UNIVERSITY OF CAMBRIDGE

COMPUTER SCIENCE TRIPOS Part IB

Wednesday 3 June 2015 1.30 to 4.30 pm

COMPUTER SCIENCE Paper 5

Answer ***five*** *questions.*

Submit the answers in five ***separate*** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* ***all*** *attempted questions, and circle the number of the question attached.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

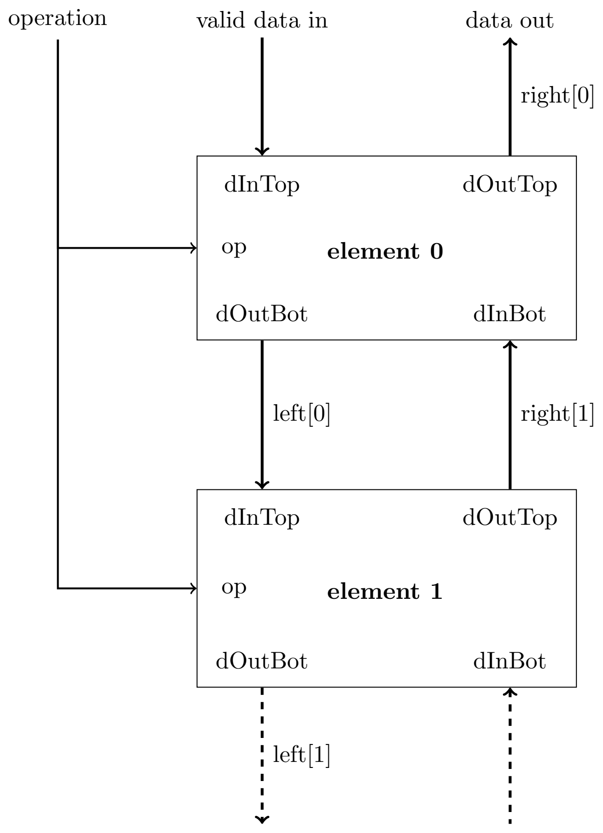
STATIONERY REQUIREMENTS	SPECIAL REQUIREMENTS
Script paper	*Approved calculator permitted*
Blue cover sheets	
Tags	
Rough work pad	

1 Computer Design

Consider the following SystemVerilog code containing the module **element**.

```
typedef struct packed {
   logic       v; // valid bit (1=valid, 0=invalid)
   logic [7:0] d; // data
} itemT;

typedef enum {opNone, opRead, opWrite} opT;

function automatic logic swapToggle(itemT a, itemT b);
   return (a.v && !b.v) || (a.d < b.d);
endfunction

module element(
     input        clk,
     input        rst,
     input  opT   op,
     input  itemT dInTop,
     output itemT dOutTop,
     input  itemT dInBot,
     output itemT dOutBot);

   itemT   d[1:0];
   logic   swap;

   always_comb
     begin
        dOutTop = d[swap];
        dOutBot = d[!swap];
     end

   always_ff @(posedge clk)
     if(rst)
       begin
          d[0] <= itemT'{d:0, v:0};
          d[1] <= itemT'{d:0, v:0};
          swap <= 0;
       end
     else
       case (op)
         opRead:
           begin
              d[swap] <= dInBot;
              swap <= swap ^ swapToggle(d[!swap],dInBot);
              // Note: ^ is SystemVerilog for XOR
           end
         opWrite:
           begin
              d[!swap] <= dInTop;
              swap <= swap ^ swapToggle(dInTop, d[swap]);
           end
       endcase
endmodule
```

The figure below shows that the **element** module can be instantiated many times over to form a linear structure (technically called a *systolic array*). Valid data (i.e. data with the v bit set) is input on the top left of element 0 during a write operation (op=opWrite). Data is read out from element 0 top right during a read operation (op=opRead). Clock (clk) and reset (rst) signals have been omitted for clarity.



(a) If there are N elements, how many data items (of type itemT) can be stored by this structure? [4 marks]

(b) Four valid data items are written in the sequence: 4, 2, 3, 1. What will be the state of the outputs of element 0 (left[0], right[0]) and element 1 (left[1], right[1]) after each write clock cycle? Clearly enumerate the state changes, including changes to the swap bits. (e.g. via a state transition table) [6 marks]

(c) If there are then four read operations, in what sequence will data be read out? [6 marks]

(d) What function does this systolic array perform and what is its space and time complexity for processing N items by first writing all N items and then reading N items? [4 marks]

2 Computer Design

(*a*) What is the von Neumann bottleneck and why can it limit performance on today's RISC machines? [4 marks]

(*b*) What computer architecture techniques are used to mitigate the effects of the von Neumann bottleneck on RISC machines as compared to early computers like EDSAC? [4 marks]

(*c*) How do RISC machines enforce memory protection for applications with disjoint data sets? [4 marks]

(*d*) What is segmented addressing (e.g. as used on x86 machines)? [4 marks]

(*e*) If non-volatile random access memory had similar performance and cost to DRAM, how would this change the memory hierarchy? Justify your answer. [4 marks]

3 Computer Design

(*a*) Consider a multicore processor running the MSI cache coherence protocol in each core's private caches. The caches are connected to each other and memory via a snoopy bus.

(*i*) What is a cache coherence protocol and in what systems is it needed? [4 marks]

(*ii*) What events would lead a cache to issue a bus transaction if it holds a block of data in state S? [4 marks]

(*iii*) When and why does data need to be flushed back to memory in this protocol? [4 marks]

(*b*) (*i*) What are the semantics of load linked and store conditional instructions? [4 marks]

(*ii*) Describe the synchronisation method that the following code performs by adding comments to it.

```
        membar
label1: ll   r2, 0(r1)
        sub  r2, r2, #1
        sc   r2, 0(r1)
        beqz r2, label1
label2: load r2, 0(r1)
        bneq r2, label2
```

[4 marks]

4 Computer Networking

The following equation provides a simple way to estimate the throughput of a TCP connection, as a function of the loss probability p, the round-trip time RTT, and the maximum segment size MSS.

$$\text{TCP Throughput} = \sqrt{\frac{3}{2}}\frac{\text{MSS}}{\text{RTT}\sqrt{p}}$$

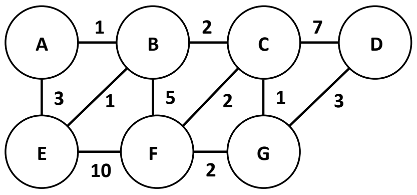
(*a*) Alice wants to send a large amount of data to Bob over a network path with RTT = 100 ms, $p = 0.01$, and MSS = 10,000 bits. What is the expected throughput in Mbit/s? [2 marks]

(*b*) With the aid of a clearly labelled diagram showing window-size versus time, derive the above equation. [10 marks]

(*c*) Alice has two options to improve the throughput: halving either the RTT or the loss probability p. If both cost the same, which is more cost effective and why? [2 marks]

(*d*) Consider your derivation of the equation in part (*b*). State three assumptions that are made and describe when these assumptions may not hold in reality. [6 marks]

5 Computer Networking



(*a*) In the above network, use Dijkstra's shortest-path algorithm to compute the shortest path from E to all network nodes. Show your working in a table: each column indicating a destination node, each row indicating an iteration of the algorithm. [10 marks]

(*b*) In 2008 an ISP was reported to have hijacked traffic for YouTube causing traffic for YouTube to be diverted into the ISP's network.

At the time, YouTube used only three IP addresses; 208.65.153.238, 208.65.153.251 and 208.65.153.253, announced as a single prefix 208.65.152.0/22.

Despite YouTube using only three addresses, each browser's YouTube URL requests are ultimately routed to the closest of over a dozen data-centres Google operates world-wide.

(*i*) Describe two concepts from the course that make this possible. [2 marks]

(*ii*) State the smallest advertised netblock that would identify all YouTube addresses. [1 mark]

(*iii*) In an attempt to resolve the problem, YouTube advertised the netblock 208.65.153.0/24, but this was the same netblock as advertised by the rogue ISP. Why would this not solve the problem? [2 marks]

(*iv*) YouTube advertised two smaller netblocks, each one half of 208.65.153.0/24. Why should this now work? [2 marks]

(*v*) BGP networks may optionally filter netblocks that are below a given size. This filtering affected the YouTube fix in (*b*)(*iv*), but not that in (*b*)(*iii*). Estimate the size of the netblock filter. [1 mark]

(*vi*) Why does BGP implement such filtering? [2 marks]

6 Computer Networking

(*a*) An older home-network router has an upload bandwidth of 1Mbit/s to the Internet, and a 100kbyte first-in first-out (FIFO) buffer for packets awaiting transmission. Packets have a maximum transmission unit (MTU) of 1500 bytes.

(*i*) If the buffer is completely full, how long does it take the router to transmit all of the bytes in the buffer? [2 marks]

(*ii*) Suppose the router supports two FIFO queues, one high-priority for interactive applications (like Voice over IP) and the other lower-priority for all remaining traffic. If a VoIP packet arrives when the queue for interactive applications is empty, what is the maximum time before the router starts transmitting the VoIP packet? (Assume that the router does not preempt any ongoing packet transmission.) [2 marks]

(*b*) Consider the delay at each node (router or switch) in a network given by this equation:

$$d_{\text{node}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

For each term: d_{proc}, d_{queue}, d_{trans} and d_{prop}

(*i*) explain what it represents;

(*ii*) state one way to reduce it; and

(*iii*) indicate a typical range of values in a 1Gbit/s local area network with link-length less than 500m.

State your assumptions throughout. [4 × 3 marks]

(*c*) A lecturer remarks that "centralised multiplexing" offers potential gains in efficiency over non-centralised multiplexing.

Give *two* reasons why this could be so. In each, state clearly what feature must be *centralised* to achieve these gains. [4 marks]

7 Concurrent and Distributed Systems

(*a*) (*i*) Name the four necessary conditions for deadlock. [2 marks]

(*ii*) Which of these conditions is frequently precluded in operating-system kernel designs in order to prevent deadlocks and why? [2 marks]

(*b*) Deadlocks are not limited to locks; cycles of waiting on *condition variables* can also lead to the "deadly embrace".

(*i*) Explain why it might be more difficult to debug deadlocks involving condition variables than those simply involving locks. [2 marks]

(*ii*) Briefly describe a condition-variable API change that might allow this problem to be solved in some cases; explain why it cannot always help. [2 marks]

(*c*) FreeBSD's WITNESS feature checks statically defined and dynamically discovered lock orders. Each time a lock is acquired, any previously undiscovered graph edges involving lock types currently held by the thread and the newly acquired lock type will be added to the graph. Cycle detection is performed, and debug information is printed if a previously unreported cycle is discovered.

(*i*) Describe a common code structure in which programmers are likely to be able to define a static order between two lock types. [2 marks]

(*ii*) Describe a common case in which programmers are likely to rely instead on dynamic discovery of an order between two lock types. [2 marks]

(*iii*) Unlike the deadlock-detection algorithm presented in lecture, the WITNESS algorithm does not remove edges when locks are released. Explain why WITNESS's behaviour might be more useful in practice. [2 marks]

(*iv*) WITNESS is subject to *false positives*: warnings can be emitted due to legitimate cycles even though, by design, the cycle could never trigger an actual deadlock. Describe a situation in which this might arise, and explain why deadlock could never occur. [3 marks]

(*v*) WITNESS, as written, is intended to be used with *mutexes* and other lock types providing mutual exclusion. A developer might naïvely extend WITNESS to support reader-writer locks (**rwlock**) by introducing graph edges for both read and write acquires as it does for mutex acquires. Explain why this might not always lead to the desired result. [3 marks]

8 Concurrent and Distributed Systems

(*a*) (*i*) Define the term *capability*. [2 marks]

(*ii*) What two fields must RBAC-based ACL entries always contain? [2 marks]

(*b*) Network-Attached Secure Disks (NASD) utilise *file managers* and *block servers*. File-manager RPCs exchange an authorised user ID, password, and object ID for a keyed cryptographic capability granting block access: $f(k, ObjID, rights)$.

(*i*) Describe the consequences of a user learning the value of key k. [2 marks]

(*ii*) Alice obtains a capability for object O_i. Bob then issues an RPC to the file manager revoking Alice's access to O_i. Describe what occurs when Alice performs her next block-server read on O_i. [2 marks]

(*iii*) Explain why it might be desirable, from a security perspective, to add a timeout field t, protected by the keyed hash, to the capability. [2 marks]

(*iv*) Developers extend NASD to support Quorum-replicated block servers. What new failure mode may arise during a Quorum block write, relative to unmodified NASD capabilities, in adding capability timeouts? [2 marks]

(*c*) The Andrew File System (AFS) is authenticated and encrypted using Kerberos; ACLs expressing positive and negative rights for users and groups. *Multiuser AFS clients* (e.g., UNIX servers) build a secure RPC connection for each local user, authenticated with their Kerberos ticket, and issue RPCs (e.g., file **read**) on their behalf only via their own connection. If no suitable Kerberos ticket is available (e.g., the ticket has expired, the user has destroyed their ticket, or a job is running unattended), then an insecure connection is used instead.

(*i*) The group *system:anyuser* holds the union of unauthenticated (anonymous) users and all authenticated users. Explain why an ACL granting read access to *system:anyuser* via a positive entry, but denying read access to user *rnw* via an overriding negative entry, might prove problematic. [2 marks]

(*ii*) Describe the consequences to AFS authentication and authorisation of a malicious local user gaining root access on a multiuser client. [2 marks]

(*iii*) An AFS client uses the unauthenticated Network Time Protocol (NTP) to synchronise its clock with the AFS server. Attacker Mallory is able to inspect, drop, and insert packets between the AFS client and server (e.g., by controlling a network switch). Describe an attack that allows Mallory to inject malicious content into the client's AFS cache, but that does not allow Mallory to write content directly to the AFS server. [4 marks]

9 Concurrent and Distributed Systems

These questions relate to *reliable multicast* and *distributed transactions.* Answers may include timelines of message transmissions/deliveries or transaction submissions/commits. For each event in the timeline, show a physical timestamp (T_1, T_2, ...), process numbers (P_1, P_2, ...), operation ('transmits', 'delivers', 'submits', 'commits'), and a numbered message (m_1, m_2, ...) or numbered transaction (x_1, x_2, ...). For example, "T_7: P_1 transmits m_4". In this context define:

(*a*) (*i*) *FIFO ordering* [1 mark]

(*ii*) *causal ordering* [1 mark]

(*iii*) *total ordering* [1 mark]

(*iv*) *strong consistency* [1 mark]

(*v*) *weak consistency* [1 mark]

(*b*) (*i*) Does causal ordering imply total ordering? If so, explain why; if not, show a counterexample, labelling and explaining the violating event. [2 marks]

(*ii*) Does total ordering imply causal ordering? If so, explain why; if not, show a counterexample, labelling and explaining the violating event. [2 marks]

(*c*) A replicated database is implemented using totally ordered reliable multicast. Clients may submit transactions to any process in the group. When process P_x receives a new transaction x_i from a client, it will multicast the transaction to all processes, including itself. As x_i is delivered by multicast, each process submits the transaction to a local ACID database. P_x returns the result (abort or commit) to the client; other processes discard the transaction result.

(*i*) This model works well if queries do not contain the SQL **time** keyword, which is substituted with the current time when a transaction is evaluated. Explain why using **time** might be a problem and describe a solution. [4 marks]

(*ii*) In the first release of the database, processes submit received multicast transactions synchronously, one at a time, to the local database. In a later version, to improve performance, processes are allowed to submit multiple transactions at a time asynchronously to the local database. Why does this fail to provide strong consistency for distributed transactions? Describe a solution that might allow limited (but useful) local concurrency to be supported. [3 marks]

(*iii*) Describe changes to the design to support weak consistency, and describe two reasons why this might improve performance. [4 marks]

END OF PAPER