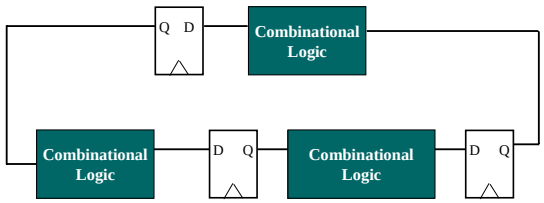
CprE / ComS 583 Reconfigurable Computing

Prof. Joseph Zambreno
Department of Electrical and Computer Engineering
Iowa State University

Lecture #14 – Placement and Routing

Register Transfer-Level Design

- A register-transfer machine has combinational logic connecting registers:



FPGA Design Flow with Test

Design and implement a simple unit permitting to speed up encryption with RC5-similar cipher with fixed key set on 8031 microcontroller. Unlike in the experiment 5, this time your unit has to be able to perform an encryption algorithm by itself, executing 32 rounds…..

Specification

```
Library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity RC5_core is
        port(
                clock, reset, encr_decr: in std_logic;
                data_input: in std_logic_vector(31 downto 0);
                data_output: out std_logic_vector(31 downto 0);
                out_full: in std_logic;
                key_input: in std_logic_vector(31 downto 0);
                key_read: out std_logic;
        );
end RC5_core;
```

VHDL description

Functional simulation

Synthesized Circuit

Post-synthesis simulation

FPGA Design Flow with Test (cont.)

Synthesized Circuit

Post-synthesis simulation

Implementation

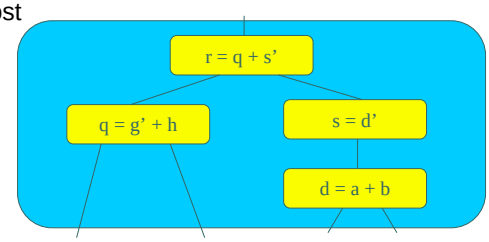
Timing simulation

Configuration

On chip testing

LUT-based Logic Synthesis

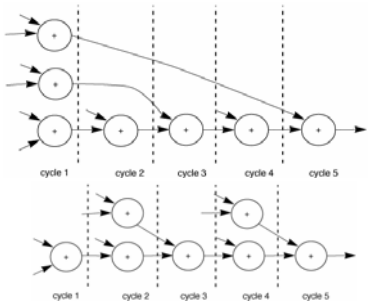
- Cost metric for static gates is literal
  - ax + bx' has four literals, requires 8 transistors
- Cost metric for FPGAs is logic element
- All functions that fit in an LE have the same cost



Allocation and Scheduling

ASAP

ALAP



Outline

- Recap
- Placement and Routing Metrics
- FPGA Placement Techniques
  - Iterative partitioning
  - Simulated annealing
- FPGA Routing Techniques

Placement and Routing

- Two critical phases of layout design:
  - **Placement** of components on the chip
  - **Routing** of wires between components
- Placement and routing interact, but separating layout design into phases helps us understand the problem and find good solutions

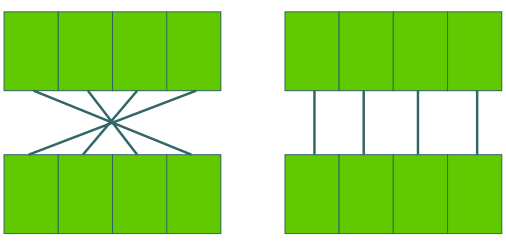
Placement Metrics

- Quality metrics for layout:
  - Area
  - Delay
  - Power
- Area and delay determined partly by wiring
- How do we judge a placement without wiring?
  - Estimate wire length without actually performing routing
- Design time may be important for FPGAs

FPGA Issues

- Often want a fast answer
  - May be willing to accept lower quality result for less place/route time
- May be interested in knowing wirability without needing the final configuration
- Fast placement: constructive placement, iterative improvement through simulated annealing

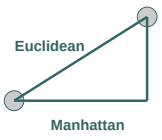
Wire Length as a Quality Metric



Bad Placement **Good Placement**

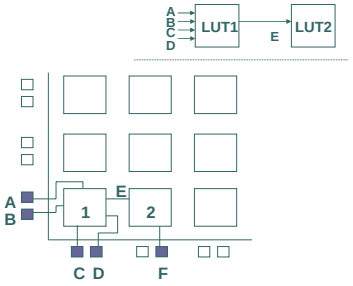
Wire Length Measures

- Estimate wire length by distance between components
- Possible distance measures:
  - Euclidean distance (sqrt($x^2 + y^2$))
  - Manhattan distance (x + y)
- Multi-point nets must be broken up into trees for good estimates



Placement

- Placement has a set of competing goals
- Can't optimize locally and globally simultaneously
- Use heuristic approaches to evaluate quality



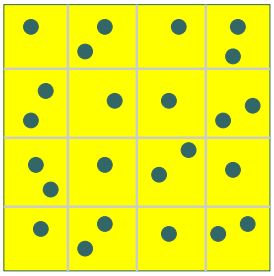
Placement Techniques

- Can construct an initial solution, improve an existing solution
- Pairwise interchange is a simple improvement metric:
  - Interchange a pair, keep the swap if it helps wire length
  - Some heuristic determines which two components to swap

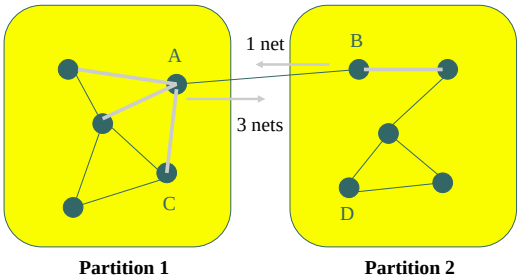
Placement by Partitioning

- Works well for components of fairly uniform size
- Partition netlist to minimize total wire length using **min-cut** criterion
- Partitioning may be interpreted as 1-D or 2-D layout

Recursive Partitioning



Min-Cut Bisecting Partitioning



Min-Cut Partitioning (cont.)

- Swapping A and B:
  - B drags 1 net
  - A drags 3 nets
  - total cut increase: 3 nets
- Conclusion: probably not a good swap, but must be compared with other pairs

Kernighan-Lin Algorithm

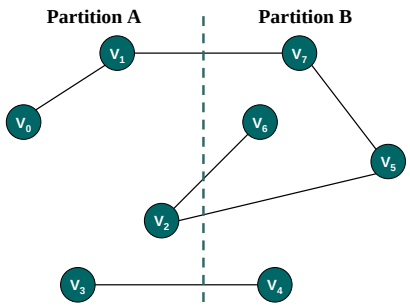
- Compute min cut criterion:
  - Count total net cut change
- Algorithm exchanges sets of nodes to perform hill-climbing—finding improvements where no single swap will improve the cut
- Recursively subdivide to determine placement detail

Kernighan-Lin Algorithm (cont.)

1. Make an initial guess partition into two subsets of equal sizes, and unlock all the vertices in the graph
2. Associate a cost D with every vertex i, where D(i) = E(i) – I(i)
   - I(i) is the number of edges that do not cross the bisection boundary
   - E(i) is the number of edges that cross the boundary
3. Calculate the gain G for all possible swaps between unlocked vertices a_i and b_i
   - $G(a_i,b_i) = D(a_i) + D(b_i) – 2C(a_i, b_i)$
   - $C(a_i, b_i)$ is the weight of the edge between (a_i,b_i)
4. Make the swap for the max G, and lock the nodes
5. Iterate until no more swaps can be made

In-Class Exercise

- Step through Kernighan-Lin on the following circuit:



Simulated Annealing

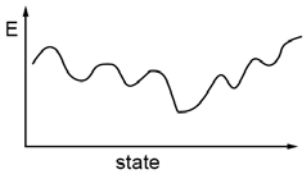
- Powerful but CPU-intensive optimization technique
- Analogy to annealing of metals:
  - Temperature determines probability of a component jumping position
  - Probabilistically accept moves
  - Start at high temperature, cool to lower temperature to try to reach good placement

Physical Annealing Analogy

- Take a metal and heat to high temperature
- Allow it to cool slowly; metal is annealed to a low temperature
- Atoms in the metal are at lower energy states after annealing
- Higher the temperature initially and slower the cooling, the tougher the metal becomes
- Atoms transition to high energy states and then move to low energy

Simulated Annealing (cont.)

- Generate random moves
  - Initially, accept moves that decrease and increase cost
  - Local minimum versus global minimum
- As temperature decreases, the probability of accepting bad moves decreases
- Eventually, default to greedy algorithm



Annealing Wrap-up

- Big-hammer for hard optimization problems
- General cost model - accommodates most any constraints
- If cool slowly enough, will get good results
- Finesse in working out parameters
- Cost should be cheap to update
- Annealing schedule can be tricky to optimize (balance speed versus quality)
- ...generally takes a long time... (...why PPR is slow)

Imagine … (Routing)

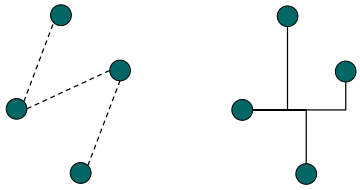
- You have to plan transportation (i.e. roads and highways) for a new city the size of Chicago
- Many dwellings need direct roads that can't be used by anyone else
- You can affect the layout of houses and neighborhoods but the architects and planners will complain
- And … you're told that the time along any path can't be longer than a fixed amount
- What are some of your considerations?

Some Considerations

- How many levels do the roads need to go?
  - Remember: higher is more expensive
- How to avoid congestion?
- What basic structure do I want for my roads?
  - Manhattan?
  - Chicago?
  - Boston?
- Automated routing tools have to solve problems of comparable complexity on every leading-edge chip

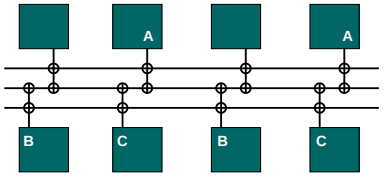
Routing Sub-Problems

- Shortest Path (two-pin nets – $O(N_3)$)
- Steiner Tree (easy for *n*-pin where *n* <= 5; NP-complete in general)
- Compatibility (NP-complete)



Routing Compatibility

- Example: satisfy three simultaneous net connections (A–A, B–B, C–C)
- A–A cannot use middle track
- Greedy approach will not be sufficient

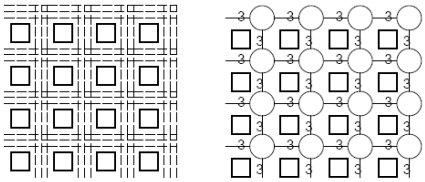


Standard Approach

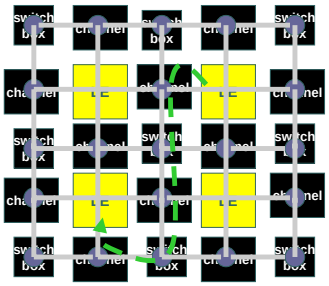
- Major phases in routing:
  - **Global routing** assigns nets to routing areas
  - **Detailed routing** designs the routing areas
  - One phase routers – channel assignment and wire selection happens in one routing pass
- Two phase routers were initially popular
  - Simpler to write and faster to execute
  - More closely models ASIC routing techniques
- One phase routers shown to give MUCH better performance
- Net ordering is a major problem
  - Order in which nets are routed determines quality of result
  - Net ordering is a heuristic

Global routing

- Choose a sequence of channels
  - Not tracks within a channel
- Must take capacity into account
- Channel graph allows path algorithms to be used for global routing

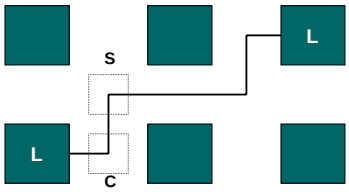


Channel Graph

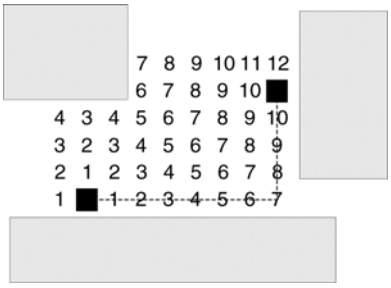


Maze Routing

- Will find shortest path for a single wire, if such a path exists
- Two phases:
  - Label nodes with distance, radiating from source
  - Use distances to trace from sink to source, choosing a path that always decreases distance to source



Lee (Wavefront) Router



Summary

- Placement
  - Placement and clustering of modules critically important for subsequent routing step
  - Often initial placement performed and then iteratively improved
  - Mincut partitioning approaches sometimes used for initial placement
  - Can benefit from simulated annealing approaches, given an accurate cost function
- Routing
  - Routing a difficult problem based on device size, complexity
  - Hard part of routing is the compatibility problem
  - Can be attacked using iterative or simulated annealing approaches