

PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS JANUARY 2002

SWANSEA

Computer Science

CS 113 From Language to Hardware

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University through the invigilators

CS_113
FROM LANGUAGES TO HARDWARE
(Attempt 2 questions out of 3)

Question 1.

- (a) Draw symbols for the following logic gates: AND, OR, NAND. Write down a truth table, which determines the output of these gates depending on their inputs.

[3 marks]

- (b) Draw a circuit, which determines the result of the following Boolean expression

$$x \cdot y + \bar{x} \cdot \bar{y}$$

depending on inputs x, y .

[5 marks]

- (c) Draw the configuration of an R-S-latch. Which outputs represent memory value 1? Which inputs are used in order to store this memory value in it? Describe, possibly using a table, how the S-R-latch adapts when the inputs for storing memory value 1 are applied to it. (The state of the R-S-latch before setting these input values is supposed to be unknown).

[10 marks]

- (d) Determine a Boolean formula which, depending on variables x, y, z , has result $f(x,y,z)$, where f is given by the following table:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Try to find a solution which requires as few as possible logical operations.

[7 marks]

Question 2.

- (a) Describe for the following three addressing modes, how the effective address of the operand is calculated depending on the address fields of the operand:
- (i) Direct Addressing,
 - (ii) Register indirect addressing.
 - (iii) Relative addressing.

You might make use of a picture, or describe it with the help of a formula.

[6 marks]

- (b) Explain how a stack can be implemented using registers and main memory. Fix a concrete implementation with a concrete stack size and concrete memory addresses and registers. Assume that your stack is initially empty and that the following operations are carried out *in sequence*: PUSH(2), PUSH(3), POP. Determine the content of the registers and memory locations used by your implementation initially and after the execution of each of these instructions.

[12 marks]

- (c) Translate the following high level language instruction into typical assembly language code:

```
for i=0 to 10 do b:= b+ a[i];
```

Here i and b should be integer variables and a be an array of integers, with range $0, \dots, 10$. a and b should be stored in main memory. The variable i is only used in this statement (especially its value after the loop has terminated doesn't matter), and you can therefore choose where i is to be stored. The address of b and the address of $a[0]$ should be known when writing this program. One integer should be stored as 4 bytes and addresses should refer to bytes (ie. individual bytes can be addressed). You don't need to minimize the use of registers.

[7 marks]

Question 3.

- (a) There are three main groups of bus lines. Name each of them and give a short explanation about the kind of information transmitted via them.

[6 marks]

- (b) Consider a simple architecture in which the CPU, main memory and I/O devices are connected via the system bus. What is the disadvantage of such an architecture? Describe three methods used in order to improve it.

[5 marks]

- (c) Assume an architecture, having one general purpose register AC, an instruction register (IR) and a program counter (PC). The architecture should have a word size of 2 bytes. Assume that the instruction set has an instruction LOAD A meaning “load the word at memory location A into register AC”. Assume an address length of 12 bits, and that instructions are encoded as 16 bits. The leftmost 4 bits of the instruction code determine the type of instruction (where 0x0 means LOAD) and the remaining 12 bits form the address. So 0x0700 encodes LOAD 700. Both AC and IR store 2 bytes and the PC stores 12 bits. Assume the following content of main memory

Address	Content
0x500	0x0700
0x700	0x0003

and of the registers

Register	Content
PC	0x500
AC	0x0000
IR	0x0000

Assume that the CPU is about to fetch the instruction with address given in the PC. Describe the content of the registers and of main memory after the fetch- and after the execution-cycle for this instruction. Explain the movement of data between the registers and main memory during these cycles.

[10 marks]

- (d) Assume a variable is stored at address 0x1234 and the content at this address is 0x4567. Assume that a function is called with this variable as parameter. Assume the compiler represents this function call as a subroutine call and stores the parameter on the stack. Which number are pushed as parameter on the stack, if it is a
- call by value parameter,
 - call by reference parameter?

Explain your solution. An explanation is mandatory in order to obtain marks.

[4 marks]