System LSI

Reed Solomon IP CoreProgram Version 1.0

For OpenCores

2011 July 26

History

Ver.	YYYY/MM/DD	Author	Comment
1.0	2011/07/26	Kazunari Hayashi	1st Create

Contents

1. Reed Solomon Function

1.1 General Description

This tool is used to generate Reed Solomon IP which includes Encoder/Decoder RTL and testbench. Basically Reed Solomon uses unit of symbol by ‘r’ bits. One block consists of N symbols. N symbols called code symbols includs K symbols of Source and (N-K) symbols of Redundancy. (c.f. Fig.1-1)

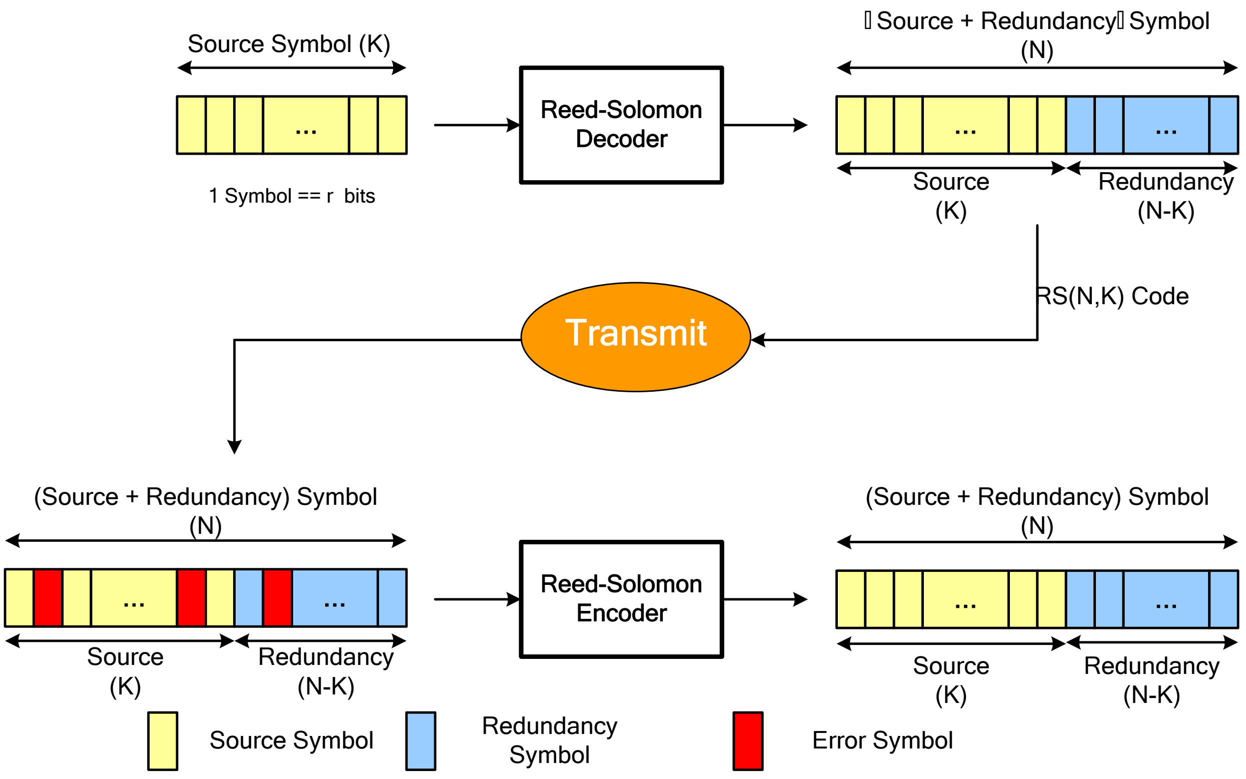

Fig.1-1 Reed Solomon System Block

1.2 Parameter Requirement

RS(N, K)

- Lenght of Symbol = r bits
- N: (Source + Redundancy) Symbols
- K: Source Symbols

Rule1: $0 < K < 2^{(r-1)}$

Rule 2: $0 < N \leq 2^{(r-1)}$

Rule 3: $K < N$

Rule 4: (N-K) modulo 2 = 0

Rule 5: If Erasure correction, ability of correction is $\frac{(N-K)}{2}$

IF disable Erasure correction, ability of correction is

$(2xErr + Era) \leq \frac{(N-K)}{2}$ Err : Number of Error symbols
Era : Number of Erasure symbols

Fig.1-2 Parameter Requirement

2. IP CoreProgram

2.1 Environment

The working of this tool is confirmed on command prompt of Windows XP.
Already confirmed compilers are:

- Microsoft Visual Studio .NET 2003
- Microsoft Visual C++ 2008 Express Edition
- Microsoft Visual Studio 2010 Express Edition

There are many options when the command is invoked.

```
DataSize          = atoi(argv[1]);  // Source Symbol K, range = [1..(2^r)-3]
TotalSize         = atoi(argv[2]);  // Total Symbol N, range = [3..(2^r)-1]
PrimPoly          = atoi(argv[3]);  // Primitive Polynomial
ErasureOption     = atoi(argv[4]);  // Erasure Decoding (0: No, 1:Yes)
BlockAmount       = atoi(argv[5]);  // [sim]RS Decoder Block Amount
ErrorRate         = atoi(argv[6]);  // [sim]Error rate, range = [0..99]
PowerErrorRate    = atoi(argv[7]);  // [sim]Power of Error Rate (0: 1, 1: 10^-1, 4: 10^-4)
ErasureRate       = atoi(argv[8]);  // [sim]Erasure rate, range = [0..99]
PowerErasureRate  = atoi(argv[9]);  // [sim]Power of Error Rate (0: 1, 1: 10^-1, 4: 10^-4)
bitSymbol         = atoi(argv[10]); // bit of Symbol, range = [3..12]
errorStats        = atoi(argv[11]); // Error Status (0: No, 1:Yes)
passFailFlag      = atoi(argv[12]); // Pass/Fail Flag (0: No, 1:Yes)
delayDataIn       = atoi(argv[13]); // Delayed Data (0: No, 1:Yes)
encDecMode        = atoi(argv[14]); // Select Mode (1: enc only, 2:dec only, 3: enc & dec)
encBlockAmount    = atoi(argv[15]); // encoder block amount
ipCustomerKey     = atoi(argv[16]); // Revision ID=27
```

Example: in MS Command Prompt

```
> RsIpEngine.exe 233 255 285 1 10 25 2 3 0 8 1 1 1 3 3 27
```

2.2 Program Files

Table2-1 Description of Program Files

FILE NAME	COMMENT
RsIpEngine.cpp	top file (main)
RsMrefTab.cpp	Mref table of Galois Field
RsPrefTab.cpp	Pref table of Galois Field
RsEncode.cpp	Encoder Top
RsDecode.cpp	Decoder Top
RsSimBench.cpp	[RTL] Testbench generator
RsGfMultiplier.cpp	Multiplier of Galois Field
RsGfInverse.cpp	Inverse of Galois Field
RsEncodeTop.cpp	[RTL] Encoder top (generate RsEncodeTop.v)
RsEncodeMakeData.cpp	[RTL] Encoder testpattern (generate RsEncIn.hex and RsEncOut.hex)
RsDecodeSyndrome.cpp	[RTL] Decoder Syndrome calc (generate RsDecodeSyndrome.v)
RsDecodeErasure.cpp	[RTL] Decoder Erasure calc (generate RsDecodeErasure.v)
RsDecodePolymul.cpp	[RTL] Decoder polymul calc (generate RsDecodePolymul.v)
RsDecodeEuclide.cpp	[RTL] Decoder Euclide Algorithm (generate RsDecodeEuclide.v)
RsDecodeShiftOmega.cpp	[RTL] Decoder Omega shifter (generate RsDecodeShiftOmega.v)
RsDecodeDegree.cpp	[RTL] Decoder Degree calc (generate RsDecodeEuclide.v)
RsDecodeChien.cpp	[RTL] Decoder Chien Search (generate RsDecodeChien.v)
RsDecodeInv.cpp	[RTL] Decoder Inverse (generate RsDecodeInv.v)
RsDecodeDelay.cpp	[RTL] Decoder Delayed data (generate RsDecodeDelay.v)
RsDecodeDpRam.cpp	[RTL] Decoder DPRAM (generate RsDecodeDpRam.v)
RsDecodeTop.cpp	[RTL] Decoder top (generate RsDecodeTop.v)
RsDecodeMul.cpp	[RTL] Decoder Multiplier of Galois Field (generate RsDecodeMult.v)
RsDecodeMakeData.cpp	[RTL] Decoder testpattern (generate RsDecIn.hex and RsDecOut.hex)
RsDecodeEmulator.cpp	Decoder emulation: top file
RsDecodeSyndromeEmulator.cpp	Decoder emulation: Syndrome
RsDecodeErasureEmulator.cpp	Decoder emulation: Erasure
RsDecodePolymulEmulator.cpp	Decoder emulation: polymul
RsDecodeEuclideEmulator.cpp	Decoder emulation: Euclide Algorithm
RsDecodeShiftOmegaEmulator.cpp	Decoder emulation: Omega shifter
RsDecodeDegreeEmulator.cpp	Decoder emulation: Degree
RsDecodeChienEmulator.cpp	Decoder emulation: Chien Search

2.3 Program Hierarchy

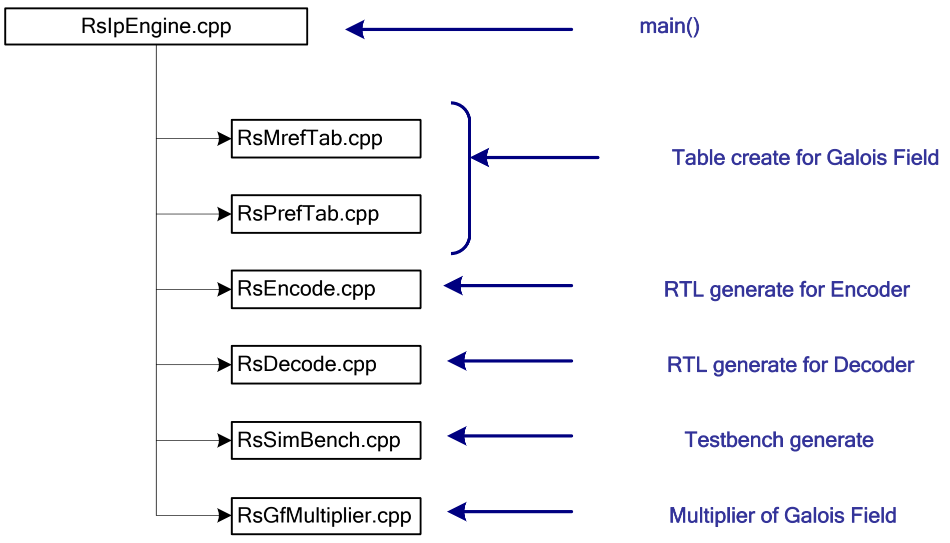

Fig.3-1 TOP structure

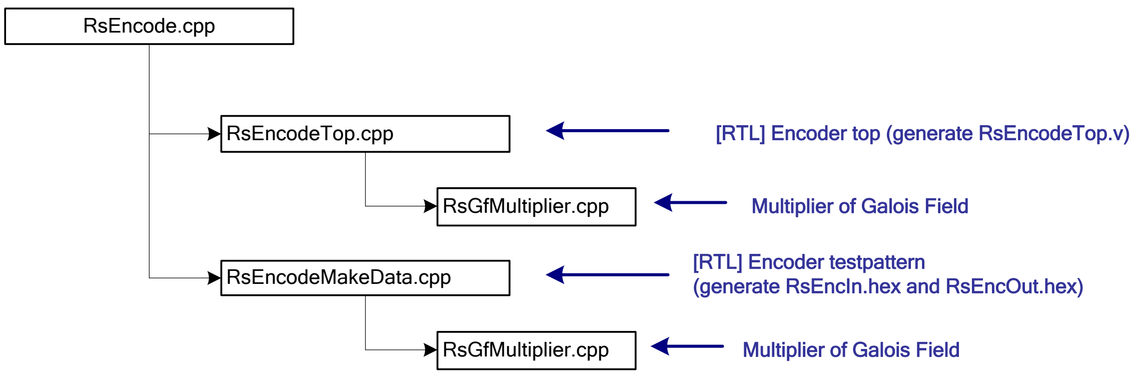
RsEncode.cpp
RsEncodeTop.cpp
[RTL] Encoder top (generate RsEncodeTop.v)
RsGfMultiplier.cpp
Multiplier of Galois Field
RsEncodeMakeData.cpp
[RTL] Encoder testpattern
(generate RsEncIn.hex and RsEncOut.hex)
RsGfMultiplier.cpp
Multiplier of Galois Field

Fig.3-2 RsEncode structure

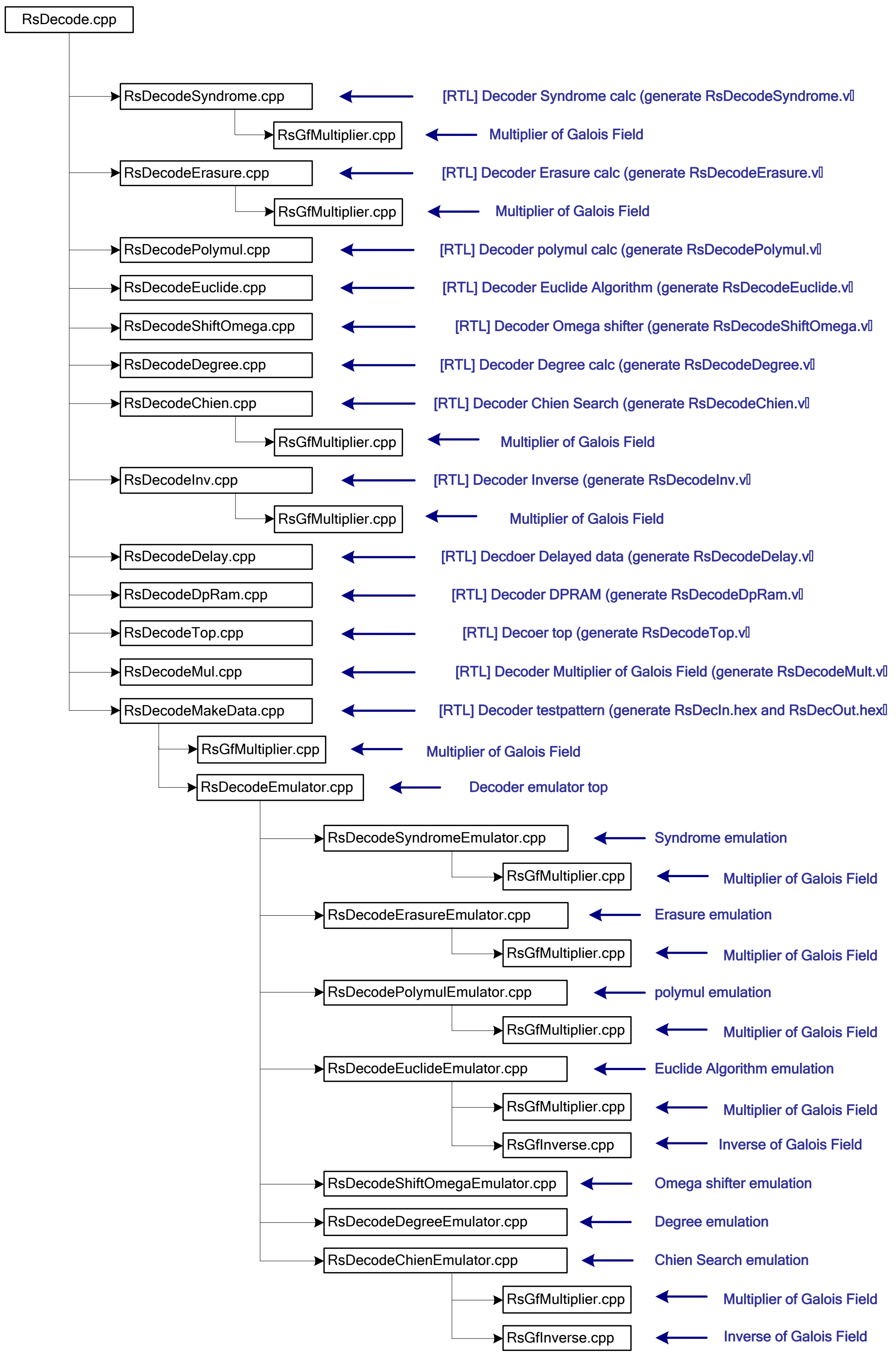
RsDecode.cpp
RsDecodeSyndrome.cpp
[RTL] Decoder Syndrome calc (generate RsDecodeSyndrome.v）
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodeErasure.cpp
[RTL] Decoder Erasure calc (generate RsDecodeErasure.v）
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodePolymul.cpp
[RTL] Decoder polymul calc (generate RsDecodePolymul.v）
RsDecodeEuclide.cpp
[RTL] Decoder Euclide Algorithm (generate RsDecodeEuclide.v）
RsDecodeShiftOmega.cpp
[RTL] Decoder Omega shifter (generate RsDecodeShiftOmega.v）
RsDecodeDegree.cpp
[RTL] Decoder Degree calc (generate RsDecodeDegree.v）
RsDecodeChien.cpp
[RTL] Decoder Chien Search (generate RsDecodeChien.v）
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodeInv.cpp
[RTL] Decoder Inverse (generate RsDecodeInv.v）
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodeDelay.cpp
[RTL] Decdoer Delayed data (generate RsDecodeDelay.v）
RsDecodeDpRam.cpp
[RTL] Decoder DPRAM (generate RsDecodeDpRam.v）
RsDecodeTop.cpp
[RTL] Decoer top (generate RsDecodeTop.v）
RsDecodeMul.cpp
[RTL] Decoder Multiplier of Galois Field (generate RsDecodeMult.v）
RsDecodeMakeData.cpp
[RTL] Decoder testpattern (generate RsDecIn.hex and RsDecOut.hex）
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodeEmulator.cpp
Decoder emulator top
RsDecodeSyndromeEmulator.cpp
Syndrome emulation
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodeErasureEmulator.cpp
Erasure emulation
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodePolymulEmulator.cpp
polymul emulation
RsGfMultiplier.cpp
Multiplier of Galois Field
RsDecodeEuclideEmulator.cpp
Euclide Algorithm emulation
RsGfMultiplier.cpp
Multiplier of Galois Field
RsGfInverse.cpp
Inverse of Galois Field
RsDecodeShiftOmegaEmulator.cpp
Omega shifter emulation
RsDecodeDegreeEmulator.cpp
Degree emulation
RsDecodeChienEmulator.cpp
Chien Search emulation
RsGfMultiplier.cpp
Multiplier of Galois Field
RsGfInverse.cpp
Inverse of Galois Field

Fig.3-3 RsDecode structure

3. Verilog-RTL Modules

3.1 Encoder

Table 3-1 Encoder Module I/O

Signal Name	I/O	Comment
CLK	入力	system clock (Active on rising edge)
RESET	入力	Async. reset (Active on negative)
enable	入力	system enable (Active Hi)
startPls	入力	Start pulse
dataIn [r-1:0]	入力	Source data
dataOut [r-1:0]	出力	Encoded data

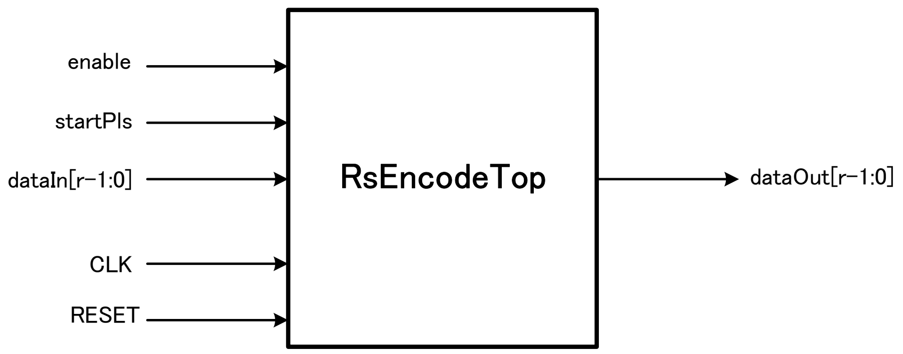

Fig.3-1 Encoder I/F

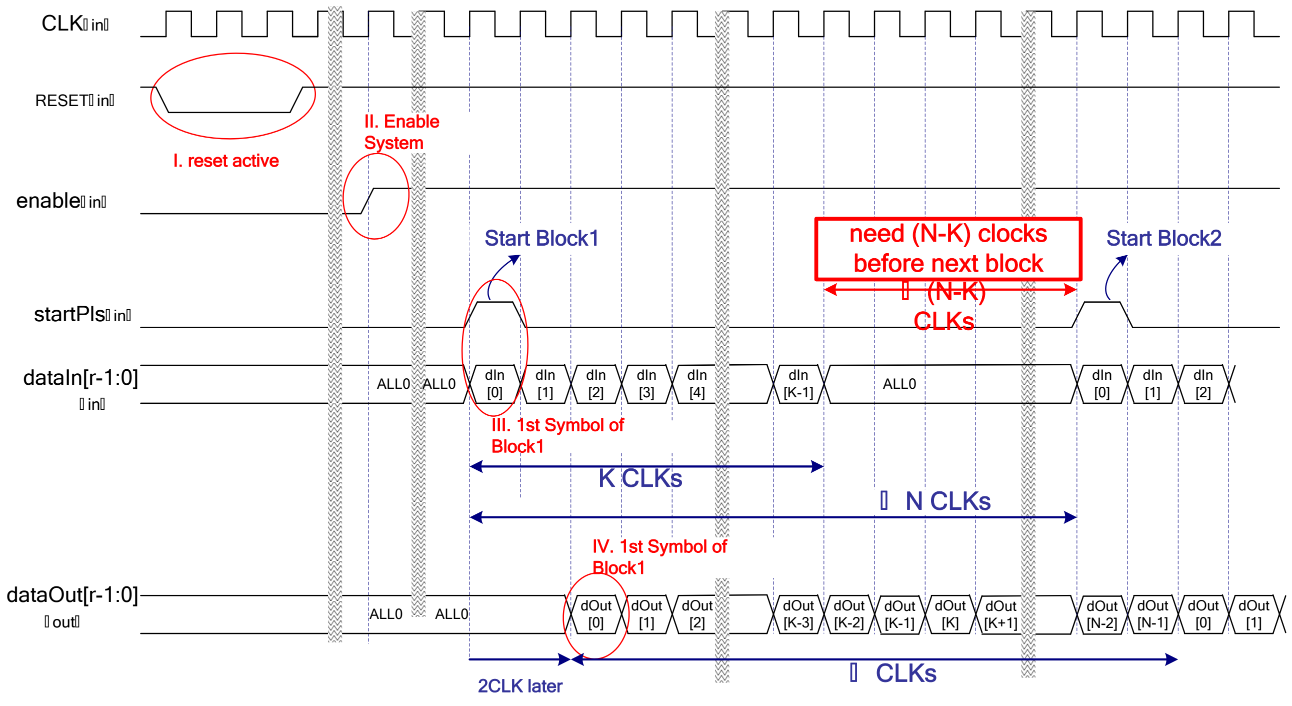

Fig.3-2 Encoder Waveform

3.2 Decoder

Table 3-2 Decoder Module I/O

Signal Name	Option	I/O	Comment
CLK	No	入力	system clock (Active on rising edge)
RESET	No	入力	Async. reset (Active on negative)
enable	No	入力	system enable (Active Hi)
startPls	No	入力	Start pulse
dataIn [r-1:0]	No	入力	Encoded data
outDdata [r-1:0]	No	出力	Decoded data
outEnable	No	出力	Enable of Decoded data
outStartPls	No	出力	Start pulse to Decoded data
Outdone	No	出力	Indicator of last data
erasureIn	Option	入力	Erasure (0:disable 1:enable)
Fail	Option	出力	Decode result (0:success 1:fail)
errorNum[r-1:0]	Option	出力	number of correction if Fail=0
erasureNum[r-1:0]	Option	出力	number of erasure if Fail=0
delayedData[r-1:0]	Option	出力	mirror of Encoded data

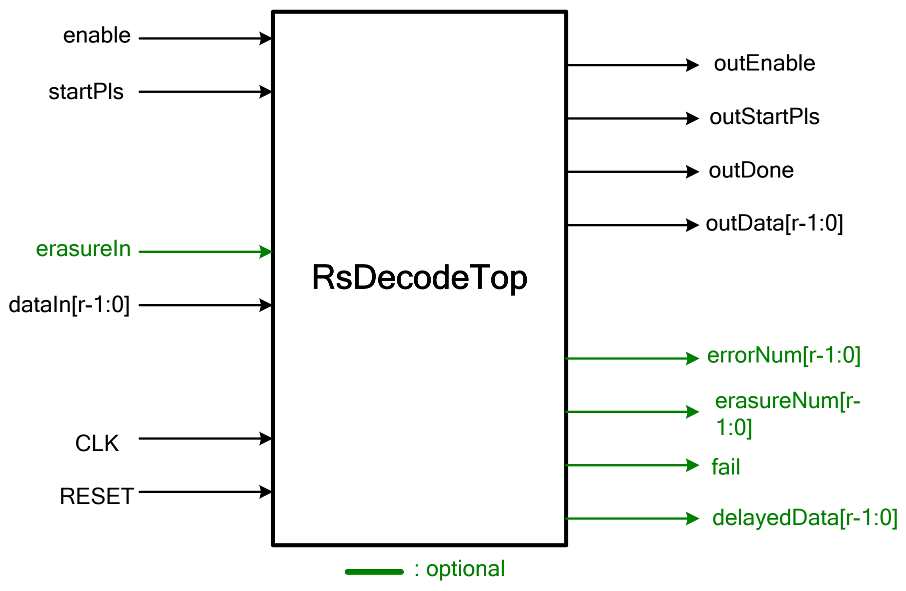

Fig.3-3 Decoder I/F

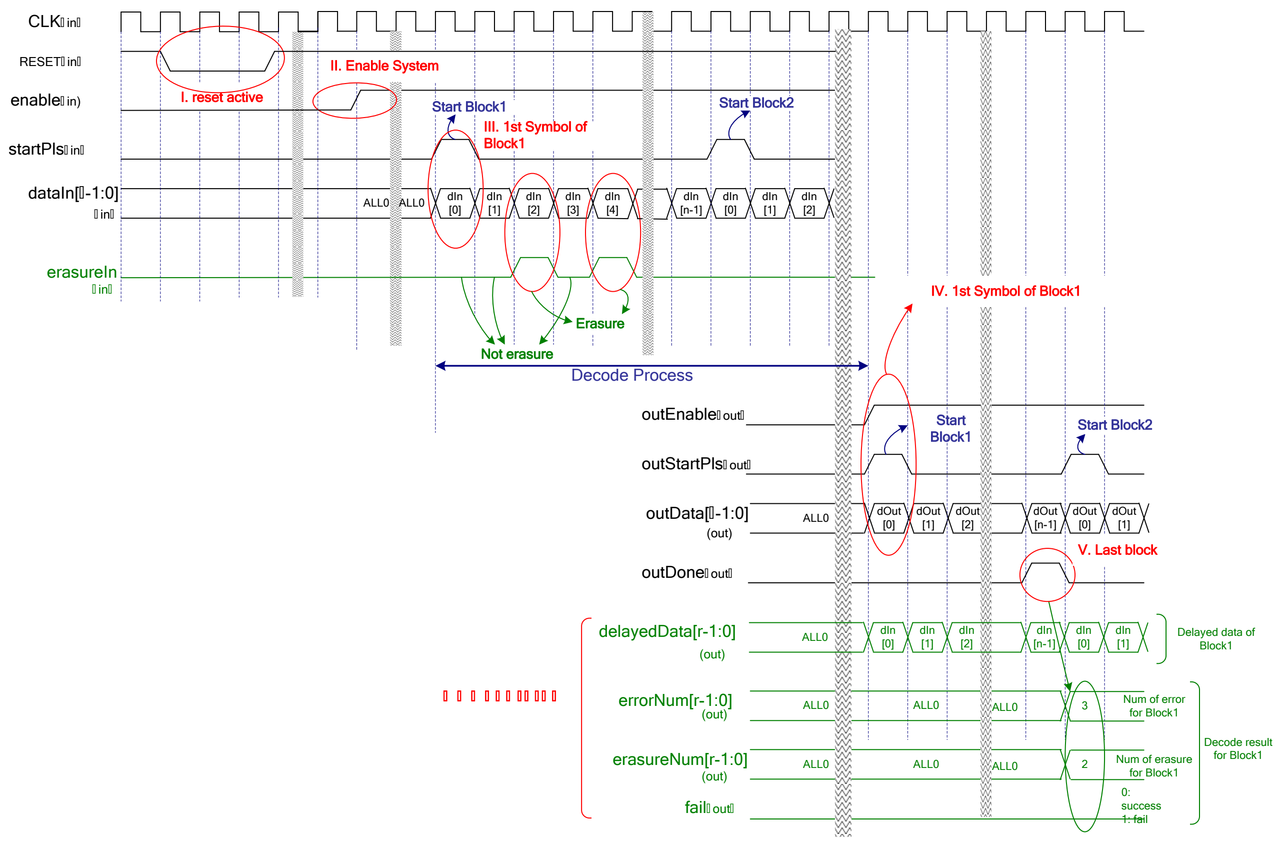

Fig.3-4 Decoder Waveform

3.3 Implementation

Table 3-3 Encoder implementation for FPGA

XC5VLX30-3-FF324	
LUT/FF pairs	154
LUT	69
FF	7
fMAX	380MHz

Table 3-4 Decoder implementation for FPGA

XC5VLX30-3-FF324	
Source Symbol (K)	188
Encoded Symbol (N)	204
Primitive Polynomial	285
Erasure Option	Disable
Error Status	Disable
Pass/Fail Flag	Disable
Delayed Data	Disable
LUT/FF pairs	938
LUT	1380
FF	255
fMAX	368Mhz

Table 3-5 Code Coverage of Encoder

Source Symbol (K)	188
Encoded Symbol (N)	204
Primitive Polynomial	285
Block Coverage	100%

Table 3-6 Code Coverage of Decoder

Source Symbol (K)	188
Encoded Symbol (N)	204
Primitive Polynomial	285
Erasure Option	Disable
Error Status	Disable
Pass/Fail Flag	Disable
Delayed Data	Disable
Block Coverage	100%