

Two hours

An additional answersheet is provided for Question 2.  
Please remember to complete the additional answersheet  
with your University ID number and attach it to your answerbook(s).

**UNIVERSITY OF MANCHESTER  
SCHOOL OF COMPUTER SCIENCE**

Processor Microarchitecture

Date: Tuesday 15th January 2013

Time: 14:00 - 16:00

---

**Please answer any THREE Questions from the FOUR Questions provided**

---

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are  
not programmable and do not store text.

**[PTO]**

1.

- a) Explain the difference between continuous, blocking and non-blocking assignments in the Verilog hardware description language.  
(3 marks)
- b) How do you identify blocking and non-blocking assignments in Verilog code?  
(2 marks)
- c) With the aid of a diagram discuss how blocking and non-blocking assignments are scheduled in the simulator.  
(7 marks)
- d) What is the difference between a task and a function in Verilog?  
(1 mark)
- e) A simple RISC processor has the following branch conditions specified by a 3-bit condition field in the instruction:

BAL – 000 – always branch  
 BNV – 001 – never branch  
 BEQ – 010 – branch if equal to zero  
 BNE – 011 – branch if not equal to zero  
 BCC – 100 – branch if carry clear  
 BCS – 101 – branch if carry set  
 BHI – 110 – branch if higher than  
 BLS – 111 – branch if lower than/same as

and has a status register containing the following status flags:

Z – zero flag  
 C – carry flag

Produce a Verilog function, `takebranch`, which is used to determine whether the branch should be taken or not. The inputs to the function should be the 3-bit condition field from the instruction and the 2-bit status register, [Z, C].

How would you use the function in your processor module?

(7 marks)

[PTO]

2.

- a) In a simple RISC processor there are three stages to the execution of an instruction. Identify these three stages and discuss the operation in the processor in each stage.

At what point in the instruction execution will the Program Counter (PC) be updated?

(4 marks)

- b) The STUMP processor has a different control compared to the simple RISC processor referred to in Q2a). Highlight the difference in the case of the STUMP.

(2 marks)

- c) What is special about the register r0 in the register bank of the STUMP processor? Why is it required?

(2 marks)

- d) Explain the difference between Type 1 and Type 2 instructions in the STUMP processor. Explain the additional operation that can be applied in the case of a Type 1 instruction.

(5 marks)

- e) Figure Q2.1 illustrates an architectural view of the STUMP processor.

Using the additional answer sheet provided shade the path usage for each stage of the instruction execution for the instructions at lines 4, 5 and 6 in the code sequence given in Figure Q2.2. Clearly identify the type of instruction (i.e. Type 1, 2 or 3) being processed on your answer sheet.

(7 marks)

```

1    ands r6, r6, #15
2    add r3, r3, r0
3    sub r5, r4, #2
4    add r5, r2, #8
5    st r7, [r0, r5]
6    and r4, r3, r0
7    subs r5, r5, r0
8    and r4, r4, #3

```

Figure Q2.2

**IMPORTANT:** Please do not forget to complete your additional answer sheet for question 2 with your University id number and attach it to your answer booklet.

(Question 2 continues on the following page)

(Question 2 continues from the previous page)

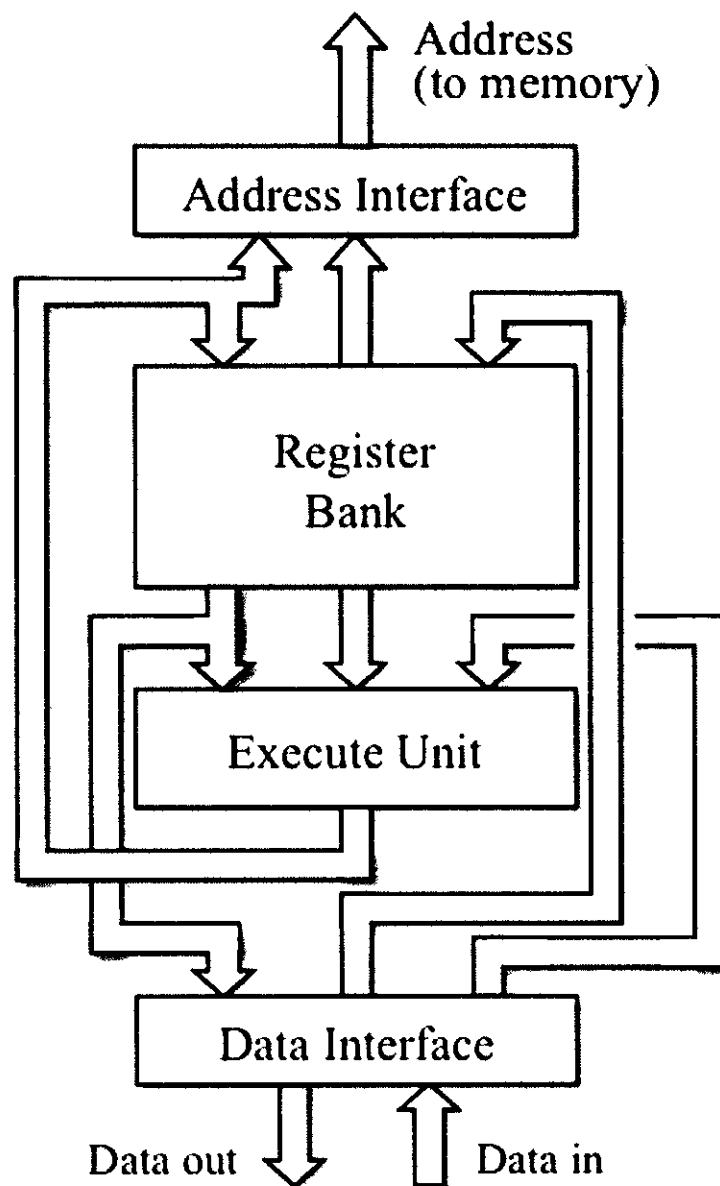


Figure Q2.1

[PTO]

3.

- a) What is meant by an Instruction Set Architecture (ISA)? How does this differ from a processor's *microarchitecture*?

(4 marks)

- b) Some ISAs use fixed-length instructions, others allow instructions of many different lengths. Compare and contrast these two philosophies highlighting a potential advantage of each.

(4 marks)

The IA-32 (x86) instruction: `ADD 0x12000(ESI), 0x1234` adds an immediate value (0x1234) to a memory location. The address of the location is the sum of an offset in the instruction (0x12000) and the contents of a processor register (ESI). The value added is 0x1234.

- c) Outline the *sequence* of operations the processor must go through to fetch and execute this instruction.

(6 marks)

The ARM processor does not have instructions like the IA-32 one given above.

- d) i) Why is this *not possible* with the ARM's 32-bit instructions?

(2 marks)

- ii) Even if such an operation was possible, why might it be *undesirable* in a modern RISC microprocessor? What might be done instead?

(4 marks)

[PTO]

4.

- a) Why is *simulation* important when preparing an ASIC design for production?  
(2 marks)
- b) Simulation may be performed at many 'levels' of the design process. State a purpose for:
  - i) Behavioural RTL simulation
  - ii) Post-layout simulation

In each case outline the class(es) of potential problems that simulations should aim to discover.

(4 marks)

- c) With both types of simulation in part b (above) some *performance* data can be obtained. What information about performance can be determined from:
  - i) Behavioural RTL simulation (2 marks)
  - ii) Post-layout simulation (2 marks)

The ASIC design will be described in Verilog, synthesised and implemented as a CMOS chip.

- d) What limits the minimum clock period? In your answer mention the output of the synthesis tool and any physical/electrical factors involved.  
(4 marks)
- e) If the minimum clock period is too long (i.e. the chip is going to be too slow) what might you do to improve performance:
  - i) in the Verilog description? (2 marks)
  - ii) by influencing the synthesis process? (2 marks)
- f) Why might the resultant ASIC not be run at its maximum possible clock frequency some, or even all, of the time?  
(2 marks)

**END OF EXAMINATION**