Computer Design Supervision Problems

Kevin Zhou

October 14, 2015

1 Logic Minimisation

Simplify the following expression

\[ F(A, B, C, D) = (A \cdot B) \oplus (C + D) \]  

where \( \oplus \) is the XOR operation.

(a) Derive the truth table of the expression

(b) Use the Karnaugh-map to minimize the expression.

(c) Use the Quine McCluskey method to minimize the expression

2 System Verilog

(a) Given a positive reset signal, how is an asynchronous reset described in SystemVerilog?

(b) In SystemVerilog, what is the purpose of an “initial” block and how is timing specified?

(c) What is the difference between production test and functional test?

(d) For each of the following six \texttt{always} blocks, what sequence or error will be produced? 
   You should assume that registers are reset to zero at the start (as they are for FPGAs) and that clk is a clock.

```verilog
reg [2:0] counterA, counterB, counterC, counterD, counterE, counterF;

always @(posedge clk) begin
  counterA <= counterA+1;
  if(counterA==5) counterA <= 1;
end

always @(posedge clk) begin
  if(counterB==5) counterB <= 1;
  counterB <= counterB+1;
end

always @(posedge clk) begin
  if(counterC==5) counterC = 1;
end
```
counterC = counterC+1;
end

always @(*) counterD <= counterE+1;

always @(posedge clk)
counterE <= (counterD==5) ? 1 : counterD;

always @(*)
begin
    if (counterF==5) counterF <= 1;
    counterF <= counterF+1;
end

(e) A novice SystemVerilog programmer has written the following decimal counter module which should zero the decimal_count on reset and then, when enabled, increment modulo 10 the decimal_count on every positive clock edge.

module count_decimal_wrong(
    input logic clk;
    input logic reset;
    input logic enable;
    output logic decimal_count);

always_comb @(posedge clk or reset)
if(enable)
begin
    decimal_count = decimal_count+1;
    if(decimal_count>9)
        decimal_count = 0;
end
elsif(reset)
    decimal_count = 0;
endmodule // count_decimal_wrong

What bugs exist in the code and how can they be rectified?

3 Computer Architecture Fundamentals

Search from the Internet, suggested text books or from other sources, answer the following question:

(a) What does ISA in (computer architecture) stand for?

(b) What are CISC and RISC architectures? What are the advantages and disadvantages of both groups?

(c) List five different ISAs you found and typical CPUs that were built on your mentioned ISA.

(d) (i) Briefly describe the CPU that was installed in your mobile phone.
(ii) How many transistors and what CMOS manufacturing process it is built on (if known)?

(iii) Why your mobile phone could not run binary executables that were compiled for desktop computers?

(iv) Do you think the CPU in your phone performs better than the old desktop PC computers (For example: Intel Pentium 4/Single Core/2.0GHz/130nm manufactured in 2000) and why do you think so?

(v) Why does reducing the size of transistors in the CPU lead to improved performance and power efficiency?

(vi) If you were a testing engineer in the company that manufactured the chip of your mobile phone, what would you do if you were told to compare the performances between your mobile and old desktop computers?

(vii) What are the major merits that CPU vendors would rely on when they are trying to sell their CPU products to the markets?

End of Problems