Optimization of Routing Algorithms

Summer Science Research

By: Kumar Chheda
Research Mentor: Dr. Sean McCulloch

What is optimization?

- Program optimization
  Process of removing or replacing parts of an existing program that make it slower, time intensive and space consuming.
- A program can be optimized by:
  - Changing data structures and other parts of the code to make it smaller and faster.
  - Modifying the algorithm or coming up with a better one.

What is routing?

- Routing:
  Process of finding the best route to reach or interconnect several given objects.
- Several logic gates on a chip need to be connected in an efficient way so that
  - All such sets of interconnected logic gates can be feasibly connected.
  - The placement and the number (channel width) of the wires is optimum.

Background Information

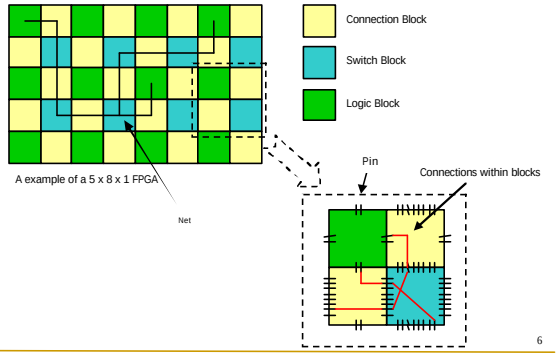
- Dr. Sean McCulloch's research thesis:
  - University of Virginia
  - Auction based routing in Field Programmable Gate Arrays (FPGAs).
- Quark – an auction-based router
- On-going project, currently in its 2nd year at Ohio Wesleyan University.

Understanding Quark

Basic Components of FPGA as viewed by 'Quark':

- Auction Pin
  - A pin is defined as a wire connecting two blocks in a FPGA.
  - A auction pin is the abstraction of the pin in 'Quark'. It is the object which all nets bid on to realize their path.
- Auction Block
  - Allows restrictive connections between wires.
  - The pins described above connect these blocks.
- Auction Net
  - A net is a path between the set of blocks (usually Logic Blocks).
  - An auction net is an abstraction of a regular net in 'Quark', and plays the main role in this algorithm.

Understanding Quark



Different Personalities

- A personality defines a bidding strategy which the nets can use
- 3 Main Personalities
  - Baseline
  - Split
  - Focused

Tasks for each personality

- Each personality is responsible for the following:
  - Place initial bids on pins needed
  - Bidding during the auction
  - Implement various strategies when a net is losing a critical pin required for its routing.
- These issues can be handled differently and hence every personality is defined by the set of rules it uses to perform them.

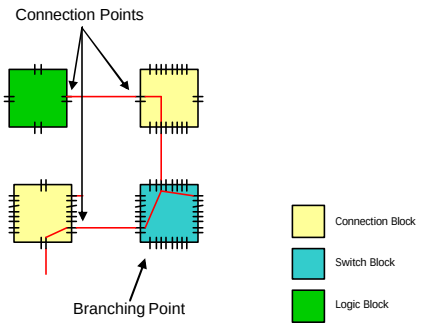
Focused Personality

- Most complicated of all personalities
- Difficult to implement
- Lot of areas where focused personality can be improved by inserting additional algorithms to cases differently.

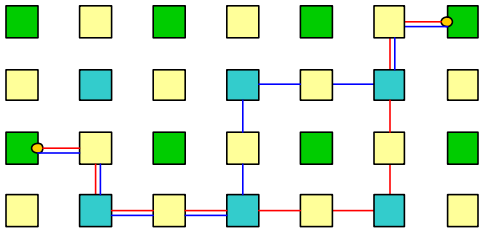
Focused Personality

- Strategy: “Focus attention on a set of pins that are important to win.”
- We identify two kinds of objects
  - Branching Points: On a multi-terminal net, branching points are the ones that have more than two edges incident on them.
    - Bid heavily on the branching points to win them.
  - Connection Path: Everything else.
    - Fill in this part of the path however it can be filled.
    - The length of the path does not matter as long as it fits the budget constraints

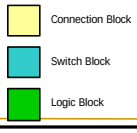
Focused Personality



Connection Paths



The red and the blue paths are just two of the many ways to obtain a connection path between two pins. This gives us flexibility because we can choose a cheaper path to make sure we can use the money saved to bid for expensive branching points.



Major Problem with Focused Personality

- Bidding wars between two nets
  - War for a branching point

ex: Net 5 wins BP A.

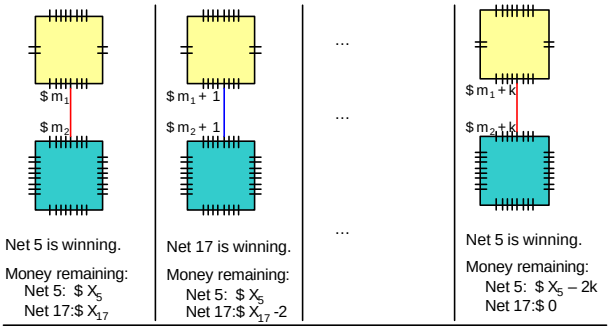
Net 17 also wants BP A, bids more & wins.

Net 5 needs A and is currently losing. Net 5 frees up some more money & bids.

This goes on forever.

(until some net runs out of money).

Bidding War (example)



Net 5 is winning.

Money remaining:
Net 5: $ X_5
Net 17:$ X_{17}

Net 17 is winning.

Money remaining:
Net 5: $ X_5
Net 17:$ X_{17} -2

Net 5 is winning.

Money remaining:
Net 5: $ X_5 – 2k
Net 17:$ 0

Net 5 ($ X_5 at start) Net 17 ($ X_{17} at start) $X_5 = X_5

One solution

- Mechanism to detect such bidding war
  - Eminent Domain
    - If net P needs a point A but ran out of money to bid for it ⟶ Give it a one-time grant.
    - If later on net P loses point A to some net Q and needs it back, but can't free up enough money ⟶ P goes to eminent domain. Eminent Domain realizes a bidding war and quits!
    - Eminent Domain is so useful a concept that it is provided for all other nets as last resort as well.

Another thought

- While eminent domain is important, we can only call it once per net.
- So we think of a different strategy.
- Eminent domain still stays as last resort.
- However, the net should have some way to abandon an overly expensive pin and look for other cheaper pins.

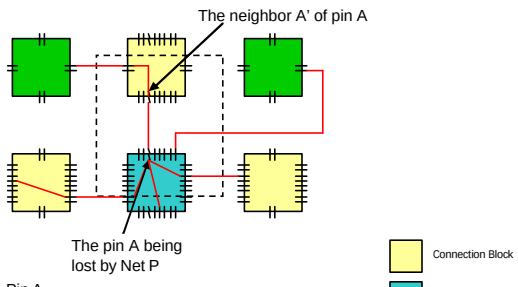
Solution

- So when a net P runs out of money, it tries to:
  - restructure and re-route
  - ask for a grant from eminent domain only if it fails.
- We need to shift the originally preset branching point.
  - i.e. If a net needs a branching point and is losing an important pin even after using up all its money, it should try to find nearby branching points and re-route.

Continued…

- The process is complicated by the fact that all pins are inter-connected. Changing one pin on a block requires changing quite a few pins on neighboring blocks and thus the need to find new pins that would minimize this change.

Routing before the new algorithm

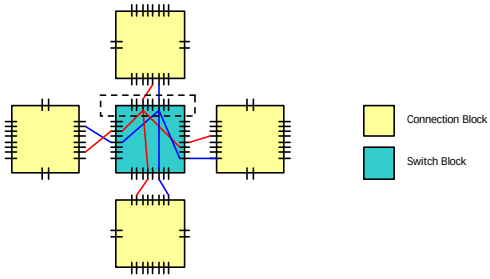


Bid on M_A Pin A

Bid $M_{A'}$ on Pin A'

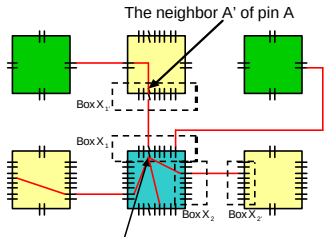
Total Bid on losing pins = $(M_A + M_{A'})$

Inner Workings of a Switch Block



Inside a switch block, there are only a limited number of legal connections. Hence changing a pin on one side may require us to change the pins that are being used on other sides and their neighboring pins on connection blocks as well.

Cascading effect



Need to find a pin within the box X_1 whose neighbor is also within box $X_{1'}$

Need to find a pin within the box X_2 which is connected to pin A and whose neighbor is also within box $X_{2'}$ and so on for every side…

Bid on M_A Pin A

Bid $M_{A'}$ on Pin A'

……

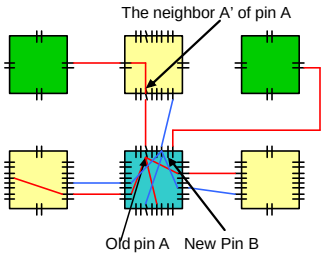
Total Bid on losing pins = $(M_A + M_{A'} + …)

Connection Block

Switch Block

Logic Block

Need a new pin like pin B



Freed M_A Pin J

Freed $M_{A'}$ on Pin J'

……

Total money freed = $(M_A + M_{A'} + …)

Connection Block

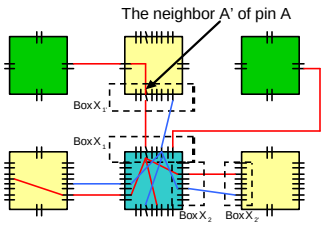
Switch Block

Logic Block

Need a new pin like pin B

- After finding an alternate pin B on the same side as A that connects to other pins that are similar to the pins that A connects to in terms of their side as well as their neighbors, we need to make sure that the money required to win all those pins is at most as much as we can spare by replacing the older pins.

Need a new pin like pin B



Connection Block

Switch Block

Logic Block

Freed M_A Pin J

Freed $M_{A'}$ on Pin J'

……

Total money freed = $(M_A + M_{A'} + …)

=

Bid M_B Pin J

Bid $M_{B'}$ on Pin J'

……

Total money bidded = $(M_B + M_{B'} + …)

If we find a new pin…

- … on the same side as pin A with all the above characteristics, we continue trying to find all such sets of pins possible.
- This helps us to compare all the newly found sets of pins price-wise.
- We pick the set of pins that is going to cost the cheapest to the net without changing too much.

Record Keeping

- After picking a new set of pins, its time to replace.
- It is very important that we cancel our bids from all the pins that are going to be replaced.
- The money gets returned to total pool and only then it is possible to bid on the new pins.

Results & Problems

- The focused personality now goes much further in evaluation and creation of a new routing.
- In the process, we found other areas where work can be focused in the future.
  - Handling of data structures needs to be re-thought. Under certain cases, some branching points can also be connection points, thus leading to problems.

Future Work

- Optimizing focused personality further by adding the capability to find newer pins on different blocks.
- Building new data structures for optimum handling of branching and connection points.
- Building new faster personalities.
- Implementation of opportunity cost model.
- Adaptation of “game-theoretic” models of economics to an implementation of Quark.

Acknowledgements

Many thanks to:

Dr. Sean McCulloch
Summer Science Research Program, OWU
The Dept. of Mathematics & Computer Science
My family and friends