A-Priori Wirelength and Interconnect Estimation Based on Circuit Characteristics

Shankar Balachandran **Dinesh Bhatia**

Center for Integrated Circuits and Systems

University of Texas at Dallas

shankars@utdallas.edu
dinesh@utdallas.edu

Outline

- Introduction
- Motivation and Prior Work
- Definitions
- Wirelength Prediction
- Routing Demand Prediction
- Conclusion

Introduction

- **Interconnect Prediction**
  - breaks repetitive design convergence loop
  - helps in performing early feasibility studies
- **A-Priori Prediction**
  - is done before placement stage
  - is used to provide congestion map, wirelength metrics
  - can be used for architecture evaluation

Scope of This Work

- A-Priori *wirelength* and *interconnect* prediction for island-style FPGAs
- Bounding box prediction for *all* wires
  - Identifying important circuit characteristics which constrain placement
  - Assumes that wirelength is minimized during placement
- Routing demand estimation
  - Channel Width calculation
- No prior characterization of placement/router

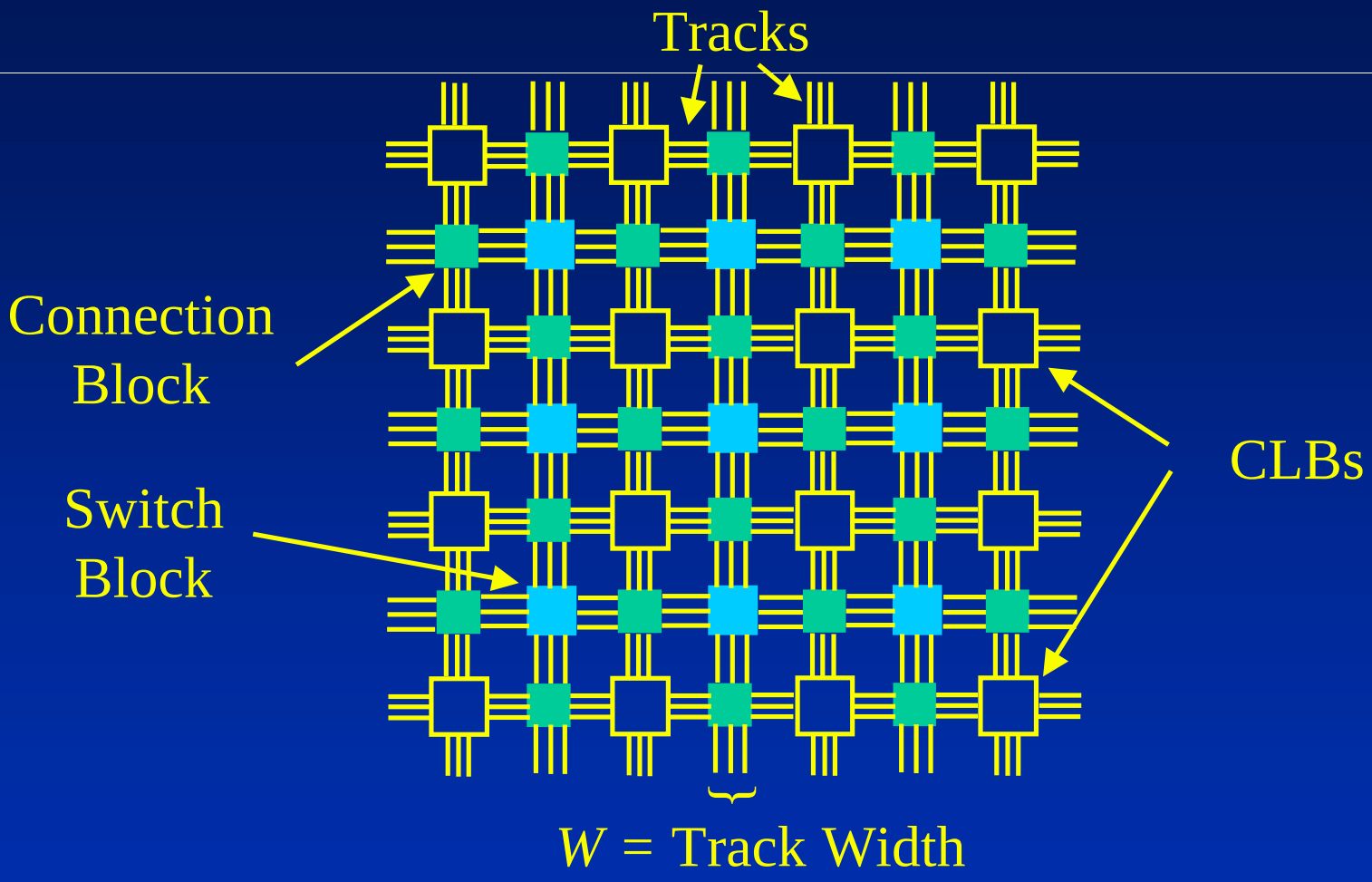
Prior Work

- Rent's Rule [TVLSI 2000, Bakoglu]
  - Interconnect prediction builds models for architecture, circuit and placement
  - Can calculate average/total wirelength, congestion etc.
- Sechen [ICCAD 87]
  - Average wirelength of optimized placements
  - For all possible bounding boxes, enumerate all possible positions for sources and sinks to calculate average wirelength of the whole netlist
- Hamada et al. [DAC 92]
  - Break down nets into cliques and perform neighborhood analysis on them
  - Placement is considered a stochastic process
  - Wirelength distribution is calculated
- Bodapati et al. [SLIP 00]
  - Bounding box estimates using structural analysis
  - Needs calibration of placement/router

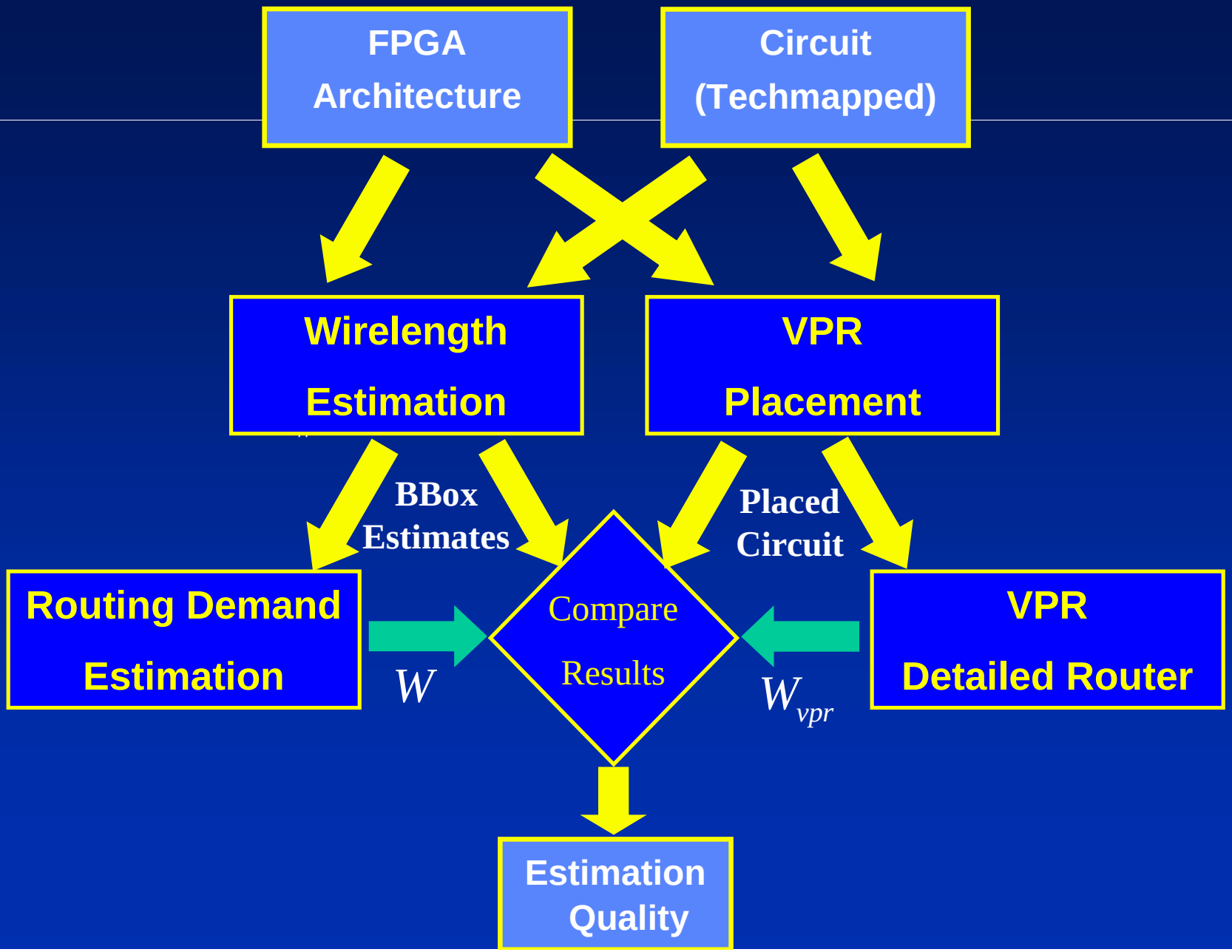
Motivation

- Average wirelength is not sufficient
  - Rent's rule, Sechen and Hamada et al. report only these figures
- Individual wirelength is useful
  - Logic synthesis, floorplanners
- Congestion metrics should be quantifiable
  - Very important for FPGAs
  - Channel Width requirements for routers
    - To avoid the chicken and egg problem

Island Style FPGA



Methodology

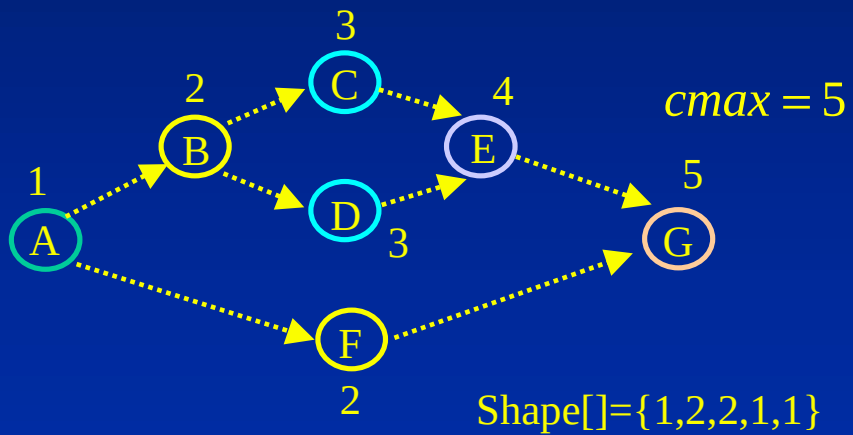


Definitions [CIRC - TCAD 98, TVLSI 02]

- **Combinational level $c(x)$ of a node is :**

$$c(x) = \max(c(x') \mid x' \in fanin(x)) + 1$$

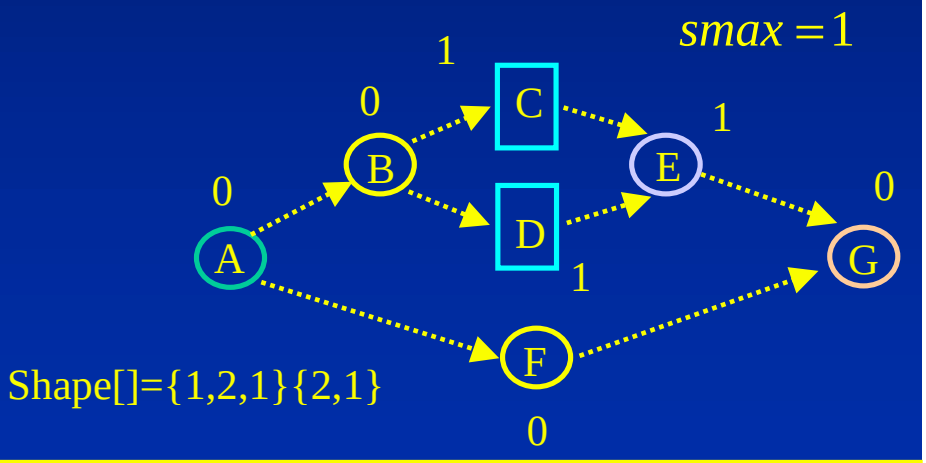
- $cmax = \max(c(x))$



- **Sequential level $s(x)$ of a node is :**

$$s(x) = \begin{cases} 0 & ; x \text{ is a PI-node} \\ s(x')+1; & x \text{ is a FF-node with input } x' \\ \min(s(x') \mid x' \in fanin(x)) & ; \text{otherwise} \end{cases}$$

- $smax = \max(s(x));$



- **Shape : A vector**
  - $Shape[i] = c_0 \mathrm{L}\ c_{cmax}$; c_i = number of nodes in level i
  - **For sequential circuits, the combinational shape vector in each level are concatenated back to back**

Definitions(2)

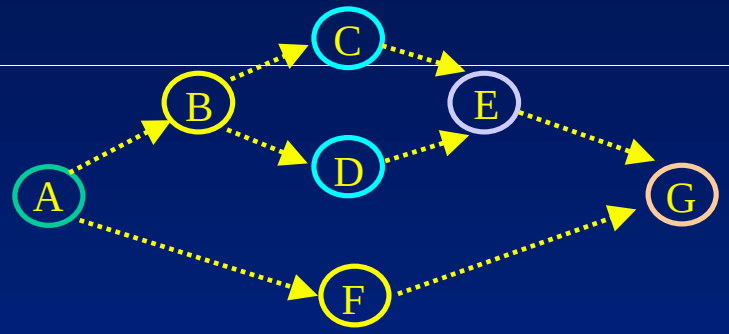
- Reconvergence R_{xy}
  - Has multiple paths from x to y
  - x is the origin of reconvergence
  - y is the destination of reconvergence
  - Always contained within one sequential level
- Number of reconvergences

RN_{xy} = Number of paths from x to y

- Length of reconvergence

$$RO_{xy}(x) = RI_{xy}(y) = \frac{1}{RN_{xy}} \sum_{p \in P_{xy}} l(p)$$

P_{xy} is the path-set from x to y
$l(p)$ is the length of path p



R_{BE} : RN = 2

$l(p_1 = B \rightarrow C \rightarrow E) = 2$

$l(p_2 = B \rightarrow D \rightarrow E) = 2$

$RO_{BE}(B) = RI_{BE}(E) = 2$

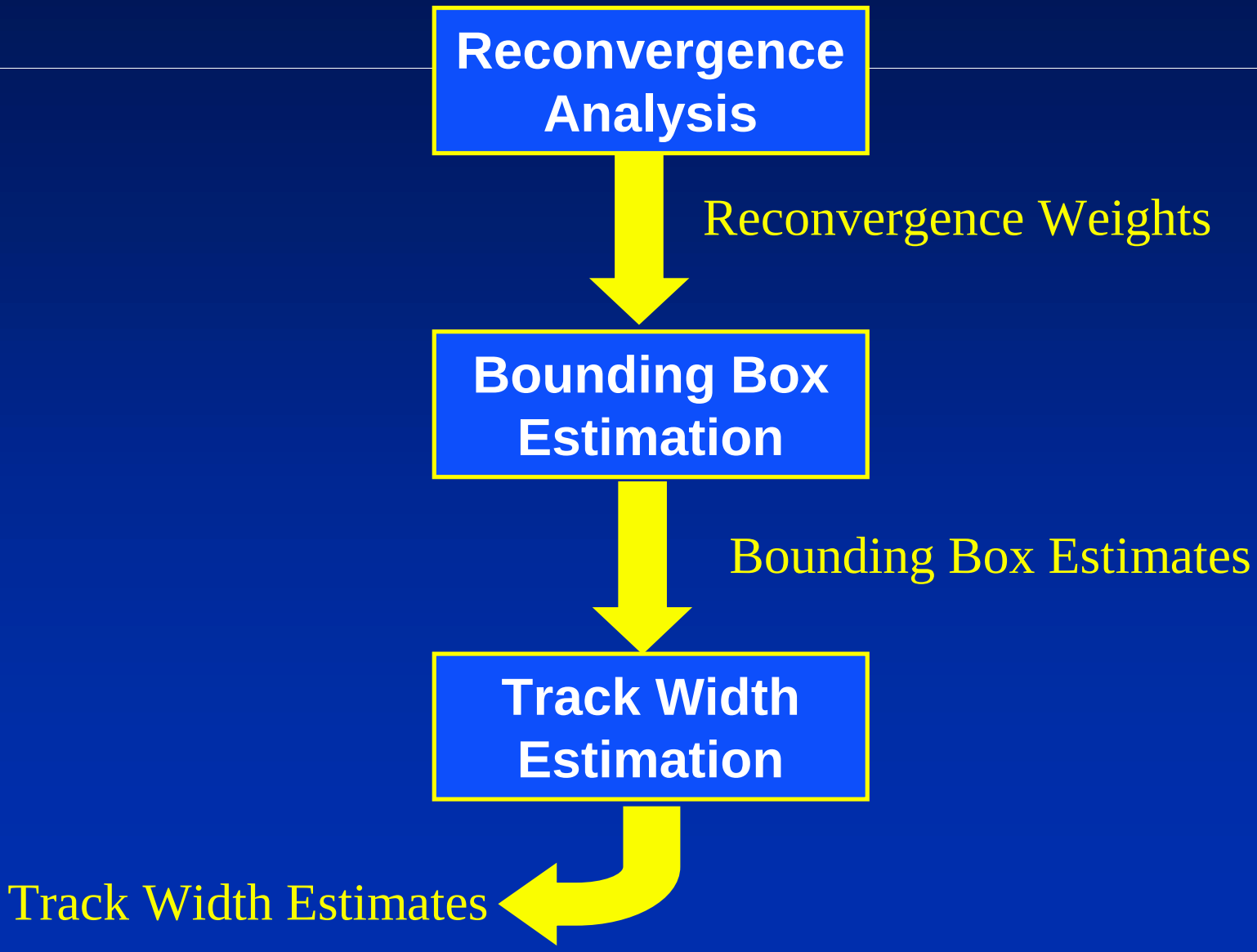
R_{AG} : RN = 2

$l(p_1 = A \rightarrow (BE) \rightarrow G) = 3$

$l(p_2 = A \rightarrow F \rightarrow G) = 2$

$RO_{AG}(A) = RI_{AG}(G) = 2.5$

Overview of Our Methodology

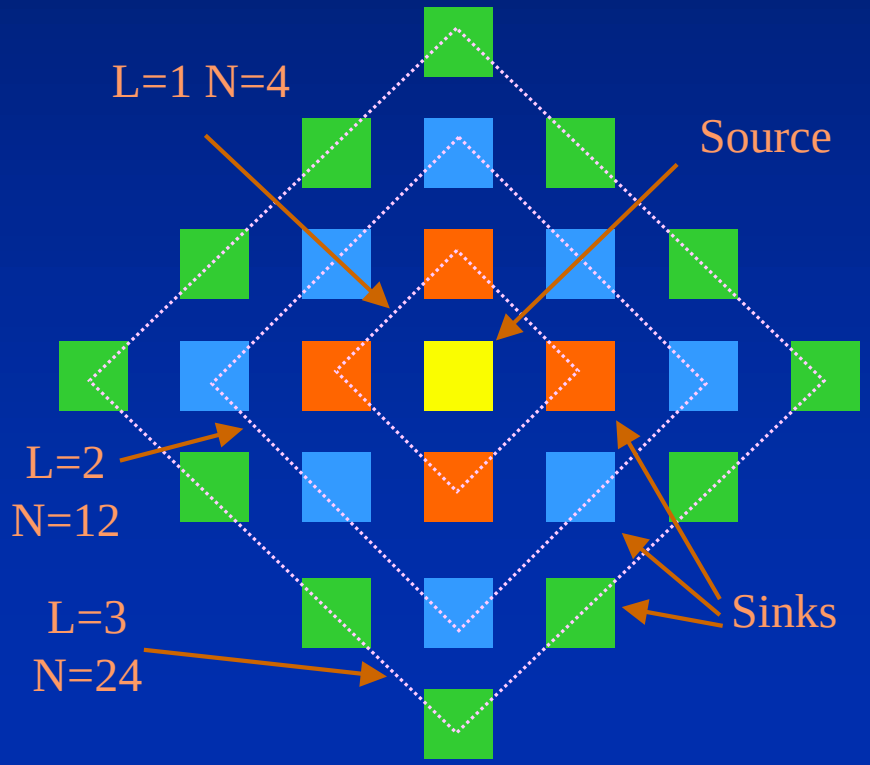


Wirelength Estimation

- Wirelength of a circuit depends on
  - Structural properties of the circuit
  - Placement of the circuit
- Nets from Input pads usually feed more nodes than the other nodes
  - Hence, classify the nets as *logic nets* and *IO-nets* and treat them separately
- Wirelength of an individual net *N* will depend on
  - Number of terminals – t_N
  - Interaction with other nets

Phase 1 - Minimum Span for Logic Nets

- Assume a net *N* is tightly placed. Wirelength is optimal when
  - Source is placed in the center
  - All sinks are tightly packed around the source
- Minimum span ***L*** is entirely dependent only on t_N

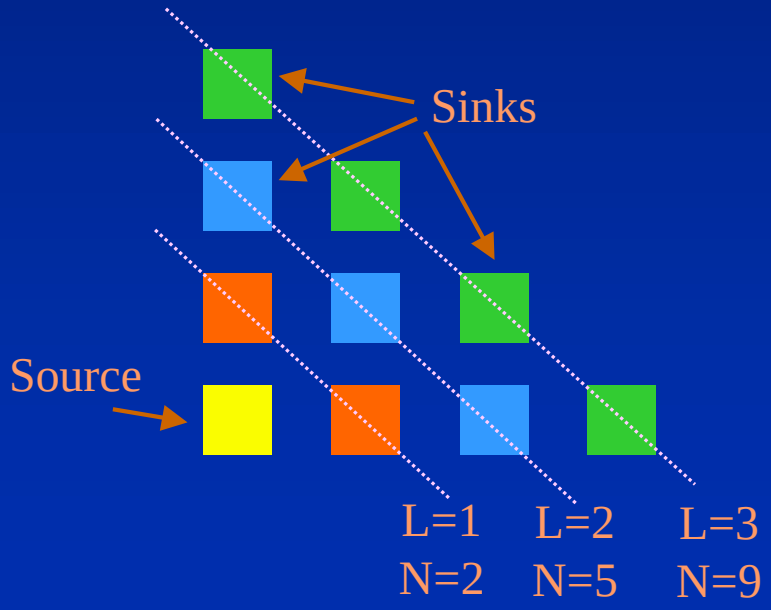


$$4+8+12+\mathrm{L}\ +4\cdot L \geq t_N$$

$$L = \frac{\sqrt{1+2\cdot t_N}+1}{2}$$

Phase 1 - Minimum Span for IO-Nets

- Input pads are placed along the periphery
- IO-Nets have many sinks usually, and the location of the Input pads cannot be guessed
- Assume the Input pad is in the corner of the FPGA
  - Worst-case calculation of wirelength
- Tight placement of sinks around the pad



$$2+3+4+\mathrm{L}\ +L \geq t_N$$

$$L=\sqrt{2\cdot t_N+9/4}-1/2$$

Phase 2 – Dilation of Nets

- Tight placement is always **not** possible
  - **Push** and **Pull** from other nets are ignored
- Net *N* has no incident reconvergences => no dilation
- *N* has incident reconvergences => other nets pull the cells away => dilation
- Net dilation is based on reconvergences on its immediate neighborhood : source, sinks, fanin
- For any node that is an origin of any reconveregnce R_{x*} , let the **out-weight** be

$$RO(x) = \frac{\sum RO_{x*}}{\sum RN_{x*}}$$

  the average length of all out-bound reconvergences
- For all node that is a destination of any reconvergence R_{*y}, let the **in-weight** be

$$RI(y) = \frac{\sum RI_{*y}}{\sum RN_{*y}}$$

  the average length of all in-bound reconvergences

Dilation Factor

- **Raw weight of a node x is** $RW'(x) = RI(x) + RO(x)$
- **Flip-Flops have many incident reconvergences, hence an adjustment w.r.to LUT-size** k

$$RW(x) = \begin{cases} \log_k RW'(x); & \text{if } x \text{ is a FF-node} \\ RW'(x) & \text{; otherwise} \end{cases}$$

- **Dilation on a net N with v_N as its source is**

$$R'(N) = \left(\sum_{x \in \text{fanin}(v_N)} RW(x) + \sum_{y \in \text{fanout}(v_N)} RW(y) + RW(v_N) \right) \Big/ t_N$$

- **Similar to flip-flops, IO-Nets have large weights, hence an empirical value for IO-Nets**

$$R(N) = \begin{cases} 3/\sqrt{2} & \text{; if } N \text{ is a IO-Net} \\ R'(N) & \text{; otherwise} \end{cases}$$

Phase 3 – Uniform Distribution

- Let *p* be the position in which *Shape* vector has the maximum value
- The nodes in this level are so many in number that they are expected to be uniformly distributed in the layout
  - The nodes in this level are not connected to each other
  - They are however strongly connected to the other nodes
- If a node has more than one such cell, the net will dilate more

Spread Due To Uniform Distribution

- ***SP*** **= Set of nodes in the peak level**
- N **= Minimum required FPGA size**
- **Construct a hypothetical grid which has only one cell from *SP***
- **The hypothetical grid size is related to the FPGA dimension as** $G = \mathrm{N} / \sqrt{|SP|}$
- **If a net N has some nodes in *SP* then the span must respect the uniform distribution assumption**
- **The uniformity factor is calculated as**

$$U = \sqrt{|\, SP \cap fanout(v_N) \,|} \cdot G$$

Bounding Box Span of a Net

- The bounding box span of the net N depends on
  - L – the minimum span of the net
  - $\mathrm{R}(\mathrm{N})$ – the dilation of the net due to reconvergences
  - U – the uniformity factor
- The horizontal span of the net N is

$$HSpan(N) = \begin{cases} \max(L, U) & ; \text{if } \mathrm{R}(\mathrm{N}) < 1 \\ \max(\mathrm{L} \times \mathrm{R}(\mathrm{N}), \mathrm{U}) & ; \text{if } \mathrm{R}(\mathrm{N}) > 1 \end{cases}$$

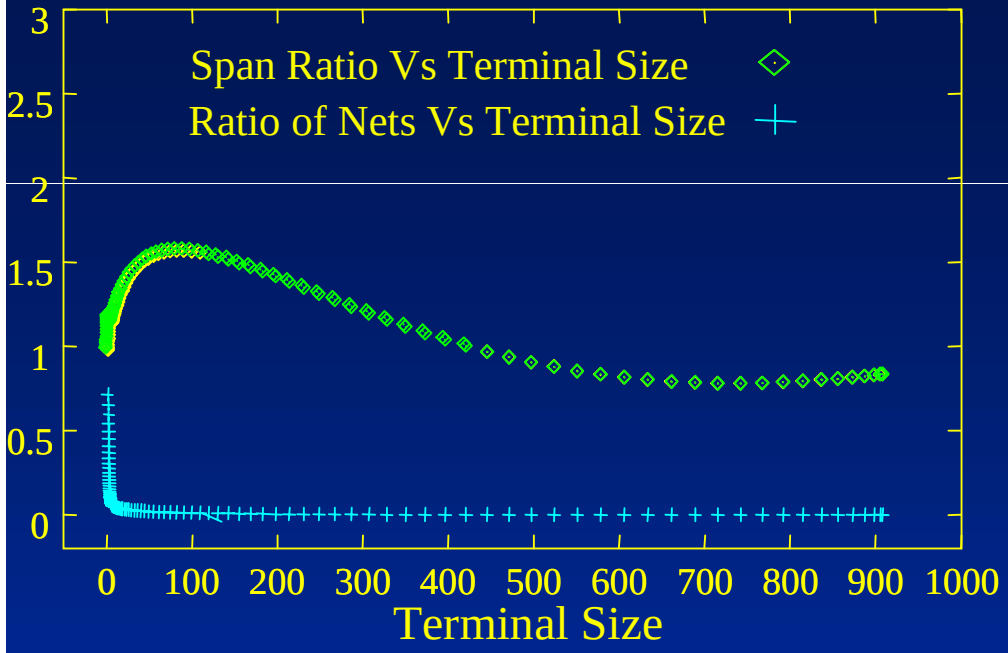
- The vertical span of the net is same as ***HSpan***
- The total span of the net is

$$Span(N) = HSpan(N) + VSpan(N) = 2 \cdot HSpan(N)$$

Results for Wirelength Estimation

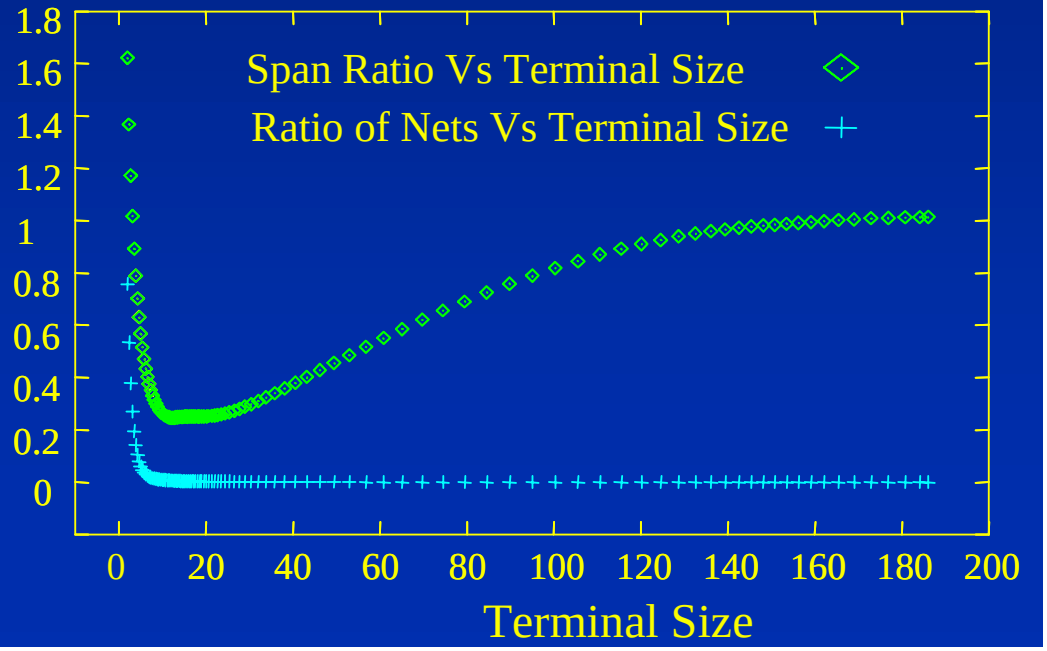
Circuit	Total Error (%)	I/O Nets Error (%)	#Nets Err < N/4	Nets Err > N/4	Total Error w/o R_N (%)	I/O Error w/o R_N(%)
alu4	**2.95**	**-2.24**	**1299**	**223**	**48.45**	**48.48**
apex2	**-19.68**	**29.10**	**1616**	**262**	**53.15**	**66.58**
bigkey	**-21.45**	**34.37**	**1699**	**8**	**-21.42**	**47.48**
dsip	**11.08**	**-4.07**	**1367**	**3**	**10.94**	**-3.46**
misex3	**4.49**	**6.77**	**1170**	**227**	**54.11**	**55.69**
pdc	**13.44**	**-14.7**	**4051**	**524**	**65.01**	**45.93**
s298	**-5.15**	**0**	**1837**	**94**	**20.86**	**38.16**
s38417	**-32.42**	**-4.84**	**5955**	**451**	**37.82**	**50.57**
seq	**4.66**	**13.24**	**1522**	**228**	**56.11**	**58.8**
spla	**-0.68**	**-14.34**	**3329**	**361**	**58.29**	**46.09**
Totals	**-**	**-**	**23845**	**2194**	**-**	**-**
Avg.	**11.6%**	**12.4%**	**-**	**-**	**42.61**	**46.13**

Individual Wirelengths

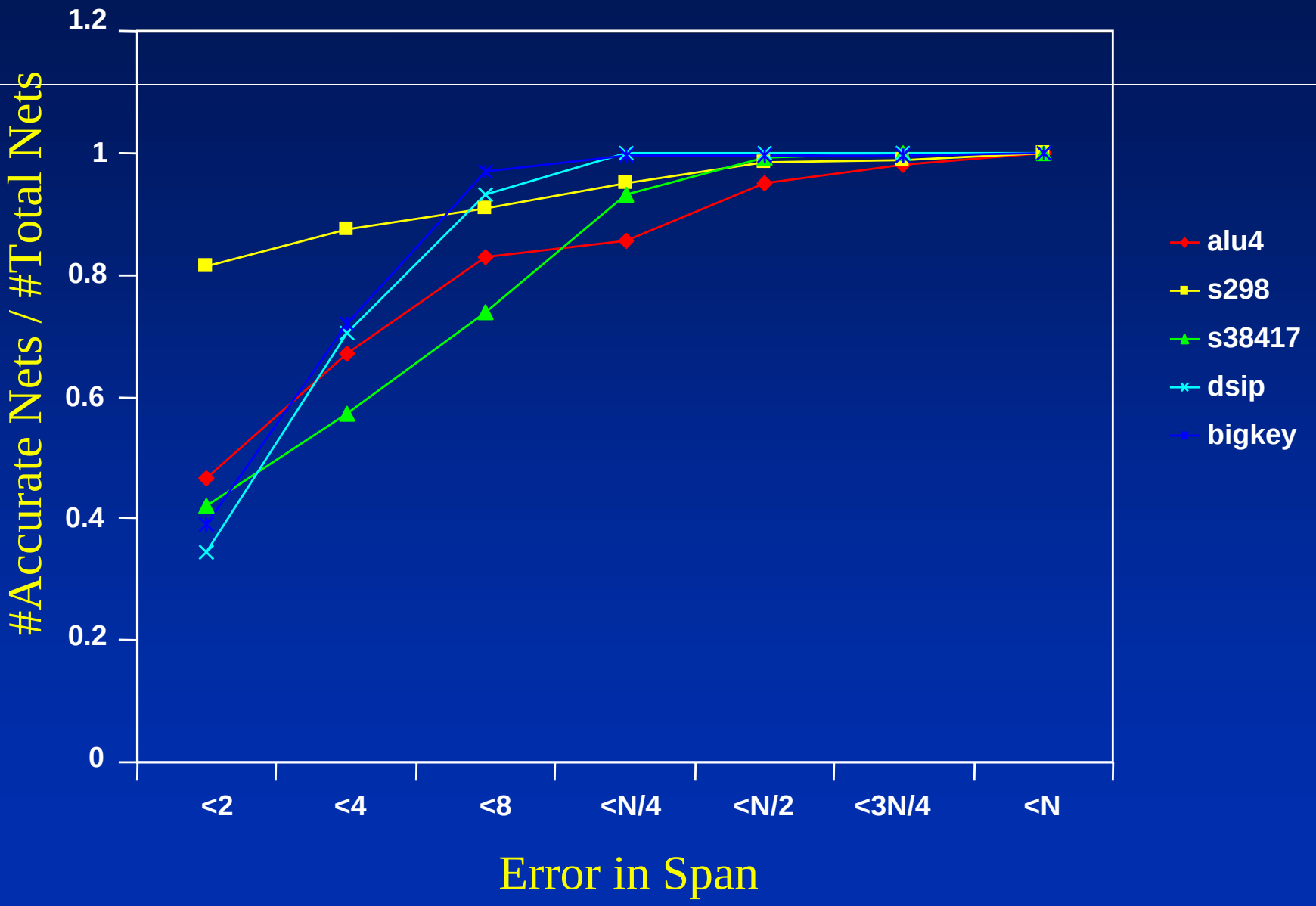


Circuit : *dsip*

Circuit : *misex3*



Error in Span Vs Number of Nets



Routing Demand Estimation

- We use RISA to calculate number of routing elements needed
  - An empirical technique based on wirelength of nets with various terminal sizes
  - The routing demand is based on two factors
    - q - an empirical factor dependent on t_N
    - Bounding Box Sizes
- The actual routing demand for a net N is calculated as

$$D_h^N = q \times \frac{1}{HSpan(N)}; D_v^N = q \times \frac{1}{VSpan(N)}$$

Definitions

- C = Number of Logic Blocks
- nIO = Number of I/O blocks
- If the circuit is placed in the smallest possible device, its width (also height) is given as

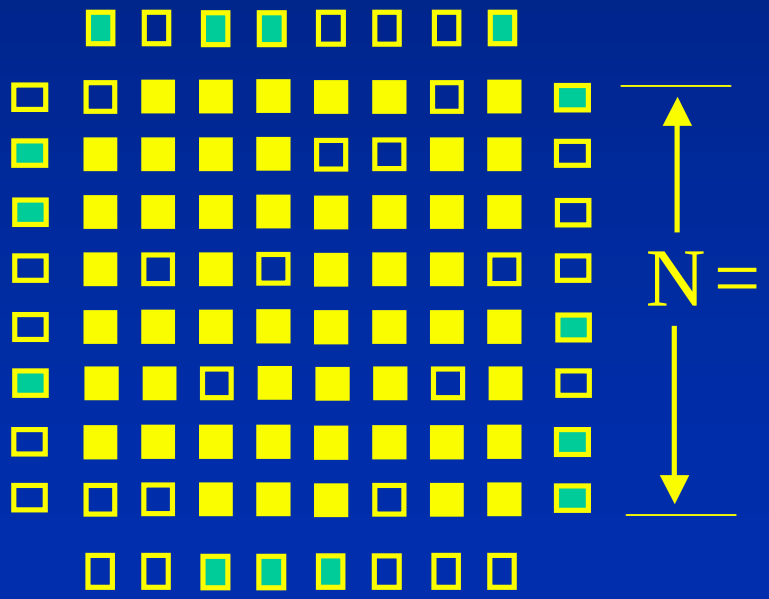
$$N \equiv \max(nIO/4, \sqrt{C})$$

- TD = Total Number of Routing Elements Needed

$$TD = \sum_{N} D_h^N + D_v^N$$

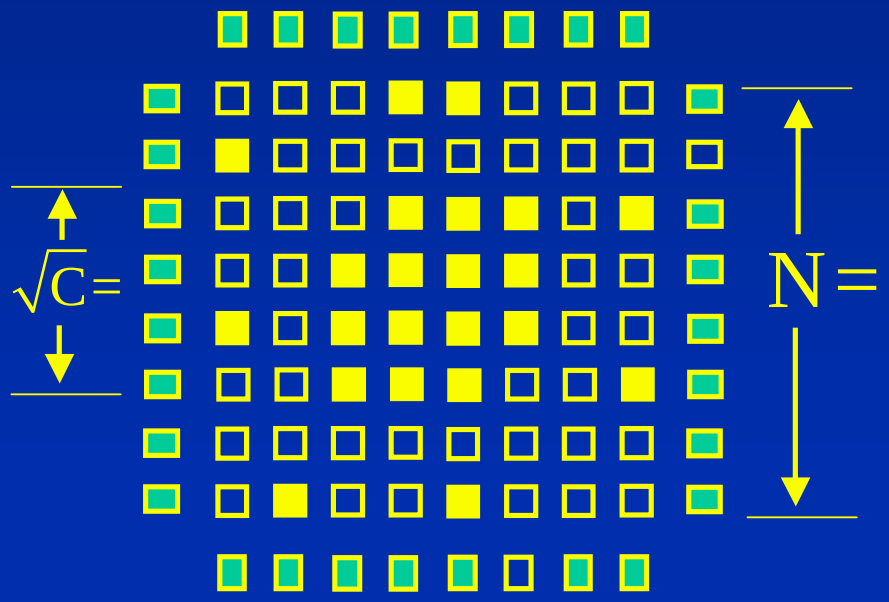
Channel Width Estimation for Pad Unconstrained Circuits

- **Pad-Unconstrained Circuits**
  - $N = \sqrt{C}$
  - ***TD*** **routing elements are uniformly distributed across the device**
  - **Channel width *W* is calculated as** $W = \frac{TD}{C} = \frac{TD}{N \times N}$



Channel Width Estimation for Pad Constrained Circuits

- **Pad-Constrained Circuits**
  - $\mathrm{N} \equiv nIO/4$
  - **Assume that all the logic blocks are placed in the center – consistent with modern placers**
  - **However, *TD* routing elements should be distributed across the whole device**
  - **Channel width *W* is calculated as** $W = \frac{TD}{\mathrm{C}} \times \frac{\sqrt{\mathrm{C}}}{\mathrm{N}} = \frac{TD}{\sqrt{\mathrm{C}} \times \mathrm{N}}$



Experimentation - Other Methods Compared

- RISA [ICCAD 94, DAC 2002]
  - Post-placement technique
  - Add up demands for different sites in the layout and find the maximum channel width
- Yang et al. [ISPD 2001]
  - Rentian Method
  - Extended for FPGAs in [8]
  - Recursive partitioning of circuit and layout
  - Worst-case congestion analysis on the boundaries

Results for Channel Width Estimation

Circuit	W_{VPR}	W	T	W_{RISA}	T_{RISA}	W_{RENT}	T_{RENT}
alu4	11	11.322	0.139	13.506	0.012	10.717	1.54
apex2	12	12.981	0.234	14.911	0.022	21.322	2.49
bigkey	9	5.7	0.385	12.105	0.027	4.761	2.6
dsip	7	6.452	0.298	9.699	0.019	4.176	1.97
misex3	11	11.252	0.156	13.649	0.015	11.682	1.37
pdc	16	11.991	5.034	20.418	0.103	19.067	14.61
s298	8	8.27	1.06	9.963	0.010	14.15	2.33
s38417	8	10.544	8.712	13.192	0.035	14.963	21.43
seq	12	11.826	0.208	14.53	0.019	13.468	2.07
spla	15	11.593	3.148	17.663	0.06	35.149	9.42
Total	109	102.29	19.38	139.63	0.324	149.455	59.86
Error	-	6.1%	-	28.1%	-	37.1%	-

Summary

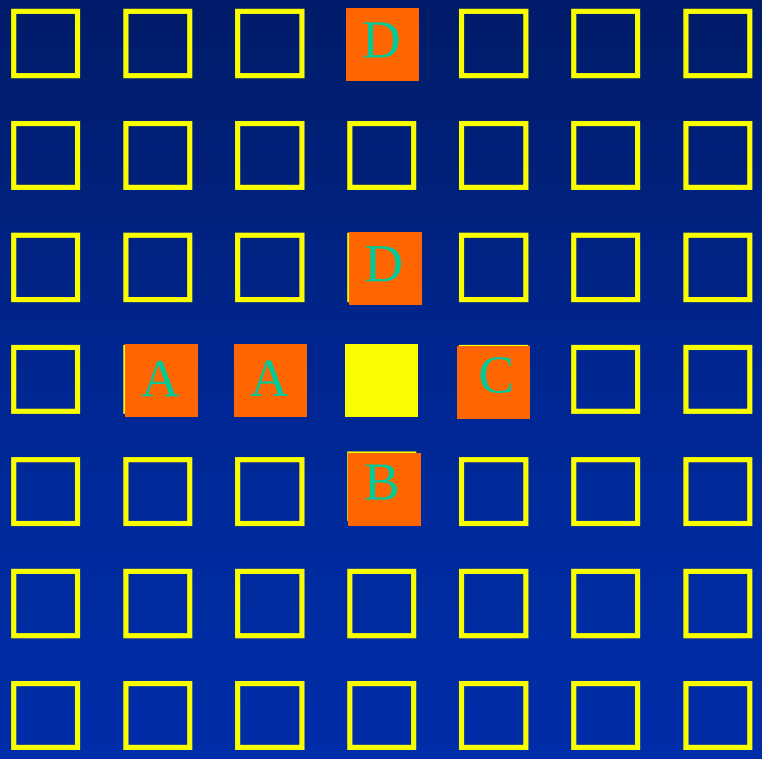
- Identified some important circuit characteristics which dictate placement
  - Push and Pull from reconvergences stretch wires
  - Reconvergences capture more than the local neighborhood of cells
  - 30% more accuracy with reconvergences factored in
- Bounding box prediction is accurate within 11.6% of post-placement lengths
- Channel widths are predicted within 6% of post-route results


THANK
YOU!

Illustration of Bounding Box Calculation

Phase 3

Node D and Node C are in peak level and hence should spread out



Phase 2

Node A has high reconvergence weight. Pulled away from the net

Phase 1

Sinks are tightly placed around the source node

Overview of Our Methodology

Reconvergence Analysis

- Perform reconvergence analysis within different sequential levels
- Assign weights to nodes based on reconvergences

Bounding Box Estimation

For every net

- Calculate the minimum possible bounding box
- Find dilation factor using reconvergence weights
- Uniformly distribute *peak nodes*
- Calculate the actual span using these 3 factors

Track Width Estimation

- Calculate the number of routing elements required using RISA for every net
- Calculate the total number of routing elements
- Distribute this routing demand evenly in the layout to obtain maximum channel width