COMPUTER SCIENCE TRIPOS Part IB

Monday 2 June 2008 1.30 to 4.30

PAPER 3

*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
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
None

1 ECAD

(*a*) For the two mystery Verilog modules below, what output sequences do they produce assuming that registers are initially reset to zero? Explain your answer. [6 marks]

```
module mystery0(input a, output reg [4:0] b);
  always @(posedge a) begin
    b[0] = !b[0];
    b[1] = b[0]    ^ b[1];
    b[2] = &b[1:0] ^ b[2];
    b[3] = &b[2:0] ^ b[3];
    b[4] = &b[3:0] ^ b[4];
  end
endmodule

module mystery1(input c, output reg [4:0] d);
  always @(posedge c)    d[0] <= !d[0];
  always @(posedge d[0]) d[1] <= !d[1];
  always @(posedge d[1]) d[2] <= !d[2];
  always @(posedge d[2]) d[3] <= !d[3];
  always @(posedge d[3]) d[4] <= !d[4];
endmodule
```

(*b*) If the modules were to be implemented in terms of two-input AND, OR and XOR gates, NOT gates, and D flip-flops, what would be the minimal circuits? [6 marks]

(*c*) Which module could be clocked most quickly if it were implemented on an FPGA with 4-input look-up tables (LUTs)? Explain your answer. [4 marks]

(*d*) What are the timing issues for any circuit looking at the outputs of the two mystery modules? [4 marks]

2 Floating-Point Computation

(*a*) Write a function in a programming language of your choice that takes a (32-bit IEEE format) `float` and returns a `float` with the property that: given zero, infinity or a positive normalised floating-point number then its result is the smallest normalised floating-point number (or infinity if this is not possible) greater than its argument. You may assume functions `f2irep` and `irep2f` which map between a `float` and the same bit pattern held in a 32-bit integer. [6 marks]

(*b*) Briefly explain how this routine can be extended also to deal with negative floating-point values, remembering that the result should always be greater than the argument. [2 marks]

(*c*) Define the notions of *rounding error* and *truncation error* of a floating-point computation involving a parameter h that mathematically should tend to zero. [2 marks]

(*d*) Given a function f implementing a differentiable function that takes a floating-point argument and gives a floating-point result, a programmer implements a function

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

to compute its derivative. Using a Taylor expansion or otherwise, estimate how rounding and truncation errors depend on h. You may assume that all mathematical derivatives of f are within an order of magnitude of 1.0. [8 marks]

(*e*) Suggest a good value for h given a double-precision floating-point format that represents approximately 15 significant decimal figures. [2 marks]

3 Programming in C and C++

A hardware engineer stores a FIFO queue of bits in an `int` on a platform with 32-bit `int`s and 8-bit `char`s using the following C++ class:

```
class BitQueue {
  int valid_bits; //the number of valid bits held in queue
  int queue;      //least significant bit is most recent bit added
 public:
  BitQueue(): valid_bits(0),queue(0) {}
  void push(int val, int bsize);
  int pop(int bsize);
  int size();
};
```

(*a*) Write an implementation of `BitQueue::size`, which should return the number of bits currently held in `queue`. [1 mark]

(*b*) Write an implementation of `BitQueue::push`, which places the `bsize` least significant bits from `val` onto `queue` and updates `valid_bits`. An exception should be thrown in cases where data would otherwise be lost. [5 marks]

(*c*) Write an implementation of `BitQueue::pop`, which takes `bsize` bits from `queue`, provides them as the `bsize` least significant bits in the return value, and updates `valid_bits`. An exception should be thrown when any requested data is unavailable. [4 marks]

(*d*) The hardware engineer has built a communication device together with a C++ library function `send` to transmit data with the following declaration:

```
void send(char);
```

Use the `BitQueue` class to write a C++ definition for:

```
void sendmsg(const char* msg);
```

Each of the characters in `msg` should be encoded, in index order, using the following binary codes: `'a'`=`0`, `'b'`=`10`, `'c'`=`1100`, and `'d'`=`1101`. All other characters should be ignored. Successive binary codes should be bit-packed together and the code `111` should be used to denote the end of the message. Chunks of 8-bits should be sent using the `send` function and any remaining bits at the end of a message should be padded with zeros. For example, executing `sendmsg("abcd")` should call the `send` function twice, with the binary values `01011001` followed by `10111100`. [10 marks]

4 Computer Graphics and Image Processing

(*a*) Most liquid crystal displays divide a pixel into three sub-pixels coloured red, green, and blue. Explain why this is so. [4 marks]

(*b*) Some liquid crystal displays divide a pixel into four sub-pixels coloured red, green, blue, and white. Explain why this might be useful, what advantages it has, and what limitations it has. [6 marks]

(*c*) Compare and contrast half-toning and error diffusion. Include in your answer an explanation of the situations in which each is superior to the other. [6 marks]

(*d*) One method of anti-aliasing is to sample at high resolution, $n \times n$ higher than the final image, and then to average each block of $n \times n$ pixels to give a single pixel value. Discuss the advantages and disadvantages of using

(*i*) Gaussian blurring, and

(*ii*) median filtering

in place of simple averaging. [4 marks]

5 Mathematical Methods for Computer Science

(*a*) Define the Fourier transform, $\mathcal{F}_{[f(x)]}(w)$, of a function $f(x)$ and the inverse transform to construct $f(x)$ in terms of $\mathcal{F}_{[f(x)]}(w)$. [2 marks]

(*b*) Show that

$$\frac{d}{dw}\left(\mathcal{F}_{[f(x)]}(w)\right) = \mathcal{F}_{[-ixf(x)]}(w)\,.$$

[4 marks]

(*c*) Define the *convolution* $f(x) * g(x)$ of two functions $f(x)$ and $g(x)$. State and prove the *convolution theorem.* [4 marks]

(*d*) Consider the function $f_a(x)$ in the case where a is a positive constant defined by $f_a(x) = e^{-ax}$ for $x \geq 0$ and zero for $x < 0$. Derive the Fourier transform of $f_a(x)$. [4 marks]

(*e*) Use the convolution theorem to determine the convolution $f_a(x) * f_b(x)$ where a and b are positive constants when

(*i*) $a \neq b$ [3 marks]

(*ii*) $a = b$. [3 marks]

[Note: You may assume that any appropriate integrals exist and that the order of integration and differentiation may be interchanged as necessary.]

6 Logic and Proof

(*a*) Draw the BDDs for the formulae $(P \wedge Q) \rightarrow R$ and $(P \vee Q) \rightarrow R$, ordering the variables alphabetically. [2+2 marks]

(*b*) Combine those BDDs to obtain the BDD for $[(P \wedge Q) \rightarrow R] \leftrightarrow [(P \vee Q) \rightarrow R]$. Briefly explain your working. [5 marks]

(*c*) Use the DPLL method to determine whether or not the following set of formulae is consistent.

$$(Q \rightarrow R) \vee P$$

$$R \rightarrow (\neg P \vee Q)$$

$$(\neg P) \leftrightarrow Q$$

$$P \rightarrow R$$

[6 marks]

(*d*) Use the sequent calculus to determine whether or not the following sequent is valid:

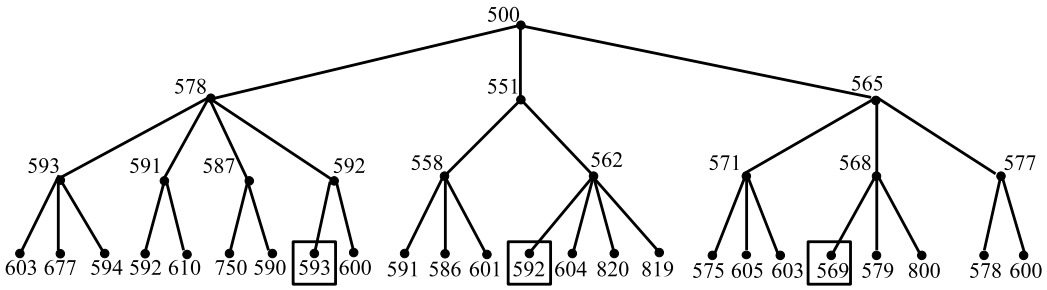
$$\forall x\, [P(x) \vee Q(f(x))], \exists y\, \neg P(y) \Rightarrow \exists y\, Q(y)$$

[5 marks]

7 Artificial Intelligence I

(*a*) Give a general description of the operation of the *Recursive Best-First Search* (*RBFS*) algorithm. [6 marks]

(*b*) Consider the following search tree.



The numbers by the nodes denote the sum of some path cost and heuristic. The boxed nodes are goals. Describe in detail the way in which the RBFS algorithm searches this tree. Your answer should indicate the order in which nodes are expanded, the reason that this order is used, and should state which of the three goals is found and why. Note that smaller numbers represent more desirable nodes. [12 marks]

(*c*) What shortcoming of the $A^\star$ algorithm does the RBFS algorithm address, and how does it achieve this? [2 marks]

8 Introduction to Security

(*a*) A source of secure, unpredictable random numbers is needed to choose cryptographic keys and nonces.

(*i*) Name *six* sources of entropy that can be found in typical desktop-computer hardware to seed secure random-number generators. [4 marks]

(*ii*) What sources of entropy can a smartcard chip, like the one in your University Card, access for this purpose? [4 marks]

(*b*) As Her Majesty's prime hacker "001", on a mission deep inside an enemy installation, you have gained brief temporary access to a secret chip, which contains a hardware implementation of the DES encryption algorithm, along with a single secret key. You connect the chip to your bullet-proof laptop and quickly manage to encrypt a few thousand 64-bit plaintext blocks of your choice, and record the resulting 64-bit ciphertext blocks. You are unable to directly read out the DES key K used in the chip to perform these encryptions and you will not be able to leave the site without knowing K. But you know that all S-boxes in the last DES round are supplied in this chip via a *separate* power-supply pin. When you create a short-circuit on that pin, the encryption progresses as normal, except that the output of all S-boxes in the last round changes to zero.

(*i*) Explain briefly the role of an S-box and the structure of a single round in DES. [4 marks]

(*ii*) How can you find K, considering that your available time and computing power will not permit you to search through more than 10^9 possible keys? [8 marks]

END OF PAPER