UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2001

BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part IV
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C318=I3.4=I4.18

CUSTOM COMPUTING

Tuesday 8 May 2001, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

This page is intentionally left blank.

Section A (Use a separate answer book for this Section)

1a Consider a floating point number representation with 2 bits of unsigned mantissa and 2 bits of unsigned exponent. Write down all the values that can be represented with this representation with and without denormals.

b Show a numeric example for a computation for which denormals in (a) enable a correct solution while the representation without denormals fails.

c Consider the Residue Number System (moduli 32, 31, 15). What is the range of values that can be represented with this moduli set? Convert decimal numbers 123 and 283 to residue digits in (32,31,15). Show the computations of 123-283 and 123*283.

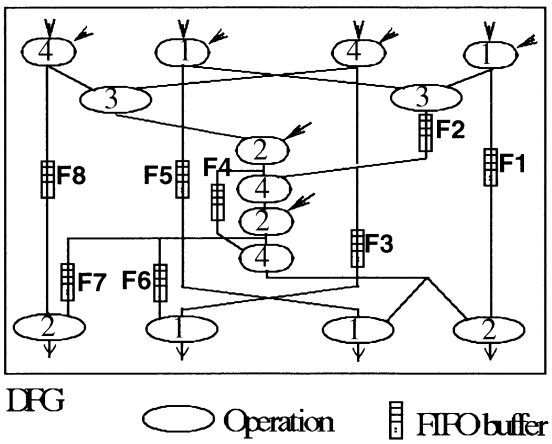
d Find the minimal number of 4-LUTs to implement a unit that counts the total number of '1's in an 8-bit binary number. Explain your design.

e Pipeline an 8-bit ripple-carry adder by inserting a flip-flop into the carry chain (between bit 3 and 4) to create a two-stage pipelined adder. Insert additional flip-flops to assure that all the output bits are produced in the same cycle. Provide a diagram of your design.

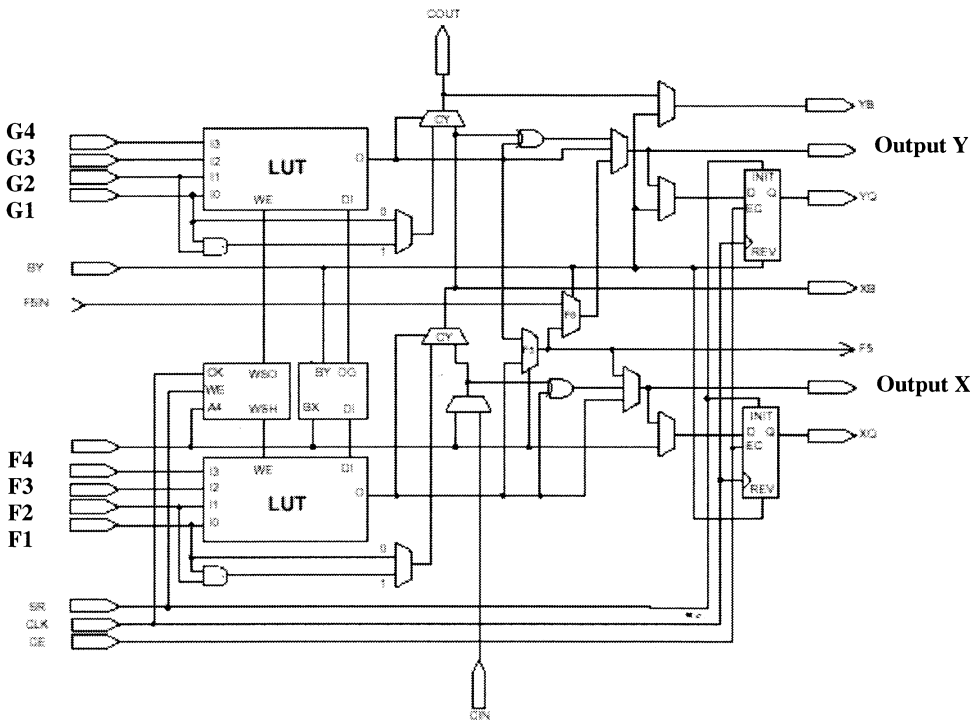
The five parts carry, respectively, 20%, 10%, 25%, 20% and 25% of the marks.

2a The "= =" operator in C compares two input values and returns a '1' if they are equal and a '0' if they are not equal. Describe how the "= ="operator and the "<=" operator can be implemented in hardware. Which of the two operators results in smaller hardware?

b Given the following dataflow-graph, and the latencies for the various nodes are shown in the graph, assign an integral number of delay cycles to each of the eight FIFO buffers (F1-F8) to guarantee correct pipelined execution.



c Given the Virtex Configurable Logic Block (CLB) in the figure below, find a mapping of an unsigned subtractor a-b = c for binary numbers, i.e. connect a[i] and b[i] to the F- inputs, and define the function implemented by the LUT.



d For each of the following two pieces of code, decide if you would implement the functionality in a systolic array, cellular automaton, or stream architecture. Explain your answer.

```
// Code A
// very large N
for (i = 0; i < N; ++I) {
      Xl = Xl ^ P[i];
      Xr = F(Xl) ^ Xr;

      temp = Xl;
      Xl = Xr;
      Xr = temp;
}
```

```
// Code B
N=640;
for (i = 0; i < N; ++i) {
     for (int j=0; j<N; j++){
       if (a[i+1][j+1]*a[i][j+1]*
           a[i+1][j]==1)
         a[i][j] = 1;
     }
}
```

The four parts carry, respectively, 30%, 20%, 30% and 20% of the marks.

Section B (Use a separate answer book for this Section)

3a The design $F_n = (\mathit{fork}; \mathsf{fst}\,\mathcal{D}; \mathit{add})^n$ is a filter that computes, given $n = 3$ and input x_t, the output $y_t = x_t + 3x_{t-1} + 3x_{t-2} + x_{t-3}$. Provide a diagram of F_3, and trace signal values to show how the design produces the above output.

b State, without proof, a theorem that can be used to fully pipeline F_n to give G_n, and compare the latency and the number of latches in F_n and G_n.

c The design $FB_{n,m}$ is the bit-level version of F_n, with each adder being implemented by an array of m fulladders *fadd* and carry propagates downwards. Express $FB_{n,m}$ in Ruby using the grid combinator.

d State, without proof, a theorem involving the trail combinator that can be used for partially pipelining the design $\mathsf{grid}_{km,kn}R$ for timeless R. Use this theorem to derive $GB_{k,n,m}$, a bit-level version of $FB_{kn,km}$, which is pipelined by every k fulladders.

e How many latches, excluding those for skewing the inputs and outputs, are there in $GB_{k,n,m}$?

The five parts carry, respectively, 20%, 20%, 20%, 25% and 15% of the marks.

4a The inductive definition for a binary-tree-shaped architecture is given by:

$$\begin{aligned} \mathsf{btree}_1 R &= R, \\ \mathsf{btree}_{n+1} R &= \mathit{group}_{2^n,2} \,;\, A \,;\, \mathsf{btree}_n R \end{aligned}$$

Provide an appropriate expression for A.

b An alternative definition for a binary-tree-shaped architecture has the same base case, while the induction case is given by:

$$\mathsf{btree}'_{n+1} R = [\mathsf{btree}'_n R, \mathsf{btree}'_n R] \,;\, R$$

For timeless R, provide an equation that can be used to fully pipeline $\mathsf{btree}'_n R$, and prove it by induction.

c A k-ary-tree-shaped architecture is given by:

$$\begin{aligned} \mathsf{ktree}_{k,1} R &= R, \\ \mathsf{ktree}_{k,n+1} R &= \mathit{group}_{k^n,k} \,;\, B \,;\, \mathsf{ktree}_{k,n} R \end{aligned}$$

Provide an appropriate expression for B.

d State, without proof, a theorem involving ktree that can be used to cluster $\mathsf{btree}_{mn} R$ as a k-ary-tree of $\mathsf{btree}_n R$. Provide a diagram of the expression involving ktree for $m = 2$ and $n = 3$.

e Use the theorem in part d to partially pipeline $\mathsf{btree}_{mn} R$.

The five parts carry, respectively, 15%, 30%, 15%, 20% and 20% of the marks.