University of Toronto, Faculty of Applied Science and Engineering
Department of Electrical and Computer Engineering

ECE 1387 - CAD for Digital Circuit Synthesis and Layout

Exercise #2 - Floorplanning Using Mixed Integer Linear Programming (MILP)

Spring 2009 J. Anderson

Assignment Date: March 27, 2009
Due Date: April 10, 2009 (by email to instructor)
Late Penalty: **-1 mark per day late, with total marks available = 10**

Linear programming is an optimization approach wherein the objective function is linear function of a set of variables and there are linear equality and inequality constraints on the variables. Linear programming has many applications in CAD (and also in finance and operations research). In this exercise, you will gain familiarity with linear programming by applying mixed integer linear programming to solve the floorplanning problem. "Mixed" means that some of your variables will be real-valued and some will be integer/binary variables. In general, MILP is NP-hard, however, in this exercise, problem instances will be sufficiently small.

For this exercise, use the `lpsolve` solver to solve your linear program. `lpsolve` is a popular solver that can be called as a binary or can be linked directly into C code. You may find it useful in your own research. Documentation is available on-line at: http://lpsolve.sourceforge.net/5.5/ You may download your own copy of `lpsolve` or you may use the Linux binary in my area on EECG at: `~janders/lpsolve`

Use the `LP` file format to express your linear program, as described here: http://lpsolve.sourceforge.net/5.5/lp-format.htm.

On the course webpage, you will find three test files containing floorplan problem instances. An example test file is:

```
10 10
-1
A 2 3
B 1 2
C 4 2
-1
N1 A B
N2 B C
-1
```

The first line contains the chip X and Y dimensions. All of the objects in your floorplan must be placed entirely within these dimensions, e.g. from 0,0 to 10,10 in the above example. The next line contains a -1 by itself.

The next set of lines defines a set of modules to be placed. There is one line per module. In the example, there are three modules. Each line specifies the module name, followed by the module width in the X-dimension followed by the module height in the Y-dimension. For example, module "A" is 2 units wide and 3 units high. Modules have fixed dimensions and should not be rotated. The list of modules is terminated by a -1 by itself on a line.

The last set of lines defines nets between modules. There is one line per net. In this example, there are two nets. Each net specifies a net name, followed by the names of modules on the nets. For example, net "N1" connects modules A and B. Note that nets may have more than two modules, i.e. there may be multi-fanout nets. A -1 by itself on a line terminates the test file.

What to do and what to hand-in?

1. Formulate the floorplanning problem as a linear program in the `LP` file format. Use `lpsolve` to solve the linear program. Your objective function should be the half-perimeter bounding box wirelength. Your linear program must ensure that no modules overlap with one another and that all modules are placed within the die area. Your linear program should produce a legal floorplan with the optimal half-perimeter bounding box wirelenth.
2. Hand-in a table showing the bounding box wirelength for each test file. Note that your linear program should assume that pin positions are at the centre of each module. Half-perimeter bounding box wirelength should also be computed as though pins are at module centres. The HPWL for the net in the figure below is: $|(4.5 - 1.5)| + |1 - 0.5| = 3.5$.
3. Hand-in the location, on the `EECG` or `ECF` network, of your `LP` files that you input to `lpsolve`. Please ensure that all directories are readable by me.
4. Hand-in the location, on the `EECG` or `ECF` network, your program/script that reads the input test files and produces the `LP` file format.

NOTES:

- Do not manually write your linear program in the `LP` file format. You need to write a script or a C program to parse the input files for this exercise and produce linear programs in the `LP` file format readable/solvable by `lpsolve`. For my implementation, I used a PERL script, however, you are free to use C or any other scripting language.
- **HINT:** See the Kim et al. paper on the course website published in IEEE Trans. on CAD in May 2003. You should not use the sequence pair floorplan representation, as in the paper. Your linear program will be considerably simplified vs. Kim, as you only have to deal with hard-shaped modules in a fixed die area.
- **SUGGESTION:** You may wish to plot the floorplan output by `lpsolve` to ensure it is legal and that you have formulated the linear program correctly.