OBUF_SSTL3_II_DCI

The following are DCI 3 state output buffer library components:

- OBUFT_GTLP_DCI
- OBUFT_GTL_DCI
- OBUFT_HSTL_I_DCI
- OBUFT_HSTL_II_DCI
- OBUFT_HSTL_III_DCI
- OBUFT_HSTL_IV_DCI
- OBUFT_LVDCI_15
- OBUFT_LVDCI_18
- OBUFT_LVDCI_25
- OBUFT_LVDCI_33
- OBUFT_LVDCI_DV2_15
- OBUFT_LVDCI_DV2_18
- OBUFT_LVDCI_DV2_25
- OBUFT_LVDCI_DV2_33
- OBUFT_SSTL2_I_DCI
- OBUFT_SSTL2_II_DCI
- OBUFT_SSTL3_I_DCI
- OBUFT_SSTL3_II_DCI

The following are DCI I/O buffer library components:

- IOBUF_GTLP_DCI
- IOBUF_GTL_DCI
- IOBUF_HSTL_II_DCI
- IOBUF_HSTL_IV_DCI
- IOBUF_SSTL2_II_DCI
- IOBUF_SSTL3_II_DCI
- IOBUF_LVDCI_15
- IOBUF_LVDCI_18
- IOBUF_LVDCI_25
- IOBUF_LVDCI_33
- IOBUF_LVDCI_DV2_15
- IOBUF_LVDCI_DV2_18
- IOBUF_LVDCI_DV2_25
- IOBUF_LVDCI_DV2_33

How to Use DCI in the Software

There are two ways for users to use DCI for Virtex-II devices:

1. Use the IOSTANDARD attribute in the constraint file.
2. Instantiate DCI input or output buffers in the HDL code.

IOSTANDARD Attribute

The IOSTANDARD attribute can be entered through the NCF or UCF file. The syntax is as follows:

```
NET <net name> IOSTANDARD = LVDCI_25;
```

Where <net name> is the name between the IPAD and IBUF or OPAD or OBUF. For HDL designs, this name is the same as the port name.

The following are valid DCI attributes for output drivers:

- LVDCI_15
- LVDCI_18
- LVDCI_25
- LVDCI_33
- LVDCI_DV2_15
- LVDCI_DV2_15
- LVDCI_DV2_25
- LVDCI_DV2_33

The following are valid DCI attributes for terminations:

- GTL_DCI
- GTLP_DCI
- HSTL_I_DCI
- HSTL_II_DCI

- HSTL_III_DCI
- HSTL_IV_DCI
- SSTL2_I_DCI
- SSTL2_II_DCI
- SSTL3_I_DCI
- SSTL3_II_DCI

VHDL Example

Instantiating DCI input and output buffers is the same as instantiating any other I/O buffers. Users must make sure that the correct I/O buffer names are used and follow the standard syntax of instantiation.

For example, to instantiate a HSTL Class I output DCI buffer, the following syntax can be used:

```
HSTL_DCI_buffer: OBUF_HSTL_I_DCI port map (I=>data_out, O=>data_out_DCI);
```

Below is an example VHDL code that instantiates four 2.5 V LVDCI drivers and four HSTL Class I outputs.

```
-- Module: DCI_TEST
--
-- Description: VHDL example for DCI SelectI/O
-- Device: Virtex-II Family
---------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity dci_test is
port (clk, reset, ce, control : in std_logic;
  A, B : in std_logic_vector (3 downto 0);
  Dout : out std_logic_vector (3 downto 0);
  muxout : out std_logic_vector (3 downto 0));
end dci_test;

architecture dci_arch of dci_test is

--DCI output buffer component declaration
component OBUF_LVDCI_25 port (I : in std_logic; O : out std_logic);
end component;
attribute syn_black_box of OBUF_LVDCI_25 : component is true;
attribute black_box_pad_pin of OBUF_LVDCI_25 : component is "O";

--HSTL Class I DCI output buffer component declaration
component OBUF_HSTL_I_DCI port (I : in std_logic; O: out std_logic);
end component;
attribute syn_black_box of OBUF_HSTL_I_DCI : component is true;
attribute black_box_pad_pin of OBUF_HSTL_I_DCI : component is "O";

signal muxout_int : std_logic_vector (3 downto 0);
signal dout_int : std_logic_vector (3 downto 0);

begin

process (clk, reset)
begin
  if (reset = '1') then
         dout_int<="0000";
    elsif (clk'event and clk='1') then
```

```
      dout_int<=dout_int+1;
end if;
end process;

process (controls, A, B, DOUT_INT)

begin
  if (control='1') then
    muxout_int<=A and B;
else
    muxout_int<=Dout_int;
  end if;
end process;

U0 : OBUF_LVDCI_25 port map(
  I=>dout_int(0),
  O=>dout(0));

U1 : OBUF_LVDCI_25 port map(
       I=>dout_int(1),
       O=>dout(1));
U2 : OBUF_LVDCI_25 port map(
       I=>dout_int(2),
       O=>dout(2));
U3 : OBUF_LVDCI_25 port map(
       I=>dout_int(3),
       O=>dout(3));

K0 : OBUF_HSTL_I_DCI port map(
  I=>muxout_int(0),
  O=>muxout(0));

K1 : OBUF_HSTL_I_DCI port map(
       I=>muxout_int(1),
       O=>muxout(1));
K2 : OBUF_HSTL_I_DCI port map(
       I=>muxout_int(2),
       O=>muxout(2));
K3 : OBUF_HSTL_I_DCI port map(
       I=>muxout_int(3),
       O=>muxout(3));

end dci_arch;
```

DCI in Virtex-II Hardware

DCI only works with certain single-ended I/O standards and does not work with any differential I/O standard. DCI supports the following Virtex-II standards:

LVDCI, LVDCI_DV2, GTL_DCI, GTLP_DCI, HSTL_I_DCI, HSTL_II_DCI, HSTL_III_DCI, HSTL_IV_DCI, SSTL2_I_DCI, SSTL2_II_DCI, SSTL3_I_DCI, and SSTL3_II_DCI.

To correctly use DCI in a Virtex-II device, users must follow the following rules:

1. V_{CCO} pins must be connected to the appropriate V_{CCO} voltage based on the IOSTANDARDs in that bank.
2. Correct DCI I/O buffers must be used in the software either by using IOSTANDARD attributes or instantiations in the HDL code.
3. External reference resistors must be connected to multi-purpose pins (VRN and VRP) in the bank cannot be used as regular I/Os. Refer to the Virtex-II pinouts for the

specific pin locations. Pin VRN must be pulled up to V_{CCO} by its reference resistor. Pin VRP must be pulled down to ground by its reference resistor.

4. The value of the external reference resistors should be selected to give the desired output impedance. If using GTL_DCI, HSTL_DCI, or SSTL_DCI I/O standards, then they should be 50 ohms.
5. The values of the reference resistors must be within the supported range. Availability of this range is planned for the next release of the **Virtex-II Data Sheet**. (~30 to 100 Ω)
6. Follow the DCI I/O banking rules.

The DCI I/O banking rules are the following:

1. V_{REF} must be compatible for all of the inputs in the same bank.
2. V_{CCO} must be compatible for all of the inputs and outputs in the same bank.
3. No more than one DCI I/O standard using Single Termination type is allowed per bank.
4. No more than one DCI I/O standard using Split Termination type is allowed per bank.
5. Single Termination and Split Termination, Controlled Impedance Driver, and Controlled Impedance Driver with Half Impedance can co-exist in the same bank.

The behavior of DCI 3-state outputs is as follows:

If a LVDCI or LVDCI_DV2 driver is in 3-state, the driver is 3-stated. If a Driver with Single or Split Termination is in 3-state, the driver is 3-stated but the termination resistor remains.

The following section lists any special care actions that must be taken for each DCI I/O standard.

LVDCI_15, LVDCI_18, LVDCI_25, LVDCI_33

Using these buffers configures the outputs as controlled impedance drivers. The number extension at the end indicates the V_{CCO} voltage that should be used. For example, 15 means V_{CCO}=1.5 V, etc. There is no slew rate control or drive strength settings for LVDCI drivers.

LVDCI_DV2_15, LVDCI_DV2_18, LVDCI_DV2_25, LVDCI_DV_33

Using these buffers configures the outputs as controlled drivers with half impedance. The number extension at the end indicates the V_{CCO} voltage that should be used. For example, 15 means V_{CCO}=1.5 V, etc. There is no slew rate control or drive strength settings for LVDCI_DV2 drivers.

GTL_DCI

GTLP does not require a V_{CCO} voltage. However, for GTL_DCI, V_{CCO} must be connected to 1.2 V. GTL_DCI provides single termination to V_{CCO} for inputs or outputs.

GTLP_DCI

GTL+ does not require a V_{CCO} voltage. However, for GTLP_DCI, V_{CCO} must be connected to 1.5 V. GTLP_DCI provides single termination to V_{CCO} for inputs or outputs.

HSTL_ I_DCI, HSTL_ III_DCI

HSTL_I_DCI provides split termination to $V_{CCO}/2$ for inputs. HSTL_III_DCI provides single termination to V_{CCO} for inputs.

HSTL_ II_DCI, HSTL_ IV_DCI

HSTL_II_DCI provides split termination to $V_{CCO}/2$ for inputs or outputs. HSTL_IV_ DCI provides single termination to V_{CCO} for inputs or outputs.

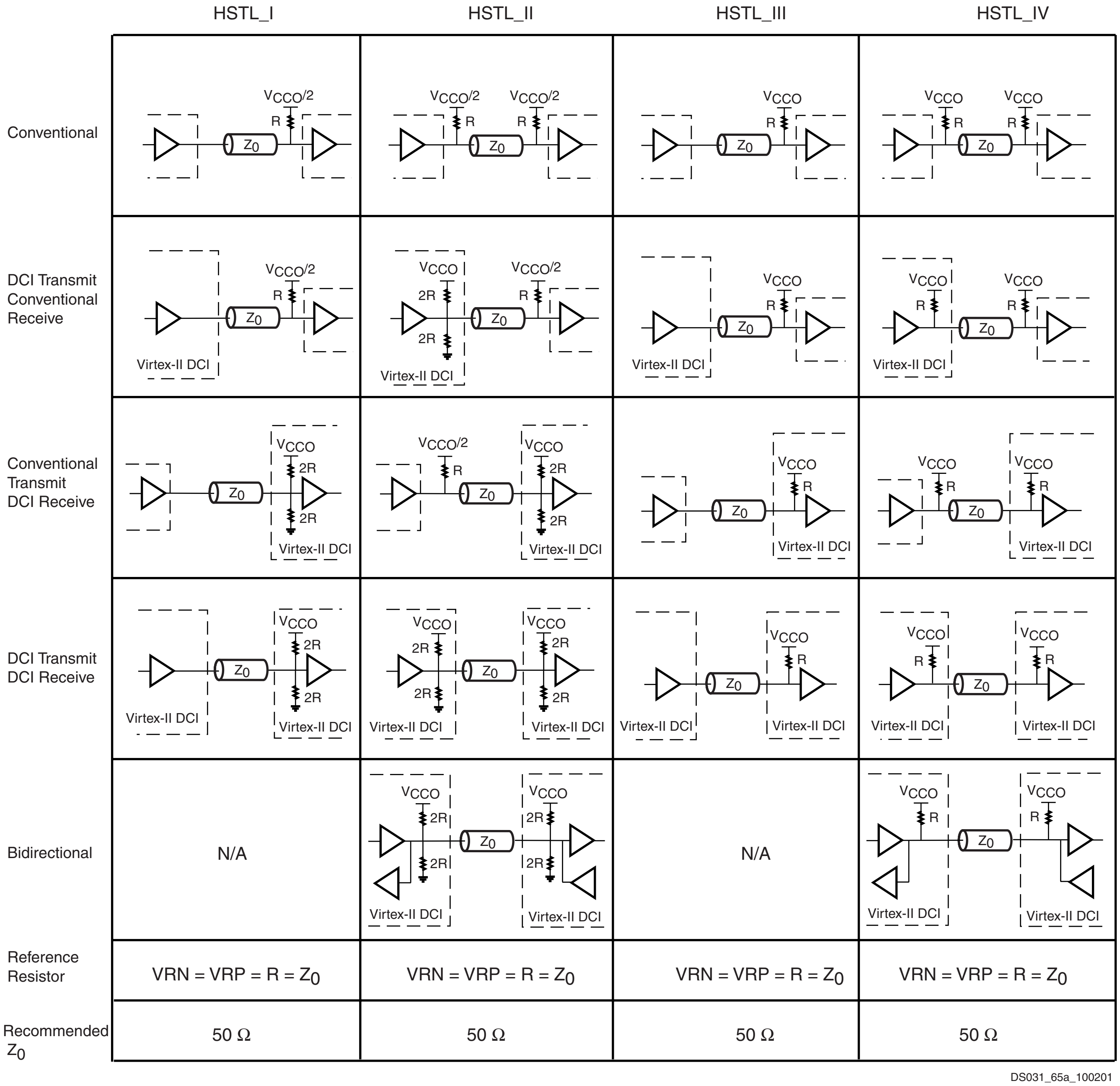
SSTL2_ I_DCI, SSTL3_I_DCI

SSTL2_I_DCI and SSTL3_I_DCI provide split termination to $V_{CCO}/2$ for inputs. Then I/O standards are SSTL compatible.

SSTL2_II_DCI, SSTL3_II_DCI

SSTL2_II_DCI and SSTL3_II_DCI provide split termination to $V_{CCO}/2$ for inputs. Then I/O standards are SSTL compatible.

Figure 2-106 provides examples illustrating the use of the HSTL_I_DCI, HSTL_II_DCI, HSTL_III_DCI, and HSTL_IV_DCI I/O standards.



	HSTL_I	HSTL_II	HSTL_III	HSTL_IV
Bidirectional	N/A		N/A	
Reference Resistor	VRN = VRP = R = Z_0	VRN = VRP = R = Z_0	VRN = VRP = R = Z_0	VRN = VRP = R = Z_0
Recommended Z_0	50 Ω	50 Ω	50 Ω	50 Ω

Figure 2-106: **HSTL DCI Usage Examples**

Figure 2-107 provides examples illustrating the use of the SSTL2_I_DCI, SSTL2_II_DCI, SSTL3_I_DCI, and SSTL3_II_DCI I/O standards.

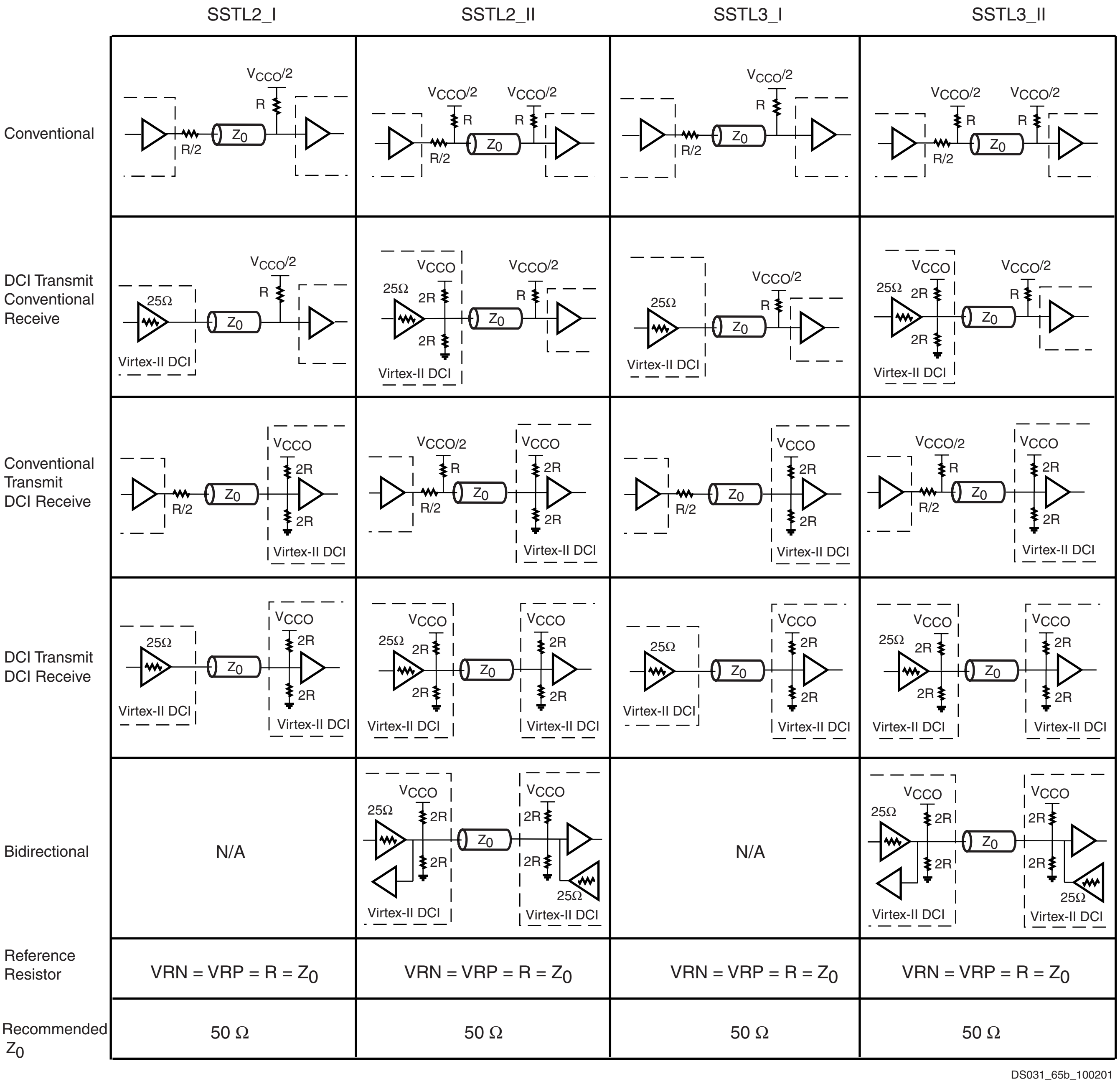


Figure 2-107: **SSTL DCI Usage Examples**