PAPER CODE NO.	EXAMINER : **Dr Dave Jackson**
COMP303	DEPARTMENT : Computer Science Tel. No. **43678**

THE UNIVERSITY
of LIVERPOOL

JANUARY 2003 EXAMINATIONS

Bachelor of Arts : Year 3
Bachelor of Engineering : Year 2
Bachelor of Science : Year 2
Bachelor of Science : Year 3
Bachelor of Science : Year 4

COMPUTER SYSTEMS AND SRCHITECTURE

TIME ALLOWED : Two Hours and a Half

INSTRUCTIONS TO CANDIDATES

Answer **FOUR** questions

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

THE UNIVERSITY
of LIVERPOOL

Answer **four** questions

1.
(a) A computer has 16K of cache memory and 16MB of main memory, i.e. the cache is about 1000 times smaller than the main store. It might be thought, therefore, that only one in 1000 memory accesses would result in a cache hit. Explain why this is not the case.

(4 marks)

(b) What is meant by the cache hit ratio? Using R to denote that ratio, M to denote the time taken to transfer a value from main memory to cache, and C to denote the time to transfer a value from cache to the CPU, derive a formula that will compute the average data access time. If R = 0.75, M = 20 and C = 5, what is the average access time?

(7 marks)

(c) Describe what is meant by the caching strategy known as fully associative mapping. What are its advantages and disadvantages in comparison to direct mapping?

(6 marks)

(d) Our computer divides main memory into 16-byte blocks and uses 24-bit addressing. If we wish to access address abcdef (hex), what tag value should we look for, assuming fully associative mapping? Where in the cache should we look for this tag? If the required item isn't in the cache, how do we decide where to put it once it has been fetched from memory?

(8 marks)

2.
(a) Briefly explain the following terms:
(i) control store
(ii) firmware engineering
(iii) bridgeware
(iv) vertical migration

(8 marks)

(b) The diagram overleaf shows the microarchitecture of the hypothetical Mic-1 computer.
The Mic-1 microcode below implements the STOL machine code instruction, which stores a value at an address relative to the stack pointer. Explain in detail what each microinstruction does.

```
STOL: A=IR B=AMASK AND C=RA
      A=RA B=SP ADD C=RA
      A=AC B=RA LDMBR LDMAR WR
                        WR JMP START
```

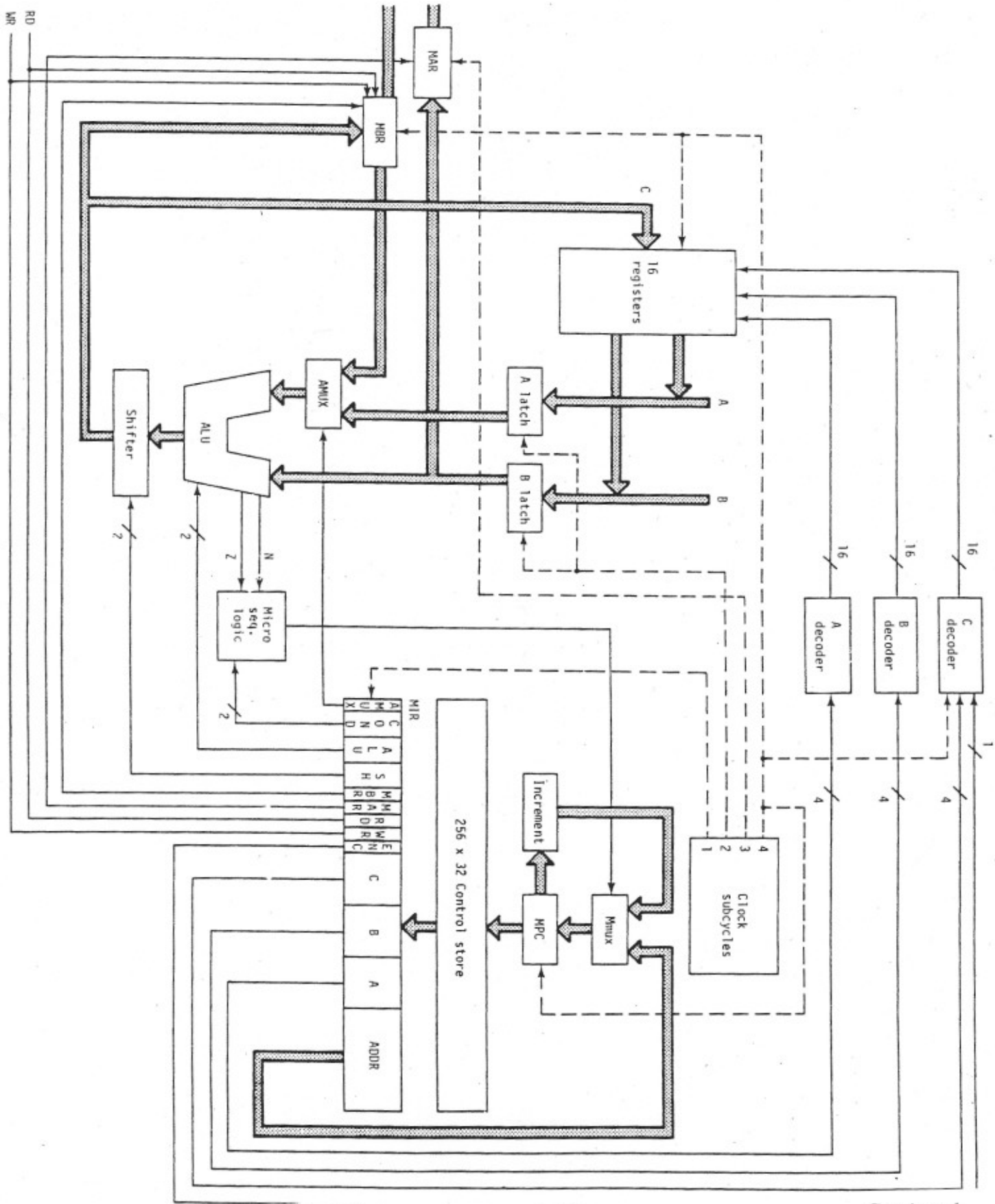
(9 marks)

(c) A machine code instruction called MUL10 has the effect of multiplying the contents of the accumulator by 10. This can be implemented in just two Mic-1 microinstructions as follows:

```
RA := AC*4
AC := (AC+RA) * 2
```

Write the Mic-1 microcode for MUL10.

(8 marks)


MR
RD
MAR
MBR
C
16 registers
A
B
A latch
B latch
AMUX
ALU
Shifter
2
2
Z
N
Micro seq. logic
2
MIR
ADDR
A
B
C
256 x 32 Control store
MPC
Mmux
Increment
Clock subcycles
1
2
3
4
A decoder
B decoder
C decoder
16
16
16
4
4
4
1

3.
(a) Explain in detail why a processor with an n-stage pipeline cannot in practice achieve an n-fold increase in execution speed.

(8 marks)

(b) A processor executes the following code:

```
inc num        // add 1 to num
load r1, num   // load num into register r1
```

What problems may arise with the execution of this sequence if the processor is pipelined?
What solutions are available?

(7 marks)

(c) Suppose the processor now encounters the instruction

```
bpl lab1       // branch if positive to lab1
```

To feed the pipeline, a decision must be made as to whether to follow the branch.
Give reasons for and against following the branch, assuming no other information is available.

(6 marks)

(d) Suppose the branch instruction in part (c) had been

```
bnz lab1       // branch if not zero to lab1
```

How might that affect the decision as to whether to follow the branch?

(4 marks)

4.
(a) What are the arguments for reducing the number of machine instructions available on computers?

(8 marks)

(b) A sub-routine definition looks as follows:

```
procedure p(int a, int b, float x)
{ …
}
```

and the following call is made:

```
p(3, 4, 7.8)
```

With the aid of suitable diagrams, describe in detail how the execution of this call on the Berkeley RISC would be made highly efficient.

(12 marks)

(c) What would happen if procedure p were highly recursive?

(5 marks)

5.
(a) Explain each of the following terms, in relation to multiprocessing systems:
(i) cache coherence
(ii) SMP
(iii) NUMA
(iv) MESI
(v) Omega network

(15 marks)

(b) Consider the following expression:

```
u*t + a*(t^2)/2
```

Draw the expression as a dataflow graph. Hence, explain how it would be executed on a computer with a dataflow architecture.
What is the main advantage of this form of execution?

(10 marks)