
ECE 636

Reconfigurable Computing

Lecture 5

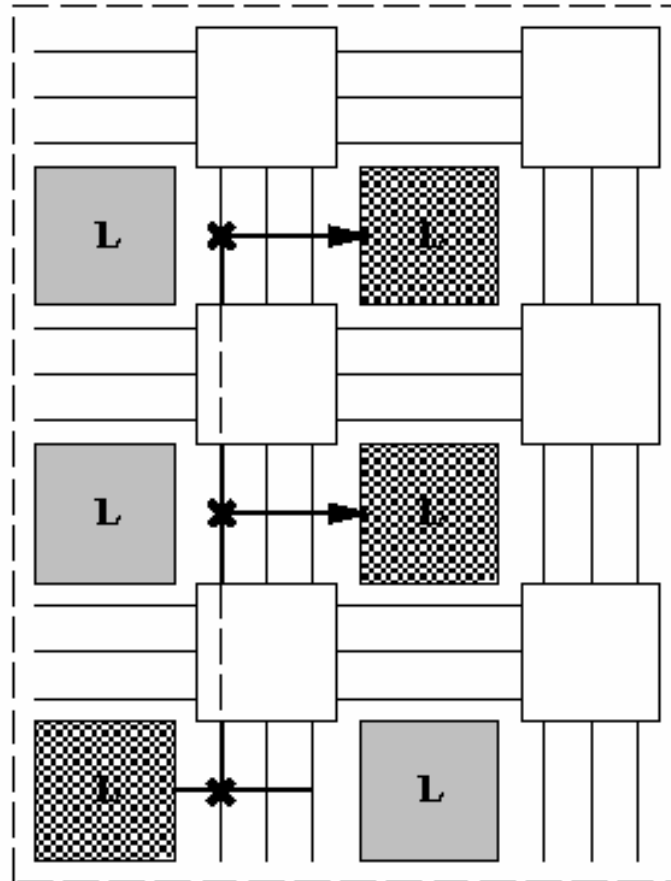
FPGA Routing



Overview

- **Routability Estimation**
- **1 step routing**
- **2 step routing**
- **Pathfinder (and Pathfinder derivatives)**

Island-Style Devices



- Two dimensional problem
- $(X+Y)!/(X!Y!)$ possible paths
- Restricted within bounding box

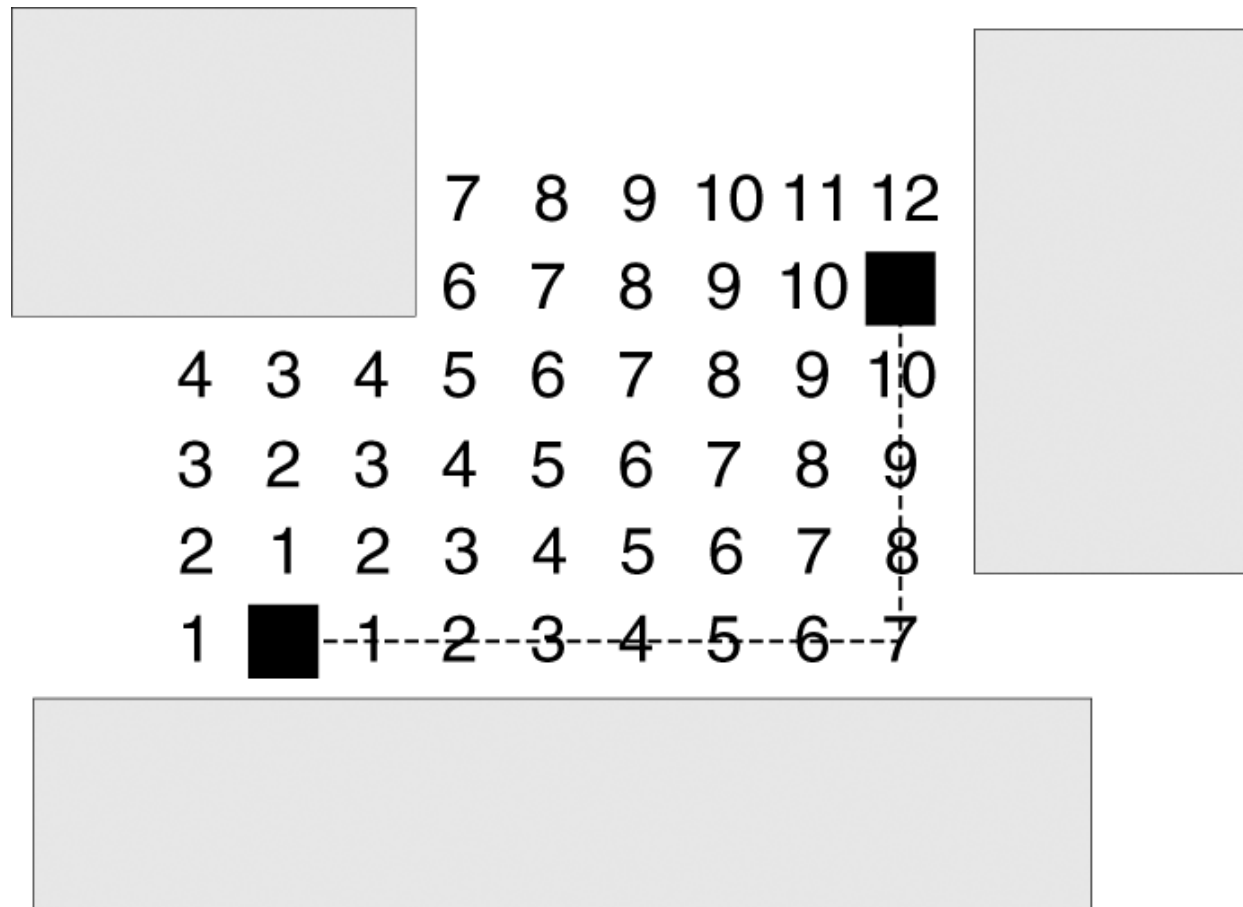
FPGA issues

- **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.**

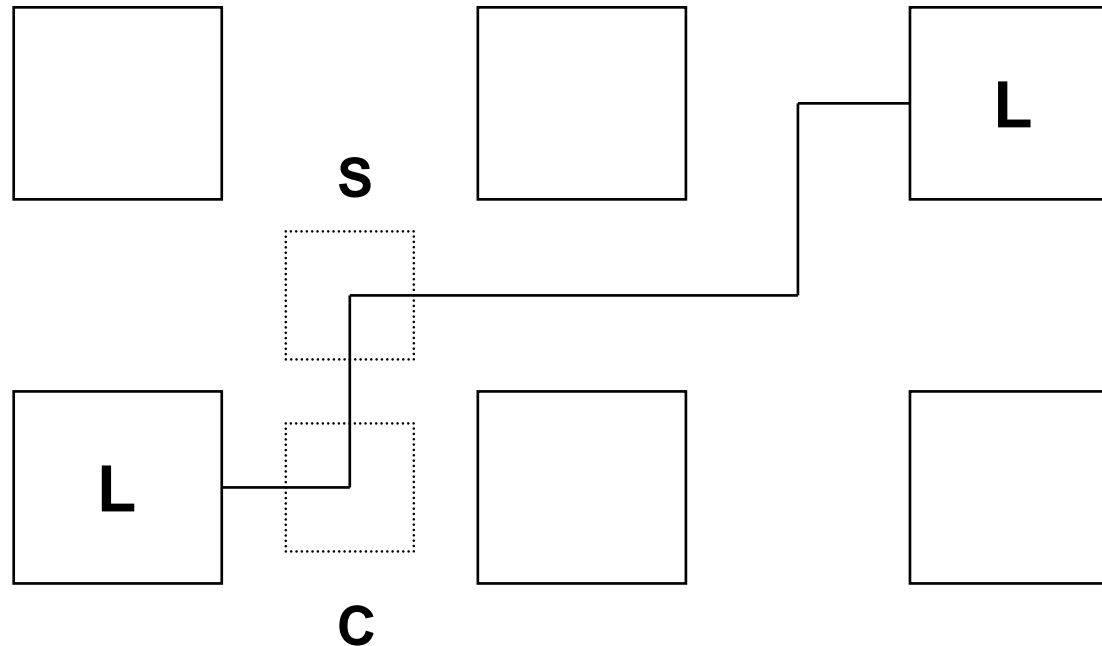
Maze routing

- **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



Maze Routing

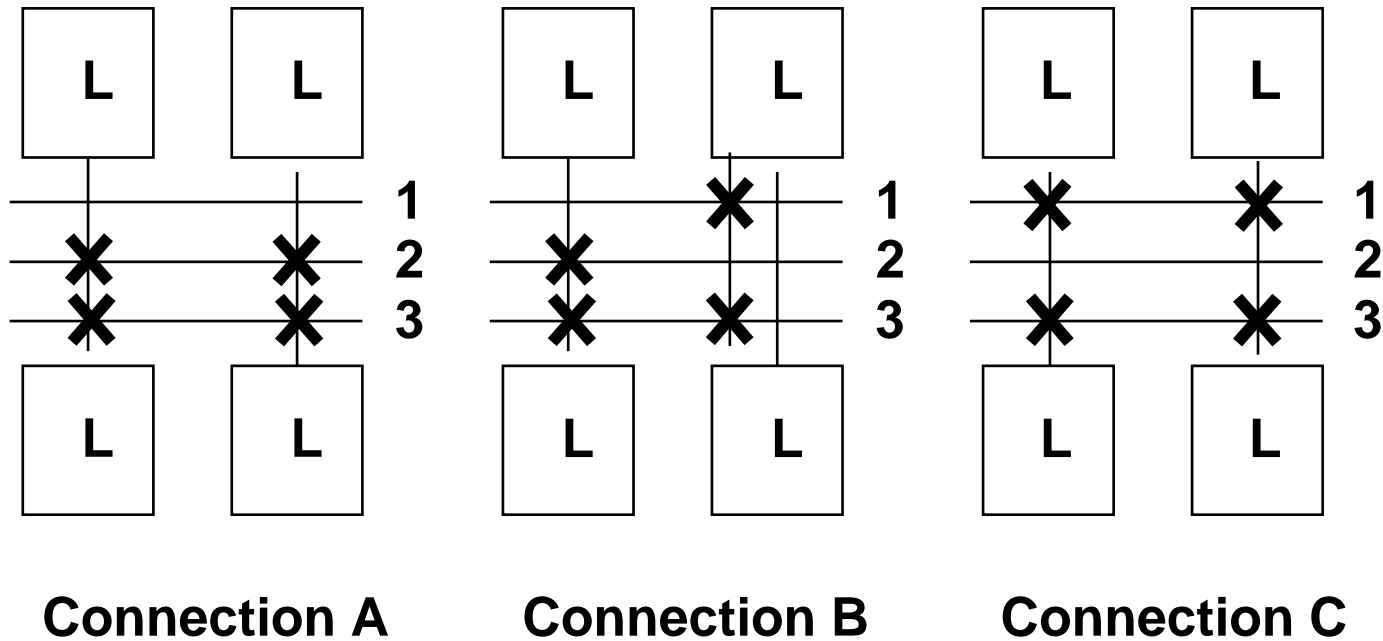


- Evaluate shortest feasible paths based on a cost function
- Like row-based device global route allocates channel bandwidth not specific solutions.
- Formulate cost function as needed to address desired goal.

Two main type of FPGA routers

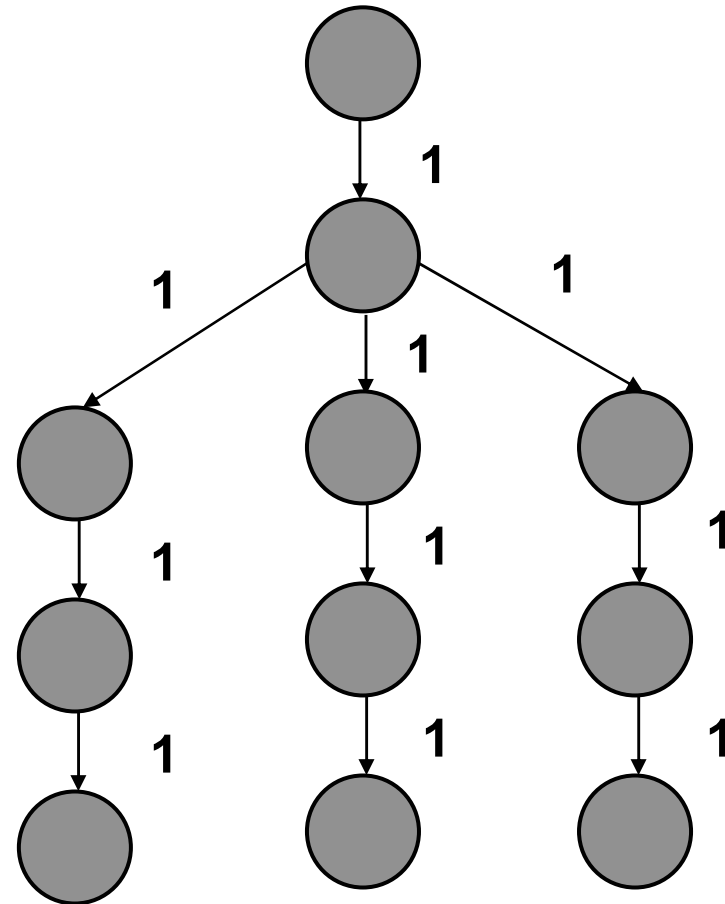
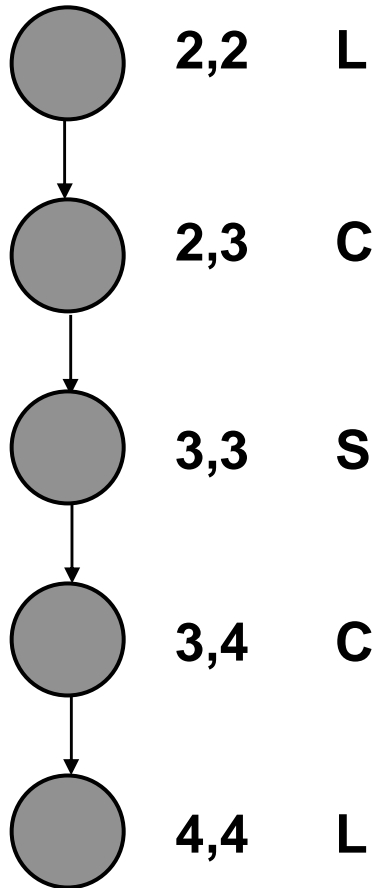
- **Two phase routers**
 - **Global router assigns net connections to specific FPGA channels**
 - **Detailed router “expands” choices in channel to select available wires**
- **One phase routers**
 - **Channel assignment and wire selection happens in one routing pass**
- **Two phase routers were initially popular**
 - **Simpler to write and faster to execute**
 - **More closely models ASIC routing techniques**
- **One phase routers shown to give MUCH better performance**

Consider This Example



- Limited resources cause dependencies
- Need to enumerate possible routes
- Prioritize based on cost factors.

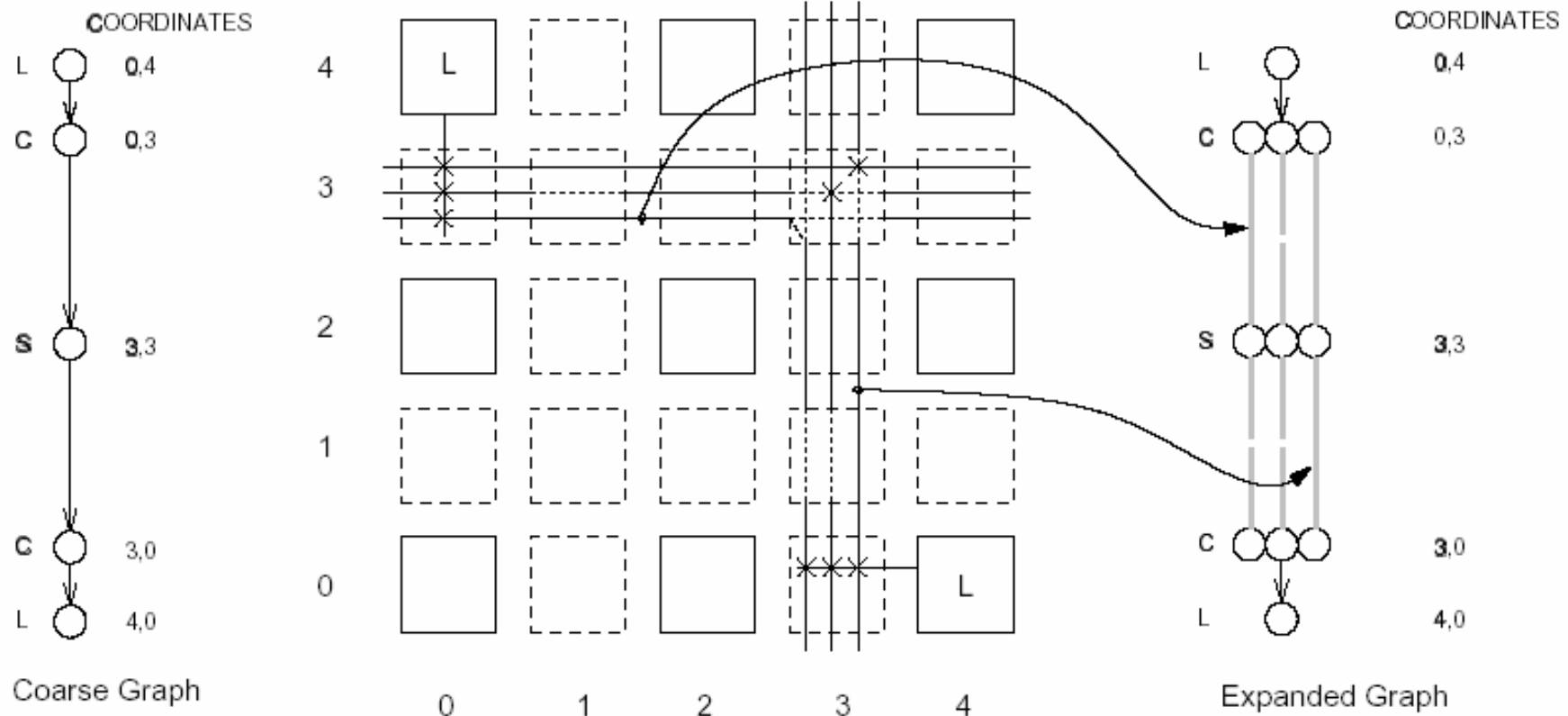
Graph Expansion



- Enumerate possible expansions
- Limited by switchbox and connection box.

Detailed graph expansion

- Consider all possible routes within selected channels
- Limited by connection block and switch block connectivity



Detailed Routing: Lemieux and Brown

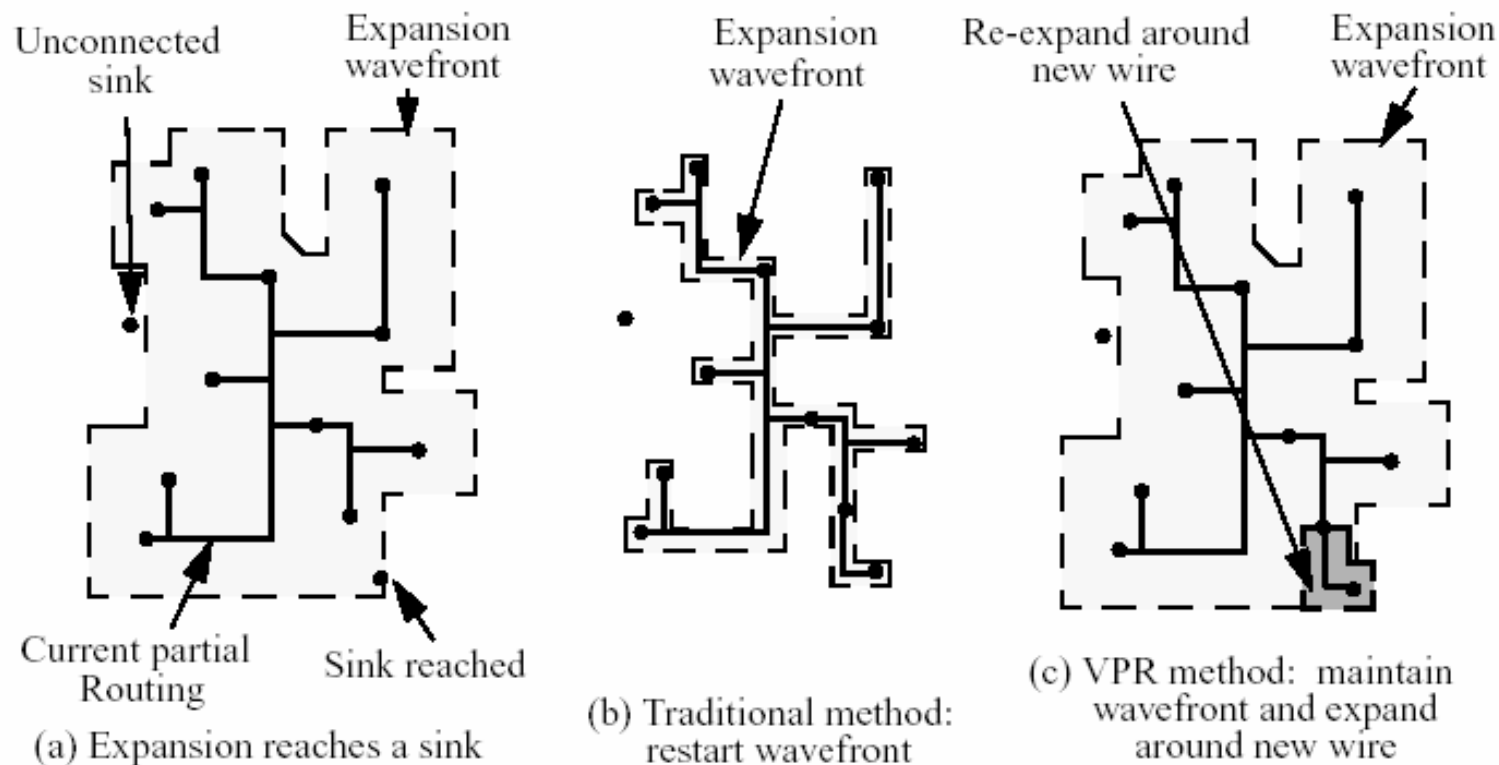
- **Issues to consider**
 - **Some wires span multiple blocks**
 - **Need to allocate long connections to long wires**
- **Step 1:**
 - **For each connection, locate all possible routes**
- **Step 2:**
 - **Use complicated cost function to route connections**
- **Many limitations**
 - **No seamless way to deal with nets with high fanout**
 - **What happens if we run out of tracks?**
 - **Doesn't support multiple iterations.**

Weighting Graph Costs

- 1. Select paths that have fewest negative effects on others.**
 - 2. Identify paths that are essential for a connection.**
 - 3. Sharing factor to enhance high-fanout**
 - 4. Selecting longer segments for nets to reduce wire delay**
- Weighting these factors is not an easy task**

VPR: One-step FPGA Router

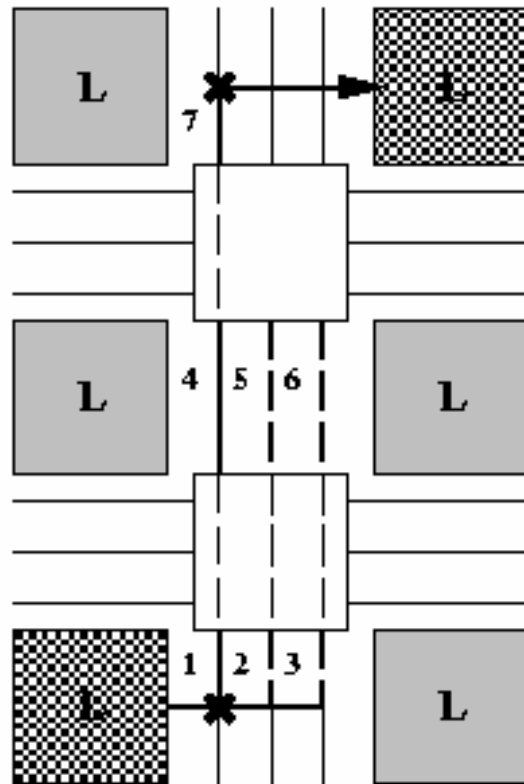
1. Combined global and detailed routing together
2. Uses “expansion wavefront” starting at net source
3. Expansion performed based on node cost



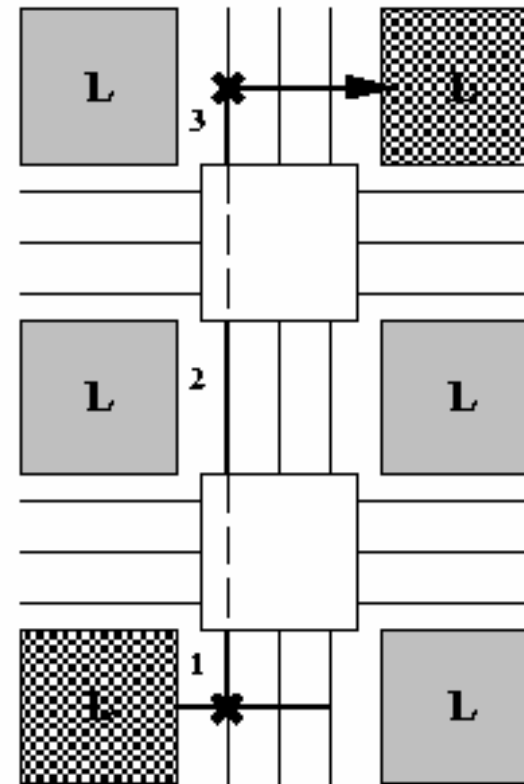
Routing Tradeoffs

- Bias router to find first, best route.
- Vary number of node expansions using:

$$pcost_i = (1 - a) \times pcost_{i-1} + ncost_i + a \times dist_i$$



Breadth-first Route



Depth-first Route

VPR versus Sega

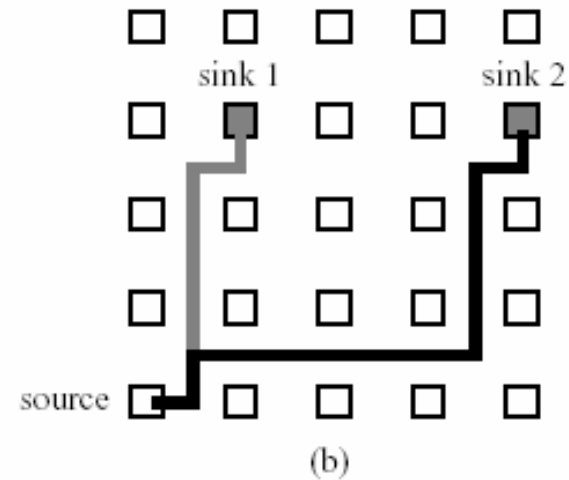
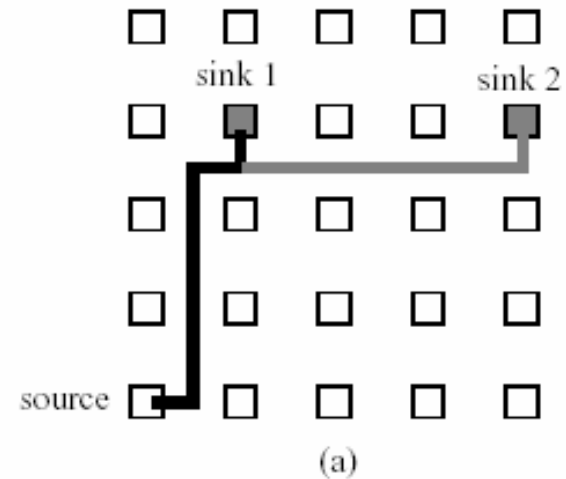
- VPR allows full search of implementation space.
- Effective for multi-fanout nets
- 40% better than Sega

Table 5. Channel widths required to place and route 20 large benchmark circuits.

Circuit	# LBs	SEGA	VPR	Circuit	# LBs	SEGA	VPR	Circuit	# LBs	SEGA	VPR
aln4	1522	16	10	dsip	1370	9	7	s298	1931	18	7
apex2	1878	20	11	elliptic	3604	16	10	s38417	6406	10	8
apex4	1262	19	12	ex1010	4598	22	10	s38584.1	6447	12	9
bigkey	1707	9	7	ex5p	1064	16	13	seq	1750	18	11
clma	8383	≥ 24	12	frisc	3556	18	11	spla	3690	26	13
des	1591	11	7	misex3	1397	17	10	tseng	1047	9	6
diffeq	1497	10	7	pdc	4575	≥ 31	16	Total	--	≥ 331	197

Making One-Step Routing Faster

- How should net fanouts be orders?
- Does this reduce area?
- What is the effect on performance?
- Relatively straightforward algorithm



Making One-Step Routing Faster

- What happens to the expansion list if there are many sinks?
- Binning helps speed up router run time.
- Many high fanout nets now handled by special global routing network.

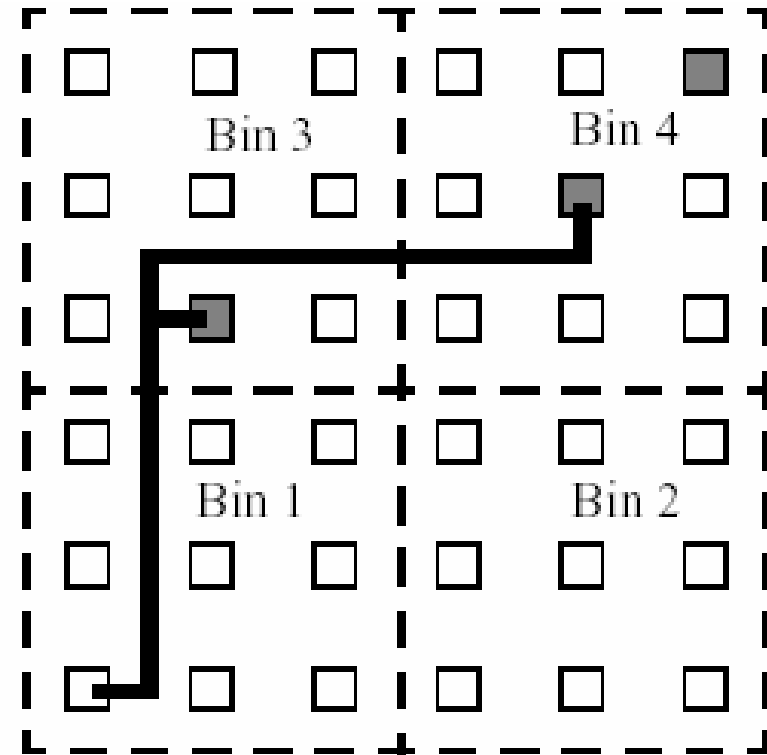


Figure 3. The Binning Technique

Dealing with Multi-Fanout Nets

- Difficult to estimate routing area associated with multi-fanout nets
- Need to modify cost estimation

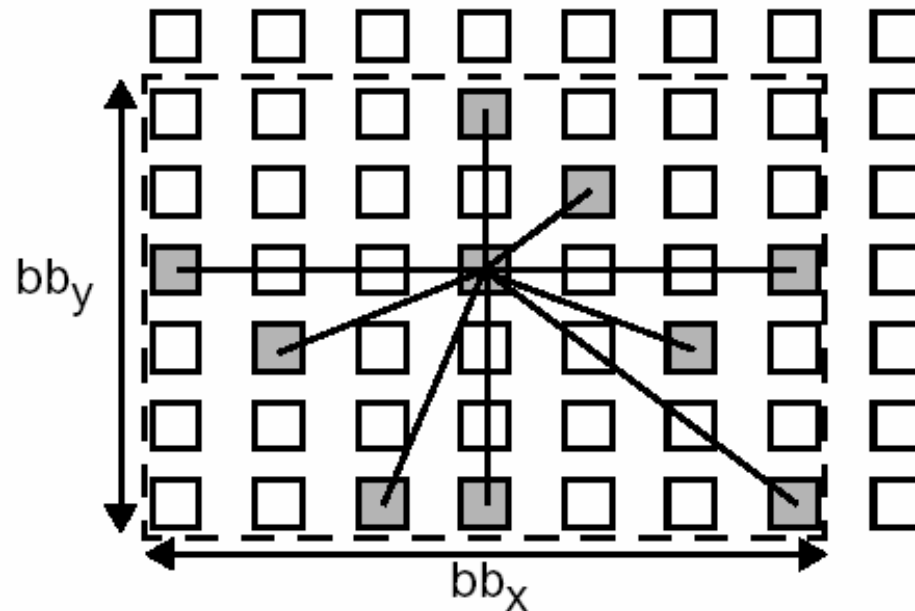


Figure 1: Example Bounding-Box of a 10 Terminal Net [1][2]

Dealing with Multi-Fanout Nets

- Use the cost factor to help scale bounding box costs associated with placement and routing estimation
- Very useful in determining likely routing success

Num Terminals	Correction Factor	Num Terminals	Correction Factor
1 ~ 3	1.00	15	1.69
4	1.08	20	1.89
5	1.15	25	2.07
6	1.22	30	2.23
7	1.28	35	2.39
8	1.34	40	2.54
9	1.40	45	2.66
10	1.45	50	2.79

Table 2: Correction Factors up to 50 [Chen94]

Dealing with Multi-Fanout Nets

- Routing becomes much easier as the amount of routing resources increases

Circuit	W_{\min}		$W_{\min}+10\%$		$W_{\min}+20\%$		$W_{\min}+30\%$		$W_{\min}+40\%$	
	Track Count	Time (s)	Track Count	Time (s)	Track Count	Time (s)	Track Count	Time (s)	Track Count	Time (s)
beast10k	22	96	25	30	27	18	29	8	31	9
beast12k	23	372	26	41	28	24	30	12	33	12
beast14k	26	175	29	31	32	15	34	15	37	16
beast16k	23	291	26	63	28	30	30	14	33	14
beast18k	26	330	29	39	32	19	34	18	37	20
beast20k	30	430	33	83	36	50	39	23	42	24
bubble sort	10	56	11	16	12	10	13	11	14	5
clma	12	909	14	33	15	36	16	12	17	6
elliptic	12	34	14	7	15	4	16	2	17	2
ex1010	14	31	16	8	17	5	19	2	20	2
frisc	12	173	14	15	15	7	16	5	17	2
pdc	16	928	18	17	20	8	21	8	23	4
s38417	8	79	9	15	10	8	11	5	12	3
s38584.1	9	33	10	16	11	19	12	9	13	4
spla	14	91	16	11	17	5	19	2	20	2

Table 4: Routing Times

Linear growth of routing time?

- In the absence of congestion, the routing problem becomes greedy
- In general wiring should increase at a more than linear rate with circuit size
- Greedy allocation of resources may obscure this.

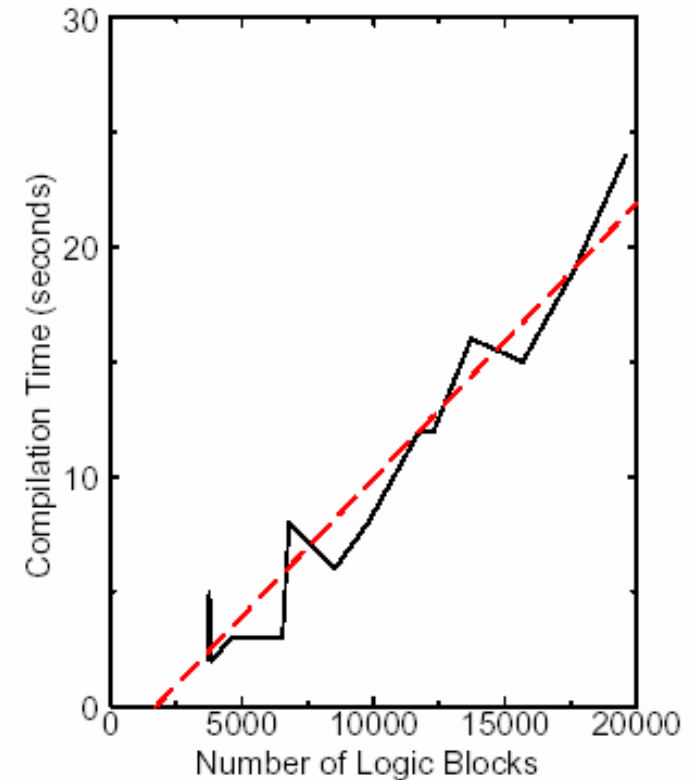


Figure 5. Compile Time vs. Circuit Size

Routing Estimation for Island-style Devices

- Develop approach to use wire length to determine routability of device.
- $W_{\text{estimate}} = \text{total wirelength}/2 * N_{\text{blocks}} * U$
- Difficult to determine U, dependent on F_c and F_s
- Real difficulty in evaluating “wire length”

Estimating Wirelength

- **Multi-fanout nets present a challenge.**
- **Use cost correction factor to adjust bounding box value (e.g. $F=4$, $C=1.08$)**
- **Not clear what to do for timing driven**
- **84% accuracy for 15 benchmarks (Rose/Swartz)**
- **Even works for segmented devices.**

Track count estimation

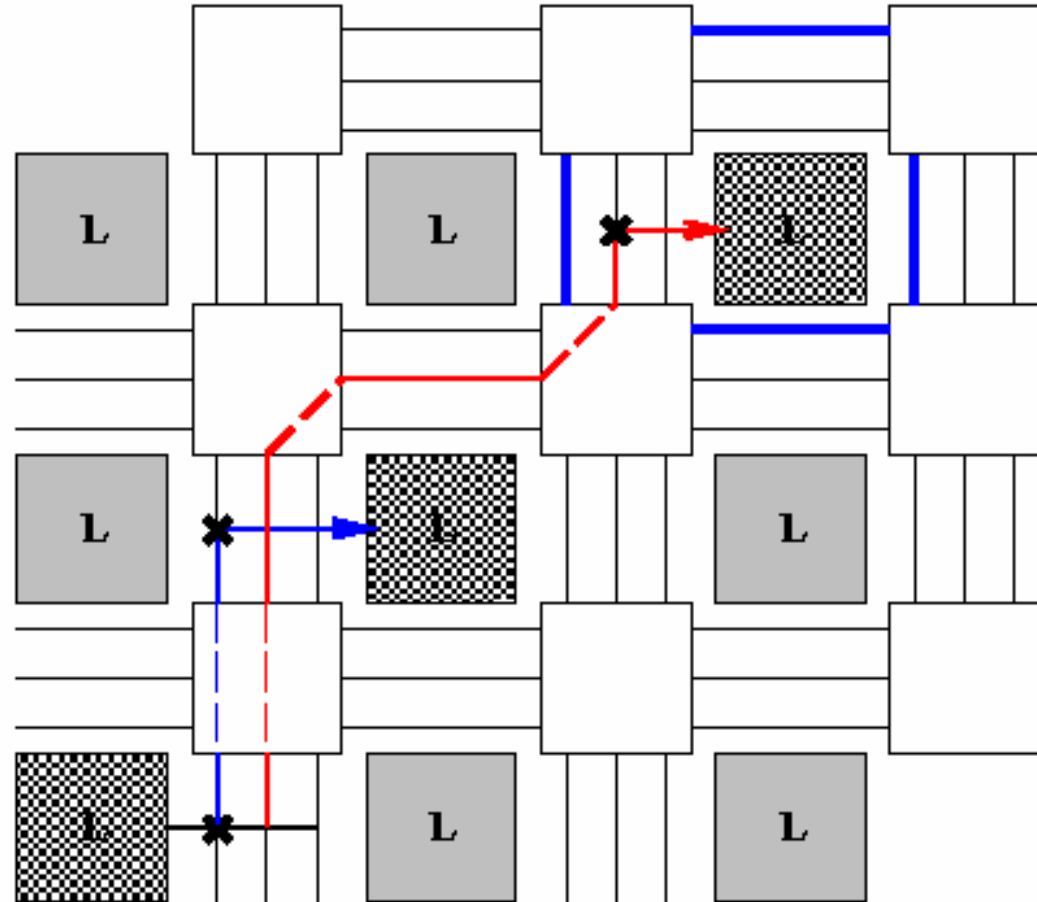
- **Use to determine if device will route successfully**
- **Less of an issue for today's commercial FPGAs**
- **Not really helpful for timing-driven compilation**

Circuit	W_{\min}	W_{estimate}	Difference
beast10k	22	22	0
beast12k	23	24	1
beast14k	26	25	-1
beast16k	23	23	0
beast18k	26	26	0
beast20k	30	29	-1
bubble sort	10	8	-2
clma	12	13	+1
elliptic	12	11	-1
ex1010	14	12	-2
frisc	12	13	+1
pdc	16	16	0
s38417	8	8	0
s38584.1	9	8	-1
spla	14	14	0

Table 7: Track Count Estimates

Architectural Limitation

- Routing architecture necessitates domain selection.
- Bigger effect for multi-fanout nets

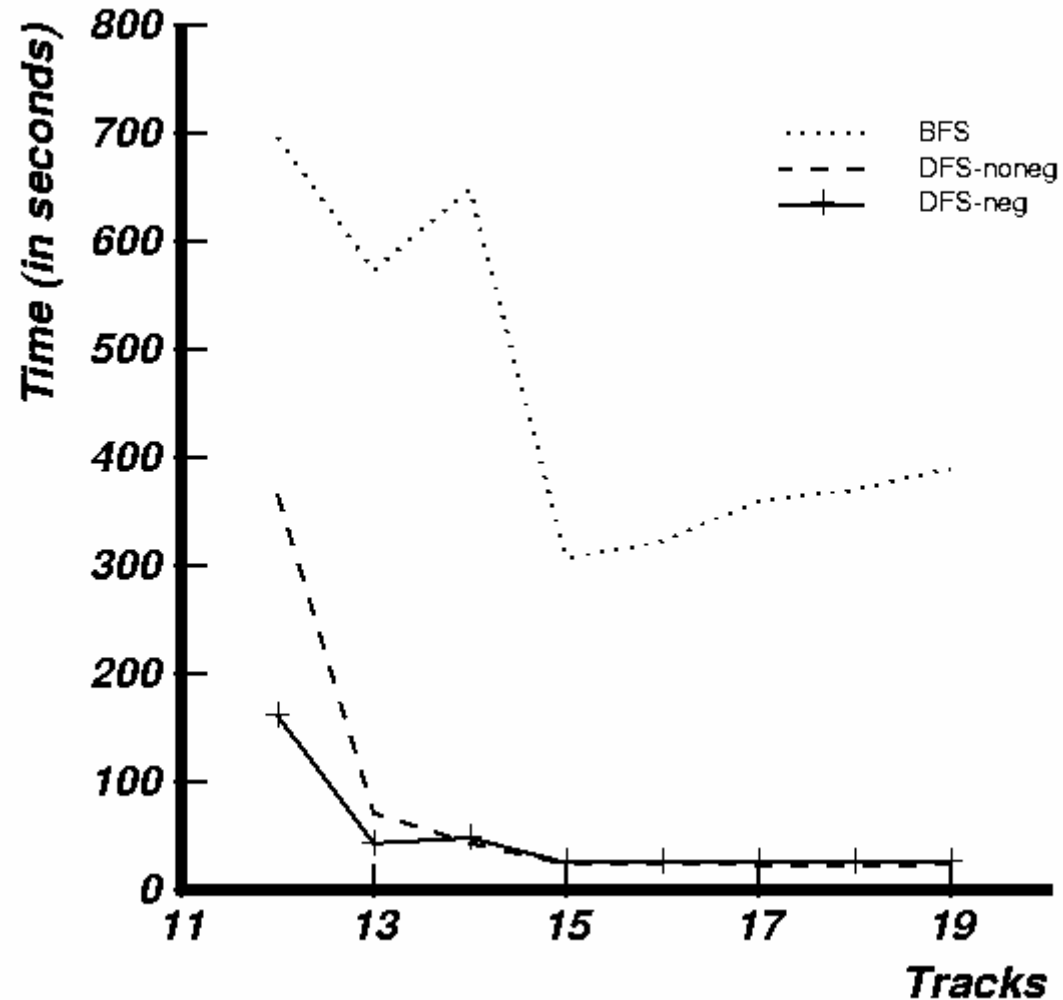


Domain Negotiation

- Evaluate occupancy of domains adjacent to outputs prior to net routing
- Rank domains by routability
- Penalize domains with little or no likelihood of successful completion.
- Add domain rank to node cost function at net output.

$$pcost_i = (1 - a) \times pcost_{i-1} + ncost_i + a \times dist_i + r_d$$

Results: Fast Routing – FFT16



- Domain negotiation has larger effect as track width is reduced.

Routing Results Summary

	Average Route Time (s)	
	Tracks : W_{\min}	Tracks: $W_{\min}+40\%$
BFS	709	269
DFS- noneg	647	20
DFS-neg	333	18

- **Low-stress case shows order of magnitude speedup**
- **Routing proportional to wire length**
- **Domain negotiation reduces difficult route time by a factor of 2.**

Difficult Routing Case Results

Circuit	Source	Logic Blocks	DFS-neg Min. Tracks	BFS Min. Tracks
fft16	RAW	11860	12	12
ssp96	RAW	12041	13	12
spm16	RAW	6632	11	11
bubble	RAW	8453	8	8
frisc	MCNC	3556	15	15
s38417	MCNC	6406	11	11
S38584.1	MCNC	6447	11	12
clma	MCNC	8383	16	16
elliptic	NCNC	3849	13	13
pdC	MCNC	4631	21	21
total			131	131

- Depth-first case needs same track count as minimum breadth-first track count.

Pathfinder

- Use a non-decreasing history value to represent congestion.
- Similarities to multi-commodity flow
- Can be implemented efficiently but does require substantial run time
- Only update after an iteration.

$$c_i = (1 + h_n * h_{fac}) * (1 + p_n * p_{fac}) + b_{n, n-1}$$

Timing-Driven Routing

- **Add delay cost component to routing.**
- **Represent delay along path as RC chain. Buffering important here.**
- **Note that timing driven routing selects most distant point for first route.**
 - **Sets upper bound on delay.**
- **Need for combined breadth-first congestion and timing-driven route.**

Timing-Driven Routing

- **Difficult to estimate remaining timing along a path**
- **Difficult to balance costs for each critical net**
- **Some routers attempt to “look-ahead” to anticipate congested or time-critical areas**
- **Optimal approaches have generally failed.**

Combined Placement and Routing

- **Used depth-first route to select initial connections**
- **Swap blocks and rip up attached nets**
- **Bias nets that span the bulk of device onto long-line resources.**
- **Took 16X longer than place and route**
 - **8% to 15% improvement.**

Summary

- **Routing a difficult problem based on device size, complexity**
- **Early 2-step approaches became obsolete as device sizes grew.**
- **Pathfinder extensively used to determine best, shortest paths.**
- **Most routers are highly advanced.**