Introduction to Digital Computers - Spring 1999

Assignment No. 1 & 2

Course homepage: http://www.eng.tau.ac.il/~guy/Digital_Computers99/dc_home.html

Firm Deadline: April 28th - before the beginning of the lecture

This assignment counts as 2 assignments due to Independence Day Holiday. (First 4 questions are assignment #1, last 4 questions are assignment #2, grades will be given separately).

1. A path in a circuit is a sequence of gates $G_0, G_1, \ldots, G_p$ such that an input of gate G_{i+1} is fed by an output of gate G_i. A cycle is a closed path, namely, a path in which $G_0 = G_p$. Prove that a combinational circuit does not contain cycles.

2. Write (on paper) an algorithm that is given as input a circuit and outputs whether it is a combinational circuit. If the circuit is a combinational circuit, then the algorithm outputs the linear order of the gates in the circuit $G_1, G_2, \ldots$ (according to the definition of a combinational circuit).

   Prove the correctness of your algorithm.

3. Give an example in which the propagation delay computed for a combinational circuit is only an upper bound on the delay and the actual delay might be much smaller. Describe the assumptions you rely on (e.g. the output of an AND-gate is zero if one of its inputs is zero, etc.).

4. Prove the correctness of the decoder presented in class.

5. Prove that all OR-tree(n) circuits have the same cost. We assume that: (a) every input is connected to exactly one input of an OR-gate; and (b) only OR-gates are used and they are connected in a tree structure.

6. Prove that if an OR-tree(n) is constructed recursively by using balanced partitions, then the delay of the tree is minimized. Formally, we construct an OR-tree(n) denoted by T by taking two trees T_1 and T_2 and connecting the outputs of two trees by an OR-gate as follows: T_1 is an OR-tree($\lfloor n/2 \rfloor$) and T_2 is an OR-tree($n - \lfloor n/2 \rfloor$). The outputs of T_1 and T_2 are fed into an OR-gate to produce the output of T. Prove that for every OR-tree(n) T' the following holds:

$$\mathit{delay}(T) \leq \mathit{delay}(T')$$

7. Can you think of a different way which also constructs minimum delay OR-trees? How many minimum delay trees are there if n is a power of 2?

8. Write all the associative Boolean functions $f : \{0,1\}^2 \rightarrow \{0,1\}$. A Boolean function $f : \{0,1\}^2 \rightarrow \{0,1\}$ is associative if

$$\forall a, b, c \in \{0,1\} \; : \; f(f(a,b),c) = f(a, f(b,c))$$