Lattice Semiconductor Corporation

Reed-Solomon Encoder

April 2003 IP Data Sheet

Features

- **3- to 12-Bit Symbol Width**
- **Configurable Polynomials**
  - Field polynomial
  - Generator polynomial
    - Starting root
    - Root spacing
- **User-defined Codewords**
  - Maximum of 4095 symbols
  - Maximum of 256 check symbols
  - Shortened codes
- **Selectable Reed-Solomon Standards**
  - OC-192
  - DVB
  - CCSDS
  - ATSC
- **Fully Synchronous**
  - User-configured latency
  - Registered input selection
- **Systematic Encoder**
- **Full Handshaking Capability**

General Description

Reed-Solomon codes are used to perform Forward Error Correction (FEC). FEC introduces redundancy in the data before it is transmitted. The redundant data (check symbols) are transmitted with the original data to the receiver. For example, a Reed-Solomon decoder is used to help recover any erred data. This type of error correction is widely used in data communications applications such as Digital Video Broadcast (DVB) and Optical Carriers (i.e. OC-192).

The codes are referred to in the format RS(*n,k*) where *k* is the number of *s*-bit wide information (data) symbols and n is the total number of s-bit wide symbols in a codeword. The Reed-Solomon encoder generates a code such that the first *k* symbols output from the encoder are the information symbols and the next *n-k* symbols from the encoder are the check symbols added for error correction. When the data output is in the same order as the input it is referred to as a *systematic encoder*. Figure 2 illustrates the operation of a systematic encoder.

Block Diagram

Figure 1. Reed-Solomon Encoder Block Diagram

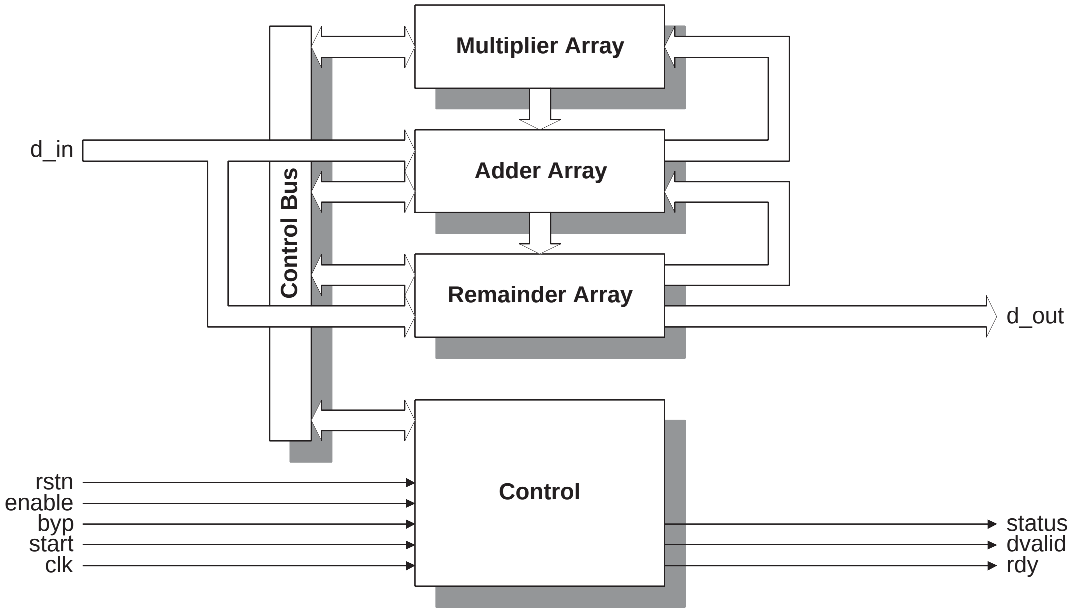
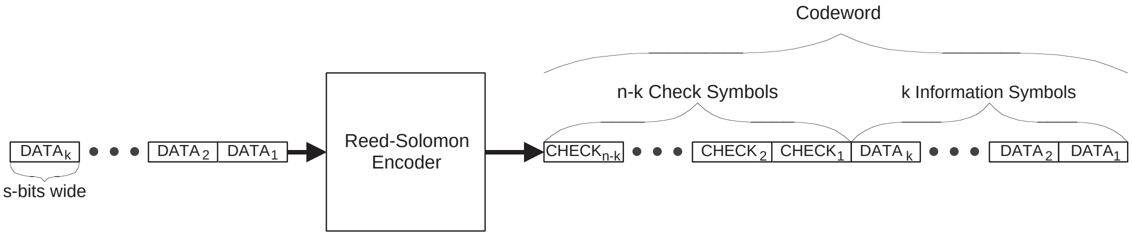


Figure 2. Illustration of a Systematic Reed-Solomon Encoder



A Reed-Solomon Decoder can correct errors and erasures. The maximum number of correctable erroneous symbols in a codeword is t = (n-k)/2, and the maximum number of correctable erasures in a codeword is t = n-k.

Reed-Solomon codes are based on finite field arithmetic, also known as Galois field. The size of the field is determined by the width of the information symbol where the field has 2^s elements. When *n* is less than the maximum value of 2^s-1, it is referred to as a *shortened code.*

Reed-Solomon codes are characterized by two polynomials: the generator polynomial and the field polynomial. The field polynomial defines the Galois field where the information and check symbols belong. The generator polynomial determines the check symbol generation, and it is a prime polynomial for all codewords (i.e. all codewords are exactly divisible by the generator polynomial). Both the field and generator polynomials are user configurable.

Field Polynomial

The field polynomial can be specified as any prime polynomial up to 2^{s+1} - 1. The field polynomial is defined by its decimal value (f). The decimal value of a field polynomial is given by setting x = 2 in the polynomial and calculating the result. For example, the polynomial $x^2 + x + 1$ in decimal value is $2^2 + 2 + 1 = 7$.

Generator Polynomial

The generator polynomial determines the value of the check symbols. The generator polynomial is defined by the starting root (gstart) and the root spacing (rootspace). The general form of the generator polynomial is given by:

$$g(x) = \prod_{i=0}^{n-k-1} (x - \alpha^{\text{rootspace} \cdot (\text{gstart} + i)}) \tag{1}$$

Shortened Codes

Shortened codes are defined as any codeword where *n* is less than 2^s - 1. For example, RS (204,188) where *s* = 8. Only the non-zero data is transmitted to the encoder (i.e. 188 in the above example). The encoder then generates the required check symbol from the non-zero data.

Functional Description

The Reed-Solomon encoder utilizes the Multiplier, Adder, and Remainder arrays to perform finite field arithmetic. The following section explains the operation of the arrays and the Control block.

Multiplier Array

The Multiplier array performs the Galois field multiplication between the generator coefficients and the addition of input data and feedback (modulo 2). This multiplication is an optimized multiplication between the generator coefficients, which are constants, and the input of the multiplier array. This optimization is done during the processing of the core.

Adder Array

The Adder array performs modulo 2 addition on the data from the previous element of the Remainder array and the result of the corresponding Galois field multiplication from the Multiplier array. The outputs from the Adder array are latched into the Remainder array on each clock cycle.

Remainder Array

The Remainder array is a shift register array. It stores the remainder polynomial after the polynomial division. The remainder polynomial becomes the check symbols once all information symbols have been processed. The Remainder array shifts in the data from the Adder array until no information symbols will be received. When all the information symbols have been received, the polynomial multiplication stops and the contents of the Remainder array are output to `d_out`.

Control

The Control block generates all control signals and determines the state of the Reed-Solomon encoder. The `rstn`, `enable`, `byp`, `start`, and `clk` inputs control the state of the encoder. The Control block uses these inputs to control the state of the Multiplier, Adder, and Remainder arrays as well as the `rdy`, `status`, and `dvalid` outputs.

Timing Diagrams

The illustrated timing examples utilize a non-continuous RS(7,3) code. The timing remains the same whether the core is continuous or not. However, when the core is continuous, the `rdy` and `dvalid` signals are not used.

Figure 3 illustrates the timing of an RS (7,3) single pipelined encoder during normal operation. The handshake signals `status`, `rdy`, and `dvalid` display the communication of the encoder with the source and destination devices.

Figure 3. Timing of an RS (7,3) Single Pipelined Encoder

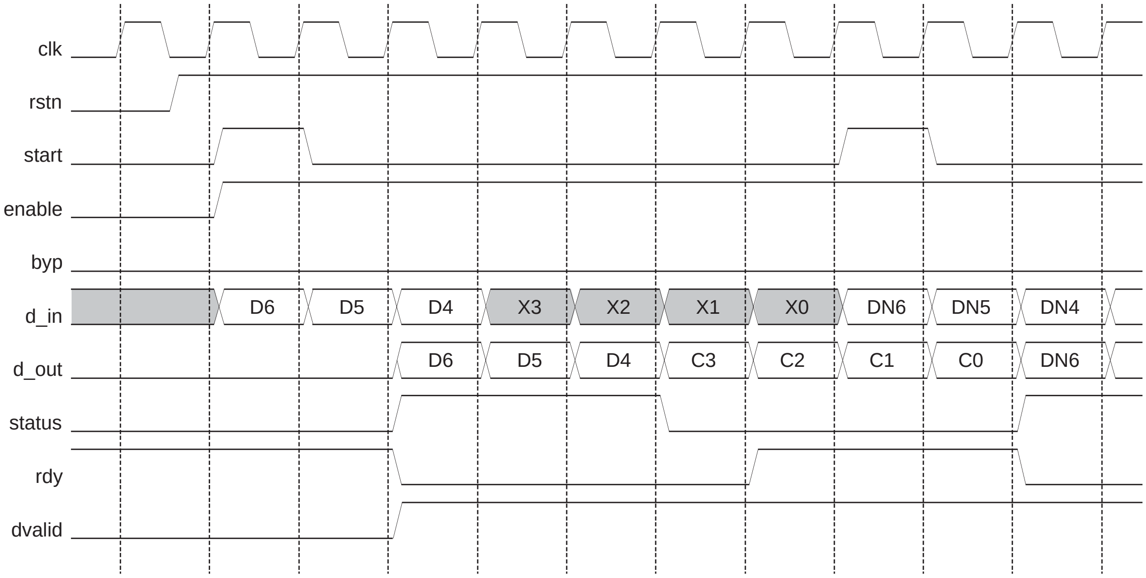


Figure 4 shows the timing of an RS (7,3) single pipelined encoder with `byp` asserted during the operation of the encoder. The handshaking signals are identical to normal operation, but the output is shifted due to the extra bypass data, which does not require check symbols.

Figure 4. Timing of an RS (7,3) Single Pipelined Encoder with* `byp` *Asserted

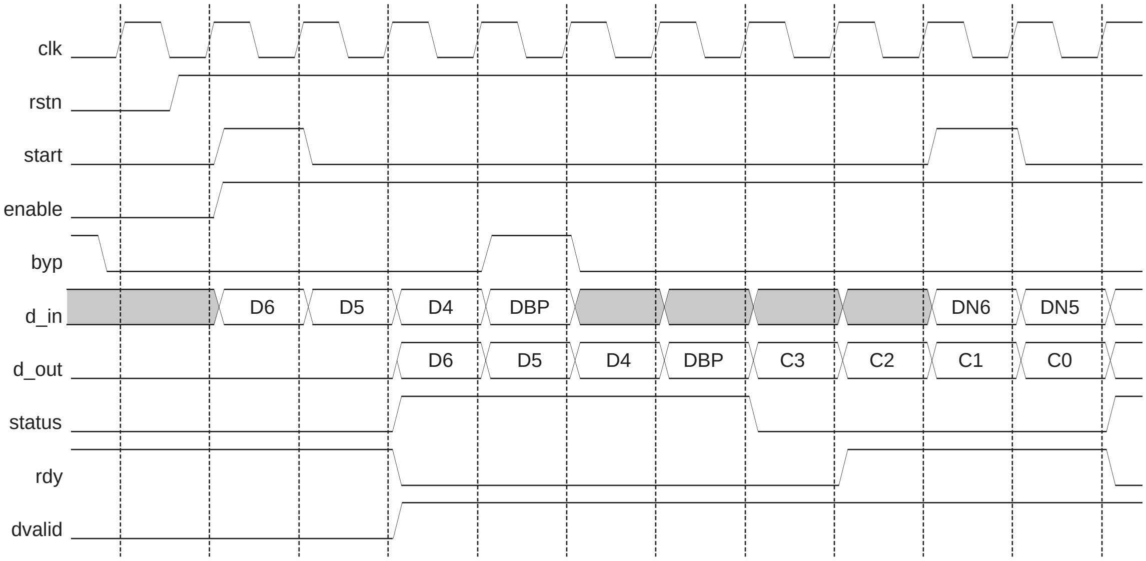


Figure 5 explains the timing of an RS (7,3) single pipelined encoder with `enable` de-asserted during the operation of the encoder. The handshaking signal, `dvalid`, indicates the data on `d_out` is invalid while the encoder maintains its state during the time `enable` is low.

Figure 5. Timing of an RS (7,3) Single Pipelined Encoder with* `enable` *De-asserted

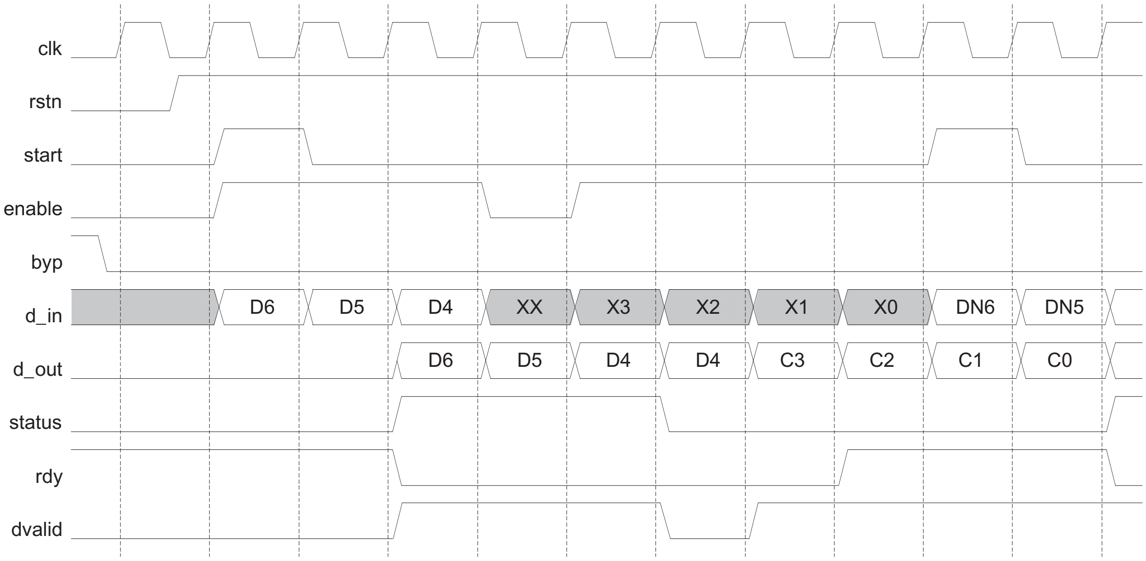


Figure 6 explains the timing of an RS (7,3) single pipelined encoder with `start` re-asserted during the operation of the encoder. The handshaking signal, `rdy`, indicates the encoder is ready to receive a new set of data when `start` is re-asserted during encoding.

Figure 6. Timing of an RS (7,3) Single Pipelined Encoder with `start` Re-asserted

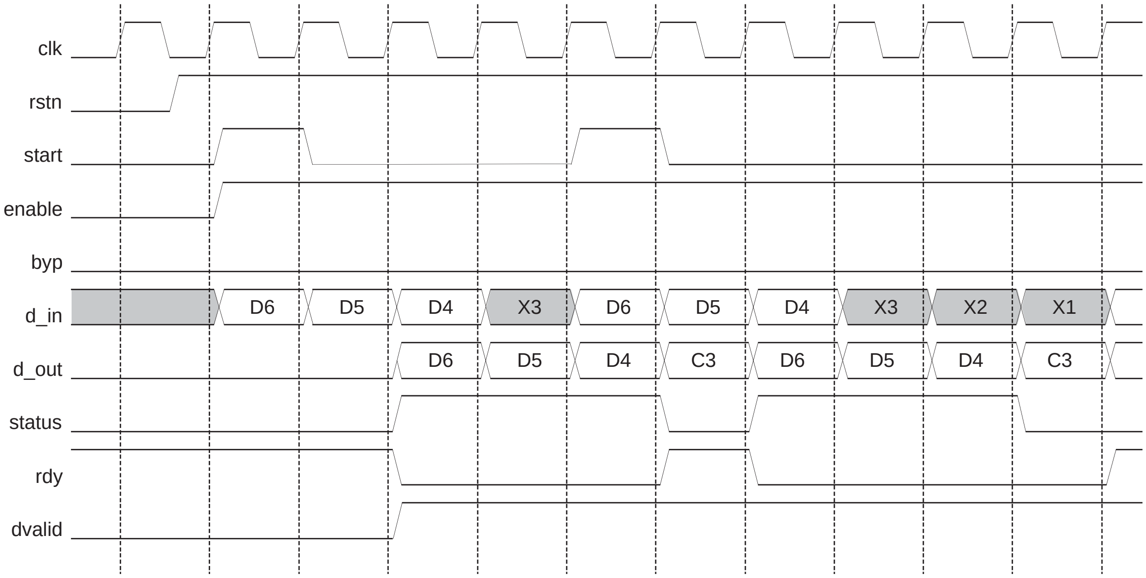


Figure 7 illustrates the timing of an RS (7,3) double pipelined encoder during normal operation. The handshake signals `status`, `rdy`, and `dvalid` display the communication of the encoder with the source and destination devices.

Figure 7. Timing of an RS (7,3) Double Pipelined Encoder

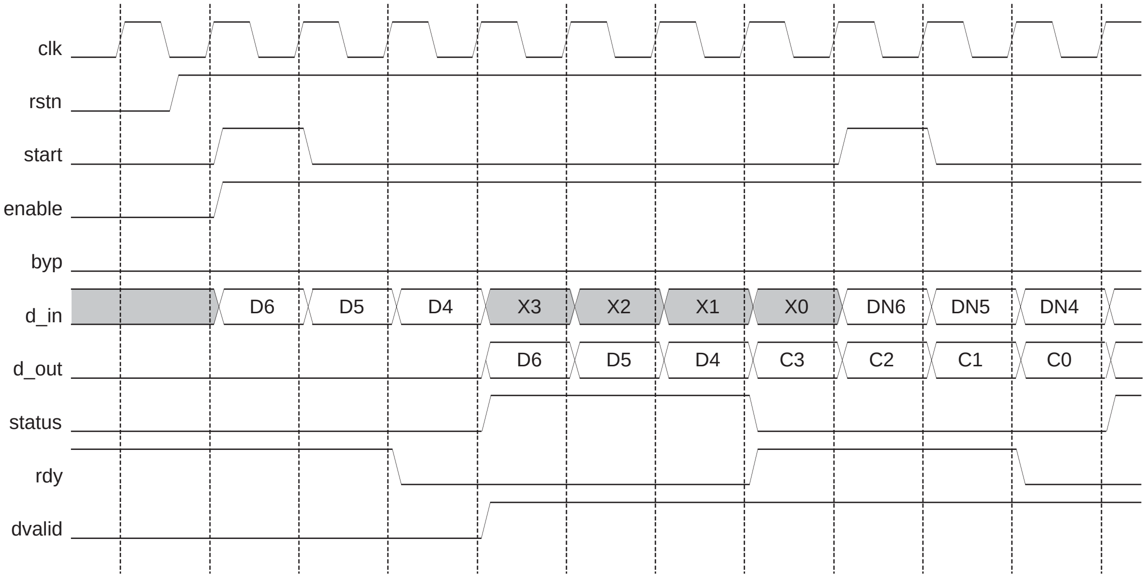


Figure 8 shows the timing of an RS (7,3) double pipelined encoder with `byp` asserted during the operation of the encoder. The handshaking signals are identical to normal operation, but the output is shifted due to the extra bypass data, which does not require check symbols.

Figure 8. Timing of an RS (7,3) Double Pipelined Encoder with* `byp` *Asserted

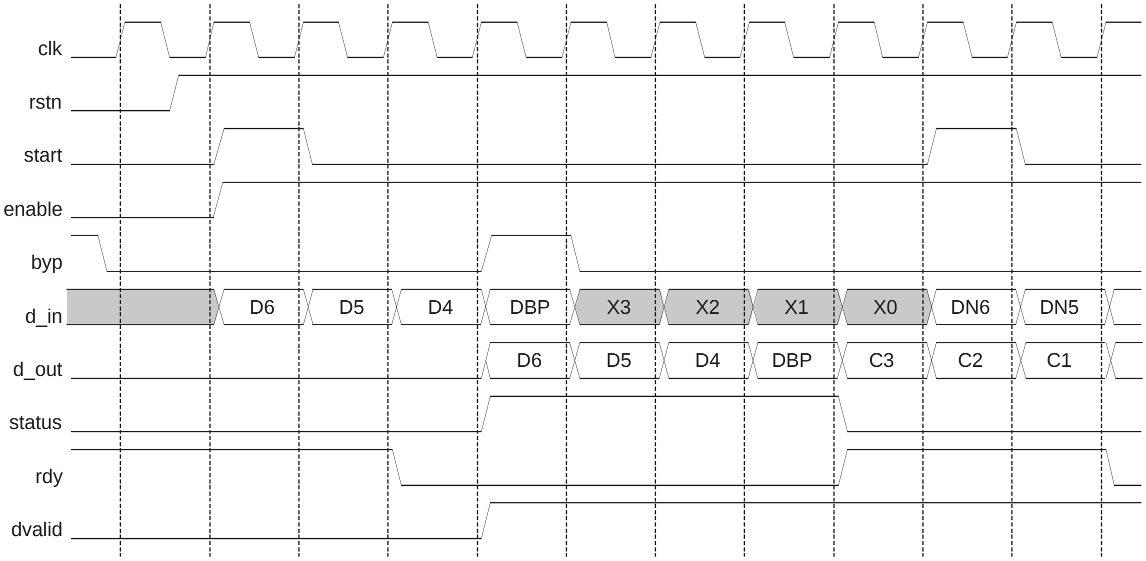


Figure 9 explains the timing of an RS (7,3) double pipelined encoder with `enable` de-asserted during the operation of the encoder. The handshaking signal, `dvalid`, indicates the data on `d_out` is invalid while the encoder maintains its state during the time `enable` is low.

Figure 9. Timing of an RS (7,3) Double Pipelined Encoder with* `enable` *De-asserted

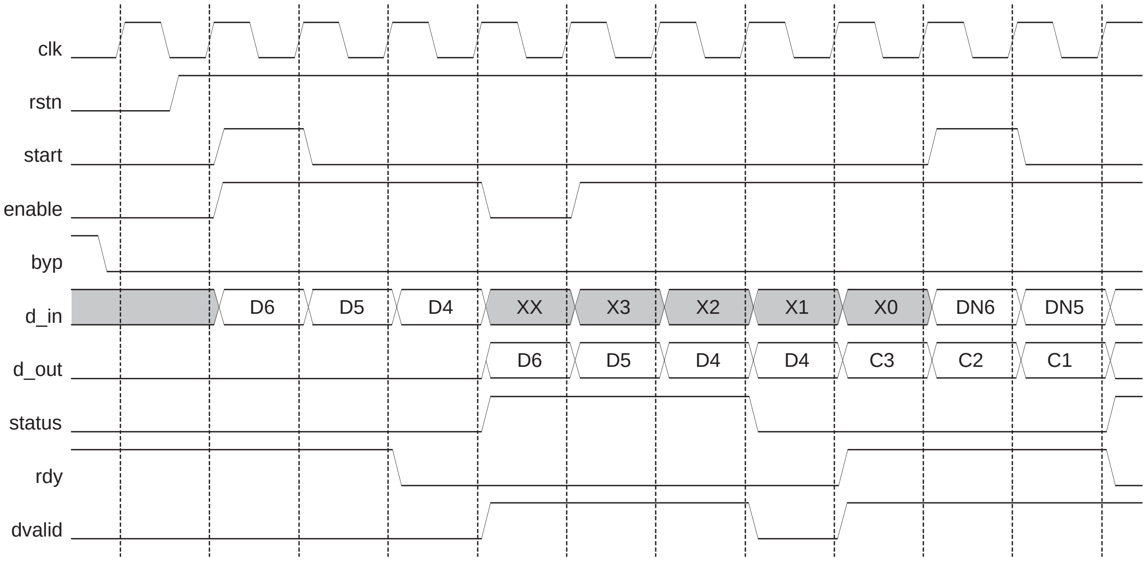
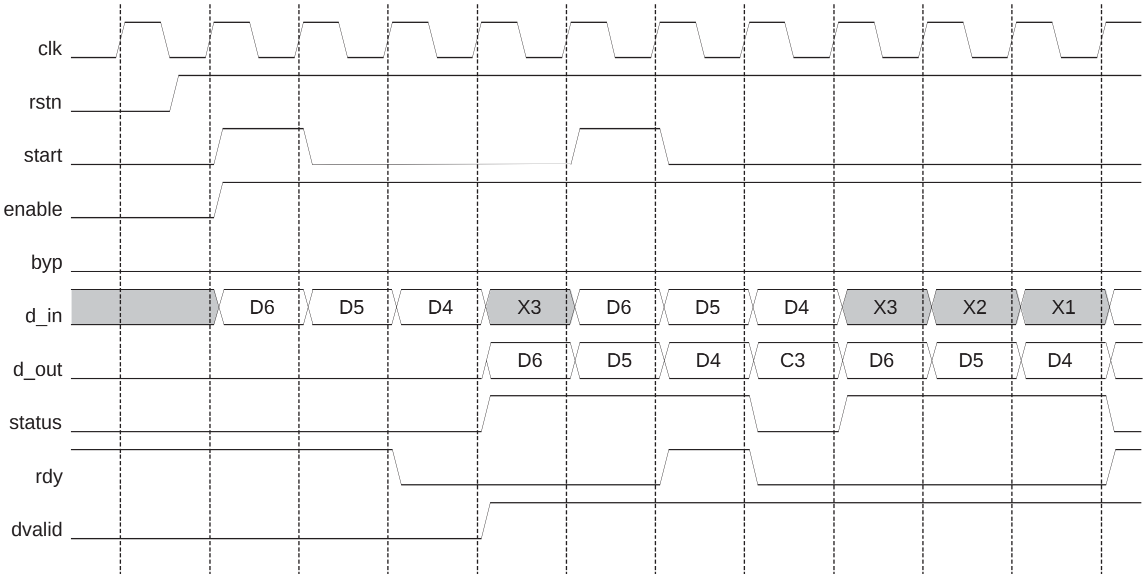


Figure 10 explains the timing of an RS (7,3) double pipelined encoder with `start` re-asserted during the operation of the encoder. The handshaking signal, `rdy`, indicates the encoder is ready to receive a new set of data when `start` is re-asserted during encoding.

Figure 10. Timing of an RS (7,3) Double Pipelined Encoder with `start` Re-asserted



Parameter Descriptions

The Reed-Solomon encoder has several parameterized values, which are user configurable. Table 1 lists the parameters and their associated descriptions.

Table 1. Reed-Solomon Encoder Parameter Descriptions

Name	Value	Default	Description
n	3 - 4095	255	Number of symbols
k	1 - 4093	239	Number of information symbols
s	3 - 12	8	Symbol width
f	11 - 8191	See Table 2	Decimal value of the field polynomial
rootspace	1 - 65535	1	Root spacing of the generator polynomial
gstart	0 - 65535	0	Offset value of the generator polynomial. The starting value for the first root of the generator polynomial is calculated as rootspace * gstart.
inreg	0, 1	1	0 = the inputs will not be registered 1 = the inputs will be registered
latency	2, 3	3	2 = the input on `d_in` will take 2 clock cycles to reach `d_out` 3 = the input on `d_in` will take 3 clock cycles to reach `d_out`
algorithm	0, 1	1	Selects between two different multiplication algorithms. Used to improve timing results
handshake	0, 1	0	1 = the core will be a non-continuous core configuration 0 = the core will be a continuous core configuration (`rdy` and `dvalid` will be used in non-continuous configuration only)

Table 2. Default Field Polynomials

Symbol Width	Default Field Polynomial	Decimal Value
3	$x^3 + x + 1$	11
4	$x^4 + x + 1$	19
5	$x^5 + x^2 +1$	37
6	$x^6 + x + 1$	67
7	$x^7 + x^3 + 1$	137
8	$x^8 + x^4 + x^3 + x^2 + 1$	285
9	$x^9 + x^4 + 1$	529
10	$x^{10} + x^3 + 1$	1033
11	$x^{11} + x^2 + 1$	2053
12	$x^{12} + x^6 + x^4 + x + 1$	4179

Signal Descriptions

Table 3 shows the definitions of the interface signals available with the Reed-Solomon encoder IP Core.

Table 3. Reed-Solomon Encoder Signal Descriptions

Signal Name	I/O	Active State	Signal Description
d_in[s-1:0]	Input	N/A	Input data
rstn	Input	Low	Asynchronous reset input
enable	Input	High	Enables the encoder to process data on d_in. When low, the input data is ignored and d_out holds its state.
byp	Input	High	Indicates the data on d_in should pass directly through to d_out after the latency. This signal is ignored if enable is low.
start	Input	High	Indicates that the data on d_in is the first information symbol of a new codeword. This signal is ignored if byp is high or enable is low.
clk	Input	Rising Edge	Master clock input
d_out[s-1:0]	Output	N/A	Output data
status	Output	High	Indicates the information symbols are present on d_out or byp is asserted.
dvalid	Output	High	Indicates valid data on d_out. Not available with continuous configuration.
rdy	Output	High	Indicates when the encoder is ready to receive data. Active when rstn is asserted or when ready to receive data or start is asserted. Inactive when sufficient data has been received and check symbols are being calculated. Not available with continuous configuration

Custom Core Configurations

For Reed-Solomon Encoder core configurations that are not available in the Evaluation Package, please contact your Lattice sales office to request a custom configuration.

Related Information

For more information regarding core usage and design verification, refer to the *Reed-Solomon Encoder IP Core User's Guide,* available on the Lattice web site at www.latticesemi.com.

Appendix for ORCA® Series 4 FPGAs

Table 4. Performance and Utilization[1]

Parameter File	Mode	PFUs	LUTs	Registers	External Pins	EBRs	f_{MAX} (MHz)
reeds_enco_o4_1_001.lpc	OC192	58	210	194	22	N/A	168
reeds_enco_o4_1_002.lpc	CCSDS	88	327	323	22	N/A	156
reeds_enco_o4_1_003.lpc	DVB	58	201	194	22	N/A	167
reeds_enco_o4_1_004.lpc	ATSC	71	233	226	22	N/A	166

[1] Performance and utilization characteristics for OR4E02-2BA352. When using other devices, performance may vary.

Table 5. Parameters for Typical Configurations

Name	CCSDS	DVB	ATSC	OC192
n	255	204	207	255
k	223	188	187	239
s	8	8	8	8
f	391	285	285	285
rootspace	11	1	1	1
gstart	112	0	0	0
inreg	1	1	1	1
latency	3	3	3	3
algorithm	1	1	1	1
handshake	0	0	0	0

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of the Reed-Solomon Encoder core targeting ORCA Series 4 devices is REEDS-ENCO-O4-N1. Table 4 lists the netlists that are available in the Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com.

Appendix for ispXPGA™ FPGAs

Table 6. Performance and Resource Utilization[1]

Parameter File	Mode	PFUs	LUTs	Registers	External Pins	EBRs	f_{MAX} (MHz)
reeds_enco_xp_1_001.lpc	OC192	86	273	248	24	N/A	166
reeds_enco_xp_1_002.lpc	CCSDS	161	504	457	22	N/A	149
reeds_enco_xp_1_003.lpc	DVB	84	273	240	22	N/A	155
reeds_enco_xp_1_004.lpc	ATSC	130	417	307	22	N/A	157

1. Performance and utilization characteristics for LFX125B-04F256C. When using other devices, performance may vary.

Table 7. Parameters for Typical Configurations

Name	CCSDS	DVB	ATSC	OC192
n	225	204	207	255
k	223	188	187	239
s	8	8	8	8
f	391	285	285	285
rootspace	11	1	1	1
gstart	112	0	0	0
inreg	1	1	1	1
latency	3	3	3	3
algorithm	1	1	1	1
handshake	0	0	0	0

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of the Reed-Solomon Encoder core targeting ispXPGA devices is REEDS-ENCO-XP-N1. Table 6 lists the netlists that are available in the Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com.