

System On Chip: Design & Modelling (SOC/DAM)

Exercises

Here is the first set of exercises. These are intended to cover subject groups 1-4 of the SOC/DAM syllabus (R, SC, SD, ESL). These questions are styled as Tripos questions, with 20 marks each, and they largely involve repeating what was lectured. (In future years, questions that require a greater amount of creative thinking might be set).

Full model answers and answer notes are available (for supervisors only).

1 R: Verilog RTL Design with examples.

Topic R: Basic RTL coding styles, simulation and synthesis algorithms.

Exercise R1 : RTL Definition

- a) Give a brief definition of RTL and Synthesisable RTL. Name two example languages. [4 Marks]
- b) 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]
- c) Outline an algorithm to convert the syntax tree of a Verilog continuous assignment into gates. [8 Marks]
- d) Give three or more reasons why the basic algorithm from *c* may not always be appropriate. [3 Marks]

Bonus: Synthesisable RTL standards require that a variable is updated by at most one thread: is this strictly necessary ?

Exercise R2 : Structural Hazards

- a) Explain the terms 'structural hazard' and 'non-fully pipelined'. [4 Marks]
- b) Sketch a micro-architecture (data path) suitable for long multiplication of 32 bit unsigned operands (using Booth's method if you recall it). *If you have the lecture notes to hand, do not copy out the answer verbatim, but instead design the controller for the micro-architecture.* [6 Marks]
- c) How many clock cycles your design use on average and in the worst case? [5 Marks]

d) If a synthesisable RTL program uses asterisks for multiplication, what is typically placed on an ASIC or on an FPGA and what problems might there be ? [5 Marks]

Exercise R3 : Compute/Commit Cycle.

a) In VHDL (and SystemC), why are both signals and variables provided and what is their difference ? [5 Marks]

b) Describe the compute/commit evaluation paradigm used with signals (also used by Verilog's non-blocking assignments). [5 Marks]

c) What are the consequences of using signals instead of variables in clock distribution trees ? [5 Marks]

d) What is meant by a 'delta cycle' in an event-driven hardware simulator (EDS). Discuss whether they are a good or bad thing to have? [5 Marks]

Exercise R4 : Communication Styles.

a) Explain when each of the following are used to communicate between components in a SoC simulation: event, variable, net, signal, transaction. [2 Marks Each]

b) Using a bus bridge as an example, explain which model of communication is best for which situations. [7 Marks]

c) Explain why it is uncommon for a SoC to have a uniform memory architecture, even if it has a single logical address space. [3 Marks]

Exercise R5 : RTL Syntax and Semantics.

a) Give a concise abstract syntax for an RTL module that uses the 'synthesisable' subset of Verilog or VHDL (structural hierarchy may be ignored). [6 Marks]

b) Describe possible sources of non-determinism that may arise in your syntax. [4 Marks]

c) Outline an algorithm for converting a set of threads in your abstract syntax into a form where each register or net is assigned from exactly one expression. (Full marks will be awarded for answers that only consider one of the following types of assignment: signal, variable, blocking and non-blocking). [6 Marks]

d) Give an example where your algorithm may fail to resolve name aliases. [4 Marks]

Exercise R6 : On Chip SRAM Memory

- a)* Why might a memory cause structural hazards and how does the number of ports on the memory affect the problem? [5 Marks]
- b)* Compare the structural hazards and other relative merits arising from on-chip RAM, off-chip ZBT RAM and off-chip DRAM. [5 Marks]
- c)* How is an on-chip RAM tested and what effect does this have on user-level RTL ? [5 Marks]
- d)* Compare a register-file synthesised from flip-flops with an on-chip SRAM macrocell ? [5 Marks]

Exercise R7 : Adder Synthesis

- a)* Implement a function that accepts a pair of lists of nets, least significant net first, and outputs the net lists for an adder with fast carry, in a similar, list form. (NB a fast carry uses gates with many inputs as compared with a ripple carry that has constant fan-in). [10 Marks]
- b)* Outline the modifications needed to your function to make it output a subtractor. [2 Marks]
- c)* Explain how you subtractor can easily implement all six common integer comparison predicates. [2 Marks]
- d)* Outline the modifications instead needed to generate a Kogge-Stone adder and say what differences in adder performance this leads to. [6 Marks]

http://www.acsel-lab.com/Projects/fast_adder/adder_designs.htm

Exercise R8 : H/W versus S/W

Summarise the main differences between synthesisable RTL and general multi-threaded software in terms of programming style and paradigms. [20 Marks].

2 SC: SystemC.

Topics: SystemC and high to low-level mapping examples.

Exercise SC1 : SystemC

- a)* Describe the principle features of SystemC. [5 Marks]
- b)* How is an RTL-style non-blocking assignment achieved in SystemC ? [5 Marks]

c) How is design module heirarchy expressed in SystemC and what sorts of 'channels' are supported between modules ? [8 Marks]

d) Why adapt a general-purpose language like C++ for hardware use when special hardware languages exist ? [2 Marks]

Exercise SC2 : C Modelling Techniques

a) List the main components of the SystemC kernel and library [5 Marks]

b) 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]

c) Give the principles of operation for a program that takes a gate-level design and generates a C/C++ model of it. Your program might or might not use a scheduler kernel. [10 Marks]

Exercise SC3 : SystemC Channels

a) Like VHDL, the SystemC version 1.0 allowed only signals to be wired between components. What is a SystemC signal and what restrictions did this limit impose ? [8 Marks]

b) Give a scenario where the use of a variable instead of a signal would result in the wrong behaviour. [3 Marks]

c) Show how an abstract datatype can be passed along a SystemC 2.0 channel by sketching the code for a packet switch, router or demultiplexer. [7 Marks]

d) Sketch a transactional code fragment where method and thread communication is used to join components. [5 Marks]

Exercise SC4 : Integer Precision & Transactors.

a) How does SystemC model registers that have widths not native to the C language ? [4 Marks]

b) 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.* [5 Marks]

c) Add a simple transactional entry point to your five-bit counter that allows a remote client to make a five bit, asynchronous parallel load of a value in the transaction. [4 Marks]

d) Restructure your answer of part *c* to use a separate transactor module so that the five-bit counter, now with parallel load, remains synthesisable. [7 Marks]

NB: Some parts of these SC exercises borrow knowledge from lecture group ESL.

Exercise SC5 : Hierarchy of Synthesisable Representations.

a) 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 sum of the five bit inputs exceeds 511. [6 Marks]

b) Give the SystemC synthesisable equivalent design. [7 Marks]

c) Give a schematic (circuit) diagram for the design (use adders and/or ALU blocks rather than giving full circuits for an such components). [7 Marks]

Exercise SC6 : Hierarchy of Synthesisable Representations.

Repeat exercise SC5 using a slightly more complex design: eg. a long division component.

Exercise SC7 : Transactors

a) Define suitable nets for a simplex interface that transfers bytes using a four phase-handshake. Describe the protocol. *Answer this part using RTL or natural language. You may assume a suitably high-frequency clock is available that will not alias the protocol.* [5 Marks]

b) Sketch RTL for a counter module that writes its output to the four-phase interface. *Precise syntax and operational details are unimportant, but a sensible answer would be a Verilog module that increments once for each output operation and wraps after decimal 255 back to zero.* [5 Marks]

c) Sketch code for a blocking transactor that writes to a four-phase, net-level interface and also some client code for it that, when the two are combined, gives it equivalent functionality to the module of part *b*. *Answer using C-like syntax, optionally using TLM 2.0 terminology.* [6 Marks]

d) Sketch further code for a transactor that owns its own thread and is a client for the answers to *b* and/or *c*. It should makes an upcall to the user's function for each byte received. [4 Marks]

3 SD: System Design and Structures

Lecture group SD covers SoC hardware structure and programming, with some forward references to TLM and other high-level models.

Topics: SoC Structure, Typical I/O components, Interrupts, Programmed I/O, DMA, Clock Domains, GPIO. modelling,

Exercise SD1 : 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 has parallel data 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]

Exercise SD2 : Interrupt Arbiter.

- a) Sketch the RTL or SystemC code for an interrupt arbiter that stores eight vectors with individual interrupt enable flags. The arbiter monitors eight interrupt inputs and presents the highest-priority, non-masked interrupt vector to the processor when the processor asserts an interrupt acknowledge signal. *Fine details will vary from answer to answer. Syntactic accuracy would not be expected in examination answers.* [10 Marks]
- b) How does the processor set up the device and what must it do after servicing an interrupt ? *Answering this part requires knowledge of operating system device drivers.* [4 Marks]
- c) How would you modify the interrupt controller to share work over two CPUs ? Is this always a good idea ? [6 Marks]

Exercise SD3 : Programmed I/O.

- a) What is meant by polled I/O and how does it compare with interrupt driven I/O ? [4 Marks]
- b) Sketch a set of typical macro definitions in C suitable for making low-level hardware access to a UART or similar device that contains status, control and

data registers. [4 Marks]

c) Give a pair of short subroutines in C that perform polled-mode, blocking read and write operations using your macros. [4 Marks]

d) Give a blocking TLM SystemC-like model of a UART device (that just does console or file I/O rather than implementing a full serial port). [4 Marks]

e) Explain how the bus interface between the hardware and the software might be bypassed in a high-level model of the system? [4 Marks]

Exercise SD4 : DMA controller.

a) 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.* [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]

c) Explain how different timing models can be used (eg. loose, approximate, cycle-accurate) in conjunction with your model and what bugs in the system architecture might be exposed by each form. [6 Marks]

d) Say with justification whether your SystemC DMA controller could be synthesised into RTL for use in a real SoC ? [3 Marks]

Exercise SD5 : Bus Bridge.

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

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

c) How might internal queue structure vary ? [3 Marks]

d) How might arbitration policy vary ? [3 Marks]

e) Sketch a SystemC or RTL 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. Syntax details are unimportant and pseudocode is acceptable.* [8 Marks]

Exercise SD6 : Network I/O, H/W and S/W.

- 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]

Exercise SD7 : Audio Output Port

- a) Define a feasible serial interface to an audio output DAC that conveys a pair of stereo channels of 16 bit precision at 44.1 ksp/s. *Hint: Three nets are normally used.* [4 Marks]
- b) 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]
- c) Describe a feasible high-level or TLM model of the same subsystem, whereby the sound can come out of the sound card on the modelling workstation. What problems might arise ? [4 Marks]

Exercise SD8 : Single-Bit DAC

- a) What is the advantage of a 'single-bit' digital to analog converter over older techniques ? [5 Marks]
- b) Give either the circuit for or RTL design of a pulse density modulator that accepts a five-bit input word. [5 Marks]
- c) Give a lower bound on the word rate at the input to five-bit modulator for CD-quality audio (44.1 ksp/s, 16 bits). [5 Marks]
- d) Give and explain the block diagram for a CD-quality delta-sigma analog to digital convertor. [5 Marks]

4 ESL: Transactional modelling (ESL).

Topics: Transactional modelling. Electronic systems level design. IP-XACT.

Exercise ESL1 : Transactions

- a)* Define a transaction in Computer Science. How does the ESL use of this term differ ? [5 Marks]
- b)* What is the difference between a blocking and non-blocking transaction in terms of implementation, efficiency and callability? When should each typically be used ? [6 Marks]
- c)* Sketch SystemC code that converts an transactional port from blocking to non-blocking, or vice versa. [5 Marks]
- d)* Give two ways that timing models can be embedded in a transactional level model ? [5 Marks]

Exercise ESL2 : TLM FIFO Design

- a)* Sketch a templated SystemC TLM model for a basic FIFO with capacity 8 items. [8 Marks]
- b)* Sketch code that will join two such FIFOs together to make a longer FIFO. [5 Marks]
- c)* Sketch Synthesisable SystemC or RTL code for a FIFO (using either a circular buffer in a RAM or else based on a multi-stage structure). [5 Marks]
- d)* Sketch code for a transactor that enables interworking between the TLM and Synthesisable FIFOs of *a* and *c*. [5 Marks]

Exercise ESL3 : TLM Memory Access

- a)* 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, master, slave). [10 Marks]
- b)* Roughly estimate how many workstation cycles used in modelling each access to the DRAM. [5 Marks]
- c)* Describe how back-door access to the DRAM might be implemented. [5 Marks]

Exercise ESL4 : Instruction Set Simulator

- a)* What is an ISS (instruction set simulator) ? [2 Marks]
- b)* What simulation performance can an ISS give and can it be faster than real time ? (mention 'JIT' mode). [5 Marks]

c) Describe ways that caches can be modelled in conjunction with an ISS. [5 Marks]

d) When an ISS is embedded in a SoC design, what differences can we expect to see when compared with a cycle-accurate model ? [5 Marks]

Exercise ESL5 : Mixed-abstraction/cross-compiled modelling.

a) Why might embedded firmware be cross-compiled to native code for a workstation ? [5 Marks]

b) Give two ways hardware device access can be modelled when firmware is natively compiled (ie. in a mixed-abstraction model). [5 Marks]

c) What issues of endianness might arise ? How can they be overcome ? [5 Marks]

d) How can dynamic code load and self-modifying code be modelled ? [5 Marks]

Exercise ESL6 : Timing Models

a) Briefly describe each of: Cycle accurate, approximately times, loosely timed, untimed. [5 Marks]

b) Why might a transactional system exhibit different behaviour on the different models ? Is this good or bad ? [5 Marks]

c) What is the purpose and effect of the timing quantum in the loosely timed model? [5 Marks]

d) Sketch two variations of a blocking TLM method where overall functionality and total reported time is identical, but where one is implemented in finer detail. [5 Marks]