Supplement to
Digital Design: An Embedded Systems Approach Using Verilog
by Peter J. Ashenden

ERRATA FOR FIRST PRINTING

6 December 2008.

You can find which printing you have by looking at the copyright page. Near the bottom is a line of numbers. The first printing has

07 08 09 10 5 4 3 2 1

The leftmost two-digit number is the year of printing. The rightmost number is the printing. If you have a later printing than the first, you should check the website for errata for your printing. The errata listed here will be corrected in later printings.

ERRATA FOR CHAPTER 2

- In the solution to Example 2.21, in the Verilog code, change two occurrences of "= =" to "==".

ERRATA FOR CHAPTER 3

- Change Exercise 3.21 to read:

We have shown that addition of two n-bit unsigned binary numbers requires $n + 1$ bits for the result to be represented without overflow. Show that addition of three n-bit unsigned binary numbers requires $n + 2$ bits.

- Change Exercise 3.22 to read:

Write a Verilog model of a circuit that adds three 12-bit unsigned binary numbers to produce a 14-bit result with no overflow detection.

- Change Exercise 3.61 to read:

Write a Verilog module definition for a component that calculates the square of a signed fixed-point number with 4 pre-binary-point and 6 post-binary-point bits. The result is unsigned, with 8 pre-binary-point and 6 post-binary-point bits.

ERRATA FOR CHAPTER 4

- In the solution for Example 4.20, change the line of Verilog code that reads

```
assign clk_100Hz = count500000 == 499999;
```

to

```
assign clk_100Hz = count500000 == 0;
```

ERRATA FOR CHAPTER 5

- In Figure 5.16, change the label "ren2" to "en2".
- Add the following sentence to the end of Exercise 5.20:

Assume that the two port share a common clock signal.

ERRATA FOR CHAPTER 6

- In the caption for Figure 6.11, change "GAL220V10" to "GAL22V10".

ERRATA FOR CHAPTER 7

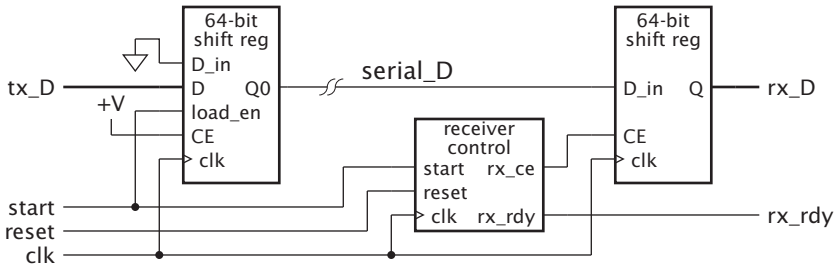
- In the solution for Example 7.10, code is missing from the always block for the data memory to set data_ack_i to 0. The corrected always block is:

```
always @(posedge clk)  // Data memory
  if (data_cyc_o && data_stb_o)
    if (data_we_o) begin
      data_RAM[data_adr_o] <= data_dat_o;
      data_dat_i           <= data_dat_o;
      data_ack_i           <= 1'b1;
    end
    else begin
      data_dat_i <= data_RAM[data_adr_o];
      data_ack_i <= 1'b1;
    end
```

```
else
  data_ack_i <= 1'b0;
```

ERRATA FOR CHAPTER 8

➤ In Figure 8.32, the connection dot is missing on the clk signal for the connection to the shift register at the left of the diagram. The diagram should be corrected as follows:



ERRATA FOR CHAPTER 9

➤ In Example 9.14, make the following changes to the testbench code:

In the task bus_write, change the blocking assignments to non-blocking assignments as follows:

```
cpu_adr_o <= adr;
cpu_sel_o <= 4'b1111;
cpu_dat_o <= dat;
cpu_cyc_o <= 1'b1; cpu_stb_o <= 1'b1; cpu_we_o <= 1'b1;
```

In the processor bus-functional model, change the initial blocking assignments to non-blocking assignments as follows:

```
cpu_adr_o <= 23'h000000;
cpu_sel_o <= 4'b0000;
cpu_dat_o <= 32'h00000000;
cpu_cyc_o <= 1'b0; cpu_stb_o <= 1'b0; cpu_we_o <= 1'b0;
```

In the processor bus-functional model, change the blocking assignments that read the status register to non-blocking assignments as follows:

```
// Read status register
cpu_adr_o <= sobel_reg_base + sobel_status_reg_offset;
cpu_sel_o <= 4'b1111;
cpu_cyc_o <= 1'b1; cpu_stb_o <= 1'b1; cpu_we_o <= 1'b0;
```

```
 @(posedge clk); while (!cpu_ack_i) @(posedge clk);
cpu_cyc_o <= 1'b0; cpu_stb_o <= 1'b0; cpu_we_o <= 1'b0;
 if (cpu_dat_i[0]) disable loop;
```

In the memory bus-functional model, change the blocking assignments to non-blocking assignments as follows:

```
mem_ack_o <= 1'b0;
mem_dat_o <= 32'h00000000;
@(posedge clk);
while (!(bus_cyc && mem_stb_i)) @(posedge clk);
if (!bus_we)
  mem_dat_o <= 32'h00000000; // in place of read data
mem_ack_o <= 1'b1;
```

➤ Change the last sentence of Exercise 9.9 to:

The software should maintain four images in memory: one being acquired from the camera, one being read by the accelerator, one being written by the accelerator, and one undergoing post-detection analysis.

ERRATA FOR CHAPTER 10

➤ In Figure 10.1, add the text “H/W Integration” to the blank grey box.