EE 577B Fall 2007: Extra Credit Homework #1

Zhiyang Ong
Ming Hsieh Department of Electrical Engineering,
Viterbi School of Engineering,
University of Southern California

Synthesizable behavorial RTL models for the encoder and decoder of the (15,11,3) Hamming code were developed in Verilog. They were synthesized into Verilog netlists, and simulation-based verification were used to check the behavioral models and the synthesized Verilog netlists.

Categories and Subject Descriptors: VLSI [**Digital IC Design**]: Communication Electronics

General Terms: Error Correction and Detection, Logic Synthesis, Functional Verification

Additional Key Words and Phrases: Error-Correcting Code, Parity-Check Matrix, Generator Matrix, Decoder,Encoder

1. INTRODUCTION

The encoder and decoder of the (15,11,3) Hamming code [Weste and Harris 2005] were developed in Verilog, using synthesizable behavorial RTL models. They were verified using circuit simulation. Subsequently, logic synthesis was performed on these models of the encoder and the decoder. The resultant synthesized Verilog netlists were verified via simulation again.

Section 1 describes the generation of the generator and parity-check matrices used for creating the encoder and the error-correcting decoder. Next, Section 2 indicates the implementation of the encoder and decoder as synthesizable behavorial RTL models in Verilog. Subseuqently, Section 4 describes how they fit into a pipelined communications channel, which is simulated. Following that is Section 5 which describes the results from performing logic synthesis on the behavioral models, and Section 6 which describes how the simulation waveforms differ in the synthesized Verilog netlists from the behavioral models. Finally, Section 7 draws conclusion on this brief report.

2. GENERATOR MATRIX G AND PARITY CHECK MATRIX H

In Table I, the relationship between data bits, B_i, and parity bits, P_i, with the encoded data bits, C_i is given. This is used to form the relationships between data bits, B_i, and parity bits, P_i, as indicated below.

P1 = B1 + B2 + B4 + B5 + B7 + B9 + B11
P2 = B1 + B3 + B4 + B6 + B7 + B10 + B11
P3 = B2 + B3 + B4 + B8 + B9 + B10 + B11

Author's addresses: zhiyango@usc.edu
Author's student number: 6004 9194 12

Table I. Table representing the relationship between data bits, B_i, and parity bits, P_i, with the encoded data bits, C_i

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
P1	P2	B1	P3	B2	B3	B4	P4	B5	B6	B7	B8	B9	B10	B11

$$P4 = B5 + B6 + B7 + B8 + B9 + B10 + B11$$

The parity check matrix H and the generator matrix G for the $m = 4$ Hamming code with $d_{min} = 3$ is given as follows:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3. IMPLEMENTATION OF ENCODER AND DECODER MODULES IN VERILOG

The encoder and decoder modules for encoding and decoding using linear block codes, (15,11,3) Hamming code, are implemented in Verilog, using behavioral RTL design style. The listings for the encoder, single-bit error-correcting decoder, and their testbenches are provided as attachments at the end of this report.

Figure 1 shows the simulation waveforms for the pre-synthesized behavioral RTL for the Hamming encoder, while Figure 2 shows the simulation waveforms for the pre-synthesized behavioral RTL for the error-correcting Hamming decoder.

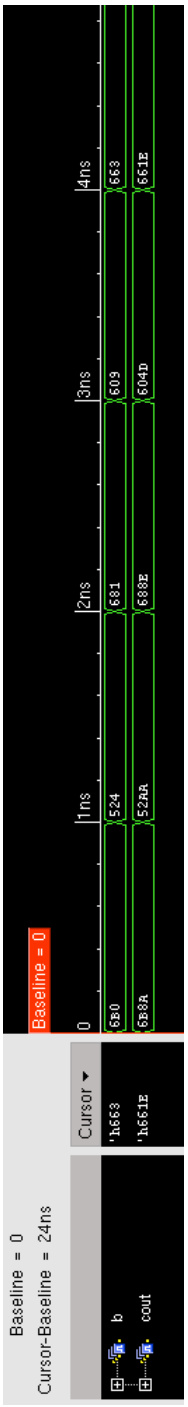

Fig. 1. Simulation Waveforms for the pre-synthesized Behavioral RTL for the Hamming Encoder.

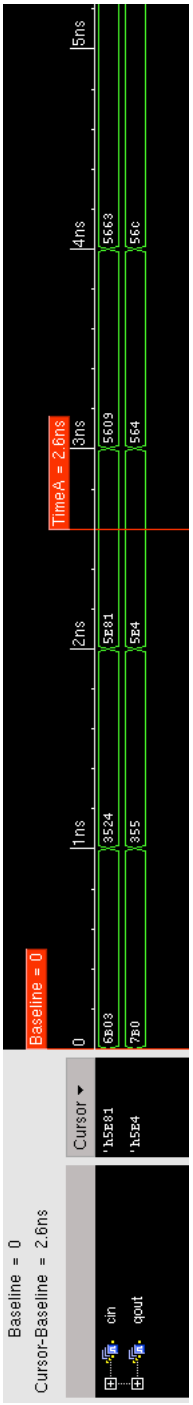

Fig. 2. Simulation Waveforms for the pre-synthesized Behavioral RTL for the Error-Correcting Hamming Decoder.

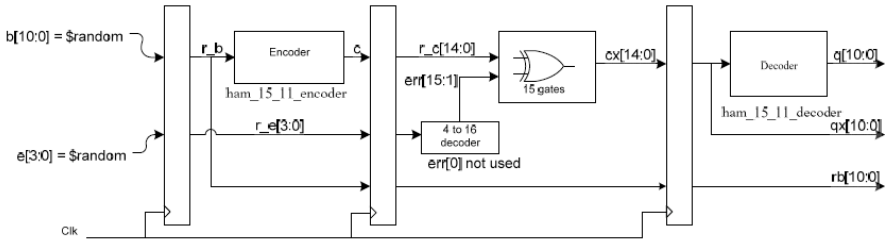

Fig. 3. Schematic modeling a communications channel.

4. DEVELOPMENT OF THE TESTBENCH THAT MODELS A COMMUNICATIONS CHANNEL

The testbench for a communications channel with three pipeline stages was developed; see Figure 3. The first stage contains the Hamming encoder, while the last stage contains the error-correcting Hamming decoder and a parity-stripper module. The parity-stripper module extracts the parity bits from the encoded data bits, which may be corrupted, transmitted through the communications channel. This results in producing the data bits (qx[10:0]), which may be corrupted, at its output.

The output of the parity-stripper module (qx[10:0]) is compared with that of the error-correcting Hamming decoder (q[10:0]), and the original data bits (rb[10:0]) that has been passed along in the communications channel. The output of the error-correcting Hamming decoder (q[10:0]) should match the original data bits (b[10:0]). If the encoded data bits are corrupted in the communications channel with a single-bit error, the error-correcting Hamming decoder should have detected it, and corrected it. Else, if more than one-bit errors are detected, errors will go uncorrected. Finally, comparing qx[10:0] with q[10:0] and rb[10:0] allows us to determine which data bits are corrupted with errors in the communications channel, and if such errors are corrected.

The second stage consist of a 4-to-16 bit decoder and a 15-bit XOR gate to model. The 4-to-16 bit decoder is used to create a 16-bit random number (err[15:1]) from a 4-bit random number (e[3:0]) that is used to represent the error signal, or noise, that is used to corrupt data bits in the communications channel. The 15-bit XOR gate is used to corrupt the encoded data bits (r_c[14:0]) with errors (err[15:1]) in the second stage of the communications channel. This models how data can get corrupted during transmission in a communications channel.

Figure 4 shows the simulation waveforms for the behavioral RTL for the 4-to-16 bit decoder. Lastly, Figure 5 and Figure 6 indicate the simulation waveforms for the pipelined communications channel. The former shows all the signals in the three-stage pipeline, while the latter shows only the input and output signals of that pipeline. It is noted that the simulation of behavioral RTL models do not use information pertinent to the interconnect and device characteristics, and parasitics to capture the delay through the digital logic. That is, these simulations are performed on ideal logic components, and cannot capture delays affecting the logic of the design.

5. LOGIC SYNTHESIS OF HAMMING ENCODER AND DECODER

The area and timing reports from performing logic synthesis on the Hamming encoder and decoder are found in the attachment. The total area for the Hamming decoder is 3041 units^2, while its delay through its critical path(s) is 2.60 units. Its total dynamic power consumption is 6.0982 mW, while its cell leakage power is 6.6074 nW.

Hamming encoder has a total area of 1008 units^2, while its delay through its critical path(s) is 0.67 units. Its total dynamic power consumption is 2.0195 mW, while its cell leakage power is 2.9028 nW.

6. SIMULATION OF SYNTHESIZED HAMMING ENCODER AND DECODER

Figure 7 shows the simulation waveforms for the synthesized Hamming encoder, while Figure 8 shows the simulation waveforms for the synthesized error-correcting Hamming encoder.

The author notes that the simulation of the synthesized Verilog netlists uses information pertinent to the interconnect and device characteristics, and parasitics to capture the delay through the digital logic. These information are provided from the SDF files, and the technology library used for logic synthesis. That, these simulations are performed on realistic logic components, and can capture delays affecting the logic of the design. For example, the correct values for the encoder appear after a significant time delay since the encoder and the decoder has different paths with various delays. Hence, the values from the input may propagate to the output on certain paths faster than other paths. Consequently, the output values are incorrect until the values have propagated from the inputs to the outputs on the critical path.

7. CONCLUSION

Synthesizable behavorial RTL models for the encoder and decoder of the (15,11,3) Hamming code were developed in Verilog. They were verified using circuit simulation. Subsequently, the encoder and the decoder underwent logic synthesis, and were verified via simulation again.

REFERENCES

Weste, N. H. E. and Harris, D. 2005. *CMOS VLSI Design: A Circuits and Systems Perspective*, Third ed. Pearson Education, Boston, MA.

Submitted November 19, 2007, 1200 hrs

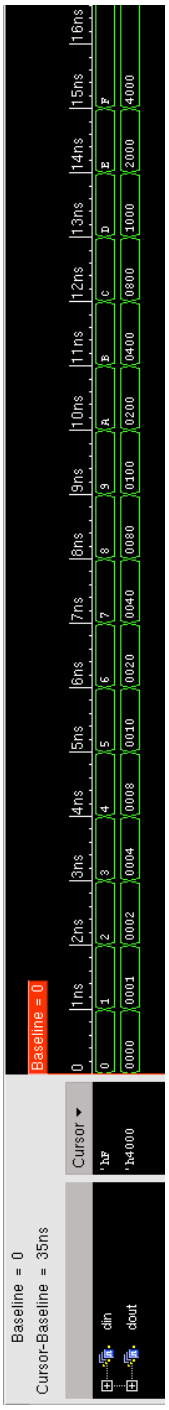

Fig. 4. Simulation Waveforms for the Behavioral RTL for the 4-to-16 bit Decoder.

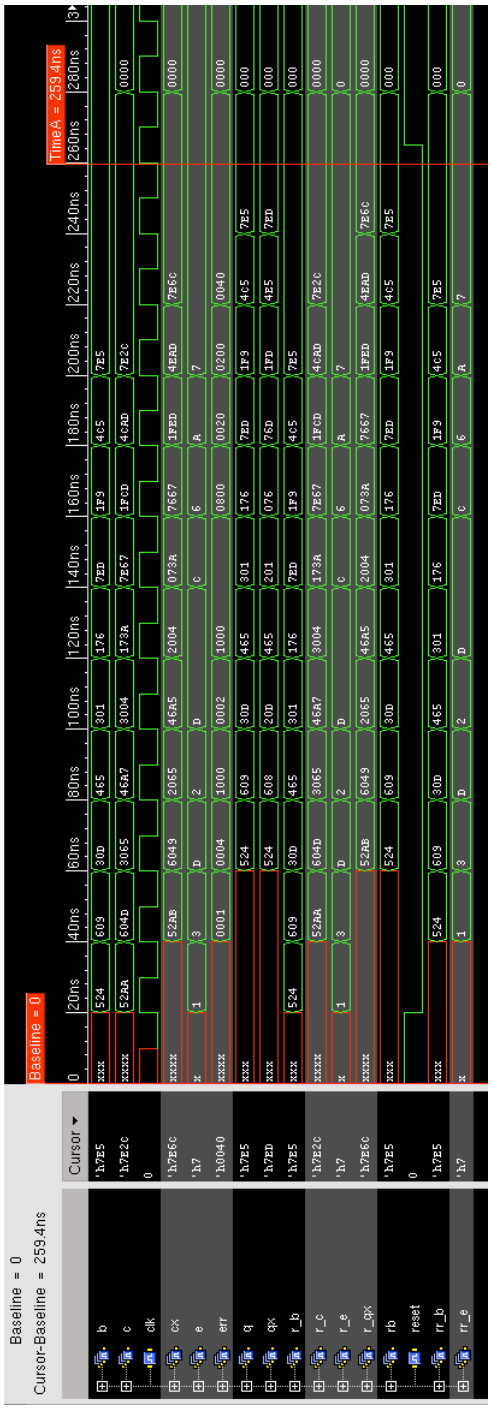

Fig. 5. Simulation Waveforms for the Communications Channel.

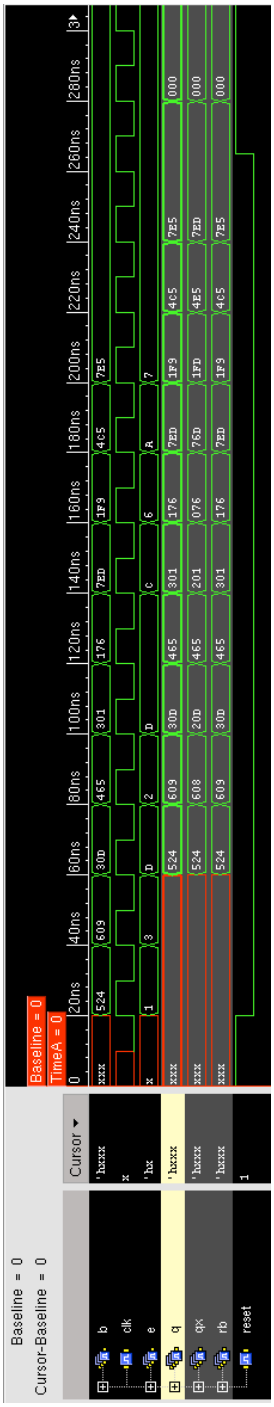

Fig. 6. Zoomed-in Display of Input and Output Simulation Waveforms for the Communications Channel.

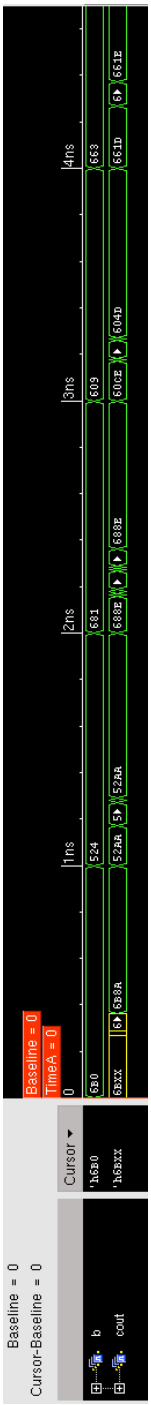

Fig. 7. Simulation Waveforms for the Synthesized Hamming Encoder.

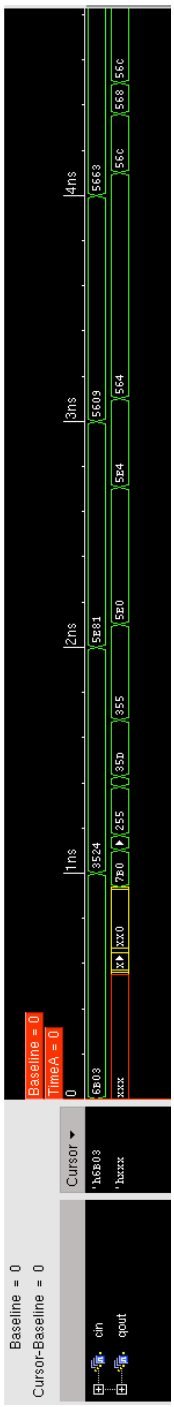

Fig. 8. Simulation Waveforms for the Synthesized Error-Correcting Hamming Decoder.