CPSC 201: Introduction to Computer Science
Carsten Schürmann
Date: March 1, 2003

Midterm Examination

Name: ______________________________

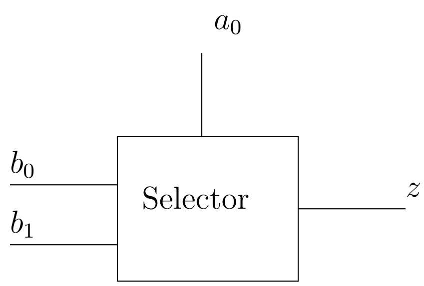
Instructions

- This is a closed book, closed notes, closed computer examination.
- There are 12 pages including 2 worksheets.
- This examination consists of 3 questions worth 100 points. The point value of each question is given with the question.
- Read each question completely before attempting to solve any part.
- Write your answers legibly in the space provided on the examination sheet. If you use the back of a sheet or a worksheet, indicate *clearly* that you have done so on the front.
- The worksheets attached to the end of this examination are for your own use; they will not be used in grading.

Problem	1	2	3	Total
Score				
Maximum	40	40	20	100

Problem 1: Boolean Logic/Circuits (40 points)

A 1-selector is a circuit that has three inputs a_0, b_0, and b_1 and one output z. If a_0 is set to 0, then the selector pushes through to z the level at b_0, otherwise that of b_1. Intuitively, you can think of this circuit to connect b_0 to z if $a = 0$ otherwise b_1.



Please answer the following four questions.

Question 1.1: (10 points)

Write out a truth table for the selector circuit.

Question 1.2: (10 points)

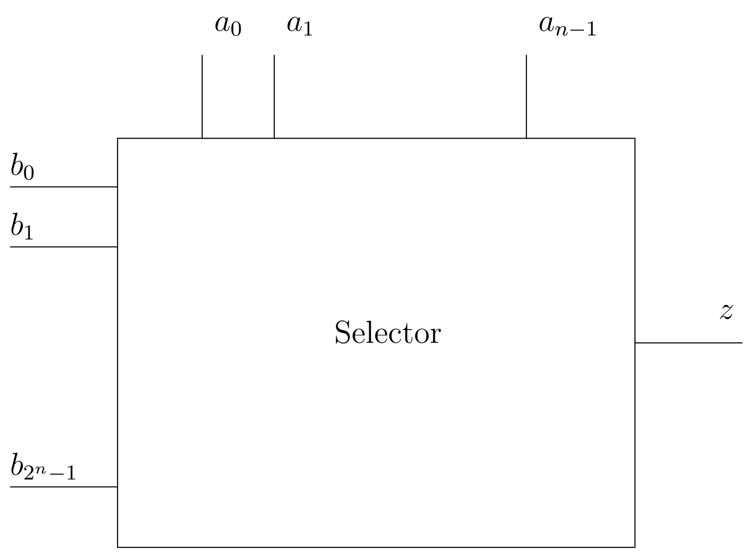
Write out Boolean expressions for z in terms of a_0, b_0,and b_1.

Question 1.3: (10 points)

Sketch the circuit that implements the 1-selector using only NOR gates.

Question 1.4: (10 points)

An n-selector is a generalization of the 1-selector, in that it has n input signals $a_0, \ldots a_{n-1}$, and 2^n input signals $b_0 \ldots b_{2^n-1}$, but still only one output line z.



Implement a 2-selector circuit exclusively in terms of 1-selector circuits. Do not use any other gates.

Problem 2: Programming in ML (40 points)

Consider the following two functions. The not so familiar `fold` function

```
fun fold f b nil = b
  | fold f b (h::l) = f (h, fold f b l)
```

and the familiar `map` function:

```
fun map f nil = nil
  | map f (h::l) = (f h) :: map f l
```

Question 2.1: (10 points)

What is the type of `fold`?

Question 2.2: (10 points)

Write out in little steps what

```
fold (fn (x, y) => x + y) 5 [3,2,1];
```

evaluates to. Use the $\stackrel{*}{\Longrightarrow}$ notation we have been using in class.

Question 2.3: (20 points)

Prove formally that for all functions `f` and `g`, for all values `b` and for all lists `L` the following hold:

```
fold f b (map g L) ≡ fold (fn (x, y) => f (g x, y)) b  L
```

Hint: The proof proceeds by induction on the structure of the list L, and $P_1 \equiv P_2$ is defined as $P_1 \stackrel{*}{\Longrightarrow} V$ iff $P_2 \stackrel{*}{\Longrightarrow} V$ for some value V.

Problem 3: Programming Machine Code (20 points)

The goal of this problem is to implement a delay function that keeps the processor busy for n seconds (where n natural number) with doing nothing. All memory related instructions, such as `LOAD`, `LOADI`, `STORE`, `STOREI` take 0.4 seconds, `JMP` and `CJMP` do not take any time at all, and all others take 0.3 seconds. You can expect the wait time preloaded in register R_1.

`LOAD` A R_i: Load the contents of memory at address A into register R_i.

`LOADI` R_i R_j : Load the contents of memory at the address contained in register R_i into register R_j.

`MOVE` R_i R_j: Move the contents of register R_i to register R_j.

`STORE` R_i A: Store the contents of register R_i into memory at address A.

`STOREI` R_i R_j : Store the contents of register R_i into memory at the address contained in register R_j.

`ADD` R_i R_j: Add contents of registers R_i and R_j and store the result in register R_0.

`MULT` R_i R_j: Multiply contents of registers R_i and R_j and store the result in register R_0.

`CONST` C R_i: Moves constant C to register R_i.

`JMP` A: Jump to memory address A and continue program execution from that location.

`CJMP` A: Jump to memory address A iff register R_0 is zero.

`OUT` R_i: Output contents of register R_i.

`HALT`: Halt the program.

Please answer the following question.

Question 3.1: (20 points)

Implement the function that computes the delay function. *Hint: You may need much less the* 19 *instructions.*

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
⋮
200 HALT

Worksheet

Worksheet