

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

This document may be downloaded free of charge; however JEDEC retains the copyright on this material. By downloading this file the individual agrees not to charge for or resell the resulting material.

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

Printed in the U.S.A.
All rights reserved

PLEASE!

**DON'T VIOLATE
THE
LAW!**

This document is copyrighted by the JEDEC Solid State Technology Association
and may not be reproduced without permission.

Organizations may obtain permission to reproduce a limited number of copies
through entering into a license agreement. For information, contact:

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 $w_0, w_1, \dots, w_{m-1}$. 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 $nbmw$, 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 $nbmw$ 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.

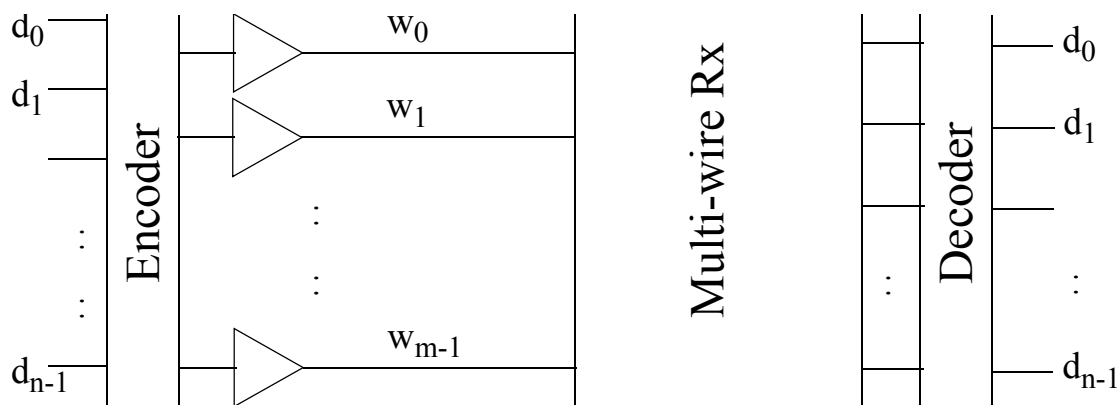


Figure 1 — Generic Architecture of a Multi-Wire Signaling Interface

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.