

Reed Solomon Decoder

April 11, 2010

Features

- Reed Solomon Decoder (204,188), with T=8.
- Input codeword length is 204 bytes and output length is 188 bytes.
- Corrects up to 8 byte errors per input codeword.
- Code generator polynomial: $(x + \lambda)(x + \lambda^2)(x + \lambda^3)\dots(x + \lambda^{16})$.
- Field generator polynomial: $x^8 + x^4 + x^3 + x^2 + 1$.
- A functional flow chart is shown in Figure 1.

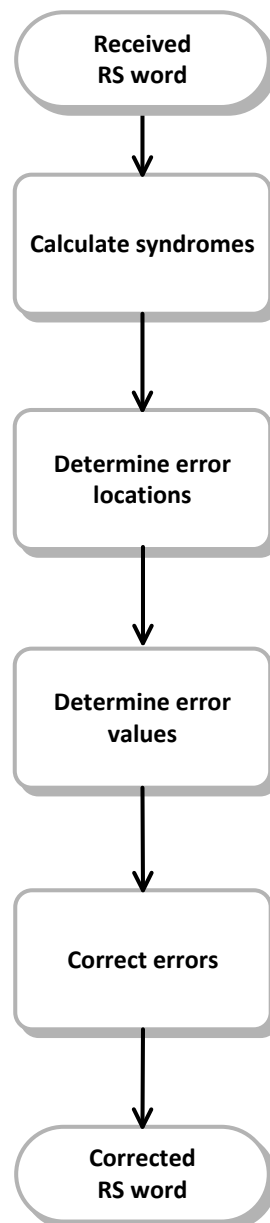


Figure 1: Functional flow chart of the Reed Solomon decoder

Steps of operation

- For every input codeword, the syndromes are calculated, and the input codeword is stored in the input pipelining memories.
- The syndromes will be used to get error location polynomial (Lambda) using Berlekamp Massey's algorithm. Coefficients will be from L1:L8 which will represent the Lambda in the following form:

$$\text{Lambda} = 1 + L_1 * X + L_2 * X^2 + L_3 * X^3 + \dots + L_8 * X^8.$$
- Lambda polynomial will be input to two blocks works in the same time, these two blocks are:
 - Find roots: find roots of Lambda polynomial (up to 8 roots).
 - Omega-Phy calculation: the syndromes is also an input to this block to calculate two polynomials
 - * Omega = Lambda * syndromes.
 - * Phy = derivative(Lambda).
- Error correction unit calculates the errors locations and values, then it will write the corrected values in the pipelining memories, as shown in Figure 2.
- The output stage block releases corrected output from pipelining memories and also generates output handshaking flags.

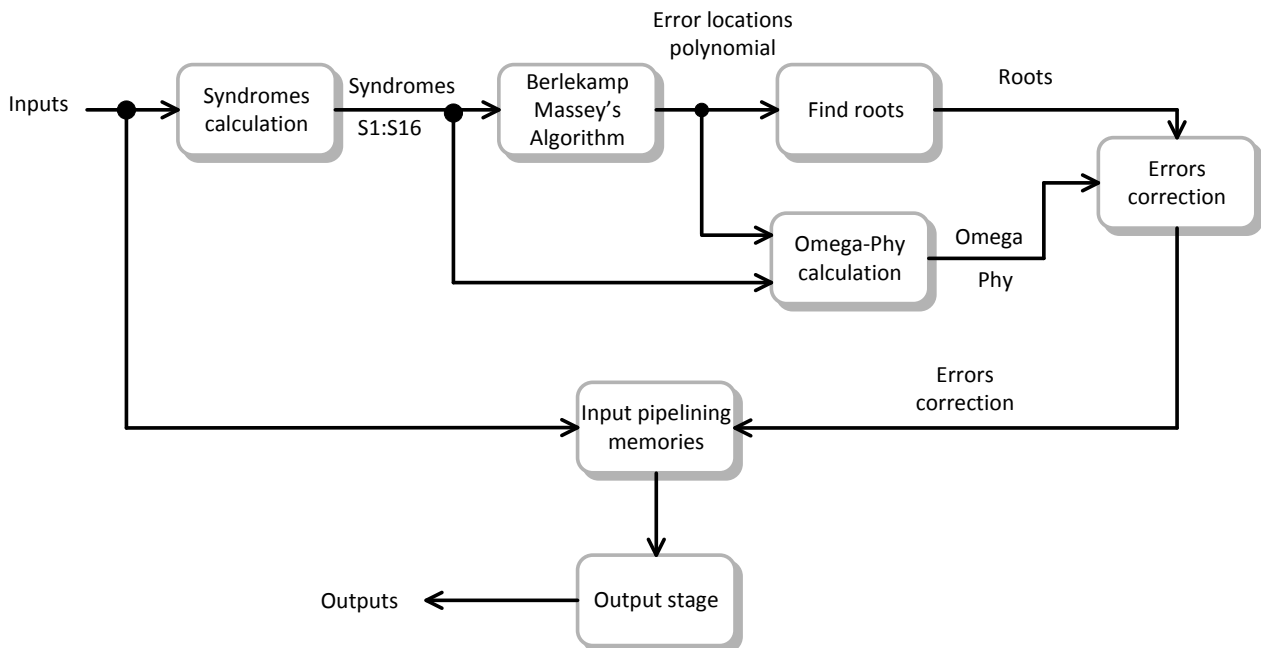


Figure 2: Block diagram of the Reed Solomon decoder

Module declaration

Listing 1: Module declaration of the Reed Solomon decoder

```

module RS_dec
(
    input clk , // input clock

```

```

input reset , // active high asynchronous reset
//active high flag for one clock with every input byte
//(minimum spacing 8 clocks)
input CE,
input [7:0] input_byte , // input byte

output [7:0] Out_byte , // output byte
// chip enable for the next block will be active high for one clock , every 8 clks
output CEO,
output Valid_out /// valid out for every output block (188 byte)
);

```

Timing diagrams

- The core is pipelined, and minimum latency between every input byte and the next input byte is 8 clock cycles, as shown in Figure 3.

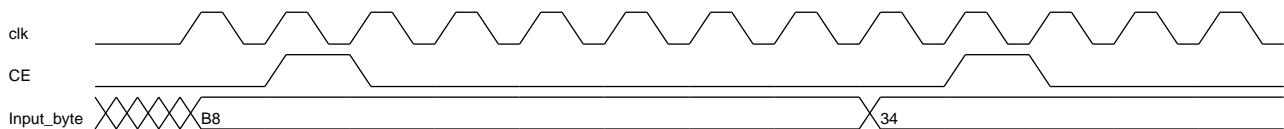


Figure 3: Input Timing Pattern

- Figure 4 is illustrating the output bytes timing, one output byte every 8 clocks.

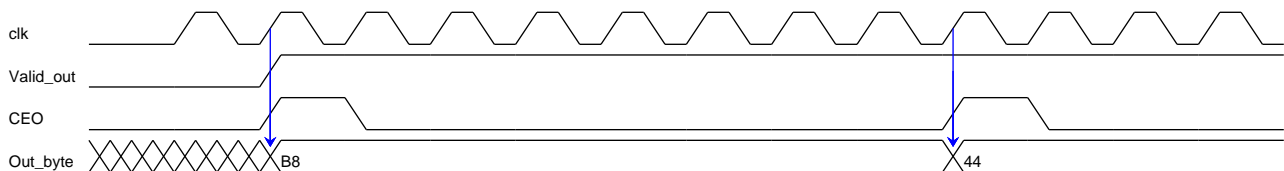


Figure 4: Output Timing Pattern

RTL Verification

Every internal block on the Reed Solomon Decoder was verified separately, also the whole design was verified using long test vectors up to 10000 input codewords with codeword errors up to 8 and all testing results were matched with the system model.

Synthesis Results

- ASIC Synthesis Results on TSMC 180 nm:
 - Total cell area (excluding memory blocks): 454,965 nm^2 .
 - Estimate gate count (excluding memory blocks): 45,592 gates.
 - Number of cells (excluding memory blocks): 15517.
 - Total memory bits are: 10,384 bits.
 - Best achievable clock is: 105.82 MHz.
- Synthesis Results on Xilinx Spartan 3A DSP:

- Number of occupied slices: 3,397/23,872 (14%).
- Best achievable clock is: 82.75 MHz.
- Total block RAMs RAMB16BWERs: 11/126 (8%).
- Synthesis Results on Altera Stratix III EP3SL150F1152C2:
 - Logic utilization: 5%.
 - Combinational ALUTs: 4,294 / 113,600 (4 %).
 - Memory ALUTs: 288 / 56,800 (< 1 %).
 - Dedicated logic registers: 3,511 / 113,600 (3 %).
 - Total block memory bits: 10,384 / 5,630,976 (< 1 %).
 - Best achievable clock is: 250 MHz.

Deliverables

- Verilog RTL files.
- Simulation test bench.
- MATLAB script to generate test vectors.
- Make scripts for synthesis and simulation.
- Documentation.

This version of the Reed Solomon core is distributed under the GPL license. An optimized and considerably more advanced version, which may be customized on request for different code generator polynomials, is available under a commercial license.

Simulation files description

- **RS_dec.tb.v:** Verilog test bench to test the Reed Solomon Decoder core, by feeding input test cases to the core and compare outputs of the core with true outputs, (using input and output files), the test bench is configured to make a functional simulation to the core, in case of post place and route simulation you should uncomment the timescale on the first line of the test bench and set the required clock value by configuring the value of the parameter pclk.
- **input_RS_blocks:** The inputs file, the test bench uses this file to feed inputs to the core, the file should contain the input bytes in binary format, every line should contain a single byte.
- **output_RS_blocks:** The outputs file, the test bench uses this file to verify the outputs from the core, the file should contain the true output bytes in binary format, every line should contain a single byte.
- **RS_test_vectors.m:** MATLAB script to generate test vectors for the core, it will generate random data and encode it using MATLAB rs_enc then put errors on the code from 0:8 byte errors on every codeword, then it will generate inputs and outputs test files to be used by the Verilog test bench.
- **wave.do:** Modelsim wave file contain inputs and outputs ports of the design.
- **do.do:** Modelsim macro to compile the verilog files, load the wave file, and start graphical functional simulation.

About Varkon Semiconductors

Varkon Semiconductors specializes in IP cores for Digital Media Solutions, and Physical Layer Design. Additionally, we offer Digital Design services for ASIC and FPGA implementation of communications transceivers for consumer electronics, including OFDM systems. Varkon Semiconductors supplies PHY layer designs to chip manufacturers, with the goal of giving our customers the lead in the market through our efficient and highly performing algorithms.

Mailing Address:

El Salam Tower, Beside El-Salaam Hospital
Cornish El-Nile, Maadi, 11431
Cairo, Egypt

Tel: +1-732-447-8611 / +20-2-252-88-225

Fax: +1-732-645-1754 / +20-2-252-88-224

www.varkonsemi.com