SoC D/M Exercise Sheet(s), 2016/2017

This sheet contains exercises of various lengths, but the very short ones are in a separate PDF file (called something like socdamquick.pdf). Many exercises are nominally allocated marks at Tripos examination level (i.e. with 20 marks making a full exam question).

There is some repetition of material between the exercises, so a suitable target is to solve approximately one third of them. Exercises marked with a ♡ form a recommended core. Some sections contain additional preference instructions.

Supervisors are recommended to mainly just set the exercises marked with the ♡ symbol, leaving other parts of the sheets for discussion or for self-study revision purposes. Example answers are available to supervisors.

The course has been rejigged a few times in the last ten years and, despite my attempts, it is likely that one or two of the questions here are now no-longer-relevant or else under the wrong heading. SystemC is no longer being lectured for net-level modelling: it is just presented as a basis for TLM modelling, so where a question suggests giving a SystemC descriptons instead of RTL, that route will not be available unless you have separately studied low-level SystemC coding.

SP3: Architectual and Tools

Architectural Exploration and Design Partition questions:

ARCH.1. ♡ : Design Partition

a) What are the major costs and risks in SoC development ? [5 Marks]

b) What factors commonly influence the choice between using standard parts and an ASIC or SoC ? [5 Marks]

c) What factors tend to make a hardware implementation preferable to a software implementation? Give an example of each approach. [5 Marks]

d) When is a standard processor preferable to a custom processor ? [5 Marks]

ARCH.2. FPGA:

a) What are the principal differences between an FPGA and a masked ASIC for implementation of a SoC ? [5 Marks]

b) How can a SoC design team use FPGAs to prototype their product before SoC fabrication ? [5 Marks]

c) When would it be sensible to ship an FPGA instead of a masked ASIC in production runs ? [5 Marks]

ARCH.3. Cost and Power

a) Summarise the historical trends that affect the relative merits of FPGA and custom silicon in consumer, professional and military, mains-powered applications [5 Marks].

b) How does the argument differ for battery-powered devices ? [5 Marks]

c) What structure or behaviour consumes power in each of FPGA, embedded processors and custom silicon ? [5 Marks]

d) Discuss whether multi-core processor chips can/should take over from FPGA and custom silicon in various applications. Consider Picochip, Zynq, and XMOS if you are familiar with them. [5 Marks]

SP4: RTL, Simulation, Hazards, Folding

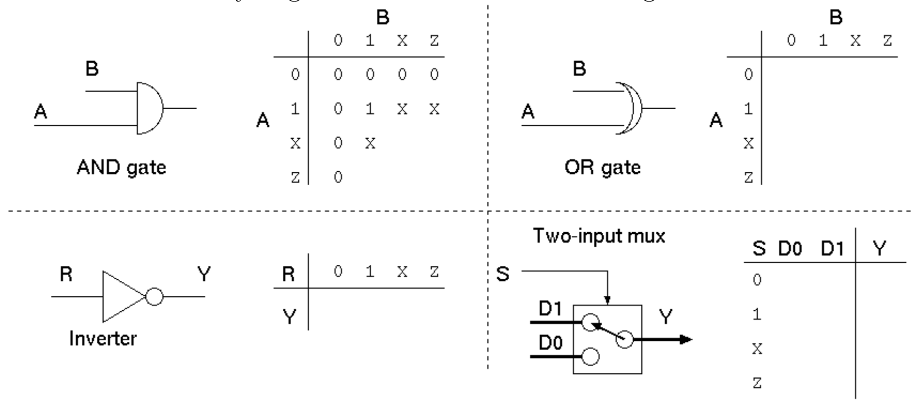
4.1 SP4: RTL, Simulation, Hazards, Folding - Full Questions

RTL1. Synthesisable RTL standards require that a variable is updated by at most one thread: give an example of a variable being updated in two `always` blocks and an equivalent combined always block or circuit that is valid/correct. [8 Marks]

RTL2. Give a schematic (circuit) diagram for the design from the quick sheet [1] that checks whether running sum of the five-bit input exceeds 511. Use adders and/or ALU blocks rather than giving full circuits for an such components. [7 Marks]

RTL3. Complete the truth tables for the 2-input AND gate (using symmetry) and the other three functions shown in this grid where the inputs range over { 0, 1, X, Z }. (X denotes don't know and Z denotes high-impedance.)

For supervision discussion: Why might we want more than four logic values in a simulation?



A \ B	0	1	X	Z
0	0	0	0	0
1	0	1	X	X
X	0	X		
Z	0			

A \ B	0	1	X	Z
0				
1				
X				
Z				

R	0	1	X	Z
Y				

S D0 D1	Y
0	
1	
X	
Z	

RTL4. Draw the gate-level circuit of an RS-latch using a pair of cross-coupled NOR gates described by the following pair of continuous assignments. Describe the complete sequence of events that occur when the the latch is initially clear and is set with a positive pulse of 5 nanoseconds between times 10 and 15. Assume the gate delay is 200 picoseconds. Your answer will essentially be a list of net/time/value triples but you also need to say when these are added and removed from the event queue. [8 Marks].

```
assign q = !(clear || qb);
assign qb = !(set || q);
```

RTL5. ♡ Identify the structural hazards in the following fragment of Verilog. What holding registers are simplistically needed if the main array (called daza) is a single-ported RAM? What if it is dual ported ? Can all the hazards be removed be recoding the algorithm ?

```
module FIBON(clk, reset);
   input clk, reset;
   reg [15:0] daza [32767:0];
   integer pos;
   reg [3:0] state;
   always @(posedge clk) begin
      if (reset) begin
         state <= 0;
         pos <= 2;
         end
      else case (state)
             0: begin
                daza[0] <= 1;
                daza[1] = daza[0];
                state <= 1;
             end
             1: begin
                daza[pos] <= daza[pos-1]+daza[pos-2];
                if (pos == 32767) state <= 2; else pos <= pos + 1;
             end
           endcase // case (state)
      end
endmodule
```

[1] Give an RTL design for a component that accepts a five-bit input, a clock and a reset and gives a single-bit output that holds when the running sum of the five-bit input exceeds 511.

RTL6. Summarise the main differences between synthesisable RTL and general multi-threaded software in terms of programming style and paradigms. [10 Marks].

RTL7. In Verilog-like RTL, write out the complete design, including sequencer, for the datapath and controlling sequencer for the Booth long multiplier following the style of the long multiplier given in the lecture notes.

```
// Call this function with c=0 and carry=0 to multiply x by y.
fun booth(x, y, c, carry) =
    if(x=0 andalso carry=0) then c else
let val x' = x div 4
    val y' = y * 4
    val n  = (x mod 4) + carry
    val (carry', c') = case (n) of
      (0) => (0, c)
     |(1) => (0, c+y)
     |(2) => (0, c+2*y)
     |(3) => (1, c-y)
     |(4) => (1, c)
    in booth(x', y', c', carry')
    end
```

It should start as follows:

```
module LONGMULT8b8(clk, reset, C, Ready, A, B, Start);
   input clk, reset, Start;
   output Ready;
   input [7:0] A, B;
   output [15:0] C;
```

(Details of language syntax are unimportant) [15 Marks]

RTL8. Optional exercise: (no further examinable ground covered): Repeat the previous exercise for long division (using either the shift-left till greater then shift right method, the fixed-latency algorith, or using Goldschmidt's method (repeatedly multiply n and d by $1 - 2d$)). [20 Marks]

-*Answer:* -TODO: Answer missing.

RTL9. Modify (fold in space) the following RTL code so that it uses half-as-many ALUs and twice-as-many clock cycles to achieve the same functionality as the following component:

```
module TOFOLD(clk, reset, start, pp, qq, gg, yy, ready);
   input clk, reset, start;
   input [7:0] pp, qq, gg;
   output reg [7:0] yy;
   output reg ready;
   integer state;
   always @(posedge clk) begin
      if (reset) begin
         state <= 0;
         ready <= 0;
         end
      else case (state)
             0: if (start) state <= 1;
             1: begin
                yy <= (pp*gg +  qq*(255-gg)) / 256;
                ready <= 1;
                state <= 2;
             end

             2: if (!start) begin
                ready <= 0;
                state <= 0;
                end
           endcase // case (state)
      end
endmodule
```

[10 Marks]

END OF DOCUMENT.