

Computer Systems Exam

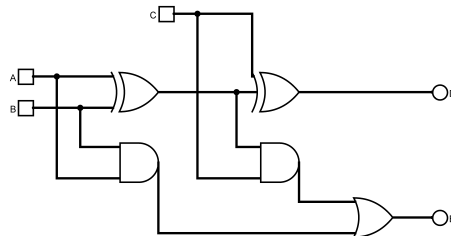
23 March 2015

This is a closed book exam: no documentation is allowed. Please make sure that your handwriting is readable and please keep the answers brief and to the point. Whenever a question states that the answer must be a single sentence, any subsequent sentences will be ignored for grading.

- Q1. **(1.2 pt)** You wrote a program in a compiled high-level language (for example C) to execute on your PC. Which steps are needed to make it run? Make a table with rows for each step and the following columns: which software/hardware performs the conversion, which language/file type is converted and into which other language/file type is it converted. A perfect answer consists of six steps.
- Q2. **(0.4 pt)** The following bit string is the Hamming encoding of some value, which has at most a single bit error. Is there such an error and, if so, in which bit? What was the original bit string before Hamming coding was applied?

0 1 1 0 0 0 1 0 0 1 0 1

- Q3. **(0.5 pt)** What does the circuit shown below do (one sentence)? Provide a truth table for the circuit.



- Q4. **(0.5 pt)** We want to make a memory cell that has 4 Gbit storage capacity (using the notation based on powers of two, so 4 Gbit = 2^{32} bits) and stores 16-bit words (it has 16 data input/output pins). The chip uses a matrix organization with RAS and CAS inputs. What is the minimal number of address lines the chip needs?
- Q5. **(0.8 pt)** Bus arbitration can be performed in various ways, including the daisy chaining approach and PCI bus arbitration. Draw two diagrams, one for each approach, showing how the request and grant lines are wired between the bus arbiter and three device controllers. Clearly label the elements, including the lines and which diagram refers to which approach. Also, for both

approaches give one example of an advantage over the other approach (one sentence each).

Q6. (1.0 pt) In the Mic1 architecture, the address of the next microinstruction (*MPC*) is selected based on the *NEXTADDR*, *JMPC*, *JAMN* and *JAMZ* fields of the microinstruction as well as the *MBR* register and the *N* and *Z* ALU outputs. For each of the four microinstruction fields (*NEXTADDR*, *JMPC*, *JAMN* and *JAMZ*), give a reason why it is needed (one sentence per field). For *MBR*, *N* and *Z* indicate how and under which conditions they are used (one sentence each).

Q7. (0.5 pt) Which IJVM instruction does the following microprogram implement?

Main1	PC = PC + 1; fetch; goto (MBR)
xxx1	MAR = SP - 1; rd
xxx2	MAR = SP
xxx3	H = MDR; wr
xxx4	MDR = TOS
xxx5	MAR = SP - 1; wr
xxx6	TOS = H; GOTO Main1

Q8. (0.9 pt) In the following x86 assembly example, which memory addresses are read from and written to and in which order? Assume that the variable *x* is stored at address 2000 and the variable *y* is stored at address 3000. Note that the JNZ instruction jumps if the flags indicate that the previous result was not zero.

```

        MOV EBX,    #1000
        MOV ECX,    #3
loop:   MOV AL,      (EBX)
        ADD AL,      x
        MOV y(ECX), AL
        ADD EBX,     #1
        SUB ECX,     #1
        JNZ loop

```

Q9. (0.5 pt) Interrupt routines have to be very careful to store all registers and flags before execution and restore them after it is done. They must also be exceptionally careful when making assumptions about the state the system is in. Why (one sentence)?

Q10. (0.5 pt) In the code example below, what problem would occur if the call to `thread_yield` were omitted (one sentence)?

```

mutex_lock:
    TSL REG, mutex
    CMP REG, #0
    JZE ok
    CALL thread_yield
    JMP mutex_lock
ok: RET

```

Q11. (1.0 pt) Your text editor is waiting for input and you type the letter 'x' on the keyboard. Describe in steps what happens from the moment you hit the key to

the moment the text editor can start processing it. Each step should indicate which information/signal is sent from which hardware/software component to which other hardware/software component (and nothing more). A perfect answer consists of 10 steps.

- Q12. **(0.6 pt)** In the “not recently used” page replacement algorithm, which two fields of the page table decide which page will be swapped out first? For both fields, which value causes the page to be swapped out early and how does this choice improve performance (one sentence each)?
- Q13. **(1.0 pt)** Assume that a particular type of harddisk has a failure rate of 10% per year and failures are independent. Failed disks are replaced by the system administrator once a year. What is the probability of failure in a year of a RAID0 configuration with two disks and a RAID1 configuration with two disks? For which two characteristics would you choose RAID0 over RAID1 (one sentence)? And for which characteristic would you choose RAID1 over RAID0 (one sentence)?

Probability theory reminder: for two independent events P and Q with probabilities p and q (both in the range $[0..1]$)

- the probability of P not happening is $1 - p$;
- the probability of P and Q both happening is $p \cdot q$;
- the probability of at least one P and Q happening is $p + q - p \cdot q$.

- Q14. **(0.6 pt)** In the lectures, it has been mentioned that a typical assembler makes two passes over the code. Under which special condition would it be possible for an assembler to make only a single pass (one sentence)? Why would it be very hard to write code under this restriction in practice (one sentence)?

— the end —