

# SoC D/M Exercise Sheet, 2015/2016

This sheet contains exercises of various lengths. 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.

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.

## SP1: SoC Components and Bus Structure Exercises

### 0.1 SP1Q: SoC Components and Bus Structure Exercises - Quick Questions

See separate PDF FILE.

### 0.2 SP1: SoC Components and Bus Structure Exercises - Full Questions

**Slide Pack SP1+SP3: This sheet should be tackled for the first supervision. Supervisors are recommended to set just the few exercises marked with the ♡ symbol, leaving the second half of this sheet for brief discussion or for self-study revision purposes.**

This year SystemC is not being lectured in detail, so please leave the SystemC parts of this sheet for now and answer again at the end according to how much was lectured.

SoC-SP1.1 What is meant by polled I/O and how does it compare with interrupt driven I/O ? [4 Marks]

SoC-SP1.2 Swapping bit order in a word requires an expensive sequence of instructions on many processors. However, it can be done instantly in wiring. It is proposed to provide an I/O device with three registers respectively of width 8, 16, 32 bits. Any word written to such a register can then be immediately read back with its bits swapped.

a) Give the programming model for such an I/O device and sketch RTL or the circuit diagram for the actual device. [4 Marks]

b) How should such a device be connected to the processor such that it actually saves time or energy compared with the sequence of instructions approach? Discuss various wiring possibilities and their pros and cons [6 Marks].

For your reference, here is some C code (from the ARM ISS), one quarter of this meets one of third of our specification:

```
switch(size_code)
{
  case 0: //Reverse byte order in a word.
    d = ((d>>24) & 0xFF) | ((d>>8) & 0xFF00) | ((d<<8) & 0xFF0000) | ((d<<24) & 0xFF000000);
    break;
  case 1: // Reverse byte order in each halfword independently.
    d = ((d>>8) & 0x00FF00FF) | ((d<<8) & 0xFF00FF00);
    break;
  case 2:// RBIT - (a full butterfly) reverse the bit order in a 32-bit word.
    d = ((d>>16) & 0x0000FFFF) | ((d<<16) & 0xFFFF0000);
    d = ((d>>8) & 0x00FF00FF) | ((d<<8) & 0xFF00FF00);
    d = ((d>>4) & 0x0F0F0F0F) | ((d<<4) & 0xF0F0F0F0);
    d = ((d>>2) & 0x33333333) | ((d<<2) & 0xCCCCCCCC);
    d = ((d>>1) & 0x55555555) | ((d<<1) & 0xAAAAAAAA);
    break;
```

```

case 3: //REVSH      Reverse byte order in the bottom halfword, and sign extend to 32 bits.
    d = ((d >> 8) & 0xFF) | ((d<<8) & 0xFF00);
    if (d & 0x8000) d |= 0xFFFF0000;
    break;
}

```

SoC-SP1.3 Sketch a set of typical macro definitions in C suitable for making low-level hardware access to the device of the previous question or else to a UART or similar device that contains status, control and data registers. [4 Marks]

SoC-SP1.4 ♡ Show how to wire up a push button to a GPIO pin and describe the action of (or sketch out the code for) a device driver that sets up the GPIO device and returns how many times it has so far been pressed. Describe firstly polled code. Then write a few sentences about how the interrupt driven solution would be implemented and when it should be used instead. (No credit is allocated for proper debouncing so feel free to neglect that aspect). [10 Marks]

SoC-SP1.5 ♡ A bus arbiter decides which initiator next makes access to the set of shared targets connected to that bus. In simple systems these have fixed behaviour and so do form part of the programming model. Here we consider an arbiter with programmable policy.

a) Sketch the circuit, RTL or SystemC code for a programmable bus arbiter that manages two initiators. Start by specifying the behaviour and then defining a suitable programming model. Then list the net-level connections to the component to draw a schematic symbol before going on to implementation details.

b) Should the programmed I/O access to the bus arbiter itself be subject to arbitration and if this were to be a problem, in what ways could higher-priority access to the arbiter's register file be granted?

c) What are the advantages and disadvantages of having an arbiter programmable? What method might a network-on-chip use to get much richer programmable arbitration?

[<http://www.cl.cam.ac.uk/research/srg/han/ACS-P35/zynq/Zynq-7000-TRM.pdf> If you look at the Zynq 7000 technical reference manual you will see a number of Quality-of-Service blocks instantiated for traffic control. An overview is given in section 5.2.2 They are programmable with maximum number of outstanding transactions, AW and AR channel peak rates, burstiness, average rates.]

SoC-SP1.6 Fully describe (using diagrams and perhaps sketches of RTL or SystemC code) an interrupt controller that stores eight vectors with individual interrupt enable flags. The controller monitors eight interrupt inputs and presents the highest-priority, non-masked interrupt vector to the processor when the processor asserts an interrupt acknowledge signal or otherwise reads the device. [Start by defining a suitable programming model and then list the net-level connections to the component to draw a schematic symbol.] *Fine details will vary from answer to answer. As always, syntactic accuracy is not required in SoC D/M examination answers.* [15 Marks]

SoC-SP1.7 How does the processor set up the interrupt distributor device of SoC-SP1.6 and what must it do after servicing an interrupt ? [4 Marks]

SoC-SP1.8 How would you make an interrupt distributor that shares work over two CPUs? What are the pros and cons of dynamic distribution of interrupts ? [6 Marks]

SoC-SP1.9 Give a programming model for a simple DMA controller with one control/status register and three operand registers for block length and source and destination addresses. *The DMA (direct memory access) controller, when active, becomes a bus master and copies a block of data from one area to another, generating an interrupt on completion. n.b. This is very similar to the dma\_controller.h example in the TOY-ESL online practical classes.* [4 Marks]

b) Sketch a full implementation of such a DMA controller that includes provision for slave access to the programmable registers, active bus mastership and interrupt generation. Memory access should use a high-level modelling style that ignores bus arbitration. *Answer preferably using SystemC syntax, or pseudocode at the same level of abstraction. Use RTL if and where needed or preferred.* [7 Marks]

SoC-SP1.10 Bus Bridge.

a) What is the function of a bus bridge in a SoC ? [2 Marks]

b) What typical and possible address translation semantics might a bus bridge implement ? [4 Marks]

- c) How might internal queue structure vary between bus bridge designs ? [3 Marks]
- d) How might arbitration policy vary between bus bridge designs ? [3 Marks]

SoC-SP1.11 Input and Output to a Network Controller

- a) Sketch the structural schematic symbol for a generic network block that is bus target only, giving full details and descriptions of the signals used to connect to a typical system bus. *The network type or internal structure does not matter, it could be Ethernet, USB, Firewire etc..* [6 Marks]
- b) What advantages are there to giving the network block the capability of being a bus master? [2 Marks]
- c) Describe the additional signals needed to make the network block a bus master. [6 Marks]
- d) Assuming the device can be a bus master, sketch the code for a typical device driver. [6 Marks]

SoC-SP1.12 Define a feasible, net-level, serial interface used between a sound controller device (That does DMA and so on) to an audio output DAC (digital-to-analog convertor). The interface conveys a pair of stereo channels of 16 bit precision at 44.1 ksps. *Hint: Three nets are normally used.* [4 Marks]

SoC-SP1.13 Sketch the block diagram or RTL for a simple audio output controller that uses DMA to send a serial audio data-stream to a DAC. Include the full programmers' model. [12 Marks]

SoC-SP1.14 Clock Domain Crossing.

- a) List basic principles used in the design of a reliable clock-domain crossing bridge to avoid metastability problems and achieve reliable transfer of data ? [6 Marks]
- b) Sketch the RTL or block diagram for a simplex clock crossing bridge that internally uses one parallel data bus and four-phase handshake ? *If giving RTL, only the receiving side logic is needed.* [6 Marks]
- c) What constraints exist for simplex protocols that cross clock domains ? [6 Marks]
- d) What constraints exists for duplex protocols that span clock domains ? [2 Marks]

SoC-SP1.15 Exercise: sketch RTL code for a non-preemptive version of the 3-input arbiter given in the handout. Alternatively, provide RTL code for a round-robin, non-preemptive version of the 3-input arbiter. *An asynchronous implementation is quite tricky unless you are experienced at logic designing with transparent latches and other level-sensitive latches, so feel free to present a synchronous design, which is just a finite-state machine.*

## SP2: Silicon Energy, Power and Technology

### 1.3 SP2: Silicon Energy, Power and Technology - Full Questions

#### Power Estimation and Control questions:

POWER.1. Dynamic Clock Gating.

- a) What is dynamic clock gating and why is it used ? [4 Marks]
- b) Describe some common clock-gate insertion transformations. [6 Marks]
- c) Compare dynamic clock gating with power isolation islands in terms of automation, scale and functionality. [6 Marks]
- d) Consider the pros and cons of coarse-grained manual clock gating logic insertion with the use of an automatic tool/compiler that inserts clock gating. [4 Marks]

POWER.2. ♡ : VLSI Energy Use. For this question, use the following figures and assume values for or look up values of any other information that you feel you need. Credit is awarded for the method and not for the numerical results.

Parameter	Value	Unit
Drawn Gate Length	0.08	$\mu\text{m}$
Metal Layers	6 to 9	layers
Gate Density	400K	gates/ $\text{mm}^2$
Track Width	0.25	$\mu\text{m}$
Track Spacing	0.25	$\mu\text{m}$
Gate Output Capacitance	0.06	fF
Gate Input Capacitance	0.03	fF
Tracking Capacitance	1	fF/mm
Core Supply Voltage	0.9 to 1.4	V
FO4 Delay	51	ps
Leakage current	21	nA/gate

A processor core in the above technology uses 200k gates, excluding cache memories. It has two operating conditions: 100 MHz at 0.9 volts or 400 MHz at 1.4 volts. The average net activity ratio during halt is negligible and 0.3 when running.

*Give all working and intermediate results. State any additional assumptions you need or use.*

- Estimate the area of the processor. [2 Marks]
- Compute the power consumed per gate at each operating condition when driving a tracks of 0 mm and 1 mm. [2 Marks]
- Estimate the power consumption of the processor core when halted and running for each operating condition. [3 Marks]
- Compared with having the processor running at full performance all the time, how much power is saved just by halting the processor when it is idle ? [2 Marks]
- How much power is saved by dynamic frequency scaling ? [2 Marks]
- How does dynamic frequency scaling compare with halting ? [2 Marks]
- How much power is saved by combined dynamic voltage and frequency scaling ? [2 Marks]
- How much power might be saved by power gating (i.e. power isolation) ? [2 Marks]
- Estimate the relative costs of performing a 32 bit addition and sending the 32 bit result 1 mm over the chip [3 Marks]

### POWER.3. : Dynamic Voltage and Frequency Scaling.

- Give a formula for the power dissipation associated with a net on a silicon chip. [3 Marks]
- What is meant by dynamic clock gating and compare this to a technique where software writes to a control register to turn off a clock generator ? [3 Marks]
- For a fixed supply voltage, quantify the power benefits of frequency scaling. In other words, compare computing quickly and halting with computing more-slowly and finishing just in time. [3 Marks]
- Give two ways that the supply voltage to a region may be varied? [3 Marks]
- Using variable supply voltages, quantify the power benefits of frequency scaling. [3 Marks]
- In supervisions, discuss the architecture of an ASIC (or part of) that uses all of these techniques. [5 Marks]

### POWER.4. : Power Consumption

*This question is primarily for discussion in supervisions.*

- What are the main components of power consumption in a laptop computer? [5 Marks]
- How does clock frequency affect power consumption ? [5 Marks]
- How might clock frequency be controller in a laptop and for what reasons ? [5 Marks]
- When viewing a DVD (including moving video and audio) on a laptop, what is the best clock frequency policy? [5 Marks]

POWER.5. Technology/Scaling.

*This question is primarily for discussion in supervisions.*

- a) What is meant by the term *feature size* in VLSI ? Give typical values. [5 Marks]
- b) What are the main consequences of moving to a smaller feature size in VLSI fabrication ? [5 Marks]
- c) What happens to the relative costs of computation and communication as features get smaller ? [5 Marks]
- d) Why has parallel computation become more important than ever before ? [5 Marks]

POWER.6. ♡ Battery Life Planning

- a) Data can be compressed before being sent by a hand-held unit over a radio link or later on at a central server. What are the advantages and disadvantages? [4 Marks]
- b) How might you estimate the battery life of a sensor device that has a 3 Watt radio transmitter, no LEDs or displays and needs to run about five minutes worth of CPU cycles per day? Would solar cells serve it well? [4 Marks]
- c) In the last decade or so, digital sensors with radio (wireless) backlinks have been installed in all rooms in all public buildings. What have been the main forward steps in technology that have facilitated this? Does the price of copper make a difference? [3 Marks]

## SP3: SoC Busses, Partition and Technology

### 2.4 SP3Q: Design Partition - Quick Questions

See separate PDF FILE.

### 2.5 SP3: SoC Busses, Partition and Technology - Full Questions

!

**Slide Pack SP2: This sheet should be tackled for the second supervision. Supervisors are recommended to set just the few exercises marked with ♡ with the remainder of the questions being available for brief discussion or self-study revision purposes.**

This year SystemC is being lectured at the end, so please leave the SystemC parts of this sheet for now and answer again after that has been lectured.

SOC-SP3.1. ♡ : SoC Structure.

- a) Sketch the block diagram for a SoC with one processor, one SRAM, one ROM, one Counter/Timer block and one PIO section, all connected to a single bus without any bus bridges. [5 Marks]
- b) List the (main) net-level interface signals needed for a bus port that enables multiple bus operations to be in flight at once (such as the BVCi port lectured, or an IP block interface of similar functionality) and explain the protocol. [6 Marks]
- c) Is it appropriate for DMA to be supported or used in the SoC of part a) ? [3 Marks]
- d) How are interrupt signals routed in the SoC of part a) ? [3 Marks]
- e) What modifications are needed if a second processor core were to be added ? Is a second bus a good idea ? [3 Marks]

SOC-SP3.2. : Multiple Busses With Bridges.

- a) In SoC terms, what is a bus and how does it compare with the 1980's concept of a motherboard bus (such as the ISA or PCI bus) ? [2 Marks]

- b) How might the destination port for a transaction over such a bus be decided ? [2 Marks]
- c) What is a bus bridge, what transactions might it support and what internal operations might it implement ? [4 Marks]
- d) If a SoC is designed with a number of bridged busses, what are the main aspects that determine the allocation of initiators and targets to the busses ? [3 Marks]
- e) Is there no real difference between a Network On Chip and a set of bus bridges ? [3 Marks]
- f) What form of bus protocol is needed for good performance on a SoC that uses a number of bridges busses or clock domains ? [3 Marks]
- g) How is contention for destinations handled in a SoC that uses a number of bridges busses compared with a NoC (network on chip) ? [3 Marks]

SOC-SP3.3. : Network-On-Chip (NoC).

- a) What is meant by the term Network-on-Chip and what are the main differences between using a number of bus bridges and a network fabric? [5 Marks]
- b) Describe two buffering techniques that might be used in a NoC ? [2 Marks]
- c) Describe a flow control technique used in a NoC ? [2 Marks]

SOC-SP3.4. : DRAM and Cache.

- a) What are the main features of DRAM and why was it not commonly integrated as part of a SoC ? [5 Marks]
- b) Why should out-of-order read responses ideally be supported by a SoC Bus or NoC ? [5 Marks]
- c) Using a system clock of 400 MHz, a 32 bit MIPS/ARM-like CPU is served without a cache by a 16-bit DRAM system with the following parameters

Operation	Clock cycles	Function
RAS	3	Sending row address,
CAS	1	Read or write 16 bits in current row,
RAS precharge	2	Write back time when finished with row.

Making some assumptions about the pattern of access that the processor will make of the memory, estimate its performance in terms of instructions per second. [5 Marks]

- d) If all instructions for inner loops are copied to a 32-bit wide on-chip SRAM (that provides true random access at 400 MHz) at code start, what is the performance now. [5 Marks]
- e) If a cache structure with 98 percent instruction and 80 percent data hit rate is applied, what processor performance is now achieved ? You may consider in-order and out-of-order processors but full credit awarded for either. [5 Marks]

SOC-SP3.5. : JTAG Port and Test Modes. *(In lectures JTAG was only briefly mentioned in relationship to debugging and GDB remote serial protocol (RSP)). This question is suitable for discussion in supervisions but the material was not lectured.*

- a) Why do ASICs commonly support special test modes? [4 Marks]
- b) Define and compare boundary scan with full scan test path [4 Marks]
- c) Briefly describe the structure and operation of the JTAG test port used on many chips. [4 Marks]
- d) How can JTAG ports be combined and is this a good idea within a single SoC ? [4 Marks]
- e) What other uses can the JTAG port frequently be put to ? [4 Marks]

SOC-SP3.6. : Cell Library.

- a) Give a short list of logic cells to be found in a standard cell library. [5 Marks]
- b) List five types of information that should be stored about each cell. [5 Marks]
- c) Name several illustrative, specialist VLSI structures or components that cannot readily be made out of standard logic cells and explain why custom design is needed. [5 Marks]

## SP4: RTL, Simulation, Hazards, Folding

### 3.6 SP4: RTL, Simulation, Hazards, Folding - Short Exercises

- RTLQ1. Give a brief definition of RTL and Synthesisable RTL. Name two example languages. [4 Marks]
- RTLQ2. Explain Verilog's blocking and non-blocking assignment statements. Show how to exchange the contents of two registers using non-blocking assignment. Show the same using blocking assignment. [6 Marks]
- RTLQ3. ♡ Convert the following behavioural RTL into an unordered list of (pure) RTL non-blocking assignments: [5 Marks]

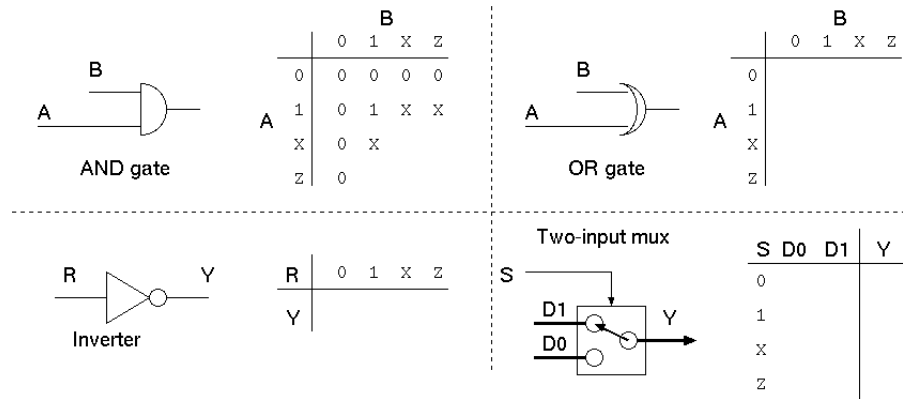
```
always @(posedge clk) begin
    foo = bar + 22;
    if (foo > 17) foo = 17;
    foo_final = foo;
    foo = 0;
end
```

- RTLQ4. Explain the terms 'structural hazard' and 'non-fully pipelined'. [4 Marks]
- RTLQ5. Give a fragment of RTL that implements a counter that wraps after seven clock ticks. [3 Marks]
- RTLQ6. ♡ Give a fragment of RTL that uses two multiply operators but where only one multiplier is needed in the generated hardware. Sketch the output circuit generated by a simplistic Verilog compiler and the optimised circuit that only uses one multiplier. [4 Marks]
- RTLQ7. 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. [6 Marks]
- RTLQ8. Show an example piece of synchronous RTL before and after inserting an additional pipeline stage. [4 Marks]
- RTLQ9. Convert the following behavioural RTL into an unordered list of (pure) RTL non-blocking assignments: [8 Marks]

```
always begin
    @(posedge clk) foo = 2;
    @(posedge clk) foo = 3;
    @(posedge clk) foo = 4;
end
```

### 3.7 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 of RTLQ7. 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 }. (This was not lectured in 2012. X denotes don't know and Z denotes high-impedance.)



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 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 by 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
```

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. (later we will do it in SystemC RTL-style and SystemC TLM-style).

```
// 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 for long division (using either the shift-left till greater then shift right method 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]

## SP5: ABD (Assertion-Based Design).

Please look mainly at those exercises marked with ♡.

ABD1. ♡ : Assertion-based design.

- What is the difference between a safety and liveness assertion over the behaviour of a system. [4 Marks]
- How does a declarative safety assertion differ from an imperative assert statement ? [4 Marks]
- How can safety and liveness assertions be used in dynamic validation ? [5 Marks]
- Give a short segment of RTL or pseudocode that contains an imperative assertion that holds and give also a pair of valid safety and liveness assertions that hold for your code. [7 Marks]

ABD2. ♡ : Black & White Box Testing

- Define the terms back box testing and white box testing [4 Marks].

Suppose a controller module has the following connections:

```
input clock;
input sensor_A;
input sensor_B;
output actuator_C;
output actuator_D;
```

- b) Give an example safety assertion that can potentially be used both with black and white-box testing. [4 Marks]
- c) Can your assertion be dynamically validated under black-box testing. [4 Marks]
- d) Can your assertion be formally proved under black-box testing. [4 Marks]
- e) Give an example liveness assertion that can potentially be used both with black and white-box testing. [4 Marks]
- f) Can your second assertion be dynamically validated under black-box testing. [4 Marks]
- g) Assuming a digital logic implementation of the controller, can your second assertion be formally proved under white-box testing. [4 Marks]

ABD3. : General ABD.

- a) What are the benefits of the assertion-based design (ABD) methodology ? [5 Marks]
  - b) Illustrate how a regular expression can be used as part of a safety assertion ? [5 Marks]
  - c) Using three or more *modelling layers*, describe the PSL reference model. [5 Marks]
- In PSL next-cycle suffix implication uses  $|=>$  and same-cycle suffix implication uses  $|->$ .
- d) Use these two different forms to give a pair of PSL expressions that have identical meaning. [6 Marks]
- See [http://www.esperan.com/tutorial/psl\\_simple.html](http://www.esperan.com/tutorial/psl_simple.html)

ABD4. Four-phase handshake.

- a) Give a temporal logic expression that defines a four-phase handshake using PSL or a PSL-like language. [12 Marks]
- b) Give the synthesisable RTL, SystemC or circuit for a monitor that checks operation of a four-phase handshake. You may assume a high-frequency clock is available that does not alias any transitions. (*Hint: an answer to this is, this-year, in the toyclasses folder.*) [6 Marks]
- c) What is automated stimulus generation and consider whether it be practically applied to interfaces such as the four-phase H/S? [3+3 Marks]

ABD5. ♡ : Protocol, Interface and Bus Monitors.

- a) What is meant by the terms 'port' and 'interface' ? [4 Marks]
- b) What is meant by the formal specification of a protocol and what is a bus monitor ? [5 Marks]
- c) How are bus monitors used in ABD and what sort of error might be detected (safety of liveness etc.) ? [5 Marks]
- d) How can a bus monitor be used to generate simulation stimulus ? What coverage might be possible ? [5 Marks]
- e) What statistics might a bus monitor collect ? [5 Marks]

ABD6. : PSL Operators and Algorithm.

- a) Why is it recommended to always use a PSL SERES as part of a suffix implication ? [5 Marks]
- b) Describe five infix operators defined in PSL. [5 Marks]
- c) Outline an algorithm for synthesising a pattern detecting automaton from the main operators in a PSL SERES (regular expression). (*This was not lectured in 09/10 but is briefly included in the additional material. In 10/11 some ML fragments were flashed up. Candidates ought to be able to do this from Ia RL&FSA material: it's a core competence.*) [5 Marks]

ABD7. : ABD Methodology.

- a) What is meant by 'Assertion Based Design' ? [5 Marks]
- b) Compare the use of assertions and yes/no test wrappers in regression testing ? [5 Marks]
- c) Explain how certain assertions can be re-used at different layers of modelling abstraction (and others not). For example, some might be used for TLM modelling as well as for pre-synthesis and post-synthesis forms of an RTL design. [5 Marks]
- d) What is meant in testing by the term 'coverage' and can this be applied to set of assertions ? [5 Marks]

ABD8. : Sequential Equivalence Checker (SEC).

- a) What is the combinational equivalence problem ? What is the role of don't cares in it ? [5 Marks]
- b) What is meant by sequential equivalence and strong and weak/stuttering bi-simulation ? [5 Marks]
- c) Why might sequential equivalence be violated in a design flow (i.e. SEC gives a negative result)? [5 Marks]
- d) Why might we see false negatives from a SEC ? [5 Marks]

ABD9. : Symbolic versus Concrete State

- a) Putting aside internal implementation details of a checking tool, what is the difference between the sort of assertion that can be expressed in a symbolic specification language compared with one that only deals with unparameterised (atomic) events and boolean values? [2 Marks]
- b) What is the essence of a 'trick' that engineers have used to implement path assertions where the tools they have (or the parts of the tool they are familiar with) only deal boolean state assertions ? [2 Marks]
- c) What is the essence of another 'trick' that engineers will use to implement symbolic assertions where the tools they have only deal with boolean states [2 Marks]

## SP6: ESL (Electronic System Level) and TLM Exercises

### (SP6a: SystemC Overview - Full Questions)

**Note: In 2015/16 - SystemC was almost entirely removed from the course. It was used in the concrete examples of TLM ESL Modelling but tutorial content on SystemC was cut down to just two or three slides. So many of the following (Sp6a) questions are not appropriate this year.**

SYSC1. Describe the principle features of SystemC. [5 Marks]

SYSC2. With what user syntax and how internally is an RTL-style non-blocking assignment achieved in SystemC ? [5 Marks]

SYSC3. How is design module heirarchy expressed in SystemC ?

- SYSC4. ♡ Why adapt a general-purpose language like C++ for hardware use when special hardware languages exist ? [2 Marks]
- SYSC5. ♡ To what level of detail can a gate-level design be modelled using SystemC, would one ever want to do this and what simulation performance might be achieved ? [5 Marks]
- SYSC6. Give a fragment of SystemC or RTL that relies on its kernel scheduler to correctly implement non-blocking updates (avoiding shoot-through) and then give an equivalent fragment of pure C that has the same behaviour but which does not need support from a scheduler or other library. [10 Marks]
- SYSC7. How does SystemC help model registers that have widths not native to the C language ? [4 Marks]
- SYSC8. Give synthesisable SystemC for a five-bit synchronous counter that counts up or down dependent on an input signal. *You should sketch C++ code that looks roughly like RTL rather than worrying about a precise definition of synthesisable for SystemC.* [5 Marks]
- SYSC9. Define suitable nets for a simplex interface that transfers packets of 3 bytes over an asynchronous eight-bit bus with a protocol that is based on the four phase-handshake. Describe the protocol. *Answer this part using RTL, timing diagrams or natural language.* [5 Marks]
- SYSC10. Sketch SystemC RTL-like code for a synthetic data generator that creates three byte packets and delivers them over the four-phase interface of SYSC9.. *Precise syntax and operational details are unimportant, but a sensible answer would be a Verilog module that puts a counting sequence in the packet payloads.* [5 Marks]

*Practical Exercise:* Using the RTL-style blocks provided in the ‘toy classes’ with the nominal processor, please experiment with various configurations and understand how you would make a more-complex system using more of the `addr_decode` and `busmux` components to address the components.

```
ssh linux.pwf.cam.ac.uk: /ux/clteach/SOCDAM/thisyears-toyclasses
```

## 5.8 SP6b: ESL (Electronic System Level) and TLM Exercises - Full Questions

Please look mainly at those exercises marked with ♡ .

- ESL1. ♡ Briefly explain how and why an ESL model that uses a TLM model of its busses can run the embedded software with no modification to its device drivers. [4 Marks]
- ESL2. ♡ Explain how the device driver for an on-chip network might be modified if the network device itself is not to be modelled and instead transactions are to be used to directly pass packets between network nodes. In the lectures notes, this was described as a *mid-level* model: what sort of model is logically above and below it? [4+2 Marks]
- ESL3. Show how a user-defined, abstract datatype can be passed along a SystemC channel by sketching several lines of code for a packet switch, router or demultiplexer. *This was lectured and illustrated in the toy classes but industrial users today would use the TLM 2.0 convenience sockets.* [7 Marks]
- ESL4. Define a transaction in Computer Science. How does the ESL use of this term differ ? [5 Marks]
- ESL5. What is the difference between a blocking and non-blocking transaction in terms of implementation, efficiency and callability? [6 Marks]
- ESL6. ♡ Sketch SystemC code for a shim function that converts a transactional port from blocking to non-blocking, or vice versa. (*n.b. One direction is harder than the other*). [5 Marks]
- ESL7. ♡ Add a simple transactional entry point to the five-bit counter RTL-counter from the SystemC exercises sheet that allows a remote client to make a five-bit, asynchronous parallel load of a value using a TLM call. [4 Marks]
- ESL8. ♡ Assume TLM calling is not synthesisable, but basic RTL-style SystemC can be converted to gates. Restructure your answer of ESL7. so that the five-bit counter has a net-level parallel load and so remains synthesisable. Then illustrate how to use a transactor to provide the TLM parallel load entry point into the now-supported, net-level parallel load. (You may ignore contention with other, simultaneous net-level operations on the counter.) [7 Marks]

ESL9. Here is some simple code for a net-level data generator consisting of a behavioural model of the data generator core and a transactor that exercises the hardware-level nets:

```
void write4p(unsigned char d)          // Transactor
{
    do (wait(clk.posedge_event()) while (ack.read())
        data = d;
        req = 1;
    do (wait(clk.posedge_event()) while (!ack.read())
        req = 0;
    }

    unsigned char x;
    while(1) instance.write4p(x++);    //And here is a client for it.
```

Sketch code for a further part of the system which is another transactor that owns its own thread and is a client for this net-level interface which makes an upcall to a user-provided function for each byte received. [4 Marks]

- ESL10. ♡ What is the advantage of putting a reference-passed delay parameter in the signature of TLM calls. [3 Marks]
- ESL11. Give two ways that timing annotations embedded in a transactional-level call can be synchronised with system global time ? [5 Marks]
- ESL12. Sketch a templated TLM SystemC model for a basic FIFO with capacity 8 items. [8 Marks]
- ESL13. Sketch code that will join two such TLM FIFOs together to make a longer FIFO. [5 Marks]
- ESL14. Sketch synthesisable SystemC or RTL-like code for such a FIFO (using either a circular buffer in a RAM or else based on a multi-stage structure). *This is rather straightforward exercise, but it is useful preparation for the next one!* [5 Marks]
- ESL15. Sketch code for a transactor (one of several possible) that enables interworking between the TLM and Synthesisable FIFOs of ESL12. and ESL14.. [5 Marks]
- ESL16. ♡ Sketch a SystemC model of a bus bridge and say what arbitration, queuing and address translation policies it implements. *Hint: a high-level model will likely lead to the shortest answer. It can be about six lines of code per direction. Syntax details are unimportant and, as always, pseudocode is acceptable.* [8 Marks]
- ESL17. Sketch a block diagram for a SoC containing at least two identical processor cores, a DRAM controller and some amount of on-chip SRAM. Mark each end of each connection with a suitable port style to be used as part of a TLM model (eg. blocking, non-blocking, initiator, target). [10 Marks]
- ESL18. Roughly estimate (order of magnitude) how many workstation instructions are used when modelling each access to the DRAM. [5 Marks]
- ESL19. Consider what simulation performance an ISS might give and can it ever be faster than real time ? [5 Marks]
- ESL20. Describe ways that caches can be modelled in a SoC [5 Marks]
- ESL21. ♡ a) Why might embedded firmware be cross-compiled to native code for a workstation ? [5 Marks]  
b) Give two or more ways hardware device access can be modelled when firmware including device drivers is cross-compiled for the modelling platform. [5 Marks]
- ESL22. What problems might arise when using high-level models of systems that use dynamic code loading and self-modifying code ? [5 Marks]
- ESL23. Briefly describe each of: cycle-accurate, approximately-timed, loosely-timed, untimed. [8 Marks]
- ESL24. What is the purpose and effect of the timing quantum in the loosely-timed model? [5 Marks]. Why might a transactional system exhibit different behaviour as the quantum is adjusted ? Is this useful or just bad ? [2 Marks]
- ESL25. ♡ How can contention for a resource be modelled with and without actual queuing of the transactions ?