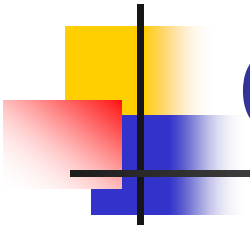
Routing-Aware Scan Chain Ordering

Puneet Gupta and Andrew B. Kahng
(Univ. of California at San Diego, La Jolla, CA, USA.),
Stefanus Mantik
(Cadence Design Systems Inc., San Jose, CA, USA.)
Email: { puneet@ucsd.edu, abk@ucsd.edu, smantik@cadence.com }

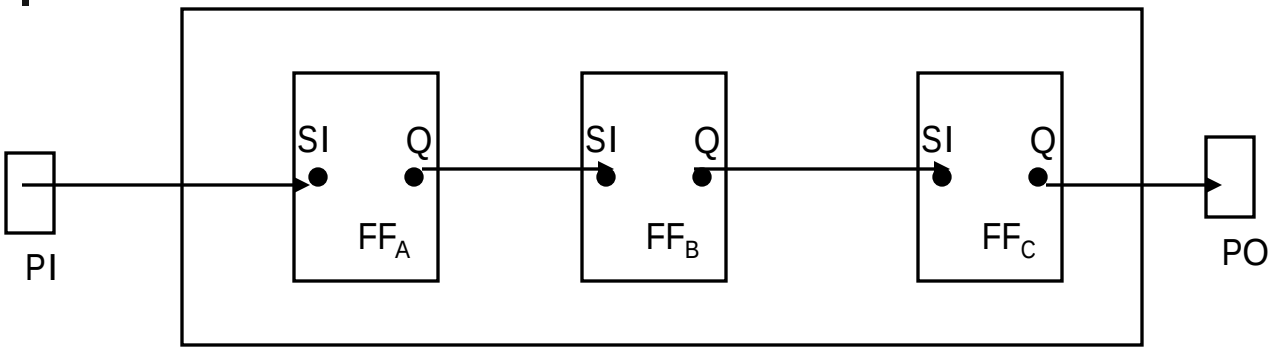
Supported by MARCO Gigascale Research Center and Cadence Design Systems



Outline

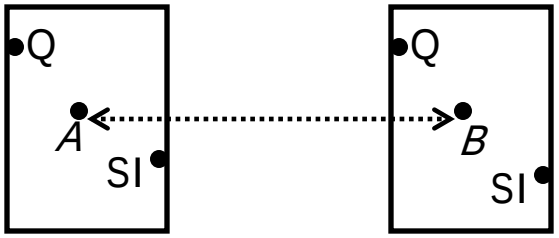
- Introduction and Previous Work
- Motivation for Routing Awareness
- Nature of the New TSP
- TSP Solvers
- Routing Aware Scan Chain Ordering
- Experiments
- Results and Conclusions

Introduction

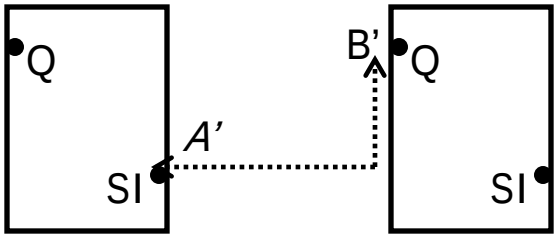


- Scan chains commonly used to enhance testability. All flip-flops chained to form a shift register.
- Minimizing wirelength overhead of scan increases routability and improves timing by reducing capacitive loading on nets that share register pins with the scan chain.
- We give a new scan chain ordering method which gives upto 85% wirelength improvements over commercial tools and previously reported methods.

Previous Work

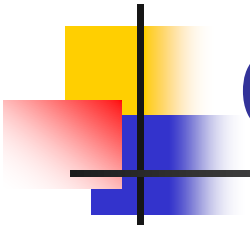


Cell-to-cell distance from FF_B to FF_A



Pin-to-pin distance from FF_B to FF_A

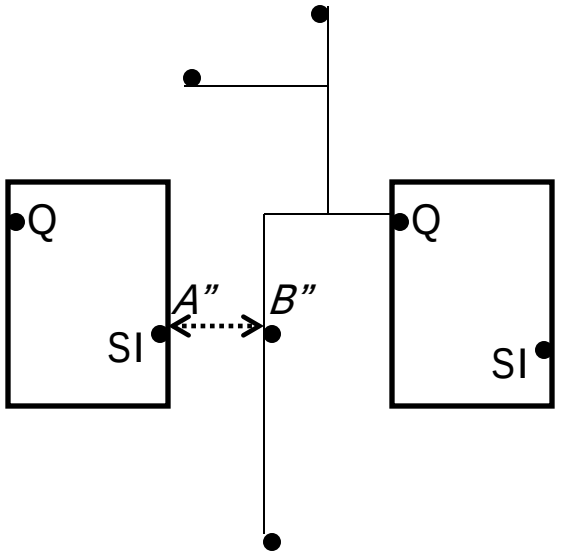
- Scan chain ordering has been modeled as a Traveling Salesman Problem (TSP) [FeuerK83]
- Previous works did placement-based ordering using cell-to-cell distance (*AB*) or pin-to-pin (*A'B'*) Manhattan distance as the TSP distance metrics
- Modified 2-opt and 3-opt heuristics for the *almost* symmetric pin-to-pin TSP were given by [BoeseKT94, KobayashiEK99]



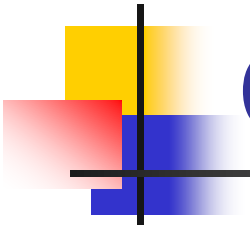
Outline

- ✓ Introduction and Previous Work
- Motivation for Routing Awareness
- Nature of the New TSP
- TSP Solvers
- Routing Aware Scan Chain Ordering
- Experiments
- Results and Conclusions

Motivation for Routing Awareness



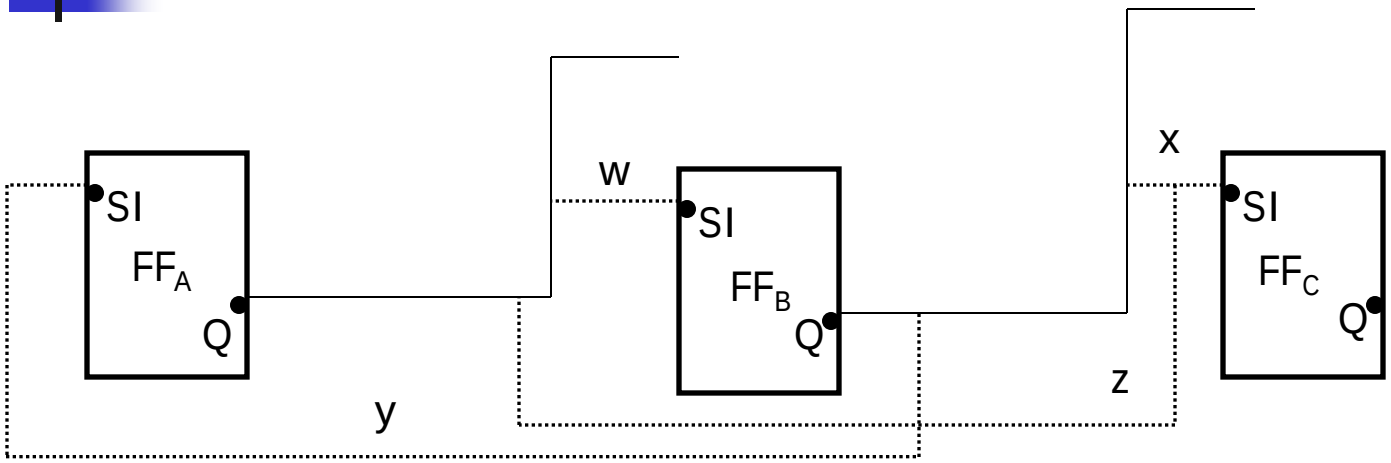
- Scan chain TSP costs should be based on wirelength estimate for the scan connection.
- A FF output pin will have a fanout routing tree.
- True routing distance to connect Q_B to SI_A will be *A”B”*.
- A routing aware scan chain ordering is likely to be different than one based on placement.



Outline

- ✓ Introduction and Previous Work
- ✓ Motivation for Routing Awareness
- Nature of the New TSP
- TSP Solvers
- Routing Aware Scan Chain Ordering
- Experiments
- Results and Conclusions

Nature of the New TSP



Note that

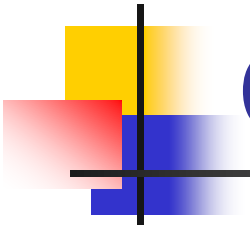
- AB ≠ BA
- AB + BC < AC

Cost of connections:

- Q_A to SI_B: w = AB
- Q_B to SI_C: x = BC
- Q_A to SI_C: z = AC
- Q_B to SI_A: y = BA

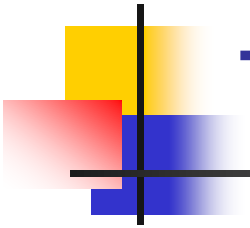
Nature of the New TSP

- Asymmetry
  - E.g. AB ≠ BA
  - Cell-to-cell distance metric was completely symmetric while pin-to-pin metric was almost symmetric
- Non-metricity
  - Triangle inequality is not obeyed. E.g. AB + BC < AC
  - Cell-to-cell metric was metric while pin-to-pin metric was almost metric
- The new TSP formulation can be highly asymmetric and very non-metric



Outline

- ✓ Introduction and Previous Work
- ✓ Motivation for Routing Awareness
- ✓ Nature of the New TSP
- TSP Solvers
- Routing Aware Scan Chain Ordering
- Experiments
- Results and Conclusions



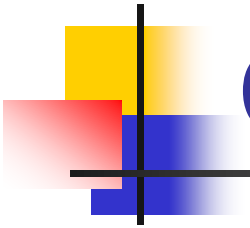
TSP Solvers

- Due to large asymmetry and non-metricity of the TSP instance, standard symmetric TSP solvers do not give good results
- Eighth DIMACS implementation challenge for ATSP ended in 2002 [JohnsonGM02]
  - Iterated Lin-Kernighan based LKH [Helsgaun00] was reported to give best tours
- We use iterated *ScanOpt* from the GSRC Bookshelf which has results comparable to LKH-1.2 and is tailored to scan chain optimization

TSP Solvers: ScanOpt

Test Case	Tour Cost (μm)		Run Time (sec.)	
	ScanOpt	LKH	ScanOpt	LKH
A (pin-to-pin)	21609	20632	1441	5670
A (pin-to-net)	9297	7511	2149	2717

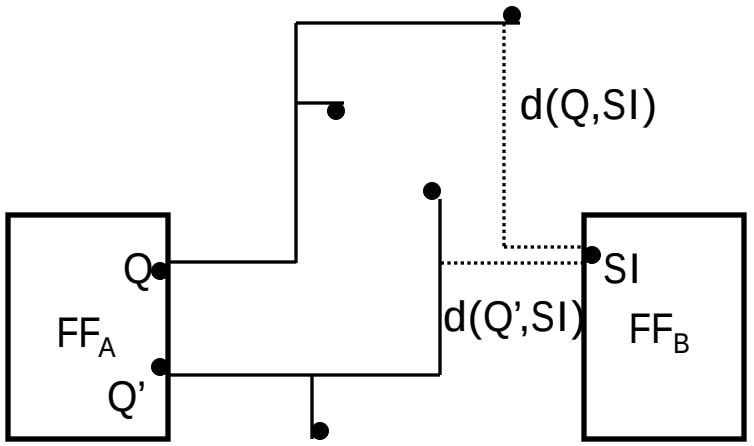
- Large step Markov chain (LSMC) methods for solving TSP alternately apply a local optimization procedure *Descent* followed by a “kick move” which perturbs the local minimum to obtain the starting solution for the next *Descent* application
- ScanOpt is a LSMC implementation based on the restricted 2,3-opt moves of [BoeseKT94] for solving an ATSP



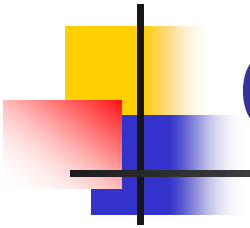
Outline

- ✓ Introduction and Previous Work
- ✓ Motivation for Routing Awareness
- ✓ Nature of the New TSP
- ✓ TSP Solvers
- Routing Aware Scan Chain Ordering
- Experiments
- Results and Conclusions

Routing Aware Scan Chain Ordering

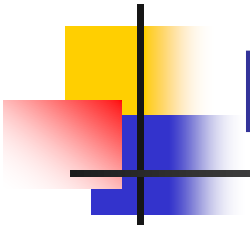


- Incremental routing cost based on existing or anticipated routing
- Considers both Q and Q' outputs for the minimum wirelength connection
- Driven by global routing or trial detailed routing
- We calculate the scan connection cost from the routed segments in the detailed routed DEF netlist



Outline

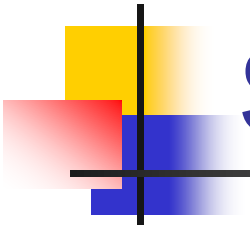
- ✓ Introduction and Previous Work
- ✓ Motivation for Routing Awareness
- ✓ Nature of the New TSP
- ✓ TSP Solvers
- ✓ Routing Aware Scan Chain Ordering
- Experiments
- Results and Conclusions



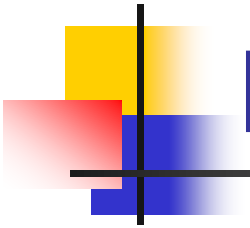
Experiments: Tools

- Placement: Cadence *Qplace v5.1.68*
- Detailed Routing: Cadence *Wroute v2.2.31*
- Our TSP Solver: *ScanOpt*
- Industry standard scan chain ordering: *QPlace* or Cadence *Silicon Ensemble v5.3.125 (SE)*
- We do not use incremental routing due to poor results of running WRoute in incremental or ECO mode: an observation confirmed in [KahngM00]

Experiments: Commercial Scan Chain Ordering



- To confirm absence of routing awareness in SE or QPlace scan stitching, we use the tools to order scan chains before and after detailed routing
- We then extract the scan orders from the routed DEFs
- Pre and post route scan chain orderings by the tools is exactly the same. Hence we infer that these tools (and to best of our knowledge, all others) do not use any routing information

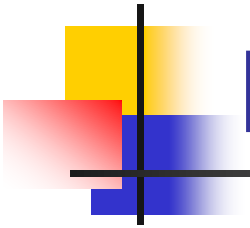


Experiments: Flows

1. The baseline place&route flow w/o scan insertion
2. Placement based scan chain ordering by *SE*
3. Placement based ordering by *QPlace*
4. Placement based ordering by *ScanOpt*
5. Routing driven scan chain ordering using *ScanOpt*

- Timing-driven and non-timing-driven versions of all place and route flows

Experiments: Routing Driven Flow

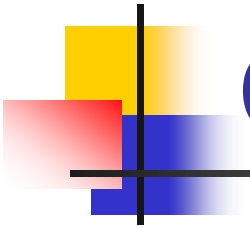


1. Trial route placed DEF netlist
2. Construct the ATSP cost matrix by computing pairwise minimum pin-to-net distances
3. Find the TSP tour using ScanOpt
4. Input the scan chain order into the *placed* DEF using the ORDERED construct of DEF
5. Attach scan nets
6. Final route the placed DEF with scan nets

Experiments: Testcases

Test Case	No. of Cells	No. of Scan FFs	#Scan Chains	Die Area mm^2	# Metal Layers
A/A_{swap}	6390	1226	2	0.526	4
A_{expand}	6390	1226	2	0.632	4
B/B_{swap}	40350	1975	1	6.875	4
B_{expand}	40350	1975	1	8.373	4
C/C_{swap}	34235	4550	10	3.846	4
C_{expand}	34235	4550	10	5.611	4

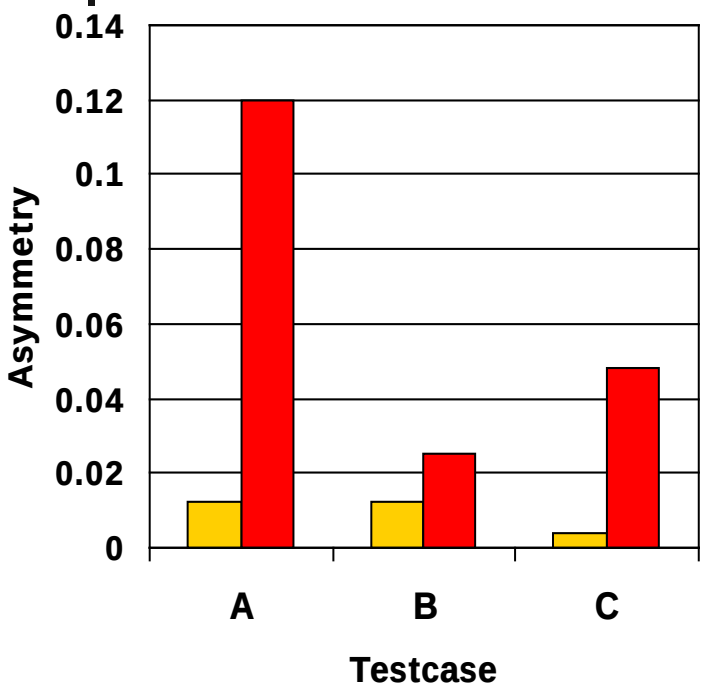
- Industry LEF/DEF testcases
- X_{swap} derived from X by random swapping of FF placements
- X_{expand} obtained from X by expanding the site-map by 20%

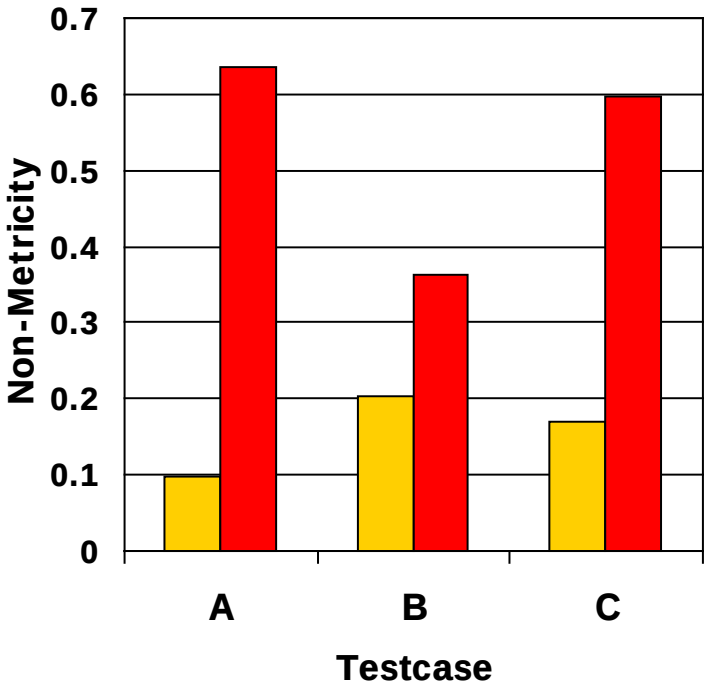


Outline

- ✓ Introduction and Previous Work
- ✓ Motivation for Routing Awareness
- ✓ Nature of the New TSP
- ✓ TSP Solvers
- ✓ Routing Aware Scan Chain Ordering
- ✓ Experiments
- Results and Conclusions

Results: Distance Metrics

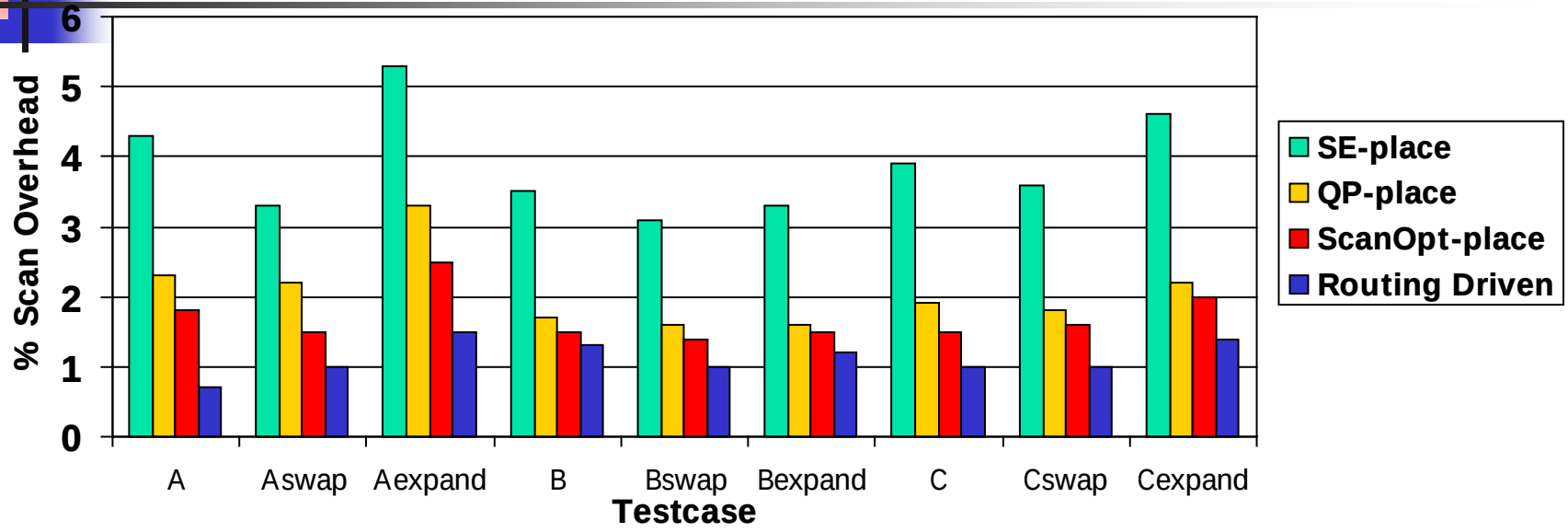




- **Asymmetry measure =**

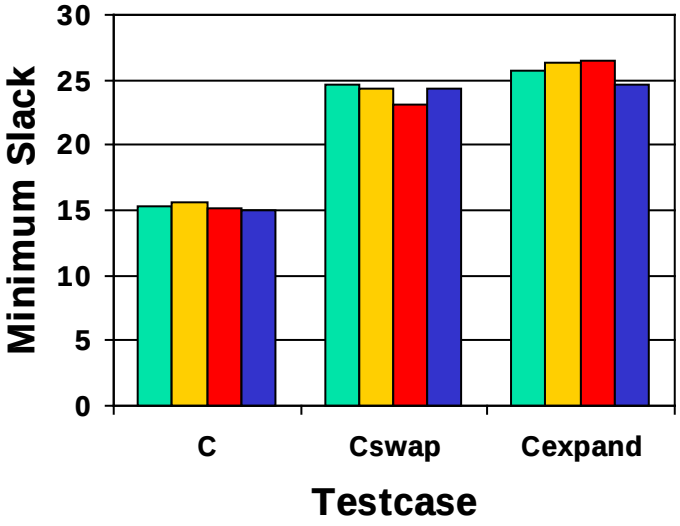
  (average(| d(i,j) – d(j,i) |)) / (average(| d(i,j) + d(j,i) |))
- **Metricity measure = average((d(i,j) -min(d(i,j),min(d(i,k)+d(k,j))) /d(i,j))**
- **Asymmetry and non-metricity for cell-to-cell distances is zero**

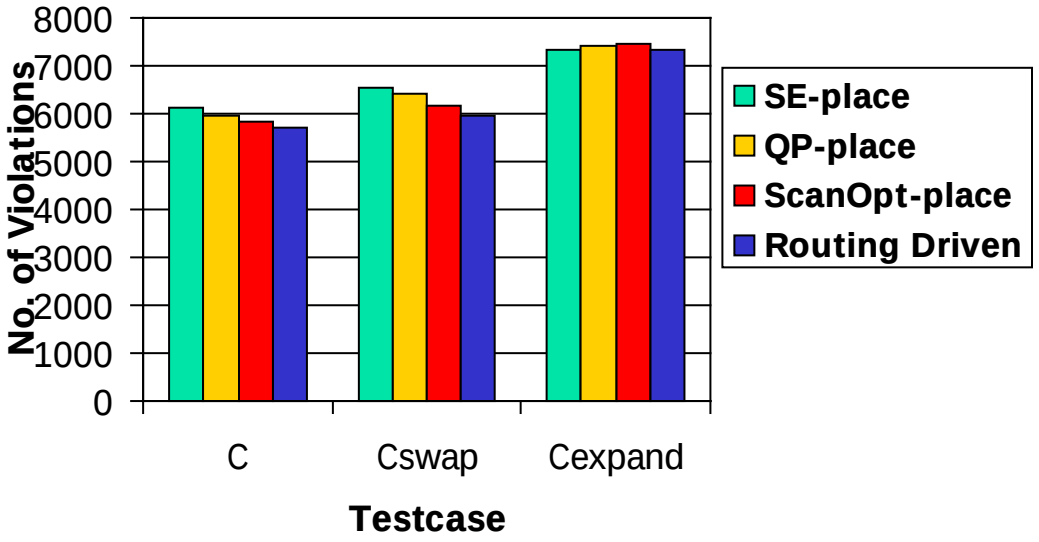
Results: Wirelength



- Flow VII (routing driven) and Flow IV (placement driven) both use *ScanOpt*
- Routing driven ordering consistently gives much smaller scan wirelength than any of the industry flows
- *QPlace* ordering is better than *SE* ordering

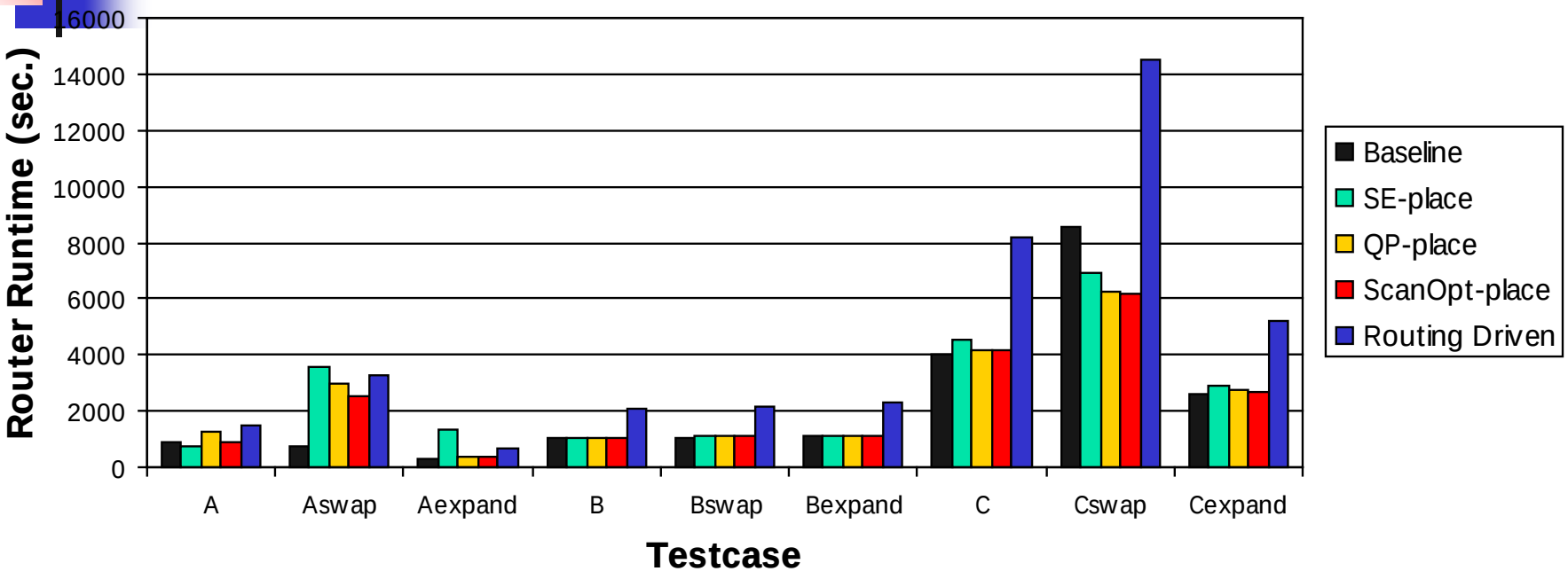
Results: Timing



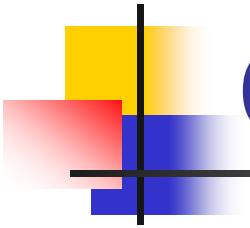


- We measure quality of timing by minimum slack and the number of timing violations
- Our aim is wirelength reduction
- Timing is not worse than the other flows

Results: Runtime



- Router runtimes normalized to 143MHz Sun Ultra-I are reported
- CPU time for Flow VII is sum of initial trial and final router runs
- For Flow VII routing is done from scratch to route the scan chain. No incremental routing is used



Conclusions

- A substantial reduction in wirelength (20%-85%) impact of scan is achieved by routing aware scan chain ordering
- Despite being timing oblivious, routing-aware flow does not significantly worsen the timing
- Runtime overheads of routing awareness can be reduced substantially if industry routers are able to deal better with incremental optimizations
- Timing aware extensions are possible but rely on controllability of the router. See the ISQED'03 publication "A Proposal for Routing-Based Timing-Driven Scan Chain Ordering"

Routing-Aware Scan Chain Ordering

Puneet Gupta and Andrew B. Kahng
(Univ. of California at San Diego, La Jolla, CA, USA.),
Stefanus Mantik
(Cadence Design Systems Inc., San Jose, CA, USA.)
Email: { puneet@ucsd.edu, abk@ucsd.edu, smantik@cadence.com }

Supported by MARCO Gigascale Research Center and Cadence Design Systems