

SoC D/M Exercises 10/11

These exercises are allocated marks at Tripos examination level, with 20 marks making a full exam question. Example answers are available to supervisors. There is some repetition of material between the exercises, so a suitable target is to solve approximately half of them. Those marked with a heart (♥) are recommended.

1 SoC Components Exercises

- SOC1. ♥ What is meant by polled I/O and how does it compare with interrupt driven I/O ? [4 Marks]
- SOC2. ♥ 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]
- SOC3. ♥ Give a pair of short subroutines in C that perform polled-mode, blocking read and write operations using your macros of SOC2.. [4 Marks]
- SOC4. ♥ Show how to wire up a push button to a GPIO pin and sketch out the code for a device driver that returns how many times it has so far been pressed. Sketch polled and interrupt driven code. Neglect debouncing. [10 Marks]
- SOC5. 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 or otherwise reads the device. [Start by defining the net-level connections to the component.] *Fine details will vary from answer to answer. Syntactic accuracy would not be expected in examination answers.* [10 Marks]
- SOC6. How does the processor set up the interrupt arbiter device of SOC5. and what must it do after servicing an interrupt ? [4 Marks]
- SOC7. How would you make an interrupt arbiter that shares work over two CPUs ? Is this always a good idea ? [6 Marks]
- SOC8. 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 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]
- SOC9. ♥ 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 between bus bridge designs ? [3 Marks]
- d) How might arbitration policy vary between bus bridge designs ? [3 Marks]
- SOC10. 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]
- SOC11. ♥ 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 kbps. *Hint: Three nets are normally used.* [4 Marks]

- SOC12. 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]
- SOC13. ♡ Clock Domain Crossing.
- 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]
 - 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]
 - What constraints exist for simplex protocols that cross clock domains ? [6 Marks]
 - What constraints exists for duplex protocols that span clock domains ? [2 Marks]
- SOC14. 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.*
- SOC15. Single-Bit DAC/ADC *Not lectured in 09/10 or 10/11.*
- What is the advantage of a 'single-bit' digital to analog converter over older techniques ? [5 Marks]
 - Give either the circuit for or RTL design of a pulse density modulator that accepts a five-bit input word. [5 Marks]
 - Give a lower bound on the word rate at the input to five-bit modulator for CD-quality audio (44.1 ksps, 16 bits). [5 Marks]
 - Give and explain the block diagram for a CD-quality delta-sigma analog to digital convertor. [5 Marks]

2 ESL (Electronic System Level) Exercises

- 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. Syntax details are unimportant and, as always, pseudocode is acceptable.* [8 Marks]
- ESL17. ♡ What is an ISS (instruction set simulator or emulator) ? [2 Marks]
- ESL18. 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]
- ESL19. Roughly estimate (order of magnitude) how many workstation instructions are used when modelling each access to the DRAM. [5 Marks]
- ESL20. Consider what simulation performance an ISS might give and can it ever be faster than real time ? [5 Marks]
- ESL21. The DMI feature of SystemC enables a client to fetch data from a modelled memory with bypass of the bus/NoC model. How might this be implemented and how might it impact modelling accuracy. [5 Marks]
- ESL22. Describe ways that caches can be modelled in a SoC [5 Marks]
- ESL23. Describe a suitable model or models of the subsystem of SOC12., whereby the audio is rendered via the sound port of the modelling workstation. What problems might arise ? *Hint: There is a TLM example of a music playing system, with TLM DAC model, in the additional material on the course web site (or last year's site).* [4 Marks]
- ESL24. ♡ 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]
 d) What issues of endianness might arise ? How can they be overcome ? [5 Marks]
- ESL25. What problems might arise when using high-level models of systems that use dynamic code loading and self-modifying code ? [5 Marks]

- ESL26. Give alternative definitions of the blocking calls of SOC3. to produce a high-level C/C++ model of a UART device (that just does console or file I/O rather than implementing a full serial port). [4 Marks]
- ESL27. Explain how firmware can be conditionally compiled to either direct calls through the code of SOC3. or instead call the code of ESL26.. (Note, there are two answers to the latter half, where the the bus interface between the components is either modelled or not) [10 Marks]
- ESL28. Briefly describe each of: cycle-accurate, approximately-timed, loosely-timed, untimed. [8 Marks]
- ESL29. Why might a transactional system exhibit different behaviour on the different models ? Is this good or bad ? [2 Marks]
- ESL30. What is the purpose and effect of the timing quantum in the loosely-timed model? [5 Marks]
- ESL31. Explain how different timing models can be used (e.g. loose, approximate, cycle-accurate) in conjunction with your answer to the DMA question (SOC8.) and what bugs in the system architecture might be exposed by each form. [6 Marks]
- ESL32. ♡ How can contention for a resource be modelled with and without actual queuing of the transactions ?