MARCH: MAze Routing Under a Concurrent and Hierarchical Scheme for Buses

Speaker:
Jingsong Chen
Authors:
Jingsong Chen, Ph.D. student, CSE, CUHK
Jinwei Liu, Ph.D. student, CSE, CUHK
Gengjie Chen, Ph.D. student, CSE, CUHK
Dan Zheng, Ph.D. student, CSE, CUHK
Evangeline F. Y. Young, Professor, CSE, CUHK

Outline

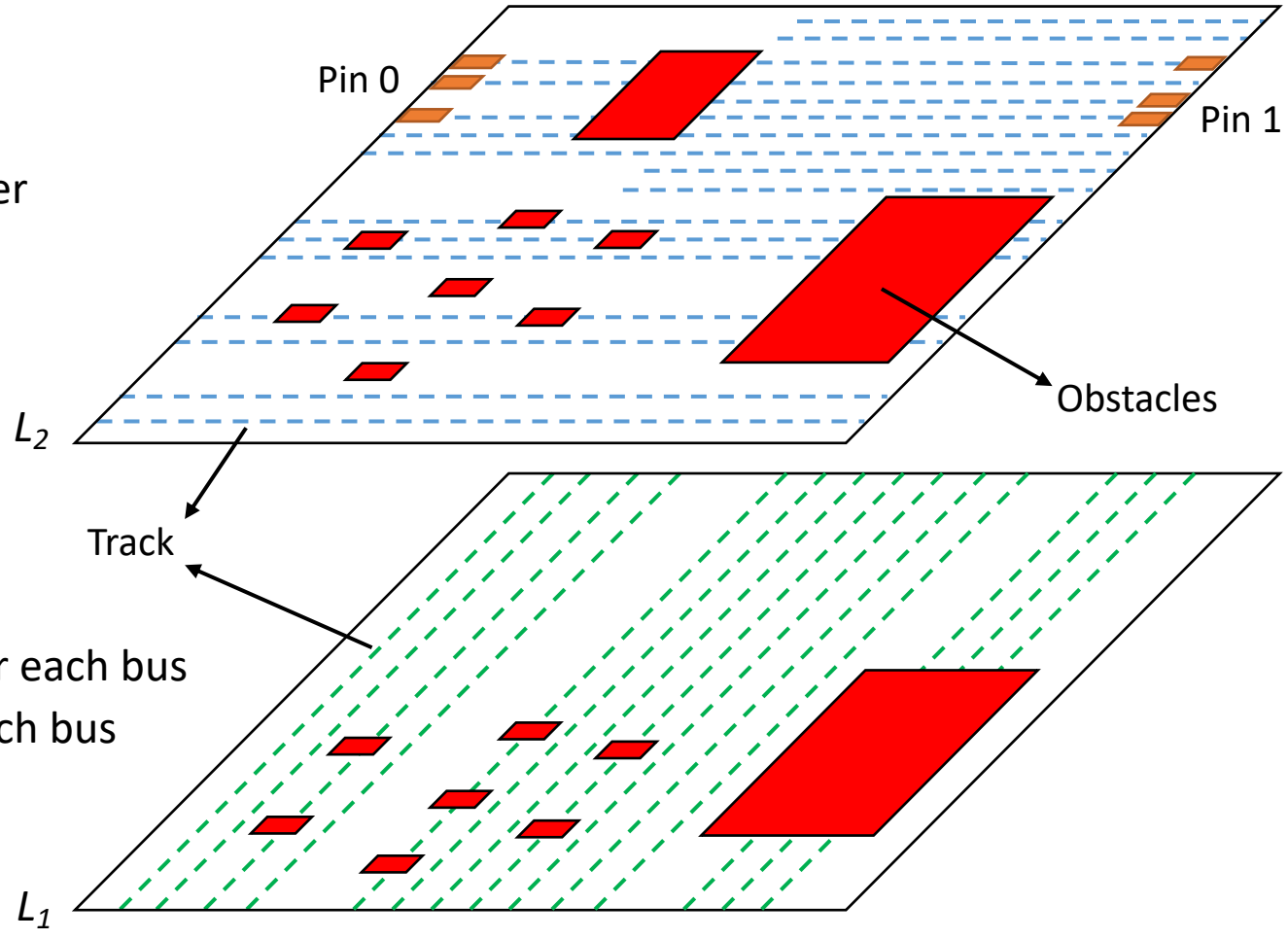
- Problem Formulation
- Our Methodology: MARCH
- Experimental Results
- Conclusion

Outline

- Problem Formulation
- Our Methodology: MARCH
- Experimental Results
- Conclusion

Problem Formulation

- Input:
  - A set of routing layers
    - With preferred track direction for each layer
    - With min spacing constraint for each layer
    - With design boundary for each layer
    - With a set of obstacles for each layer
  - A set of routing tracks
    - With wire width constraint for each track
  - A set of buses
    - With a set of bits to be routed on tracks for each bus
    - With width constraint for each layer for each bus
    - With pin shapes for each bit in a bus

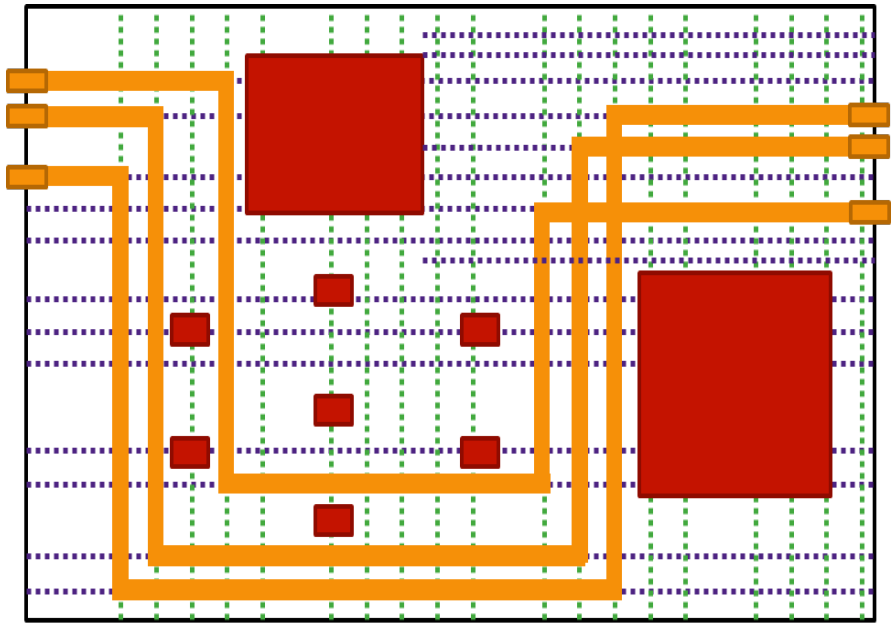


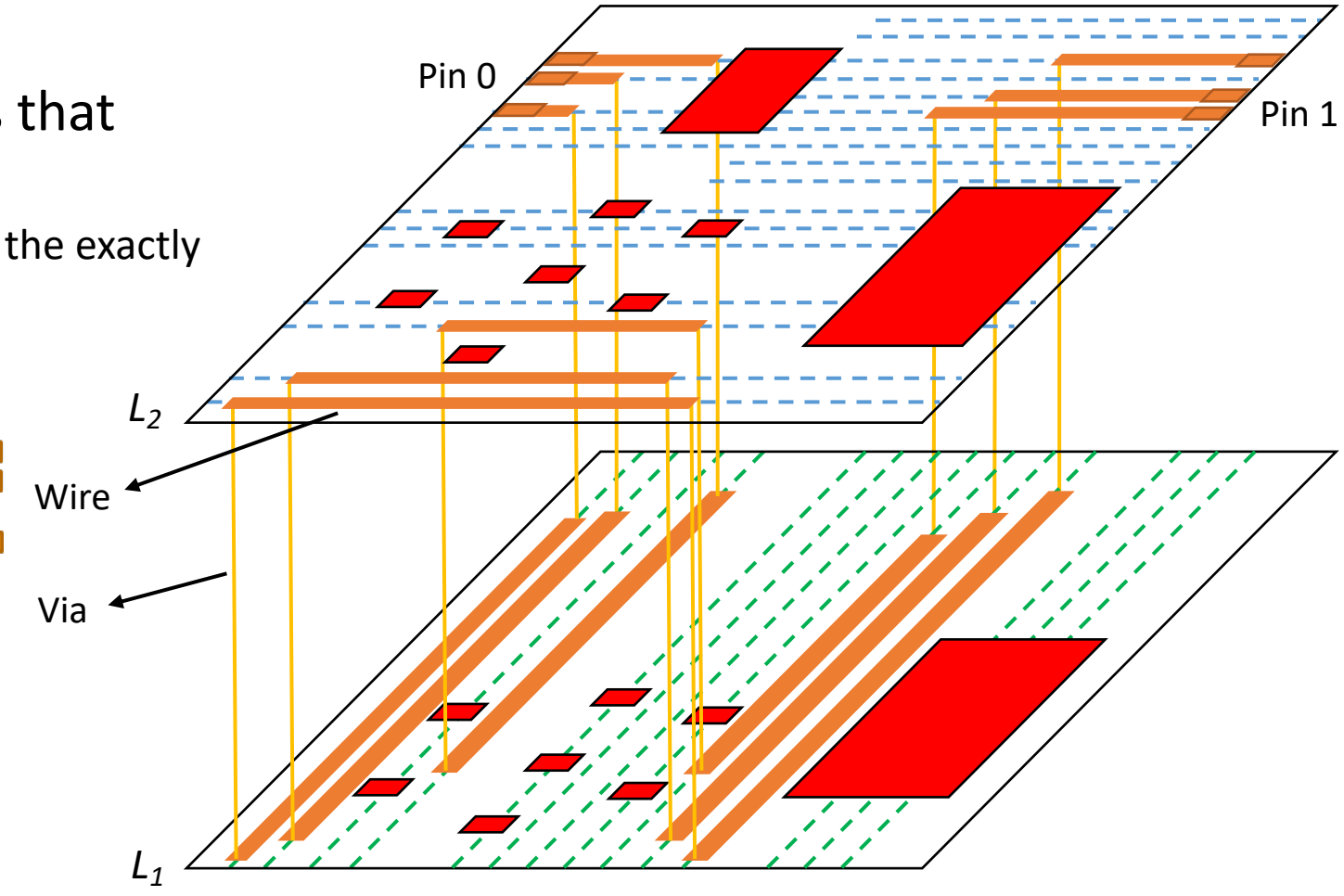
A toy case*

*This case comes from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: http://iccad-contest.org/2018/)

Problem Formulation

- Output:
  - A set of on-track wires and vias that connect pins for all buses
    - The bits in the same bus must share the exactly same topology



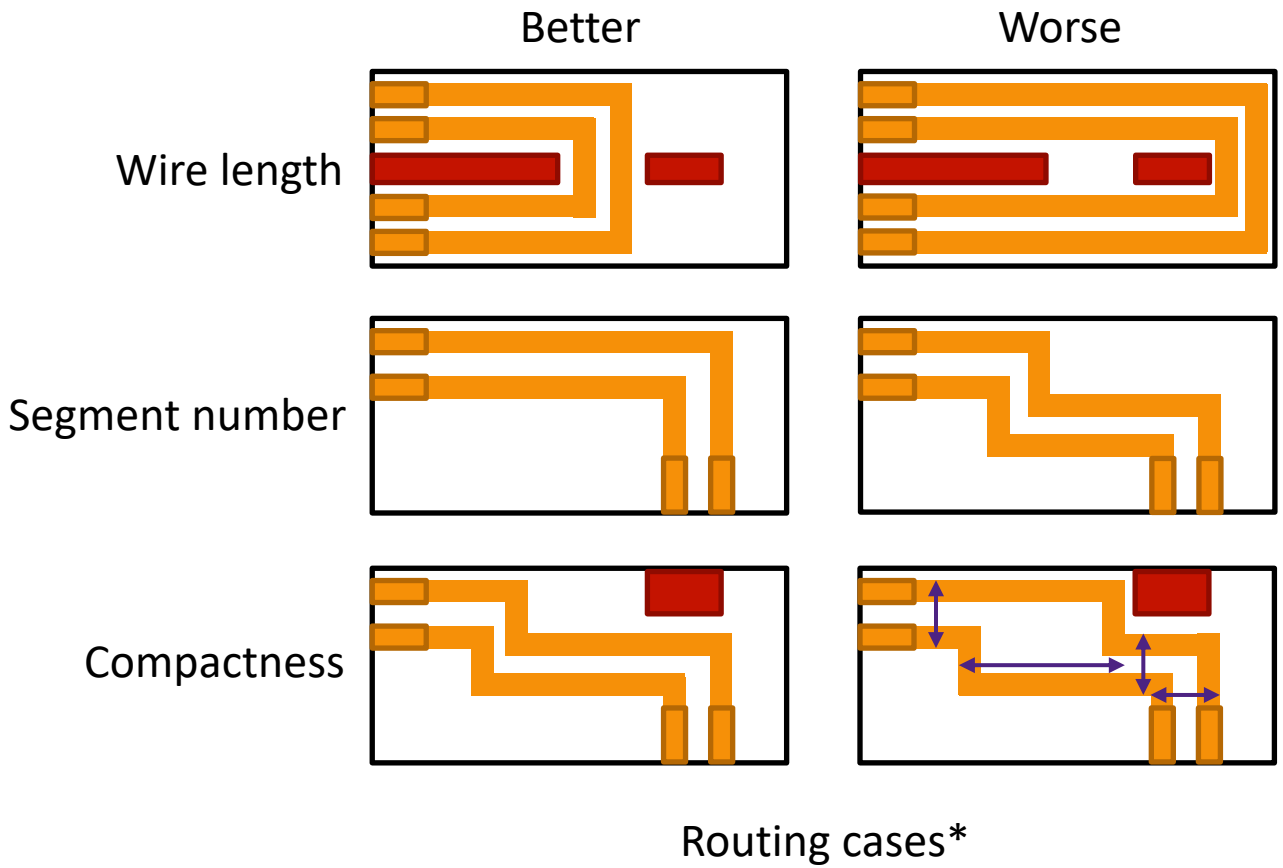


A toy case*

*This case comes from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: http://iccad-contest.org/2018/)

Problem Formulation

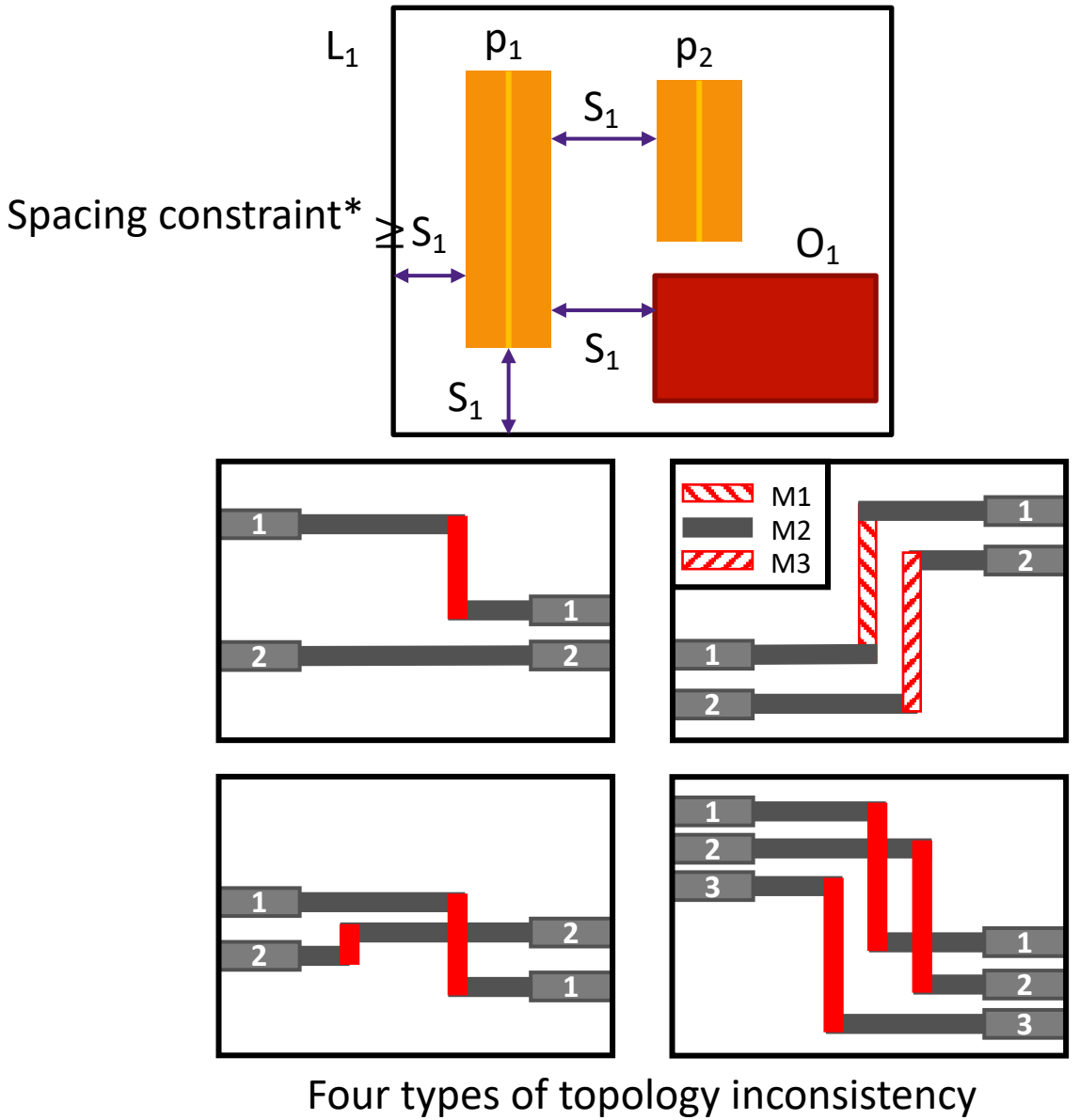
- Evaluation rule:
  - Overall cost consists of routing cost and penalty cost
    - Routing cost is the summation of :
      - Wire length cost: shorter -> better
      - Segment cost: less -> better
      - Compactness cost: more compact -> better



*These cases come from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: http://iccad-contest.org/2018/)

Problem Formulation

- Evaluation rule:
  - Overall cost consists of routing cost and penalty cost
    - Penalty cost is the summation of:
      - Spacing violation penalty
      - Routing failure penalty
        - Wire off-track
        - Track width violation
        - Bit open
        - Topology inconsistency



Four types of topology inconsistency

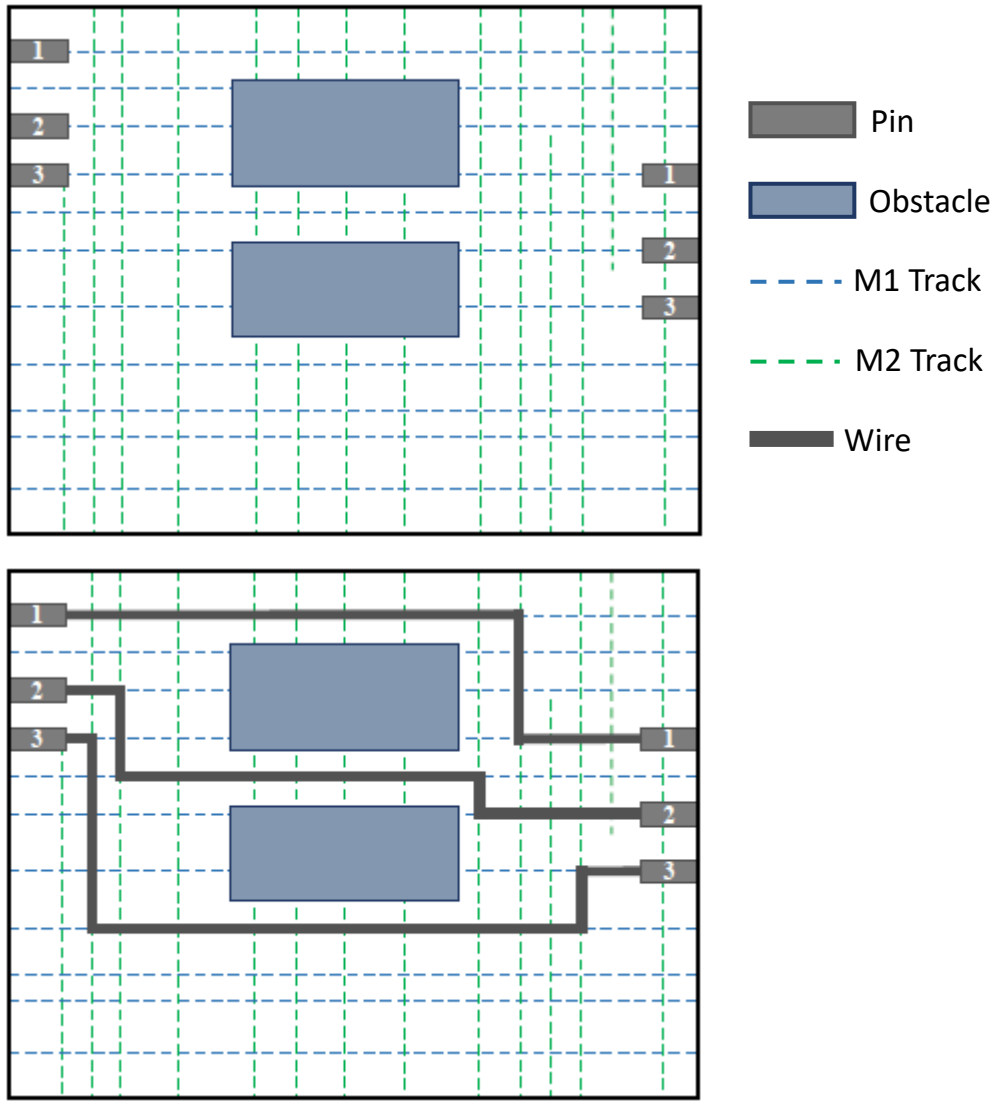
*This figure comes from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: http://iccad-contest.org/2018/)

Outline

- Problem Formulation
- Our Methodology: MARCH
- Experimental Results
- Conclusion

Our Methodology: MARCH

- If routing bit by bit:
  - Advantage: traditional routing methods can be applied naturally.
  - Disadvantage: topology consistency can hardly be maintained in a relatively complex routing environment.

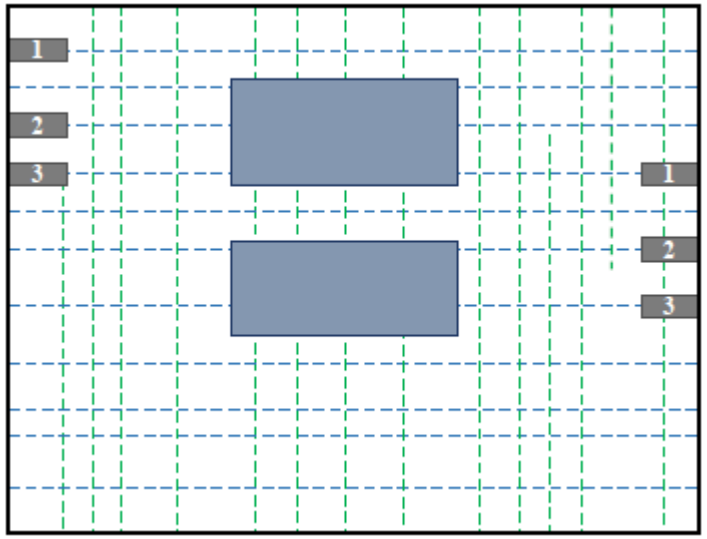


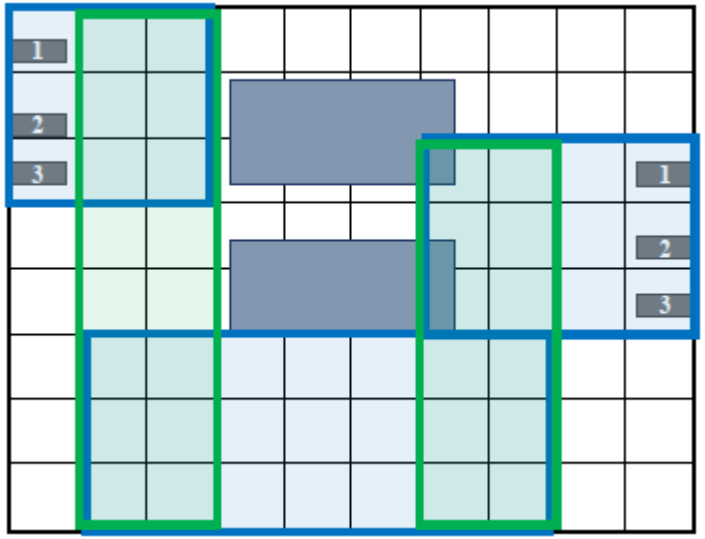
Our Methodology: MARCH

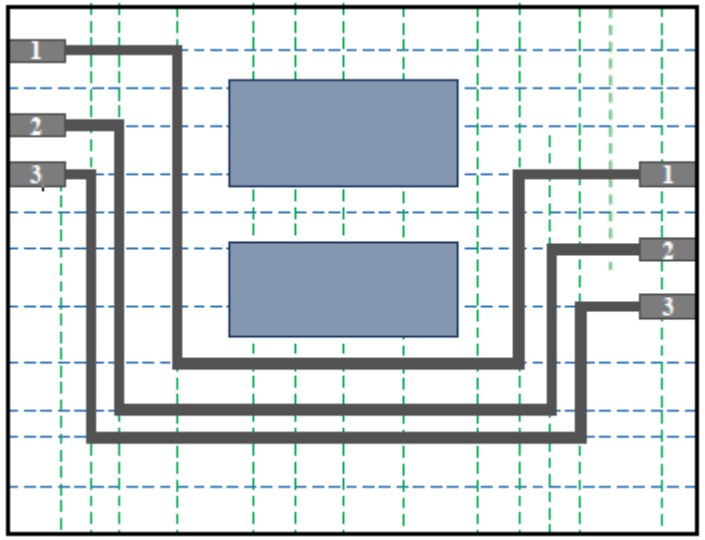
- Two key features of MARCH:
  - Hierarchically: a topology-aware path planning (coarse-grained) and a track assignment for bits (fine-grained).

    ➡ **EFFICIENCY**
  - Concurrently: route all the bits in a bus concurrently.

    ➡ **CORRECT-BY-CONSTRUCTION**

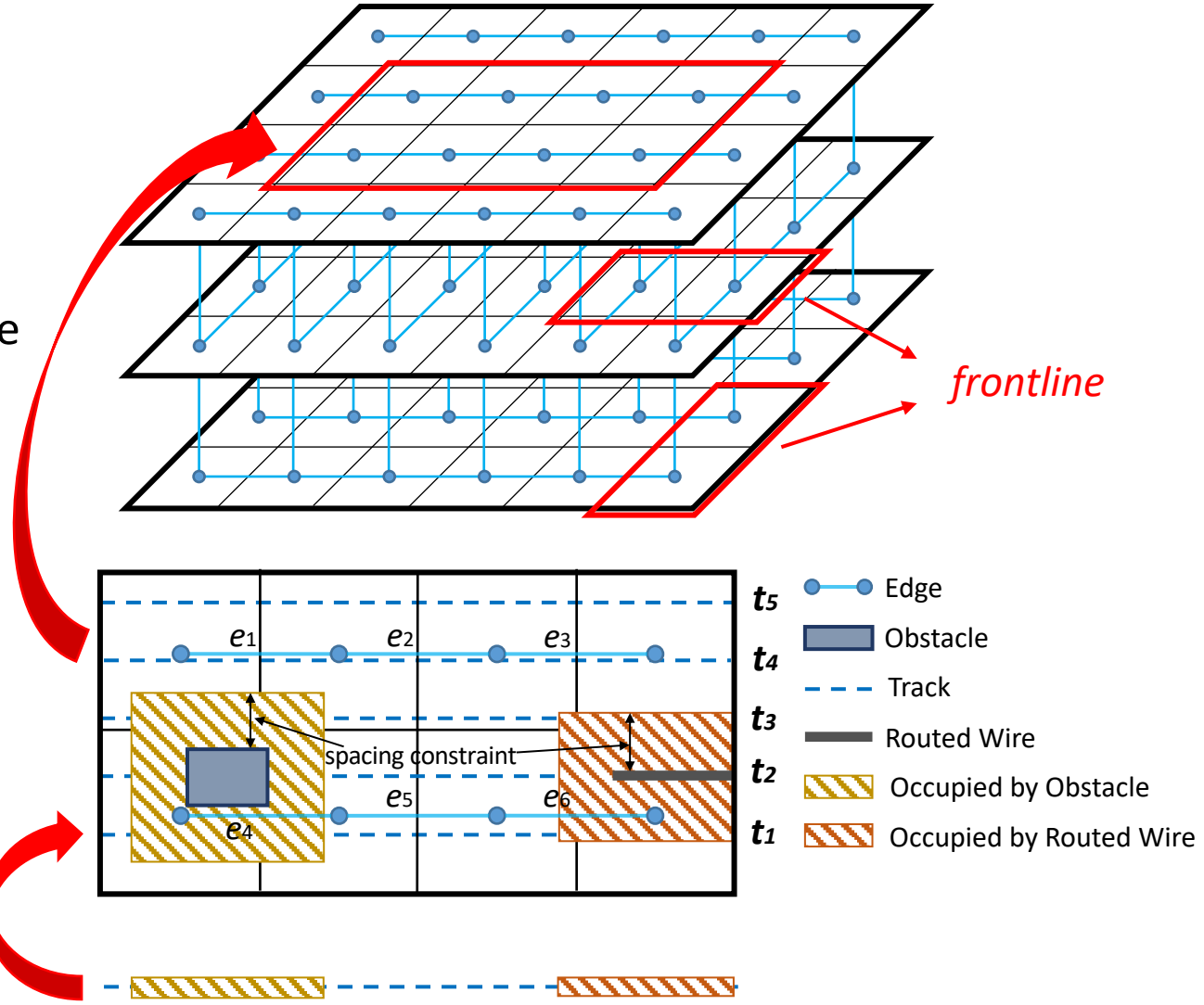






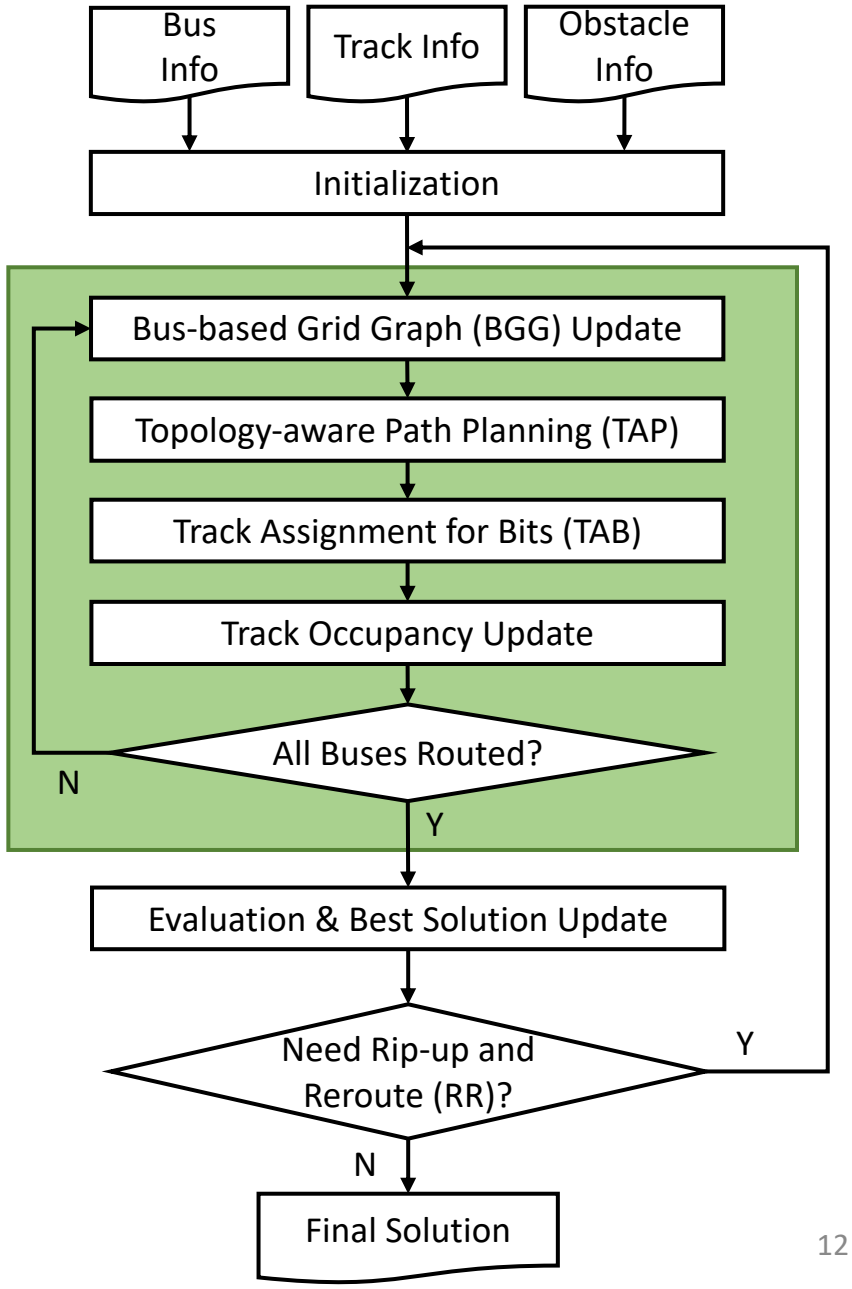
Our Methodology: MARCH

- Data Structures:
  - Bus-based Grid Graph (BGG):
    - A multilayer grid graph with uniform grid cells (G-cells)
    - A row of G-cells, named *frontline*, is the propagation unit in routing process.
    - Each edge stores edge capacity for the bus and history cost.
  - Track occupancy of each track:
    - Record the positions of the track segments which cannot be used since:
      - Occupied by obstacles
      - Occupied by the routed wires of the other buses



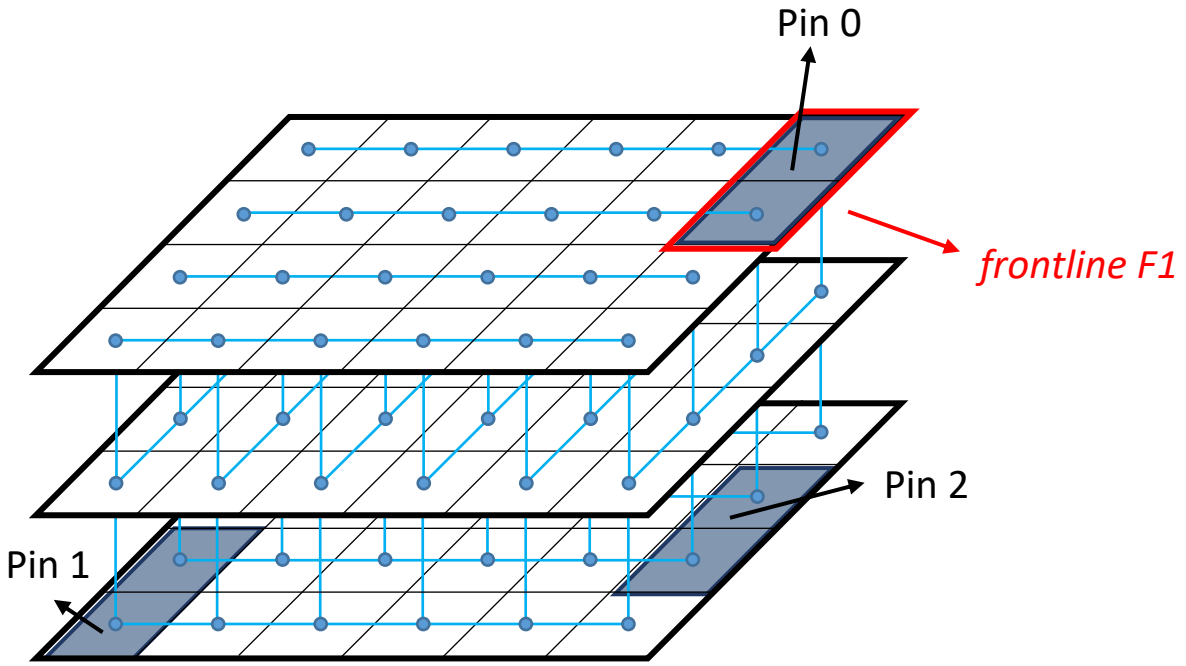
Our Methodology: MARCH

- The overall flow of MARCH:
  - Initialization
    - Construct a **BGG** with empty edge capacities
  - Inner loop for each bus *b*:
    - Update the **BGG** for *b*:
      - Edge capacities meeting width constraint
      - Pin locations
    - Generate a routing path consisting of a set of rectangular regions in topology-aware path planning (**TAP**)
    - Assign the track segments to the bits within each rectangular region in track assignment for bits (**TAB**)
    - Update the track occupancies of all the tracks by the routed *b*
  - Outer loop for rip-up and reroute (**RR**):
    - Add history cost to the edge of **BGG**
    - Enlarge the frontline size



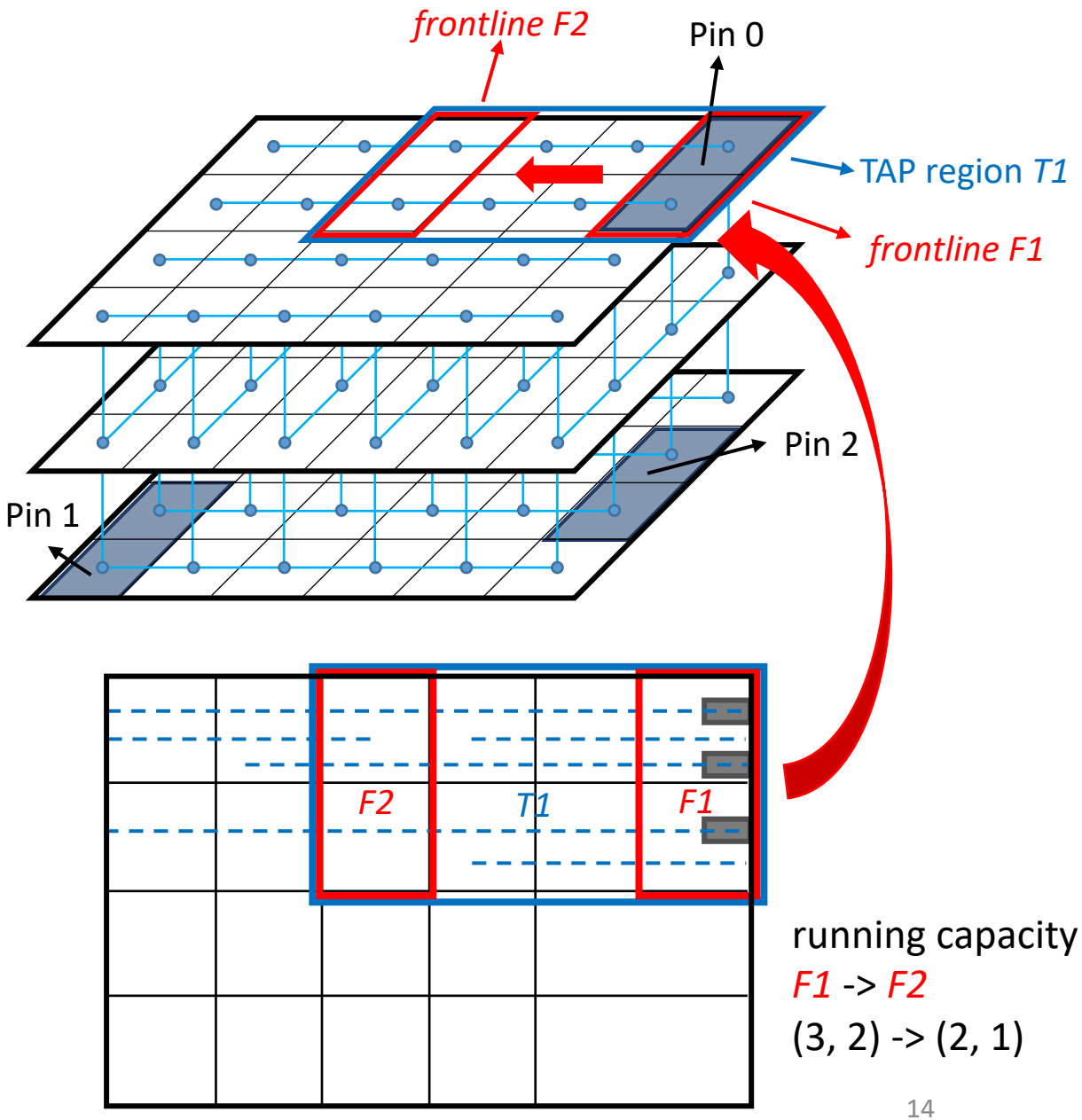
Our Methodology: MARCH

- Topology-aware Path Planning (TAP):
  - Same layer propagation
  - Layer switching
  - Build routing paths for multi-pin buses



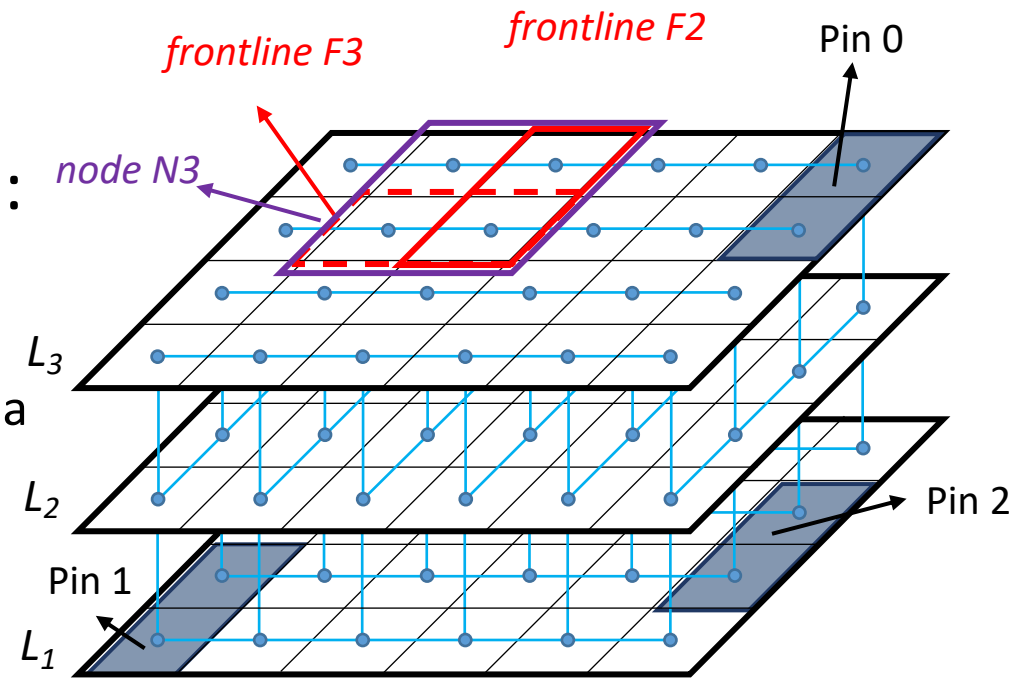
Our Methodology: MARCH

- Topology-aware Path Planning (TAP):
  - Same layer propagation:
    - Propagate the frontline along the track direction (*F1* -> *F2*)
    - Generate a TAP region (*T1*)
    - Maintain running capacity which will only decrease when some tracks are broken midway



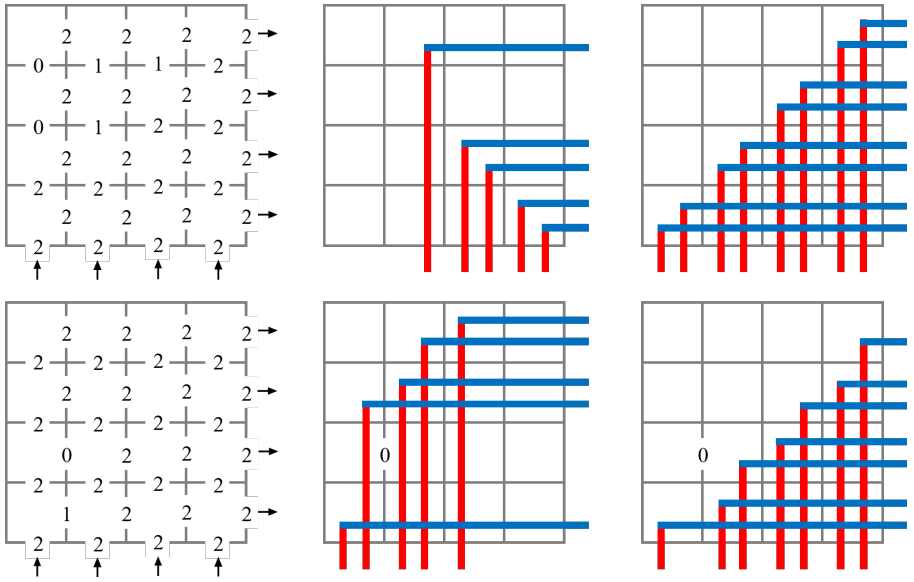
Our Methodology: MARCH

- Topology-aware Path Planning (TAP):
  - Layer switching:
    - Go from one layer to its upper layer or lower layer (*F2* on L_3 -> *F3* on L_2) through a switching *node*



Our Methodology: MARCH

- Topology-aware Path Planning (TAP):
  - Layer switching:
    - Go from one layer to its upper layer or lower layer (*F2* on L_3 -> *F3* on L_2)
    - Compute the max number of bits that can pass through the *node*
      - Without any bottleneck edge: trivial
      - With bottleneck edge(s):
        - With bit order unchanged:
          - Solved by a greedy method
        - With bit order changed:
          - Solved by the right recursive algorithm
          - The efficiency can be guaranteed by pruning

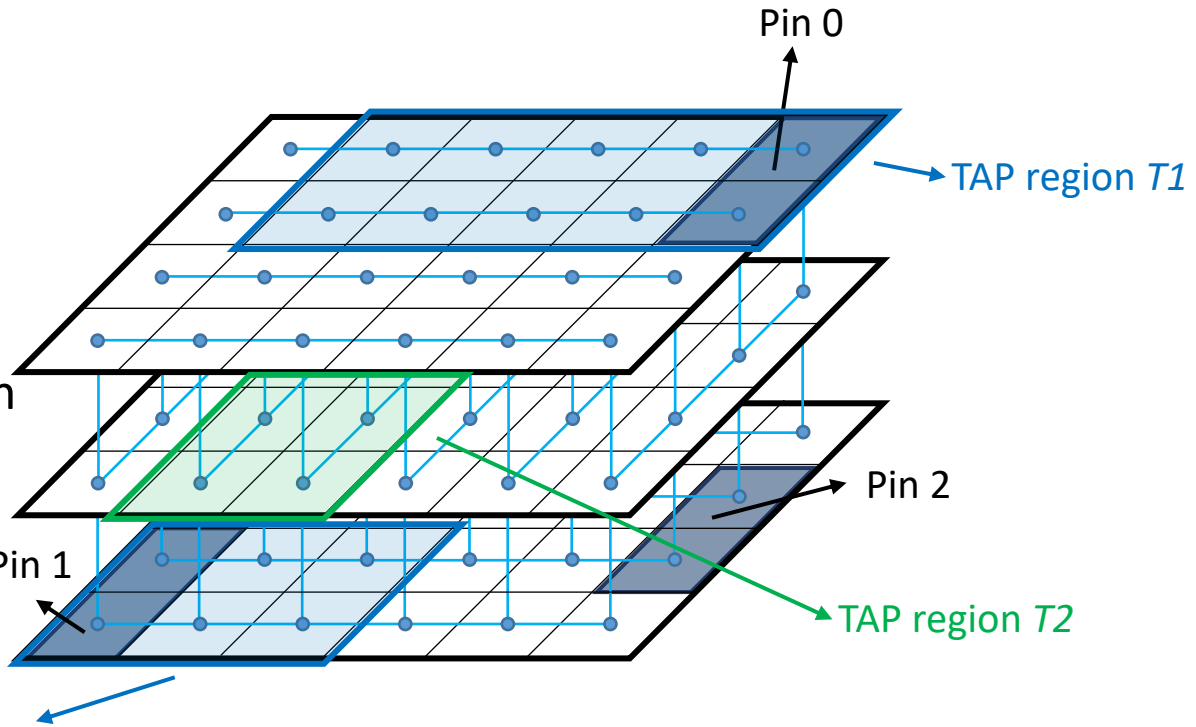


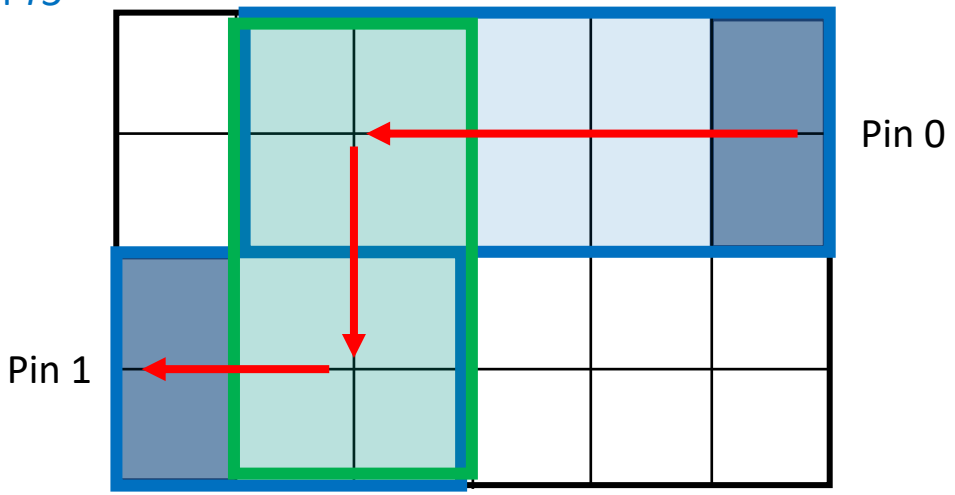
Algorithm 1 Compute $m \times n$ switching node capacity with flipped bit order

```
1:  function NodeCapacityFlip(i, j)
2:      if i >= n or j >= m then
3:          return 0
4:      capacity ← The maximum number of bits that can be routed
        from column i to row j
5:      Record capacity change
6:      if no more bits can enter column i then
7:          return capacity + NodeCapacityFlip(i + 1, j)
8:      if no more bits can exit row j then
9:          return capacity + NodeCapacityFlip(i, j + 1)
10:     return capacity + max(NodeCapacityFlip(i + 1, j),
        NodeCapacityFlip(i, j + 1))
```

Our Methodology: MARCH

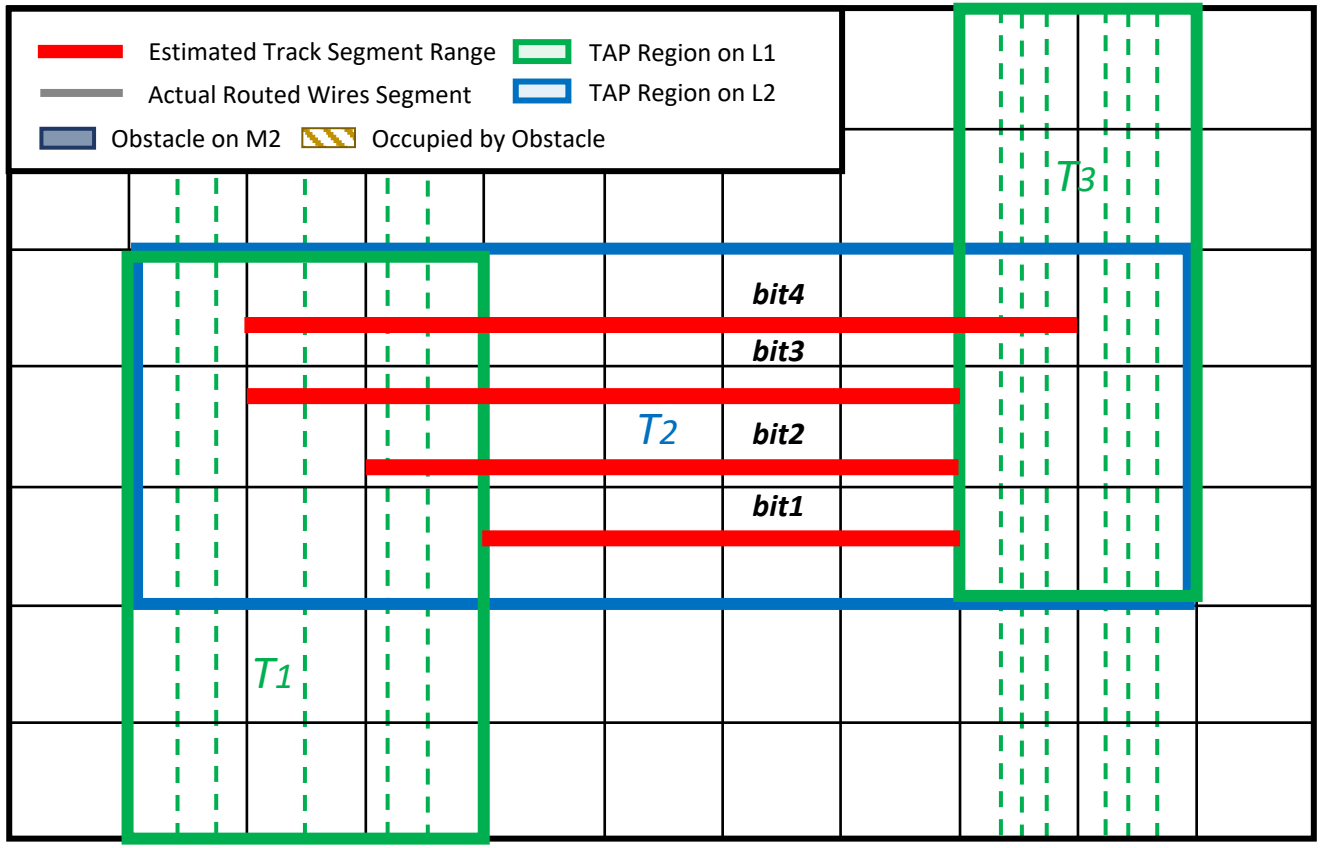
- Topology-aware Path Planning (TAP):
  - Build routing paths for multi-pin buses:
    - Find the path between the source pin and one of the sink pins through the propagation from the source pin
    - Start the propagation from the current path to connect to the next pin
    - Repeat this process until all the pins are connected





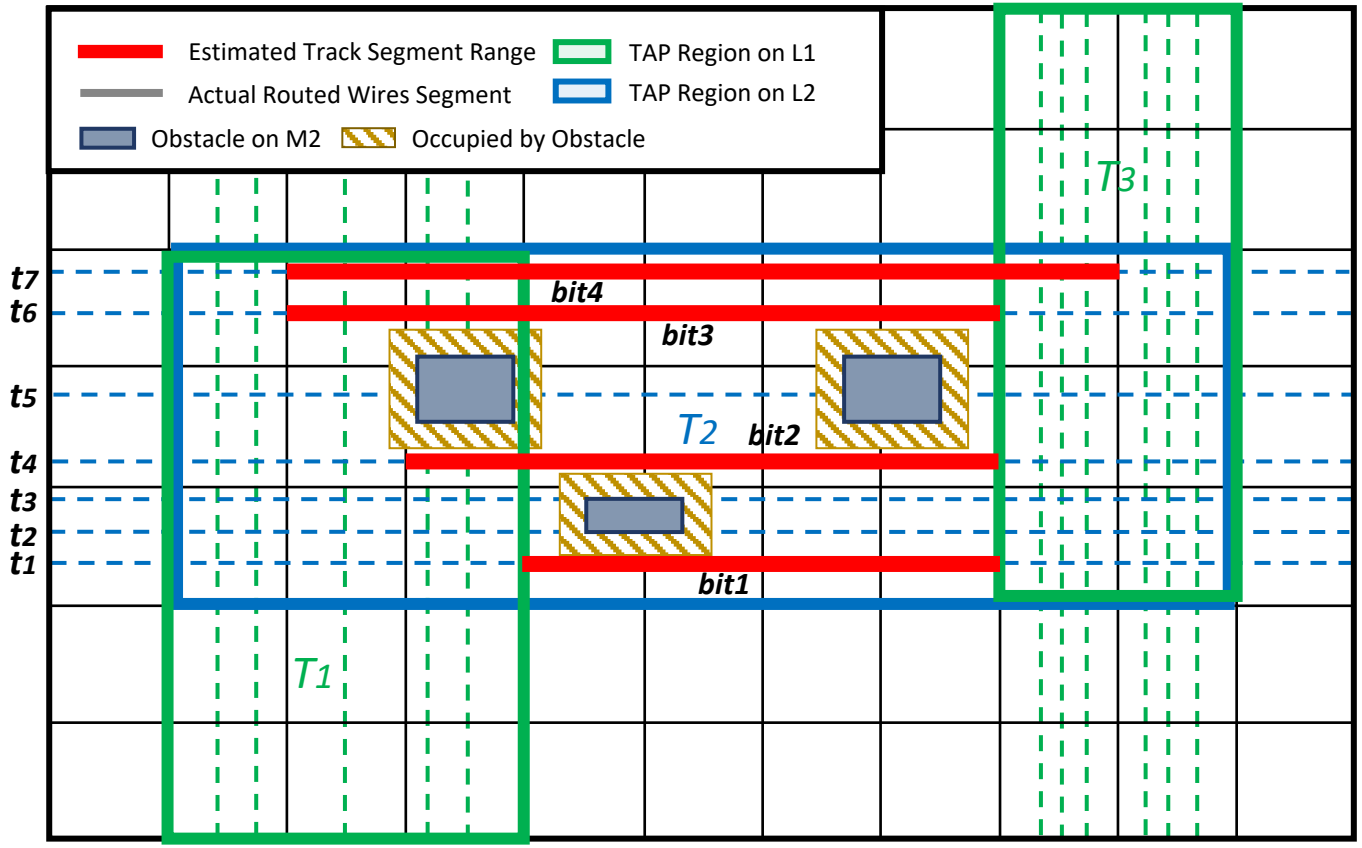
Our Methodology: MARCH

- Track Assignment for Bits (TAB)
  - Track segment range estimation:
    - By determining the column/row of G-cells where the bit will be routed



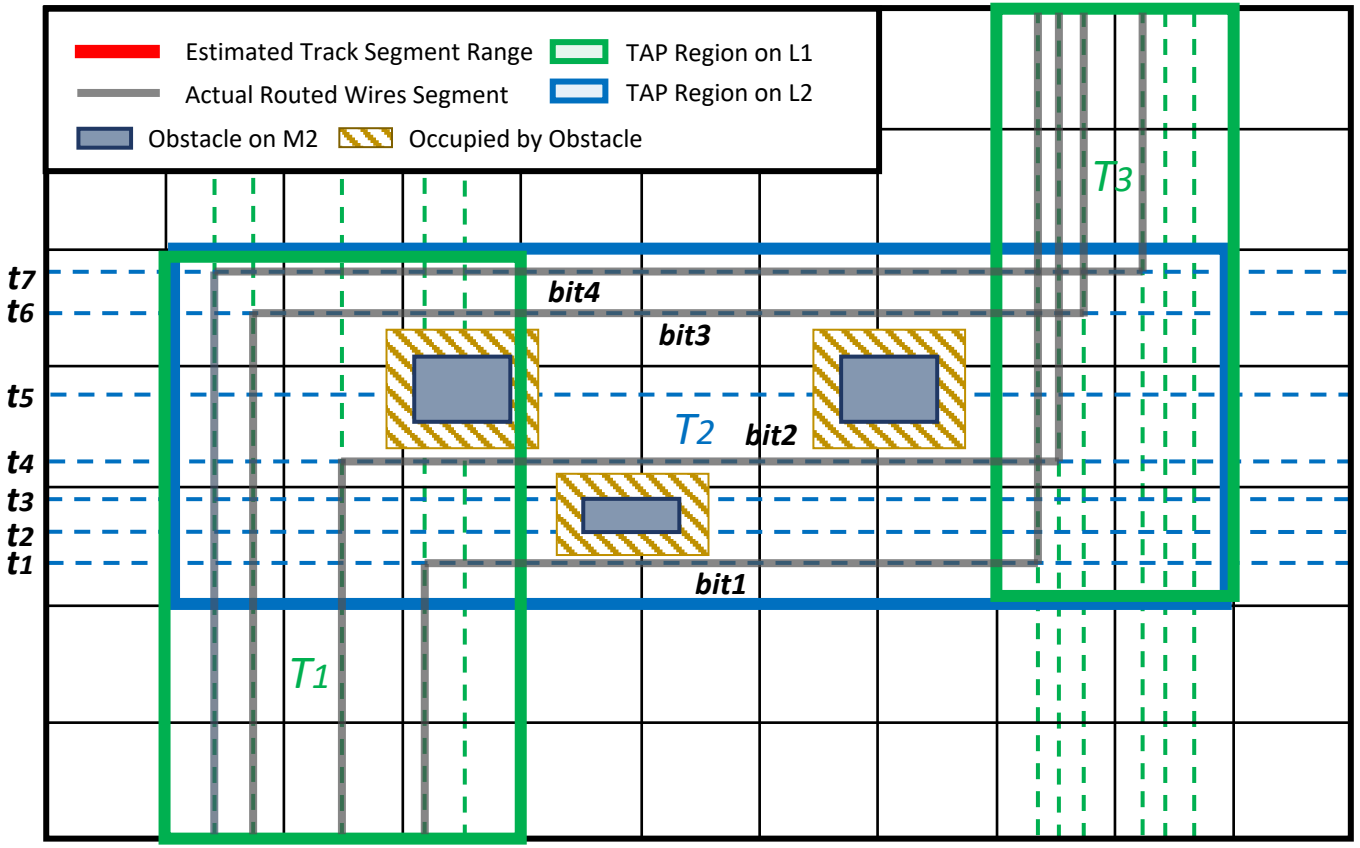
Our Methodology: MARCH

- Track Assignment for Bits (TAB)
  - Track segment range estimation:
    - By determining the column/row of G-cells where the bit will be routed
  - Exact track selection:
    - Three conditions:
      - Satisfy the width constraint
      - Have long enough track segment
      - With as less as possible spacing violations



Our Methodology: MARCH

- Track Assignment for Bits (TAB)
  - Track segment range estimation:
    - By determining the column/row of G-cells where the bit will be routed
  - Exact track selection:
    - Three conditions:
      - Satisfy the width constraint
      - Have long enough track segment
      - With less spacing violations
  - Exact track segment range assignment
    - See the actual routed wires



Our Methodology: MARCH

- Rip-up and Reroute Scheme (RR):
  - Add history cost to the edge of BGG:
    - To eliminate the congested regions on the BGG
    - $h_{new} = \alpha \cdot n_{vio} + \beta \cdot h_{old}$ where n_{vio} is the number of spacing violations on the edge, and α and β are weights.
  - Enlarge the frontline size:
    - To handle the insufficiency of violation-free routing resources
    - Enlarge the frontline size of the bus by **1** on the layer where:
      - The number of spacing violations in a TAP region is larger than the bit number in **2** successive RR iterations.

Outline

- Problem Formulation
- Our Methodology: MARCH
- Experimental Results
- Conclusion

Experimental Results

	Characteristics				Metric Weights				
	bus no.	**net no.**	**layer no.**	**track no.**	**w_{wire}**	**w_{seg}**	**w_{com}**	**w_{space}**	**w_{fail}**
beta1	34	1260	3	49209	5	1	5	8	2000
beta2	26	1262	3	49209	5	1	5	8	2000
beta3	60	665	3	22732	12	1	4	8	2000
beta4	62	698	3	22732	12	1	4	8	2000
beta5	6	1964	4	54150	8	1	5	8	2000
final1	18	1032	3	81226	10	1	5	10	2000
final2	70	1285	3	14209	10	1	5	10	2000
final3	47	852	4	21379	10	1	5	10	2000

Experimental Results

	First Place					Second Place					Third Place					MARCH				
	Croute	Cspace	Cfail	Ctotal	Time (s)	Croute	Cspace	Cfail	Ctotal	Time (s)	Croute	Cspace	Cfail	Ctotal	Time (s)	Croute	Cspace	Cfail	Ctotal	Time (s)
beta1	689	280	0	969	3600	701	5096	0	5797	-	641	8744	4000	13385	-	765	0	0	765	50
beta2	515	760	0	1275	3600	563	4904	0	5467	-	484	9472	2000	11956	-	578	0	0	578	9
beta3	1936	0	0	1936	71	2024	0	0	2024	-	1999	1928	0	3927	-	1942	0	0	1942	72
beta4	2192	0	0	2192	64	2271	0	0	2271	-	2250	1048	0	3298	-	2165	0	0	2165	39
beta5	119	1848	0	1967	3600	95	616	2000	2711	-	98	1216	2000	3314	-	118	1848	0	1966	12
final1	327	830	2000	3157	3317	367	2750	2000	5117	-	252	0	10000	10252	-	356	840	0	1196	352
final2	1824	4500	8000	14324	3600	1890	2990	8000	12880	-	1976	6910	0	8886	-	2071	1480	0	3551	199
final3	2966	490	10000	13456	3600	2678	300	2000	4978	-	4238	20	24000	28258	-	3313	150	0	3463	133
Avg. Ratio				2.130	105.45				3.731	-				7.832	-				1.000	1.000

*The scores of top 3 teams of IC/CAD 2018 contest are provided by the contest organizer. A binary is also obtained from the first place to get its runtime information.

Outline

- Problem Formulation
- Our Methodology: MARCH
- Experimental Results
- Conclusion

Conclusion

- We propose MARCH for bus routing:
  - Have a hierarchical framework (TAP & TAB) for efficiency
  - Route all the bits of a bus concurrently for topology consistency
  - Apply a RR scheme to reduce the routing congestion
  - Performance compared with the top contest teams
    - Reduce spacing violations greatly
    - Avoid any routing failure
    - Have competitive routing costs
    - Have a much shorter runtime

Thank you for attention!

Q & A