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.
1 Computer Design

(a) In SystemVerilog, what is the difference between:

(i) The ternary operator ? and if...then...else statements? [2 marks]

(ii) always_ff and always_comb? [2 marks]

(iii) Blocking, non-blocking and continuous assignment? [3 marks]

(iv) Logic values 0, 1, x and z and how these values propagate through Boolean logic gates? [3 marks]

(v) The way that synchronous and asynchronous reset are declared in an always_ff statement? [2 marks]

(b) The following module attempts to implement a reset control circuit that should have the following behaviour: when the user-controlled reset button (which needs to be debounced) is pressed the asyncButton signal is high and should result in the rst going high and remaining high for a minimum of $10^6$ clock cycles. rst should be generated immediately after the rising clock edge to allow time for it to propagate.

```verilog
module timeResetBad(
    input logic clk,
    input logic asyncButton,
    output logic rst);

logic ctr [18:0];
logic ctrAtMax;
always_comb begin
    ctrAtMax = &ctr;
    rst = !ctrAtMax;
end
always_ff @(posedge clk)
    ctr <= asyncButton ? 0 :
    !ctrAtMax ? ctr+1 : ctr;
endmodule
```

(i) What is wrong with the timeResetBad module? [4 marks]

(ii) Write a corrected version timeResetBad that makes minimal changes and adds no new modules. [4 marks]
2 Computer Design

(a) Describe each of the four models defined by OpenCL’s specification. [4 marks]

(b) Describe the different types of memory available to OpenCL kernels. [4 marks]

(c) Contrast how calls to a kernel, e.g. DAXPY, are invoked and grouped for execution in OpenCL compared with CUDA. [4 marks]

(d) Describe, with the aid of a diagram, how a GPU executes data-parallel kernels efficiently, including the two main pieces of hardware support. [4 marks]

(e) Describe the trade-offs between using a GPU or a specialised accelerator for tasks containing data-level parallelism. [4 marks]
3 Concurrent and Distributed Systems

(a) In the Network Time Protocol (NTP), a client (C) and a server (S) exchange (request, reply) messages to compute corrections to the time at C. Assume the time at S is always correct, and that C is synchronised to S at 13:30:00.

(i) Thirty days later the time at S is again 13:30:00 but C now believes the time to be 13:28:30. Define and compute skew and drift for C. [2 marks]

(ii) NTP estimates the offset and delay using four timestamps $(T_0, T_1, T_2, T_3)$ from a request-reply message exchange. Two such exchanges occur between C and S, producing timestamps $(310.000, 400.100, 400.102, 310.202)$ and $(311.000, 401.150, 401.160, 311.410)$ respectively, denoting all timestamps as seconds since a common fixed point. Show on a diagram the point in the message exchange at which each timestamp $T_0$...$T_3$ is taken. Give definitions for offset and delay, and compute both for each set of timestamps. Which of the two offsets you have computed would you prefer to use to adjust the time at C, and why? [5 marks]

(iii) What happens to your estimates of offset and delay if network delays are no longer symmetric? [2 marks]

(b) It is often necessary to agree only on the ordering of events, not their times.

(i) $x \rightarrow y$ indicates event $x$ happens-before event $y$. Define happens-before. Explain why it provides only a partial order on events. [2 marks]

(ii) Vector clocks can be used to implement happens-before. Give the vector clock values at each event, $a$...$g$, and explain whether each of the following relations is true or false: $b \rightarrow c$, $c \rightarrow e$, $c \rightarrow f$, $d \rightarrow g$. If false, give the relation that does hold between the given pair of events.

[8 marks]

(iii) An earlier approach used Lamport Clocks, defining $L(x)$ such that, for two events $x$ and $y$, $x \rightarrow y \Rightarrow L(x) < L(y)$ but $L(x) < L(y) \not\Rightarrow x \rightarrow y$. Explain how vector clocks resolve this issue and ensure $L(x) < L(y) \Rightarrow x \rightarrow y$. [1 mark]
4 Concurrent and Distributed Systems

(a) Programs with concurrency are vulnerable to classes of problems that are not exhibited in single-threaded programs.

(i) Explain the concepts of deadlock and livelock in a multithreaded program. [2 marks]

(ii) Explain the four conditions required for deadlock. A sentence explaining each condition is sufficient. [4 marks]

(b) Modern instruction set architectures provide instructions for performing atomic operations over memory locations.

(i) One class of instructions are generically referred to as Compare And Swap (CAS). Describe how the CAS instruction on the x86 architectures is used to perform an atomic operation. Briefly explain how the instruction uses the instruction operands when executed. [3 marks]

(ii) Databases often utilise a technique known as write-ahead logging to provide durability guarantees. Describe how a disk-based transaction log might be implemented, and what the atomic operation used in this technique is. [3 marks]

(c) The below snippet of C code uses pthreads for concurrent execution. It uses a mutex M and a condition variable C to ensure that the run_critical_code function only executes when the condition boolean is true.

L1: pthread_mutex_lock(&M);
L2: run_critical_code ();
L3: if (!condition)
L4:   pthread_cond_wait(&C, &M);
L5: if (!condition)
L6:   pthread_cond_broadcast(&C);
L7: pthread_mutex_lock(&M);

Unfortunately, there are four bugs in the code that prevent it from working correctly. List each of the four bugs and describe how each bug affects programme execution. Write down a new version of the code snippet with the four bugs corrected. [8 marks]
5 Semantics of Programming Languages

This question is about a simple functional programming language with the following syntax.

Expressions: \( e ::= x \mid \text{skip} \mid \text{fn } x : T \Rightarrow e \mid e\ e' \)

Types: \( T ::= \text{unit} \mid T \rightarrow T' \)

(a) Give rules defining a typing relation (\( \vdash \)) for this language. [5 marks]

(b) Give a brief illustration of the following concepts: free variables and closed expression. [2 marks]

(c) Give rules defining a transition relation (\( \rightarrow \)) for this language. Use the call-by-value evaluation order, and take care to say what the values are. [5 marks]

(d) State and prove a Type Progress theorem for this language. [8 marks]
Consider a language with the following abstract syntax.

\[
e ::= n \mid x := e \mid !x \mid e_1; e_2
\]

(a) Define a conventional deterministic small-step operational semantics \( (e, s) \rightarrow (e', s') \) for the language. Comment briefly on the choices you make. [5 marks]

(b) If your language is deterministic and terminating, the operational semantics implicitly defines a more abstract semantics: we can regard each expression as a function over stores \([e]\) that takes store \(s\) to the unique number \(n\) and store \(s'\) such that

\[
(e, s) \rightarrow^* (n, s') \land \exists e'' . \langle n, s' \rangle \rightarrow (e'', s'')
\]

This language is quite limited in expressiveness. Describe, as clearly and precisely as you can, the set of functions from stores to \((\text{number}, \text{store})\) pairs that are expressible as \([e]\) for some \(e\). [5 marks]

(c) The primitive contexts \(C\) for this language are expressions with a single hole:

\[
C ::= x := . \mid e_1; . \mid ; e_2
\]

Write \(C[e]\) for the expression resulting from replacing the hole in \(C\) by \(e\).

Say a binary relation \(\sim\) over expressions is a congruence if \(e \sim e'\) implies \(\forall C. C[e] \sim C[e']\).

Say a binary relation \(\sim\) over expressions respects final values if \(e \sim e'\) implies \(\forall s_0, n, n', s, s'. (\langle e, s_0 \rangle \rightarrow (n, s) \land \langle e', s_0 \rangle \rightarrow (n', s')) \Rightarrow n = n'\).

Use your characterisation of part (b) to define an equivalence relation over expressions that is a congruence and respects final values. Explain briefly why it has those properties. [4 marks]

(d) Define a terminating algorithm that, for any expressions \(e\) and \(e'\), computes whether \(e \sim e'\) or not. Explain informally why it is correct. \textit{Hint:} you may want to adapt your semantics from part (a) to compute symbolically. [6 marks]
7 Programming in C and C++

(a) Find at least 2 sources of undefined behaviour in the following program, and write a corrected version of this function. [5 marks]

```c
int main(void) {
    char *s = "abcde"; int len = strlen(s);
    for (int i = 0; i <= len; i++)
        s[i] += 1;
    return printf("'%s' is %d characters long\n", ++s, strlen(s));
}
```

(b) Restructure the program below to be more cache-efficient, giving the code and explaining your changes. [5 marks]

```c
typedef struct point { double x, y, z; } Point;

int find_max_x_argument(int n, Point *elems) {
    double max = 0; int max_index = 0;
    for (int i = 0; i < n; i++)
        if (max < elems[i].x) { max_index = i; max = elems[i].x; }
    return max_index;
}
```

(c) The following definition forms part of a legal C++ program:

```cpp
int foo() {
    MyClass x(1,2);
    MyClass y = C(3,4);
    MyClass z = x;
    MyClass t;
    z = x;
    z.f = x.f;
    return z.f;
}
```

(i) Give a declaration of `MyClass` which enables `foo` to compile and run, noting any methods or constructors in `MyClass` which are invoked when `foo` is called. [Note: Precise C++ syntax is not necessary to obtain full marks.] [4 marks]

(ii) Having seen your declaration of `MyClass`, a colleague points out some of the lines of `foo` may be redundant. Which are these? [2 marks]

(iii) Your boss now replaces your declaration of `MyClass`. Not having access to the new declaration, explain, giving reasons, which if any lines of `foo` are now redundant. [4 marks]
8 Programming in C

(a) The following function is specified to return the quotient of two integers, returning zero when the answer is undefined.

```c
#include <stdint.h>
#include <limits.h>

int64_t divide(int64_t x, int64_t y) {
    return x / y;
}
```

(i) Identify two bugs in this program.

(ii) Write a correct version of this program.

[6 marks]

(b) The `strlen` function takes a valid C string as an argument, and returns the length of the string up to and not including the first null character. An (erroneous) implementation is given below:

```c
#include <stddef.h>

size_t strlen(const char *s) {
    size_t i;
    while (s[i] >= 0)
        i++;
    return i;
}
```

(i) Find two errors in this program.

(ii) Give a correct implementation of this function.

[6 marks]

(c) Write a function with the prototype

```c
void rotate(int len, int *array, int k)
```

which rotates its input k elements to the right. E.g., if the input `array` is the array `[0, 1, 2, 3, 4, 5]`, then the call `rotate(6, array, 2)` should result in `array` being modified to `[4, 5, 0, 1, 2, 3]`. Assume the array length is passed in the `len` argument and $0 \leq k < len$.

[8 marks]
9 Further Graphics

(a) The challenge of simulation sickness (sim sickness) in Virtual Reality is a crucial hurdle in creating engaging virtual content.

(i) When it comes to avoiding sim sickness, what is the cardinal rule of VR? [1 mark]

(ii) Succinctly explain the triggers and effects of sim sickness. [2 marks]

(iii) List the constraints that sim sickness imposes on user interface design in VR. For each constraint give a one-sentence explanation of how developers must adapt to compensate. [5 marks]

(b) A mathematician tells you that a mystery shape is composed of polygons forming a closed, connected, manifold mesh without border. They claim that the mesh has 832 vertices, 1,648 edges, and 600 faces. The mathematician wants to know if it is possible that there is a loop of connected polygon edges in the mesh which, if all of those edges were cut apart, would not split the mesh into disconnected parts.

(i) If you say no, they will want to know why.

(ii) If you say that you cannot answer, they will want to know why not.

(iii) If you say yes, they will ask you why; and then they will ask you to find the greatest possible number of such loops that could simultaneously be cut in the mesh.

What do you tell them? [5 marks]

(ii) If the mathematician had instead said that the mesh had 832 vertices, 1,648 edges, and 900 faces, would your answer to their question have been different? If so, how? [2 marks]

(c) Implement a Signed Distance Field method \texttt{cylinder()} which returns the signed distance from point \(p\) to a finite cylinder segment. The cylinder should go from point \(a\) to point \(b\) with flat ends and radius \(r\).

\begin{verbatim}
float cylinder(vec3 p, vec3 a, vec3 b, float r) {
    // ...
}
\end{verbatim} [5 marks]
10 Foundations of Data Science

(a) Let $X_1, \ldots, X_n$ be independent binary random variables, $\Pr(X_i = 1) = \theta$, $\Pr(X_i = 0) = 1 - \theta$, for some unknown parameter $\theta$. Using Uniform$[0,1]$ as the prior distribution for $\theta$, find the posterior distribution. [Note: For your answer, and in answer to parts (b) and (d), give either a named distribution with its parameters, or a normalised density function.] [3 marks]

I have collected a dataset of images, and employed an Amazon Mechanical Turk worker to label them. The labels are binary, nice or nasty. To assess how accurate the worker is, I first picked 30 validation images at random, found the true label myself, and compared the worker’s label. The worker was correct on 25 and incorrect on 5.

(b) Let $\theta$ be the probability that the worker labels an image incorrectly. Using Beta$(0.1, 0.5)$ as the prior distribution for $\theta$, find the posterior. [3 marks]

I next ask the worker to label a new test image, and they tell me the image is nice. Let $z \in \{\text{nice, nasty}\}$ be the true label, and let the prior distribution for $z$ be $\Pr(\text{nice}) = 0.1$, $\Pr(\text{nasty}) = 0.9$.

(c) For both $z = \text{nice}$ and $z = \text{nasty}$, find

$$
\Pr(\text{worker says nice} | z, \theta).
$$

Hence find the posterior distribution of $(z, \theta)$. Your answer may be left as an un-normalised density function. [5 marks]

(d) Find the posterior distribution of $z$. [5 marks]

My colleague has more grant money and she can employ 3 workers to rate each image. On a test set of 30 images, she found that they all agreed on 15 images, worker 1 was the odd one out on 8 of the images, worker 2 was the odd one out on 4, and worker 3 was the odd one out on 3.

(e) Let $\theta_i$ be the probability that worker $i$ labels an image incorrectly. Find the posterior distribution of $(\theta_1, \theta_2, \theta_3)$. Your answer may be left as an un-normalised density function. [4 marks]

Hint. The Beta$(\alpha, \beta)$ distribution has mean $\alpha/(\alpha + \beta)$ and density

$$
\Pr(x) = \binom{\alpha + \beta - 1}{\alpha - 1} x^{\alpha-1}(1-x)^{\beta-1}, \quad x \in [0,1].
$$
11 Foundations of Data Science

The exam paper at Oxbridge Academy has three questions, and students are asked to choose two questions and answer them. Each question is marked out of 20. The results for four students were

<table>
<thead>
<tr>
<th></th>
<th>question 1</th>
<th>question 2</th>
<th>question 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>student 1</td>
<td>13</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>student 2</td>
<td>14</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>student 3</td>
<td>18</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>student 4</td>
<td>16</td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

The examiners are concerned that question 3 was harder than the other two questions. Consider the model

\[ X_{ij} \sim \text{Normal}(\alpha_i + \beta_j, \sigma^2) \]

where \( X_{ij} \) is the mark for student \( i \) on question \( j \). Here, \( \alpha_i \) represents the ability of student \( i \), and \( \beta_j \) represents the easiness of question \( j \).

(a) Write this as a linear model, and identify the feature vectors. [3 marks]

(b) Are the feature vectors in your model linearly independent? Justify your answer. If they are not independent, rewrite your model in a form with linearly independent feature vectors. [Hint: You should have 6 linearly independent feature vectors.] [5 marks]

(c) In order to grade the students fairly, the examiners wish to fill in the blanks in the table using predicted marks. Give pseudocode to find \( \alpha_1 + \beta_3 \), the predicted mark for student 1 on question 3, using your model from Part (b). Describe briefly any standard library routines you use in your answer. [4 marks]

(d) What is meant by parametric resampling? Explain how to use parametric resampling to generate a resampled version of this dataset. [4 marks]

(e) To find out whether question 3 is indeed harder, the examiners wish to find a confidence interval for \( \beta_3 - (\beta_1 + \beta_2)/2 \). Suggest a confidence interval, and give pseudocode to find its error probability. [4 marks]

END OF PAPER