Name: Matriculation Number:

Midterm Exam
General CS II (320201)

March 23, 2009

You have one hour(sharp) for the test;
Write the solutions to the sheet.

The estimated time for solving this exam is 58 minutes, leaving you 2 minutes for revising your exam.

You can reach 30 points if you solve all problems. You will only need 27 points for a perfect score, i.e. 3 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

	To be used for grading, do not write here								
prob.	1.1	2.1	2.2	2.3	3.1	3.2	3.3	Sum	grade
total	4	4	6	6	4	4	2	30	
reached									

Good luck to all students who take this test

1 Graphs

Problem 1.1 (Planar Graphs) 4pt 10min

A graph G is called planar if G can be drawn in the plane in such a manner that edges do not cross elsewhere than vertices. The geometric realization of a planar graph gives rise to regions in the plane called faces; if G is a finite planar graph, there will be one unbounded (i.e. infinite) face, and all other faces (if there are any) will be bounded. Given a planar realization of the graph G, let $v = \#(V)$, $e = \#(E)$, and let f be the number of faces (including the unbounded face) of G's realization.

Prove or refute the Euler formula, i.e. that $v - e + f = 2$, must hold for a connected planar graph.

Solution:

Proof: Proof by induction on the number of faces

P.1 If G has only one face, it is acyclic and connected, so it is a tree and $e = v - 1$. Thus $v - e + f = 2$.

P.2 Otherwise, choose an edge e connecting two different faces of G, and remove it; e can then only appear once in the boundary of each face, so the graph remains connected – any path involving e can be replaced by a path around the other side of one of the two faces. This removal decreases both the number of faces and edges by one, and the result then holds by induction. $\square$

---

2 Combinatorial Circuits

Problem 2.1 (Two's complement conversion) 4pt 8min

Let $A = 27C$ and $B = -71$ be base 13 numbers.

1. Convert the numbers into n-bit TCN. What is the minimum n for A and B?
2. Perform the binary operations $A + B$ and $A - B$ on the TCN numbers.

Solution:

- $\varphi_{27C}(13) = \varphi_{441}(10) = 0110111001_{TCN}; n_{min} = 10$
- $-\varphi_{71}(13) = -\varphi_{92}(10) = 10100100_{TCN}; n_{min} = 8$
- $A + B = 0101011101$
- $A - B = 01000010101$ In this case, there will be overflow, so the TCN Main Theorem must be used.

Problem 2.2 (Number comparator) 6pt 10min

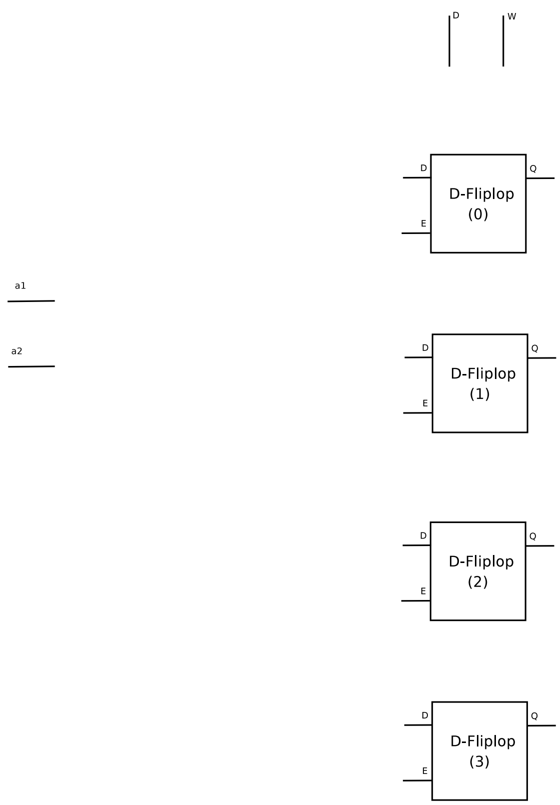
Design a combinational circuit which takes as input two n-bit numbers $\langle a_0, a_1, \cdots, a_n\rangle$ and $\langle b_0, b_1, \cdots, b_n\rangle$ and outputs also 2 n-bit numbers $\langle g_0, g_1, \cdots, g_n\rangle$ and $\langle s_0, s_1, \cdots, s_n\rangle$ representing which one is greater and which one is smaller.

Hint: Try first to design a black box which decides for the i^{th} bit of the numbers (a_i and b_i) which one belongs to the bigger number and which one to the smaller one and then show how to use several of these to solve the initial problem.

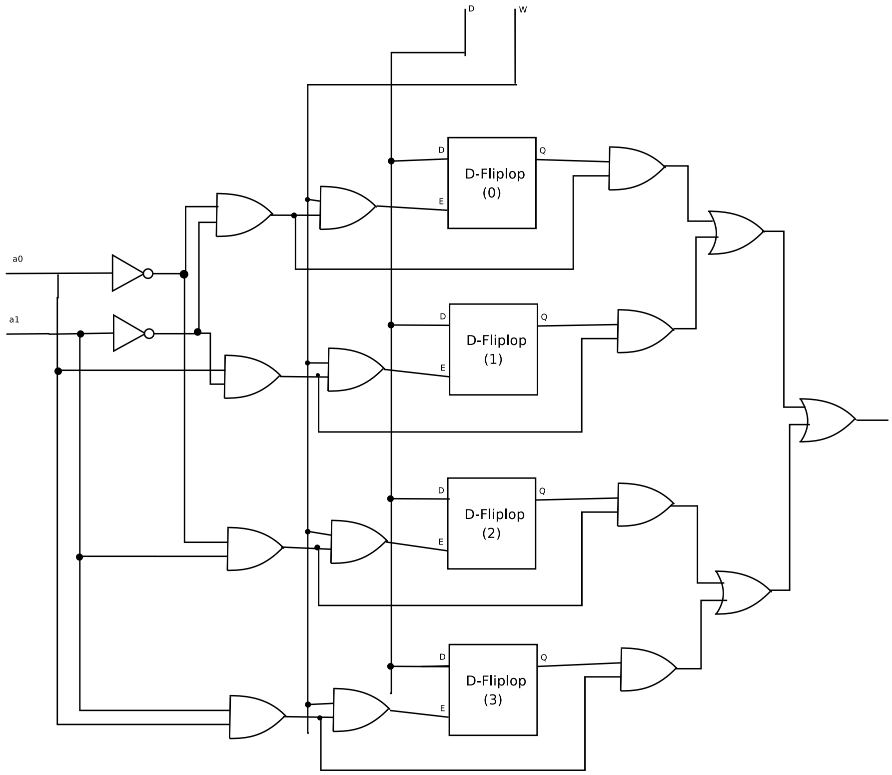
Problem 2.3 (Reading from and writing to memory) 6pt 10min

Suppose you have a 2-bit addressed memory of 4 bits managed by 4 D-Flipflops aligned as shown in the figure. The input of the circuit consists of a total of 4 bits. 2 of the bits (a_0 and a_1) provide a 2-bit address. In addition there is a data bit D and a write bit W.

Design a circuit which output should be the data memorized in the D-Flipflop addressed by $\langle a_1, a_0\rangle$. In addition if the write bit W is 1, your circuit should write the data from the data bit D to the same D-Flipflop addressed by $\langle a_1, a_0\rangle$.



Solution:


D
W
a0
a1
D
Q
E
D-Fliplop
(0)
D
Q
E
D-Fliplop
(1)
D
Q
E
D-Fliplop
(2)
D
Q
E
D-Fliplop
(3)

3 Machine Programming

Problem 3.1 (Binary to decimal) 4pt 8min

Let $P(0) = n$ contain the number of bits of a binary number stored in $P(2)...P(2+n-1)$. Each memory cell represents one bit of the number where $P(2)$ is the least significant bit and $P(2+n-1)$ is the most significant bit. Write a program that stores the corresponding decimal number in $P(1)$.

Solution:

label	instruction	comment
	`LOAD` 0	
	`MOVE` ACC $IN1$	The recursion index. Initializing $IN1 := n$
	`LOADI` 0	
	`STORE` 1	Initialize $P(1) := 0$
$\langle loop \rangle$	`MOVE` $IN1$ ACC	
	$\texttt{JUMP}_{=}$ $= \langle end \rangle$	if $IN1$ becomes 0 we are done.
	`LOAD` 1	
	`ADD` 1	
	`STORE` 1	$P(1) := 2 \cdot P(1)$
	`LOADIN` 1 1	
	`ADD` 1	
	`STORE` 1	$P(1) := P(1) + P(IN1+1)$
	`MOVE` $IN1$ ACC	
	`SUBI` 1	
	`MOVE` ACC $IN1$	$IN1--$
	`JUMP` $\langle loop \rangle$	go to next iteration.
$\langle end \rangle$	`STOP` 0	

Problem 3.2 (Multiplication and factorial) 4pt 8min

Suppose that our virtual machine does not have the `mul` instruction. Write your own static procedure that multiplies two numbers and then use it in another procedure that calculates the factorial of a number. Your procedure does not have to deal with multiplication by 0.

Solution: `proc` 2 26
`con` 1 `arg` 2 `leq` `cjp` 5 ;;$mul(a, b) =$
`arg` 1 `return` ;;if $b = 1$ then a
`con` 1 `arg` 2 `sub` `arg` 1 `call` 0 `arg` 1 `add` `return` ;;else $a + mul(a, b - 1)$

`proc` 1 25
`con` 1 `arg` 1 `leq` `cjp` 5 ;;$fact(n) =$
`con` 1 `return` ;;if $n <= 1$ then 1
`con` 1 `arg` 1 `sub` `call` 26 `arg` 1 `call` 0 `return` ;;else $mul(n, fact(n - 1))$

Problem 3.3 (Translate from $\mathcal{L}$(VM) to SW) 2pt 4min

Given the following $\mathcal{L}$(VM) program, write an equivalent SW program that does the same computation. Explain what the program would do if the stack were initialized with con (a) con (b).

```
con 3 con 4
peek 1 con 1 leq cjp (18)
peek 0 peek 0 mul poke 0
con 1 peek 1 sub poke 1
jp −21
peek 0 halt
```

Solution:

The code does 3^{2^4}, generally a^{2^b}.

```
var a:=3; var b:=4;
while 1<=b do(
    a:=a*a;
    b:=b-1;
)
return a
```
