JEDEC STANDARD

Multi-wire Multi-level I/O Standard

JESD247

JUNE 2016

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION

NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Board of Directors level and subsequently reviewed and approved by the JEDEC legal counsel.

JEDEC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint. Within the JEDEC organization there are procedures whereby a JEDEC standard or publication may be further processed and ultimately become an ANSI standard.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC at the address below, or call (703) 907-7559 or www.jedec.org

Published by
©JEDEC Solid State Technology Association 2016
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

PRICE: Please refer to the current
Catalog of JEDEC Engineering Standards and Publications online at
http://www.jedec.org/Catalog/catalog.cfm

PLEASE!

DON'T VIOLATE
THE
LAW!

JEDEC Solid State Technology Association
3103 North 10th Street, Suite 240S
Arlington, Virginia 22201
or call (703) 907-7559

MULTI-WIRE MULTI-LEVEL I/O SPECIFICATION

Contents

	Page
1 Scope	1
2 Multi-Wire Signaling Codes	2
2.1 General Architecture	2
2.2 Signal Levels	3
2.3 ENRZ Code Definition	4
2.4 CNRZ-5 Code Definition	5
3 Driver Specifications	7
3.1 Driver Test Load	8
3.2 Transmitter Return Loss	9
3.3 Common Mode Noise	9
3.4 Driver Linearity and Jitter Tests (Quaternary Drivers)	10
4 Receiver Specifications	12
4.1 Receiver Return Loss	13
4.2 Receiver Linearity and Jitter Tolerance Tests	13
5 HSpice Modeling	17
5.1 Multi-Wire Transmitters Using Quaternary Drivers	17

MULTI-WIRE MULTI-LEVEL I/O SPECIFICATION

(From JEDEC Board Ballot JCB-16-21, formulated under the cognizance of the JC-16 Committee on Interfaces Technology.)

1 Scope

This standard defines the DC and AC operating conditions, I/O impedances, termination characteristics, and compliance test methods of I/O drivers and receivers used in multi-wire, multi-level signaling interfaces. The multi-wire interfaces defined by this specification all utilize quaternary signal levels.

Multi-wire signaling encodes *n*-bits of data per symbol onto an interface consisting of *m*-wires and utilizing quaternary signal levels. Differential signaling represents a simple multi-wire code where $m = 2$. Multi-wire signaling interfaces with $m > 2$ provide noise immunity and signal integrity characteristics similar to differential signals, but with higher throughput per wire.

This standard defines (1) several multi-wire signaling codes that are supported by this standard, (2) specifications for I/O drivers and receivers that support these codes, and (3) compliance test methods used to test interfaces that utilize these I/O drivers and receivers. The specifications and compliance methods in this standard are intended to be sufficient to ensure the interoperability of devices from different manufacturers. Where possible, these test methods incorporate existing techniques used by the industry to test similar devices.

This standard defines I/O drivers and receivers that are consistent with use for a minimal loss, low skew channel between two devices that are mounted immediately next to each other within a multi-chip module. Specifications that use this standard by reference will define the maximum baud rate and channel insertion loss based on the requirements of the target application. Specifications for applications which drive outside of a package to a DIMM may specify additional requirements for channel loss, channel skew, channel crosstalk, supply offset, transmit equalization, receive equalization, and skew tolerance.

2 Multi-Wire Signaling Codes

Multi-wire signaling encodes *n*-bits of data per symbol onto an interface consisting of *m*-wires, where $m > 2$. These wires are designated as $w_0, w_1, \ldots w_{m-1}$. Each wire is driven to one of *s* wire states, where each wire state corresponds to a unique single-ended signal level. For the codes listed in this specification, the sum of the wire states across the *m* wires is zero (i.e. common mode voltage is constant), and therefore the interface exhibits noise immunity qualities similar to that of a differential signal, but with higher throughput per wire.

One method of classifying multi-wire signaling codes is by the values of *n* and *m*. The designation of *n*b*m*w, where *n* and *m* are replaced by their corresponding values, is a shorthand notation for specifying the number of bits encoded in each baud symbol and the number of interface wires on which the baud symbol is encoded, such that *n*b*m*w means that *n* bits are carried on *m* wires. The throughput efficiency of the code is defined as *n*/*m* (in units of bits per wire), which is a measure of the efficiency of the code.

This section describes the multi-wire signaling codes supported by this standard.

2.1 General Architecture

The general architecture of a multi-wire signaling interface is shown in Figure 1. The input to the encoder at the transmission end of the interface consists of *n*-bits of data, where *n* is determined by the multi-wire code being used. The encoder drives a unique codeword onto *m*-wires that is computed from the input data. The driver block puts the encoded value on the wires. Codewords are determined by the codebook for the code being used.

At the receive end of the interface, the Multi-wire Receiver (Rx) block converts the *m*-wire input to an *n*-wire output. This is accomplished by comparing the weighted average of groups of wires to the weighted average of other groups of wires as determined by a Linear Combination Table that is defined for the code. In the generic architecture, the Multi-wire Rx block may drive a decoder stage that decodes the data bits such that the d_0 to d_{n-1} bits on the decoder outputs are equivalent to the data input to the encoder. The codes specified in this document do not require a decoder stage, in which case the d_0 to d_{n-1} bits are decoded directly by the Multi-wire Rx block.

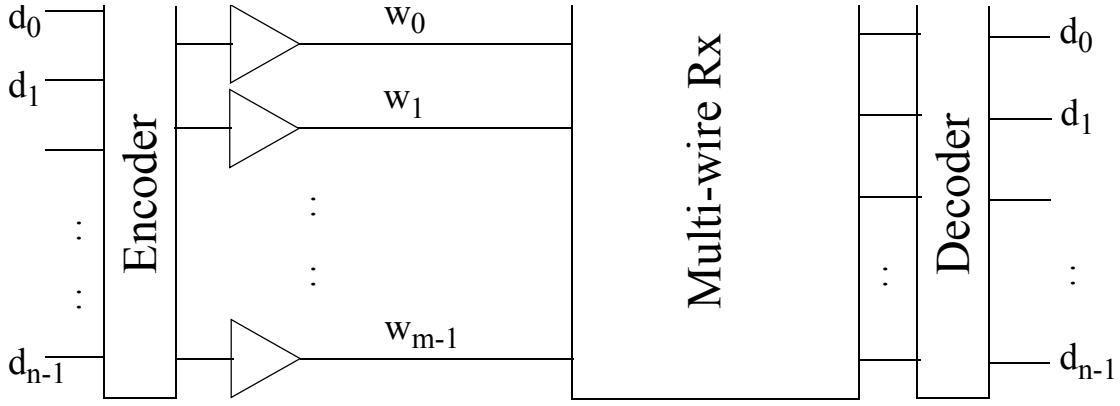

Figure 1 — Generic Architecture of a Multi-Wire Signaling Interface

One example of an implementation of the Multi-wire Rx block in Figure 1 is shown in Figure 2. The Rx Front End handles any amplification and equalization tasks. The Linear Combination matrix circuit generates weighted averages of groups of wires as determined by the Linear Combination Table defined for the code. This circuit may be implemented using a number of design approaches including passive resistor networks or multi-input amplifier circuits. The corresponding averaged values are connected to the inputs of comparator circuits, and the output of these comparators are the decoded d_0 to d_{n-1} bits. As noted previously, codes defined in this document do not require an additional digital decode stage.

2.1 General Architecture (cont'd)

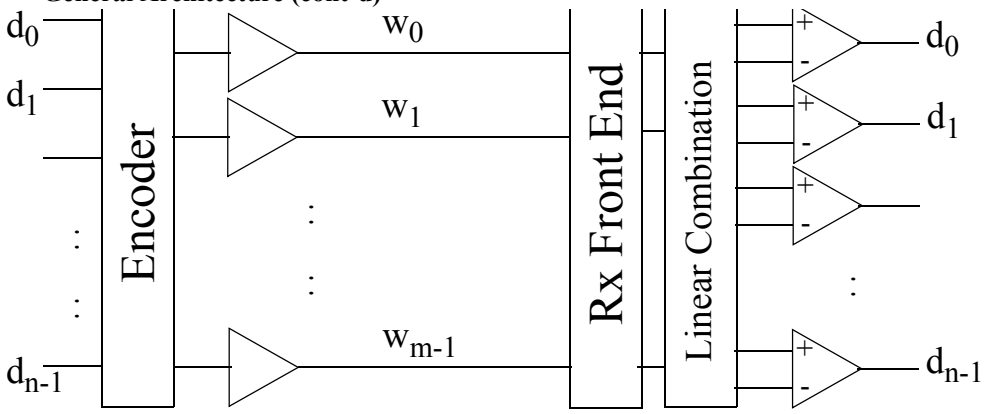

Figure 2 — Example Implementation of a Multi-Wire Signaling Interface

2.2 Signal Levels

Each of the *m* wires of the multi-wire signaling interface is driven to one of *s* wire states, where *s* is determined by the multi-wire code being used. Each of the *s* wire states corresponds to a unique single-ended signal level.

2.2.1 Quaternary Signal Levels

Codes for which *s* = 4 use quaternary I/O drivers which drive one of four signal levels. These signal levels are designated by the nomenclature {+1, +1/3, -1/3, -1}. Figure 3 illustrates the voltage relationship between these signal levels, where V_{+1} indicates a wire at signal level +1, $V_{+1/3}$ indicates a wire at signal level +1/3, and so forth.

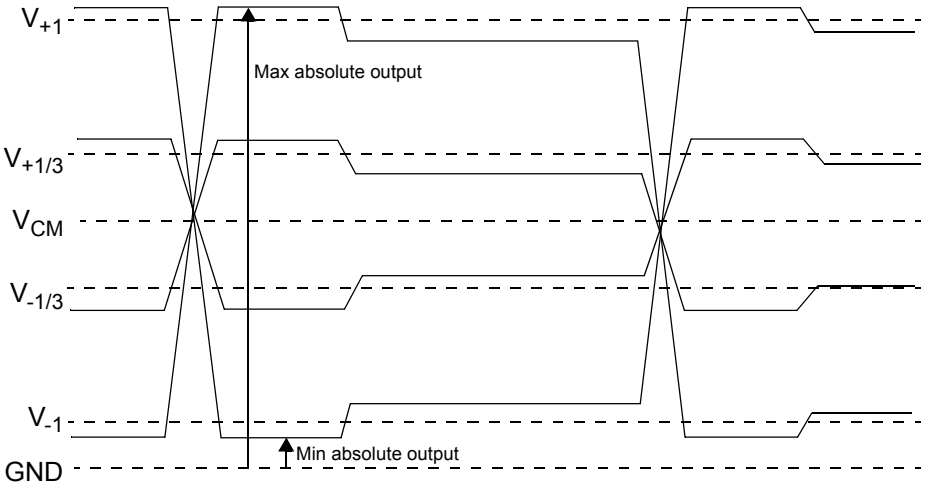

Figure 3 — Signal Levels

The single-ended voltage difference between the maximum V_{+1} voltage and the minimum V_{-1} voltage is defined as the Output Maximum Peak-to-Peak Voltage Swing (V_{SEpp}).

The voltage difference between each signal level and adjacent signal levels is roughly equal, such that the four signal levels are equally spaced across the dynamic range of the driver.

The common mode voltage (V_{CM}) is computed as the average voltage of all wires of the interface. The multi-wire codes defined in this specification are constructed such that V_{CM} is constant. This results in common mode and simultaneous switching noise immunity similar to that of differential signals.

2.3 ENRZ Code Definition

The ENRZ code maps 3 bits of data into codewords on 4 wires (3b4w), and uses quaternary signal levels ($s = 4$). The throughput efficiency of this code is 0.750.

Table 1 provides the codebook which describes the mapping between data bits and signal levels on the wires. Table 2 specifies the Linear Combination table containing the matrix of weights of each wire on the input to each comparator. A block diagram of an example implementation of an ENRZ receiver is shown in Figure 4.

Table 1 — ENRZ Code Map

Data Value (d_2, d_1, d_0)	Wire States (w_3, w_2, w_1, w_0)	Data Value (d_2, d_1, d_0)	Wire States (w_3, w_2, w_1, w_0)
0 0 0	(-1, +1/3, +1/3, +1/3)	1 0 0	(-1/3, -1/3, -1/3, +1)
0 0 1	(-1/3, +1, -1/3, -1/3)	1 0 1	(+1/3, +1/3, -1, +1/3)
0 1 0	(-1/3, -1/3, +1, -1/3)	1 1 0	(+1/3, -1, +1/3, +1/3)
0 1 1	(+1/3, +1/3, +1/3, -1)	1 1 1	(+1, -1/3, -1/3, -1/3)

Table 2 — ENRZ Linear Combination Table

	Contribution of each wire to comparator decode			
Comparator	w_3	w_2	w_1	w_0
d_0	+1/2	+1/2	-1/2	-1/2
d_1	+1/2	-1/2	+1/2	-1/2
d_2	+1/2	-1/2	-1/2	+1/2

Each comparator implements the calculation of:

$$c_i = \sum_{j=0}^{m-1} (\text{weight of wire j})(\text{signal level on wire j})$$

If c_i is greater than 0, then $d_i = 1$; if c_i is less than 0, then $d_i = 0$. Simple and fast implementations of this operation are possible. ENRZ does not require a decoder stage after the comparator stage.

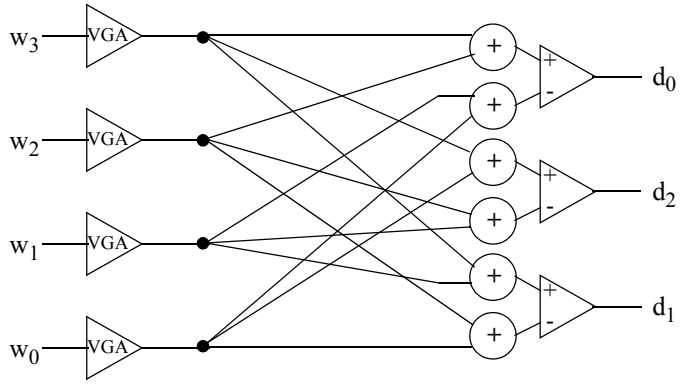

Figure 4 — ENRZ Receiver Block Diagram Example

2.4 CNRZ-5 Code Definition

The CNRZ-5 code maps 5 bits of data into codewords on 6 wires (5b6w), and uses quaternary signal levels ($s = 4$). The throughput efficiency of this code is 0.833.

Table 3 provides the codebook which describes the mapping between data bits and signal levels on the wires. Table 4 specifies the Linear Combination table containing the matrix of weights of each wire on the input to each comparator. A block diagram of an example implementation of an CNRZ-5 receiver is shown in Figure 5.

Table 3 — CNRZ-5 Code Map

Data Value (d_4, d_3, d_2, d_1, d_0)	Wire States (w_5, w_4, w_3, w_2, w_1, w_0)	Data Value (d_4, d_3, d_2, d_1, d_0)	Wire States (w_5, w_4, w_3, w_2, w_1, w_0)
0 0 0 0 0	(-1, -1/3, +1/3, +1, +1/3, -1/3)	1 0 0 0 0	(-1/3, -1, +1/3, +1, +1/3, -1/3)
0 0 0 0 1	(-1/3, +1/3, +1, +1/3, -1/3, -1)	1 0 0 0 1	(+1/3, -1/3, +1, +1/3, -1/3, -1)
0 0 0 1 0	(-1, -1/3, +1/3, -1/3, +1, +1/3)	1 0 0 1 0	(-1/3, -1, +1/3, -1/3, +1, +1/3)
0 0 0 1 1	(-1/3, +1/3, +1, -1, +1/3, -1/3)	1 0 0 1 1	(+1/3, -1/3, +1, -1, +1/3, -1/3)
0 0 1 0 0	(-1, -1/3, +1/3, +1, -1/3, +1/3)	1 0 1 0 0	(-1/3, -1, +1/3, +1, -1/3, +1/3)
0 0 1 0 1	(-1/3, +1/3, +1, +1/3, -1, -1/3)	1 0 1 0 1	(+1/3, -1/3, +1, +1/3, -1, -1/3)
0 0 1 1 0	(-1, -1/3, +1/3, -1/3, +1/3, +1)	1 0 1 1 0	(-1/3, -1, +1/3, -1/3, +1/3, +1)
0 0 1 1 1	(-1/3, +1/3, +1, -1, -1/3, +1/3)	1 0 1 1 1	(+1/3, -1/3, +1, -1, -1/3, +1/3)
0 1 0 0 0	(-1/3, +1/3, -1, +1, +1/3, -1/3)	1 1 0 0 0	(+1/3, -1/3, -1, +1, +1/3, -1/3)
0 1 0 0 1	(+1/3, +1, -1/3, +1/3, -1/3, -1)	1 1 0 0 1	(+1, +1/3, -1/3, +1/3, -1/3, -1)
0 1 0 1 0	(-1/3, +1/3, -1, -1/3, +1, +1/3)	1 1 0 1 0	(+1/3, -1/3, -1, -1/3, +1, +1/3)
0 1 0 1 1	(+1/3, +1, -1/3, -1, +1/3, -1/3)	1 1 0 1 1	(+1, +1/3, -1/3, -1, +1/3, -1/3)
0 1 1 0 0	(-1/3, +1/3, -1, +1, -1/3, +1/3)	1 1 1 0 0	(+1/3, -1/3, -1, +1, -1/3, +1/3)
0 1 1 0 1	(+1/3, +1, -1/3, +1/3, -1, -1/3)	1 1 1 0 1	(+1, +1/3, -1/3, +1/3, -1, -1/3)
0 1 1 1 0	(-1/3, +1/3, -1, -1/3, +1/3, +1)	1 1 1 1 0	(+1/3, -1/3, -1, -1/3, +1/3, +1)
0 1 1 1 1	(+1/3, +1, -1/3, -1, -1/3, +1/3)	1 1 1 1 1	(+1, +1/3, -1/3, -1, -1/3, +1/3)

Table 4 — CNRZ-5 Linear Combination Table

	Contribution of each wire to comparator decode					
Comparator	w_5	w_4	w_3	w_2	w_1	w_0
d_0	+1/3	+1/3	+1/3	-1/3	-1/3	-1/3
d_1	0	0	0	-1	+1/2	+1/2
d_2	0	0	0	0	-1	+1
d_3	+1/2	+1/2	-1	0	0	0
d_4	+1	-1	0	0	0	0

2.4 CNRZ-5 Code Definition (cont'd)

Each comparator implements the calculation of:

$$c_i = \sum_{j=0}^{m-1} (\text{weight of wire j})(\text{signal level on wire j})$$

If c_i is greater than 0, then $d_i = 1$; if c_i is less than 0, then $d_i = 0$. Simple and fast implementations of this operation are possible. CNRZ-5 does not require a decoder stage after the comparator stage.

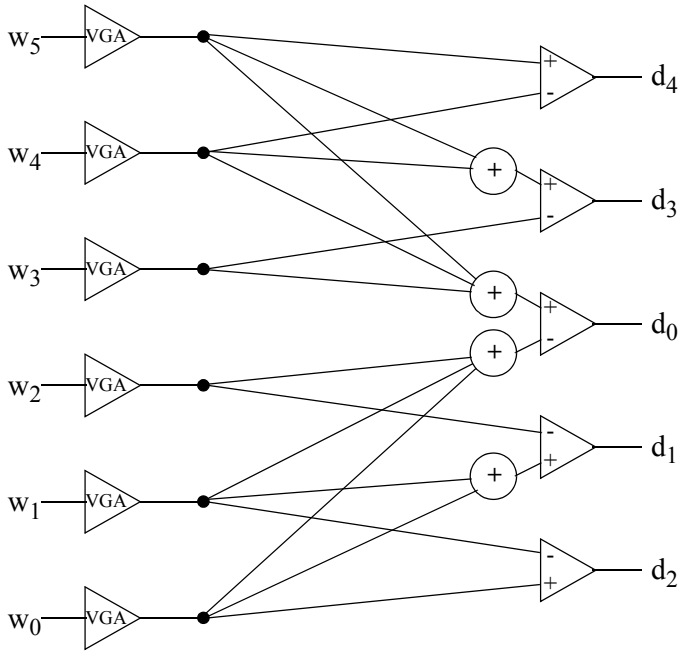

Figure 5 — CNRZ-5 Receiver Block Diagram Example

3 Driver Specifications

This specification defines I/O drivers that are consistent with use for a minimal loss, low skew channel between two devices that are mounted immediately next to each other within a multi-chip module. Specifications that use this specification by reference will define the maximum baud rate and channel insertion loss based on the requirements of the target application. Specifications for applications which drive outside of a package to a DIMM may specify additional requirements for channel loss, channel skew, channel crosstalk, supply offset, transmit equalization, receive equalization, and skew tolerance.

The multi-wire code interface transmitter consists of *m* driver I/O, where *m* is defined by the corresponding multi-wire code definition in 2. Each driver I/O shall meet the electrical and jitter specifications defined in Table 5. Measurements are performed at the maximum baud rate f_{baud} of the interface.

Table 5 — Driver Electrical and Jitter Output Specifications

Characteristic	Symbol	Condition	MIN.	TYP.	MAX.	Units
Output Max. Peak-to-Peak Voltage Swing (Single-Ended)	V_{SEpp}	Class 1 driver. See Notes 1, 2.	200		300	mVpp
		Class 2 driver. See Notes 1, 2.	100		150	mVpp
Voltage Swing Mismatch	ΔV_{SEpp}	See Note 7.	0.95		1.05	---
Output Common Mode Voltage	V_{CM}	See Note 1. AC coupled.	100		1200	mV
		See Note 1. DC coupled.	350		550	
Nominal Driver Impedance	R_{SE}	See Note 4.		75		Ω
Common Mode Noise	N_{CM}	See 3.3.			12	mV_{RMS}
Single-Ended Output Return Loss	S22	See 3.2.				
Eye Linearity	EYElin	See 3.4.3.			1.2	---
Eye Width	Hmid Hlo Hup	See 3.4.2.	0.38			UI
Eye Height	Vmid Vlo Vup	See 3.4.2. See Note 3.	$0.22 \times V_{SEpp}$		$0.40 \times V_{SEpp}$	mVpp
Total Random Jitter on Driver Output	RJ	See Note 9.			0.010	UI rms
High Frequency RJ on Forwarded Clock	RJ_{hf}	Integrated RJ above f_{baud} / 250. See Note 9.			0.008	UI rms
Voltage Level Mismatch	Vctr	See 3.4.2. See Note 8.			5	mV
Transmit Skew	T_{skew}	See Note 5.			0.06	UI
Driver Turn On/Off Time	$T_{on/off}$	See Note 6.			0.10	ns
Total Random Jitter on Forwarded Clock	RJ_{clk}				0.075	UI rms

Table 5 — Driver Electrical and Jitter Output Specifications

Characteristic	Symbol	Condition	MIN.	TYP.	MAX.	Units
High Frequency RJ on Forwarded Clock	$RJ_{clk\text{-}hf}$	Integrated RJ above f_{baud} / 250			0.009	UI rms

NOTE 1 See definition in 2.2. Measurements use the test load defined in 3.1.
NOTE 2 Drivers may be designed to meet either class 1 or class 2 limits for V_{SEpp}. Use of class 2 drivers may be used to reduce power in applications where the higher launch amplitude is not required.
NOTE 3 Eye height limits are referenced to max. V_{SEpp}, and therefore are dependent on whether the driver is class 1 or class 2.
NOTE 4 Impedance is the single-ended impedance of any driver output.
NOTE 5 Time between zero crossings of any two wires of the multi-wire interface for the same baud symbol transition.
NOTE 6 Time for driver to transition from Hi-Z state to driving valid levels, or to transition from driving valid levels to driving a high-Z state. This parameter only applies to bidirectional interfaces.
NOTE 7 Given V_{SEpp} values of each driver of the multi-wire interface, this is the ratio of the value of V_{SEpp} for any driver to the value of V_{SEpp} for any other driver.
NOTE 8 Given Vctr values of each driver of a multi-wire interface, this is the maximum difference between Vctr for any driver and Vctr for any other driver.
NOTE 9 Driver signal is referenced to the forwarded clock such that common jitter is cancels when making this measurement.

3.1 Driver Test Load

Driver specifications are verified using the test load specified by the resistor network in Figure 6. This network translates between the impedances of the driver and the impedance to ground at the probe point. For interfaces which drive outside of a package, the probe point may be a coax connection to test equipment. Resistor values are specified in Table 6. The return loss of the network should be better than 20 dB from f_{baud} / 1667 to 1.5 × f_{baud}..

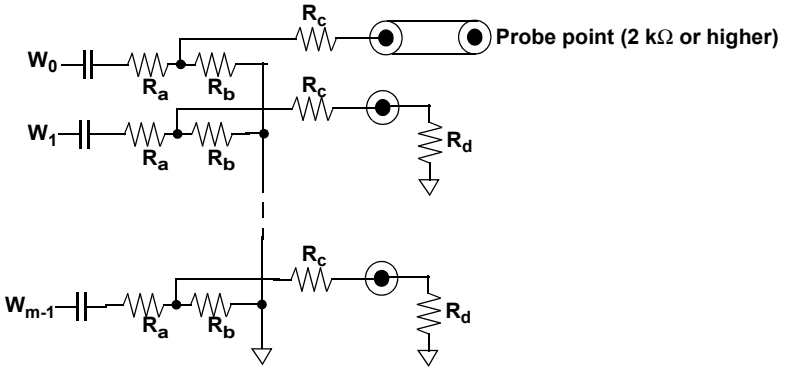

Figure 6 — Driver Test Load

Table 6 — Resistor Values for Driver Test Load

Resistor	Tolerance	ENRZ	CNRZ-5		Units
R_a	± 1%	4.64	4.64		Ω
R_b	± 1%	71.5	71.5		Ω
R_c	± 1%	71.5	71.5		Ω
R_d	± 1%	4.02	4.02		kΩ

Signal voltages at the test equipment connection point in Figure 6 depend on the impedance of the test equipment input connected to the probe point. The input impedance of the test equipment should be at least 2 kΩ to ensure the amplitude of the signal at the test equipment input is at least 90% of the source amplitude. If the input impedance of the test equipment is less than 2 kΩ, measured values should be adjusted to compensate for signal loss in the resistor network.

3.2 Transmitter Return Loss

The single-ended return loss of any driver pin of the transmitter shall be within the region specified by the graph in Figure 7.

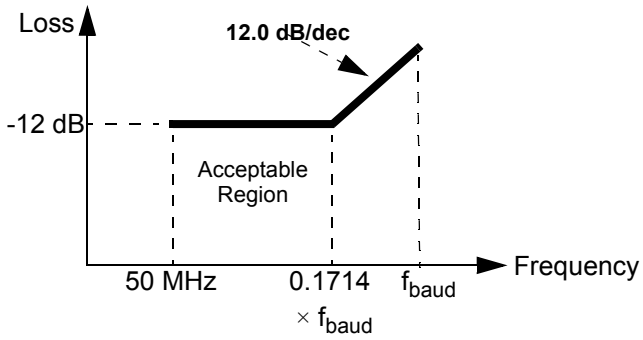

Figure 7 — Single-Ended Transmitter Output Return Loss

3.3 Common Mode Noise

Common mode noise (N_{CM}) is defined as the amplitude of the common mode voltage (V_{CM}) noise on the driver outputs. This specification may be verified using the resistor network in Figure 8 and measuring voltage noise at the probe point. Resistor values are specified in Table 7. The resistor network sums the driver outputs such that the voltage at the probe point for an ideal transmitter is V_{CM}. Noise at the probe point must meet limits described in Table 5, excluding any instrumentation noise inherent in the measurement.

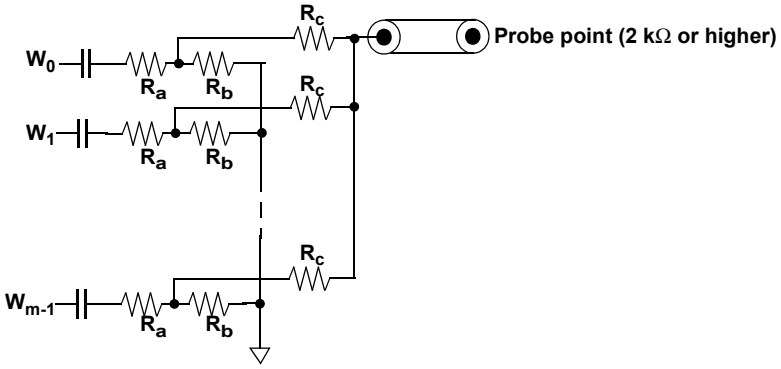

Figure 8 — Driver Test Load for Common Mode Noise Measurement

Table 7 — Resistor Values for Driver Test Load (N_{CM})

Resistor	Tolerance	ENRZ	CNRZ-5		Units
R_a	± 1%	4.64	4.64		Ω
R_b	± 1%	71.5	71.5		Ω
R_c	± 1%	71.5	71.5		Ω

Signal voltages at the test equipment connection point in Figure 8 depend on the impedance of the test equipment input. The input impedance of the test equipment should be at least 2 kΩ to ensure the amplitude of the signal at the test equipment input is at least 90% of the source amplitude. If the input impedance of the test equipment is less than 2 kΩ, measured values should be adjusted to compensate for signal loss in the resistor network.

3.4 Driver Linearity and Jitter Tests (Quaternary Drivers)

Eye measurements are performed using the test pattern described in 3.4.1 and the test load described in 3.1. Eye width and height parameters are defined in 3.4.2 and must meet limits described in Table 5.

3.4.1 Driver Test Pattern

Driver eye parameters are measured using a PRBS-13 pattern that is bit-striped across the subchannels d_{n-1} to d_0 of the interface, where n is the number of subchannels as determined by the multi-wire code definition.

3.4.2 Eye Measurements

Eye parameters are determined from 10^{-4} contours generated from oscilloscope CDF/histogram data. Eye measurements are performed by first determining the midpoint of the horizontal opening of the middle eye (*Hmid*). This sample point is then used to determine the vertical openings of the middle (*Vmid*), upper (*Vup*), and lower (*Vlo*) eyes. The midpoints of the vertical openings for the upper (*Vup*) and lower (*Vlo*) eyes is then used to determine the horizontal openings of the upper (*Hup*) and lower (*Hlo*) eyes.

Measured eye width and height parameters are defined below and are illustrated in Figure 9. These parameters must meet limits described in Table 5.

Hmid - The width of the inner eye contour constructed from 10^{-4} CDFs of eye edges at the maximum horizontal eye opening of the middle eye.

Vctr - The signal voltage corresponding to the *Hmid* measurement.

Vmid - The height of the inner eye contour of the middle eye constructed from 10^{-4} CDFs of eye edges sampled at *Hmid* / 2.

Vup - The height of the inner eye contour of the upper eye constructed from 10^{-4} CDFs of eye edges sampled at *Hmid* / 2.

Vlo - The height of the inner eye contour of the lower eye constructed from 10^{-4} CDFs of eye edges sampled at *Hmid* / 2.

Hup - The width of the inner eye contour of the upper eye constructed from 10^{-4} CDFs of eye edges at *Vup* / 2.

Hlo - The width of the inner eye contour of the lower eye constructed from 10^{-4} CDFs of eye edges at *Vlo* / 2.

AVmid - The difference between the mean levels of the $V_{+1/3}$ and $V_{-1/3}$ histograms as sampled at *Hmid* / 2.

AVup - The difference between the mean levels of the V_{+1} and $V_{+1/3}$ histograms as sampled at *Hmid* / 2.

AVlo - The difference between the mean levels of the $V_{-1/3}$ and V_{-1} histograms as sampled at *Hmid* / 2.

The *AVmid*, *AVup*, and *AVlo* parameters are used to calculate eye linearity as defined in 3.4.3.

3.4.2 Eye Measurements (cont'd)

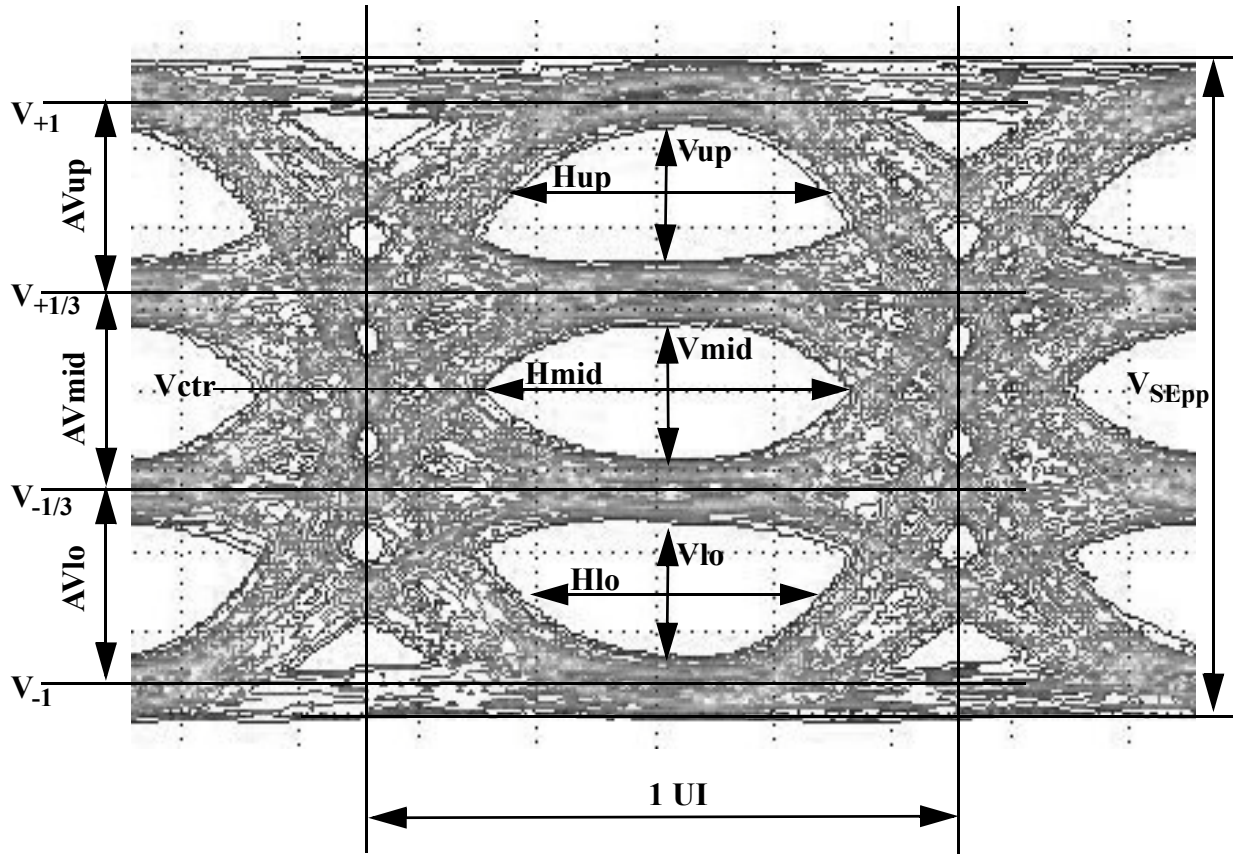

Figure 9 — Quaternary Driver Jitter and Eye Height Measurements.

3.4.3 Eye Linearity

Eye linearity is calculated using the *AVmid*, *AVup*, and *AVlo* parameters measured as described in 3.4.2. These parameters are used in the following equation:

$$EYE_{lin} = \frac{max(AVup,AVmid,AVlo)}{min(AVup,AVmid,AVlo)}$$

EYE_{lin} must be within the limits specified in Table 5.

3.4.4 Scope Connections to the Forwarded Differential Clock

The forwarded differential clock must be used for all eye measurements. This clock is driven by the transmitter and is used by the receiver to sample data. The frequency of this clock is either: f_{baud} / 2, f_{baud} / 4, or f_{baud} / 8, and a single differential clock may be shared by multiple instances of the multi-wire interface. The oscilloscope must use this clock as a trigger such that any correlated jitter is excluded from measurements.

Depending on the application, some CNRZ-5 interfaces may embed the differential clock on either subchannel d_2 or d_4. If the clock is embedded on subchannel d_2, then wires w_0 and w_1 are directly connected to the plus and minus trigger inputs of the oscilloscope. If the clock is embedded on subchannel d_4, then wires w_5 and w_4 are directly connected to the plus and minus trigger inputs of the oscilloscope.

4 Receiver Specifications

This specification defines I/O receivers that are consistent with use for a minimal loss, low skew channel between two devices that are mounted immediately next to each other within a multi-chip module. Specifications that use this specification by reference will define the maximum baud rate and channel insertion loss based on the requirements of the target application. Specifications for applications which drive outside of a package to a DIMM may specify additional requirements for channel loss, channel skew, channel crosstalk, supply offset, transmit equalization, receive equalization, and skew tolerance.

The multi-wire code interface receiver consists of *m* receiver I/O, where *m* is defined by the corresponding multi-wire code definition in 2. Receivers of the ENRZ multi-wire code implement the Linear Combination Table described in 2.3. Receivers of the CNRZ-5 multi-wire code implement the Linear Combination Table described in 2.4.

Receivers are tested using a multi-wire transmitter that complies with the requirements in 3, driven through an interconnect such that the signal at the receiver input pins is within the tolerance limits specified in Table 9. The channel is assumed to be passive and have linear behavior.

Receivers shall comply with the specifications in Table 8, and shall operate within the specified Bit Error Ratio when receiving signals within the tolerance limits specified in Table 9. Measurements are performed at the maximum baud rate f_{baud} of the interface.

Table 8 — Receiver Electrical Specifications

Characteristic	Symbol	Condition	MIN.	TYP.	MAX.	Units
Input Voltage Swing (Single-Ended)	V_{SEpp}	See Notes 1, 2.			400	mVpp
Input Impedance	R_{din}	See Note 3.	45	75	82.5	Ω
Input Impedance Mismatch	R_m	See Note 4.			10	%
Input Common Mode Voltage	V_{CM}	See Note 1. AC coupled.	100		1200	mV
		See Note 1. DC coupled.	300		600	
Single-ended Input Return Loss	S11	See 4.1.				
Bit Error Ratio	BER	See Note 5.			10^{-15}	--

NOTE 1 See definition in 2.2. Specification applies to each wire of the interface.
NOTE 2 Receivers intended to only operate with Class 2 drivers may use lower maximum value for V_{SEpp}. See Table 5 for definition of class 1 and class 2 drivers.
NOTE 3 Impedance is between from any input of the receiver to ground.
NOTE 4 R_m is the maximum impedance difference between any two inputs of the multi-wire interface receiver.
NOTE 5 Although the interface link budget is designed to support the specified BER, it is acceptable to use BER = 10^{-12} when performing test measurements.

4 Receiver Specifications (cont'd)

Table 9 — Receiver Input Tolerance Specifications

Characteristic	Symbol	Condition	MIN.	TYP.	MAX.	Units
Eye Width	EW15	See 4.2.2.	0.60			UI
Eye Height	EH15	Class 1 Receiver. See 4.2.2 and Note 4.	120			mVppd
		Class 2 Receiver. See 4.2.2 and Note 4.	80			
Sinusoidal Jitter, Maximum	SJ-max	See Notes 1, 3.			5	UIpp
Sinusoidal Jitter, High Frequency	SJ-hf	See Notes 1, 2.			0.05	UIpp

NOTE 1 The receiver shall tolerate the sum of these jitter contributions.
NOTE 2 SJ-hf period shall be > f_{baud} / 50.
NOTE 3 SJ-max period shall be < f_{baud} / 1000.
NOTE 4 Class 1 Receivers are intended for use with Class 1 Drivers only. Class 2 Receivers may be used with either Class 1 or Class 2 Drivers.

4.1 Receiver Return Loss

The single-ended return loss of any input pin of the receiver shall be within the region specified by the graph in Figure 10.

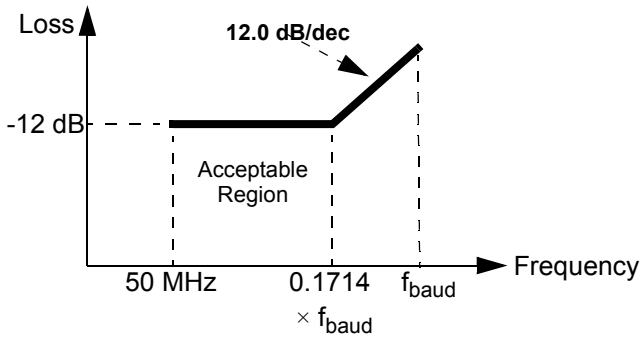

Figure 10 — Single-Ended Receiver Input Return Loss

4.2 Receiver Linearity and Jitter Tolerance Tests

Receiver linearity and jitter tolerance is tested using a transmitter that meets the specifications in 3, driven through an interconnect such that the signal at the receiver input pins is no better than the tolerance limits specified in Table 9. The receiver must operate and correctly receive data within the Bit Error Ratio (BER) limits specified in Table 8 when the received signal is at these tolerance limits.

4.2.1 Driver Test Pattern

Eye parameters at the receiver input pins are measured using a PRBS-31 pattern that is bit-striped across the subchannels d_{n-1} to d_0 of the interface, where n is the number of subchannels as determined by the multi-wire code definition.

4.2.2 Eye Measurements

Eye parameters are measured for each subchannel using the reference clock forwarded from the transmitter as a trigger. Eye parameters are determined from 10^{-6} contours generated from oscilloscope CDF/histogram data, the results of which are extrapolated to determine whether specifications are met in Table 9 for the Bit Error Ratio (BER) limit specified in Table 8.

Note that the linear combination function defined for the multi-wire code results in an NRZ eye opening at the receiver. This linear combination function must either be implemented by the test equipment or must be implemented by using passive averaging circuits when measuring characteristics of the eye opening of the signal at the receiver.

4.2.2.1 Scope Connections for ENRZ code

The eye for each subchannel d_2 to d_0 of the ENRZ interface may be determined by connecting all wires of the interface to a multi-channel oscilloscope, and configuring the oscilloscope to perform the linear combination defined in 2.3 for the subchannel being tested.

Alternatively, if a multi-channel oscilloscope is not available then the passive averaging circuit shown in Figure 11 may be used to average signals prior to the oscilloscope inputs. The inputs to the passive averaging circuit are connected as described in Table 10 for the subchannel being measured. The recommended resistor values in Figure 11 are as follows: $R_a = 4.64\Omega$, $R_{b2} = 100\Omega$, $R_c = 71.5\Omega$.

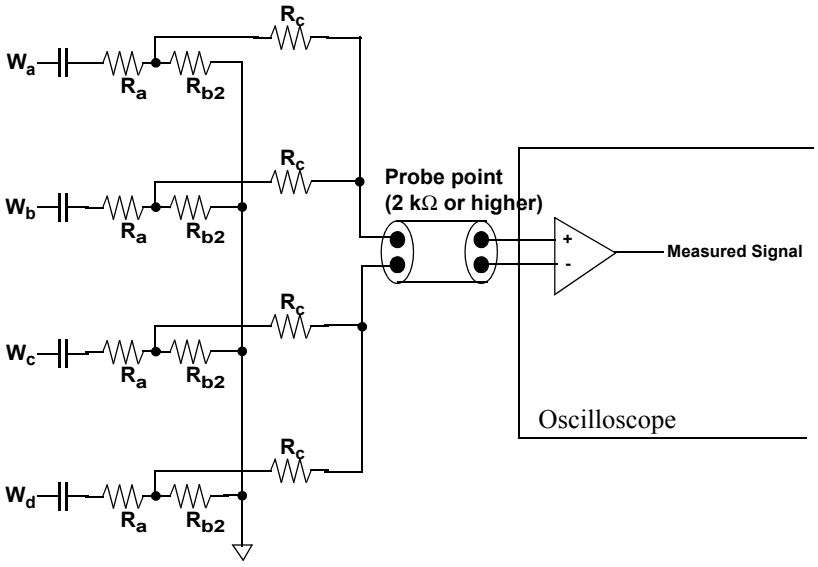

Figure 11 — Passive Averaging Circuit for ENRZ Interface Eye Measurements

Table 10 — ENRZ Passive Averaging Circuit Connection Table

To measure subchannel:	Connect w_a to:	Connect w_b to:	Connect w_c to:	Connect w_d to:
d_0	w_3	w_2	w_1	w_0
d_1	w_3	w_1	w_2	w_0
d_2	w_3	w_0	w_2	w_1

4.2.2.2 Scope Connections for CNRZ-5 code

The eye for each subchannel d_4 to d_0 of the CNRZ-5 interface may be determined by connecting all wires of the interface to a multi-channel oscilloscope, and configuring the oscilloscope to perform the linear combination defined in 2.4 for the subchannel being tested.

Alternatively, if a multi-channel oscilloscope is not available then passive averaging circuits may be used to average signals prior to the oscilloscope inputs. Passive averaging circuits are shown in Figure 12. All wires are connected through the passive averaging circuit to the oscilloscope when performing measurements on subchannel 0. When measuring subchannel 1 (or 3), wires w_5, w_4, and w_3 (or w_0, w_1, and w_2) are connected through the passive averaging circuit to the oscilloscope, and the remaining wires are terminated. When measuring subchannel 2 (or 4), wires w_5 and w_4 (or w_0 and w_1) are connected through the passive averaging circuit to the oscilloscope, and the remaining wires are terminated. The recommended resistor values in Figure 12 are as follows: $R_a = 4.64\Omega$, $R_{b1} = 71.5\Omega$, $R_{b2} = 100\Omega$, $R_{b3} = 124\Omega$, $R_c = 71.5\Omega$, $R_d = 4.02k\Omega$.

Note that the amplitude of the measured eye on subchannels d_2 and d_4 is typically less than that of the other subchannels.

4.2.2.3 Scope Connections to the Forwarded Differential Clock

The forwarded differential clock must be used for all eye measurements. This clock is driven by the transmitter and is used by the receiver to sample data. The frequency of this clock is either: f_{baud} / 2, f_{baud} / 4, or f_{baud} / 8, and a single differential clock may be shared by multiple instances of the multi-wire interface. The oscilloscope must use this clock as a trigger such that any correlated jitter is excluded from measurements.

Depending on the application, some CNRZ-5 interfaces may embed the differential clock on either subchannel d_2 or d_4. If the clock is embedded on subchannel d_2, then wires w_0 and w_1 are directly connected to the plus and minus trigger inputs of the oscilloscope. If the clock is embedded on subchannel d_4, then wires w_5 and w_4 are directly connected to the plus and minus trigger inputs of the oscilloscope.

4.2.2.4 Extrapolation to Target Bit Error Rate

Eye parameter measurements are determined from 10^{-6} contours generated from oscilloscope CDF/histogram data. To determine compliance with specifications defined in Table 9, the eye measurements must be extrapolated to the Bit Error Rate (BER) specified in Table 8.

The following procedure is used to determine the horizontal eye opening:

1. Measure the differential signal to construct CDFs of the jitter at the zero crossing using 10^{-6} contours and measure the eye width (EW6).
2. Determine the RMS value of the random jitter (RJ_{rms}) of the measured eye.
3. Calculate the eye width for the target BER as follows: EW15 = EW6 - (2 * 3.19 * RJ_{rms}).

The following procedure is used to determine the vertical eye opening:

4. Measure the differential signal to construct CDFs of the signal voltage at the center of the horizontal eye using 10^{-6} contours for both the logic one and logic zero signals as a distance in voltage from the zero crossing. Calculate the difference to determine the eye height measurement (EH6).
5. Determine the RMS value of the random noise for the logic one (RN1) and logic zero (RN0) levels.
6. Calculate the eye height for the target BER as follows: EH15 = EH6 - 3.19 * (RN0 + RN1).

4.2.2.4 Extrapolation to Target Bit Error Rate (cont'd)

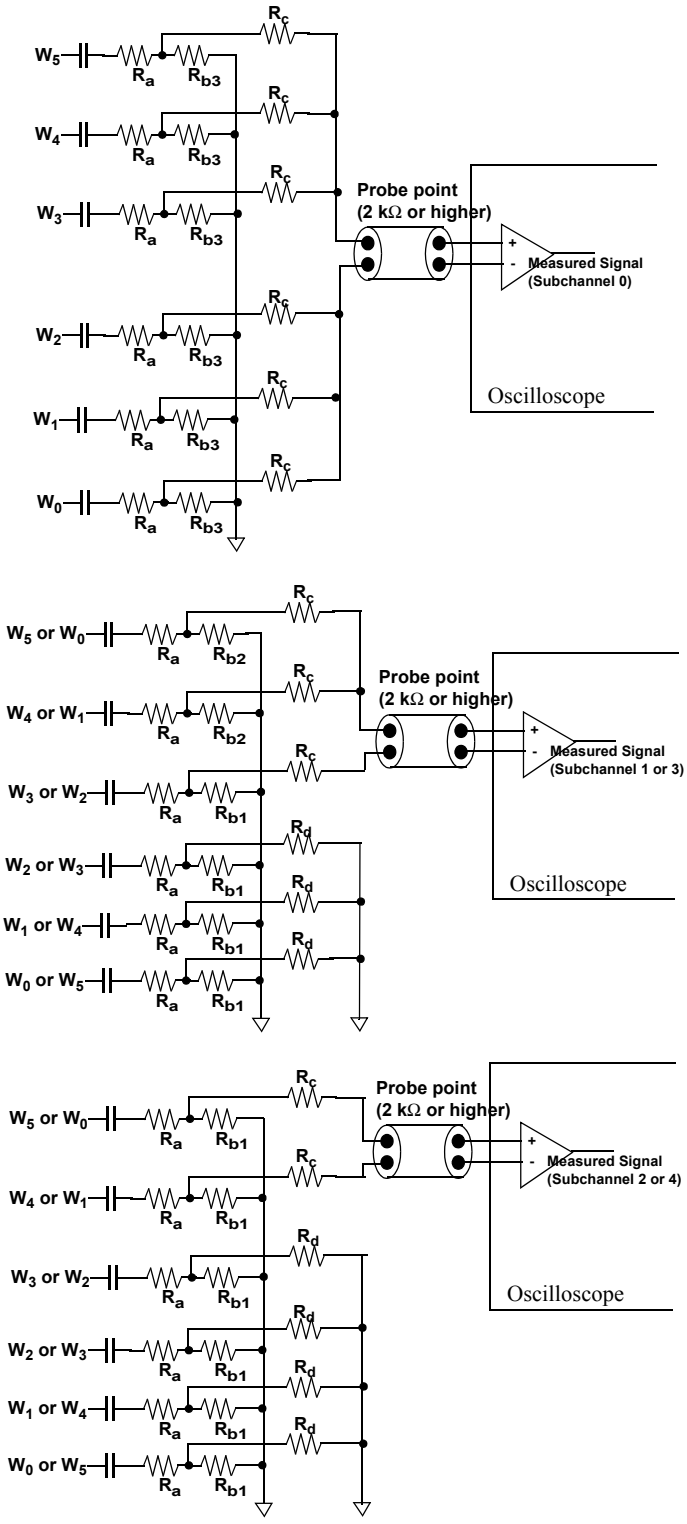

Figure 12 — Passive Averaging Circuits for CNRZ-5 Interface Eye Measurements

5 HSpice Modeling

This section describes the HSpice model structure for generalized representations of multi-wire transmitters and receivers. This description is for reference only.

5.1 Multi-Wire Transmitters Using Quaternary Drivers

The HSpice model structure for a generalized multi-wire transmitter is shown in Figure 13. This transmitter uses quaternary drivers which produce the signal levels described in 2.2.1. The number of drivers in the multi-wire transmitter is determined by the multi-wire code being implemented.

Each quaternary driver transmits one of the line levels described in 2.2.1. The driver consists of three NMOS and three PMOS current paths. For any given line level, three of these paths are turned on and three paths are turned off. The truth table shown in Figure 13 correlates line levels to the corresponding transistor gate control signals in the model.

The resistances shown in Figure 13 are defined as follows: R_c = 187.5 Ω; R_d = 125 Ω; R_{tt} = 75 Ω.

The multi-wire interface is terminated at the receiver by a single-ended termination resistance to a common signal ground point. For a DC coupled multi-wire interface, the signal ground point does not need a bias voltage. A large capacitor between the common connection point and DC ground provides the AC path to ground while allowing the common connection to bias itself to the common mode voltage.

5.1 Multi-Wire Transmitters Using Quaternary Drivers (cont'd)

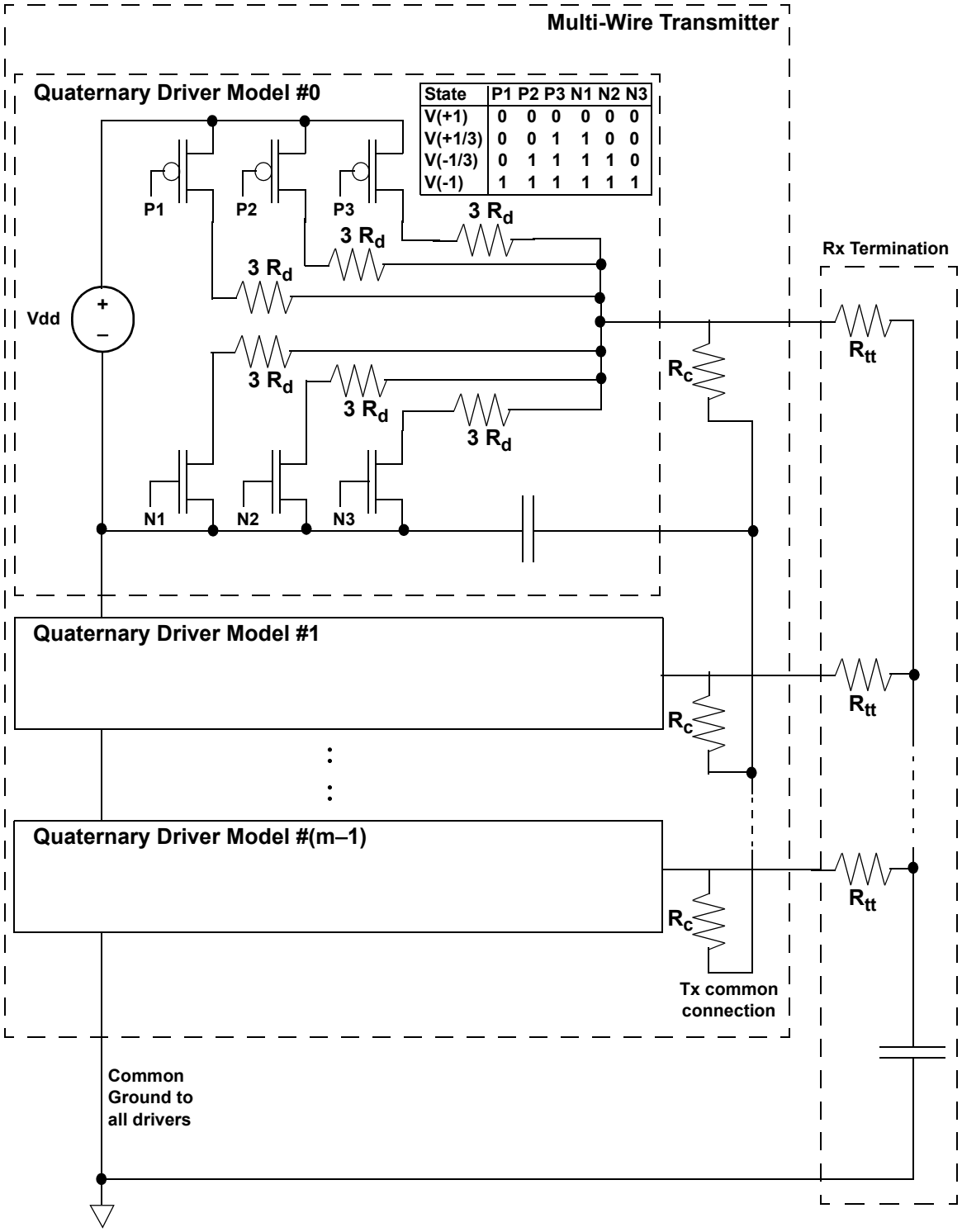

Figure 13 — Quaternary Driver and Transmitter HSpice Model

Standard Improvement Form

JEDEC _____

The purpose of this form is to provide the Technical Committees of JEDEC with input from the industry regarding usage of the subject standard. Individuals or companies are invited to submit comments to JEDEC. All comments will be collected and dispersed to the appropriate committee(s).

If you can provide input, please complete this form and return to:

JEDEC
Attn: Publications Department
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

Fax: 703.907.7583

1. I recommend changes to the following:

☐ Requirement, clause number _____

☐ Test method number _____ Clause number _____

The referenced clause number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

_____

_____

_____

_____

3. Other suggestions for document improvement:

_____

_____

_____

_____

Submitted by

Name: _____ Phone: _____

Company: _____ E-mail: _____

Address: _____

City/State/Zip: _____ Date: _____


JEDEC®