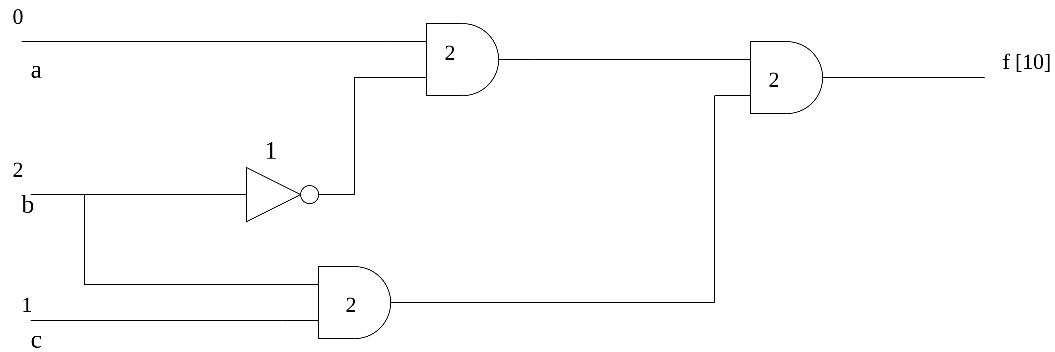
CAD of Digital Circuits — Physical Design

Spring 2014, Homework # 4
Due Date: Wed, April 30 (by 5pm, my office, or PDF by email)

1) (35 points) Consider the circuit shown below. Given the input arrival times at primary inputs a, b, c, gate delays and the output required time at $f = 10$, perform static timing analysis on the circuit to derive arrival time, required time and slack at each node. Subsequently, apply the *zero slack algorithm* to derive timing budgets on each net. Show your work.



2) *Quadratic placement:* (65 points) This assignment will give you some experience with quadratic netlength cost based placement techniques. Consider the following circuit:

Input G_0, G_1, G_2, G_3;

Output G_{17};

$G_5 = BUF(G_{10})$;

$G_6 = BUF(G_{11})$;

$G_7 = BUF(G_{13})$;

$G_{14} = NOT(G_0)$;

$G_{17} = NOT(G_{11})$;

$G_8 = AND(G_{14}, G_6)$;

$G_{15} = OR(G_{12}, G_8)$;

$G_{16} = OR(G_3, G_8)$;

$G_9 = NAND(G_{16}, G_{15})$;

$G_{10} = NOR(G_{14}, G_{11})$;

$G_{11} = NOR(G_5, G_9)$;

$G_{12} = NOR(G_1, G_7)$;

$G_{13} = NOR(G_2, G_{12})$;

These 17 nodes (including inputs and output) have to be placed on a layout area of 5×5 square units. Considering the bottom left corner of the layout as the "origin", the inputs and output pads are already placed at fixed locations: $G_0 = (0.5, 1.5), G_1 = (0.5, 3.5), G_2 = (1.5, 4.5), G_3 = (4.5, 2.5), G_{17} = (3.5, 0.5)$. Each cell/module is 1×1 in size, and you may assume its centerpoint (x_i, y_i) as the "pin location" of all it's nets. You are asked to solve the placement problem using the quadratic net-length cost model that we have studied in class.

a) (Unconstrained QP) Based on the quadratic netlength model (Sec 4.3.2, pp. 110 in the textbook), setup and solve the placement problem. If needed, you may break multi-pin nets into 2-pin nets. Your solution may require some legalization due to cell overlap. Legalize the placement using any approach (ad hoc, manual, algorithmic) and identify the longest and total netlength.

b) (QP constrained by partitioning) Now, we will further constrain the quadratic optimization subject to netlist partitioning based on the concepts we studied in class. Perform a bi-partitioning on the netlist (respect the fixed IO pad locations!), and constrain the quadratic program w.r.t. the center-of-gravity location of the left and right partition. [Refer to the paper uploaded on the class website on 4/14 titled "A path-based timing driven quadratic placement algorithm", particularly Section II]. Solve the resulting program. Does this formulation produce a different result than the unconstrained one? How does the quality of the result (in terms of worst/longest and average netlength) compare viz-a-viz the previous case?

c) You may want to make use of the Gurobi tool to solve the quadratic optimization problem. However, note that these are convex quadratic optimization problems with equality constraints, so one can easily solve them using a linear solver too. Your choice!