Routing DDR4 Interfaces Quickly and Efficiently

Michael Catrambone, Sr. Principal Product Engineer, Allegro PCB Products

PCB WEST 2016
Conference & Exhibition

cadence®

PCB Challenges as Interface Speeds Increase

- Today's challenges
  - Do more with less and deliver on time!
  - Time to market with product meeting ALL performance and design requirements
  - Increasing design complexities with advanced interfaces like XFI, XGMII, XAUI, DDR4, PCI Express® (PCIe®)
    - Requires an advanced set of electrical and physical constraints
  - The days of "connecting the dots" are long gone

- This paper will:
  - Provide an overview of DDR4 memory interfaces including topologies and constraints that need to be adhered to in order to meet timing requirements
  - Discuss new techniques designed to accelerate routing and tuning of high-speed signals quickly and efficiently
    - These techniques can be applied to component breakout, point-to-point routing, and tuning of signals

Developing an Action Plan

- Building a stable PCB design foundation
  - Careful planning of component placement and reserving space for pin escape (fanout)
  - Pin escape with interconnect topologies, routing, and any possible testing requirements in mind
  - Split plane planning during placement stage—too late once board is routed (power integrity issues)

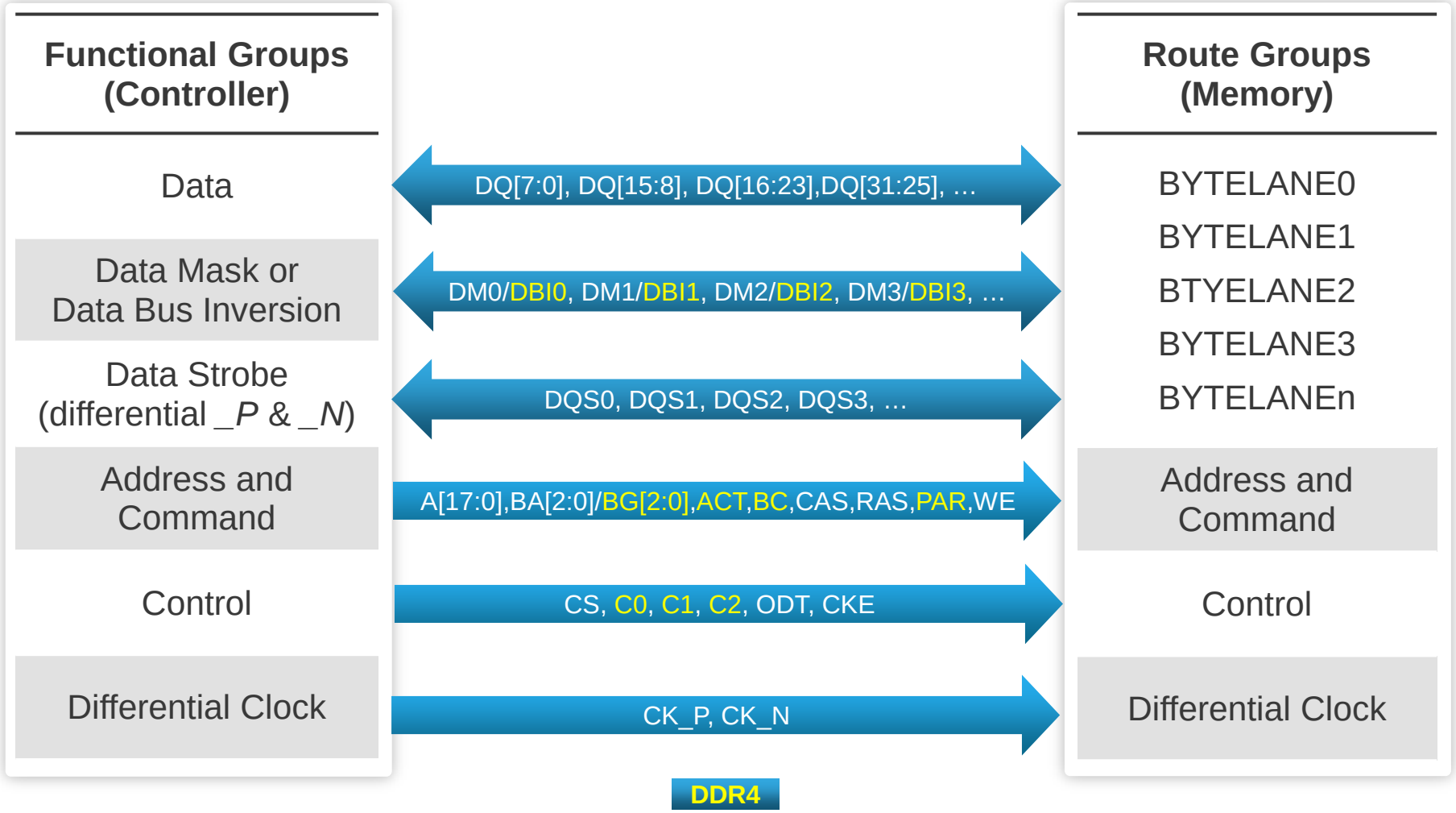
- Route flow planning between devices is critical to the success of any design
  - Most direct route path to accommodate all routing including maintaining all design clearances
  - Quick and precise component routing escape optimizing the order on both sides of the bus
  - Effective timing closure with goal guidance using interactive and automatic means

- Ultimately, today's designs dictate that you must be thinking about routing critical connections in earlier design stages while allocating adequate space to meet design/matching requirements and routing topologies

DDR4 Memory Interfaces Overview

DDR4 Memory Interfaces Overview

Functional group to route group mapping



DDR4 Memory Interfaces Overview
General design requirements

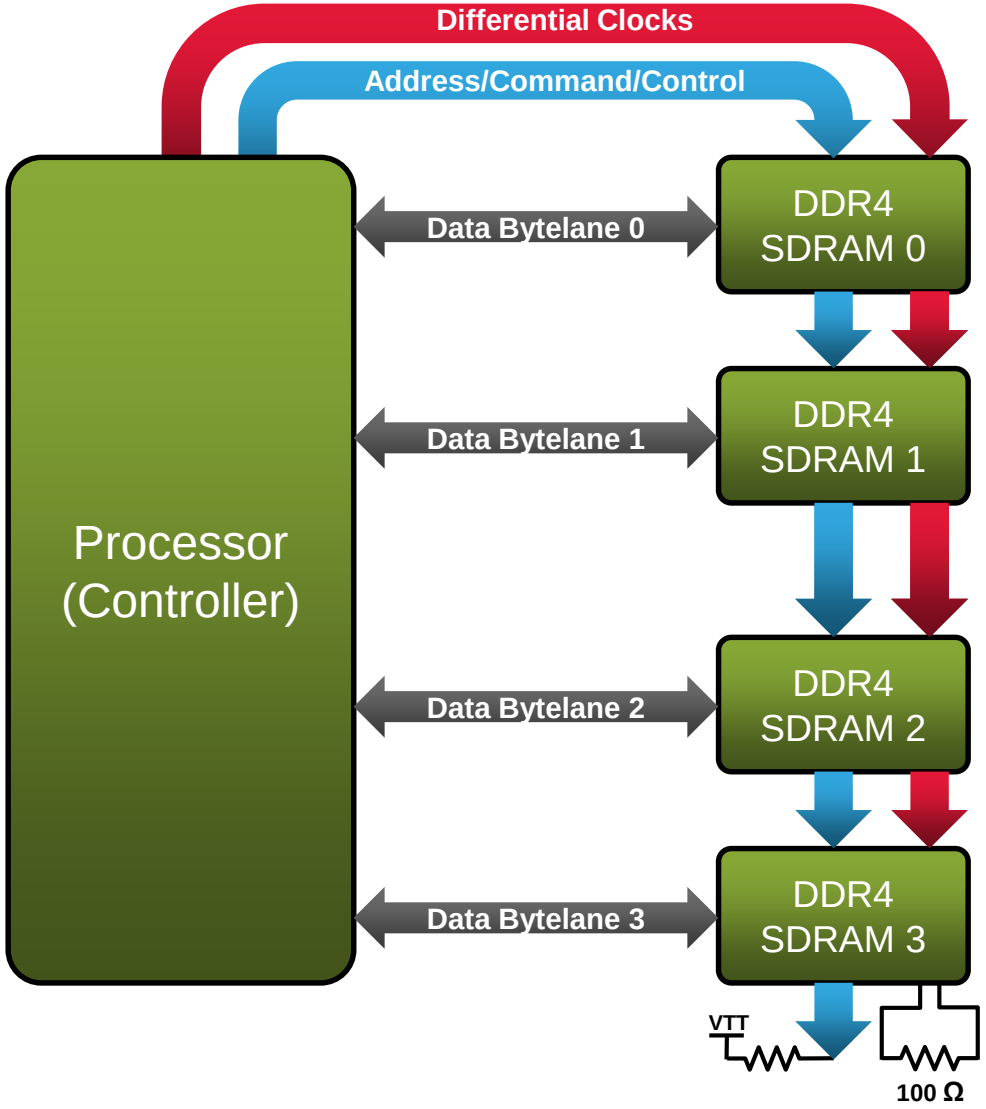
- Typical data bus structure:

BYTELANE0	DQ[7:0],	DM0/DBI0,	DQS_P0,	DQS_N0
BYTELANE1	DQ[15:8],	DM1/DBI1,	DQS_P1,	DQS_N1
BYTELANE2	DQ[23:16],	DM2/DBI2,	DQS_P2,	DQS_N2
BYTELANE3	DQ[31:24],	DM3/DBI3,	DQS_P3,	DQS_N3
BYTELANEn	DQ[<8bits>],	DMn/DBIn,	DQS_Pn,	DQS_Nn

- Data bytelane members should be routed on the same layer
- Address/command/control/differential clocks should be routed on the same layer, but if space issues arise they can be routed on different layers
  - Adjacent layers or layers referencing the same plane layer are preferred
- Address/command/control/differential clocks route topology
  - Routed using a daisy chain (fly-by) topology: Route from controller starting with Chip 0 thru Chip n routing in order by bytelane numbers
    - Chip 0 is the lower data bit (Bytelane0)/Chip n is the upper data bit (Bytelane3)

DDR4 Memory Interfaces Overview

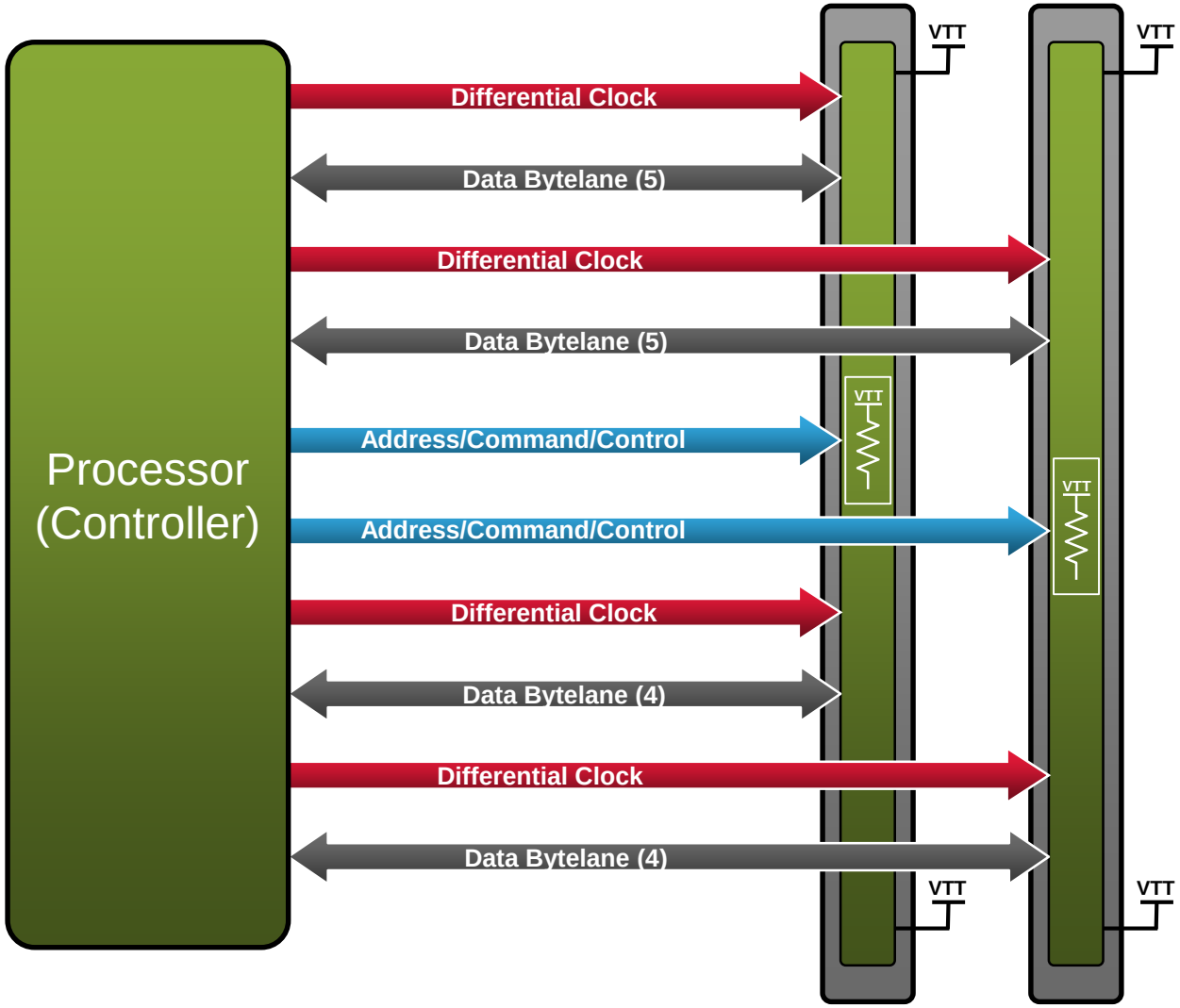
Bus topologies—On-board SDRAM



- Data bus termination
  - Series resistor termination can be used when point-to-point connection is in 2” to 2.5” range
    - Resistors located at center of transmission line
  - DRAM termination with direct connect using on-die termination (ODT)
    - Better signal quality and lower cost compared to using series resistor termination
- Clock termination
  - 100Ω differential terminator at last SDRAM device in chain

DDR4 Memory Interfaces Overview

Bus topologies—On-board two-UDIMM



- 1-cycle timing (1T) has two sets of address/command/control, driven by memory controller, connecting to each connector, as shown
- 2-cycle timing (2T) has one set of address/command/control connecting to both connectors
- Data bus is a point-to-point interface dedicated to one UDIMM module
  - One UDIMM per memory channel
- VTT termination resistors are not required on main board as they are built into DDR4 modules
- Two differential clocks per UDIMM

DDR4 Electrical Design Rules

DDR4 Design Rules
Electrical constraint targets

- Relative propagation delay
  - Data bytelane
    - *1 – 5mils* between all members inside of bytelane
  - Address/command/control
    - *100 – 200mils* between controller to first memory IC
    - *10 – 20mils* between memory ICs

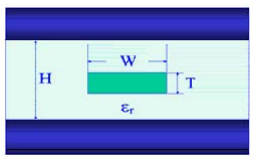
- Propagation delay
  - Normally not constraint controlled as it is driven by placement of memory ICs, which should be placed as close to controller as possible, normally between *1500 – 1750mils* from controller to first memory IC and *650 – 750mils* between memory ICs

- Differential phase tolerance
  - *1 – 5mils* for all data strobe and clock differential pairs

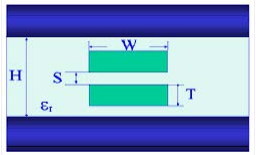
- **Disclaimer:** Design rules above are for reference only and should be treated as such—only tried and true way to determine interface design rules is with pre-/post-route simulations

DDR4 Design Rules
Impedance/design stack-up

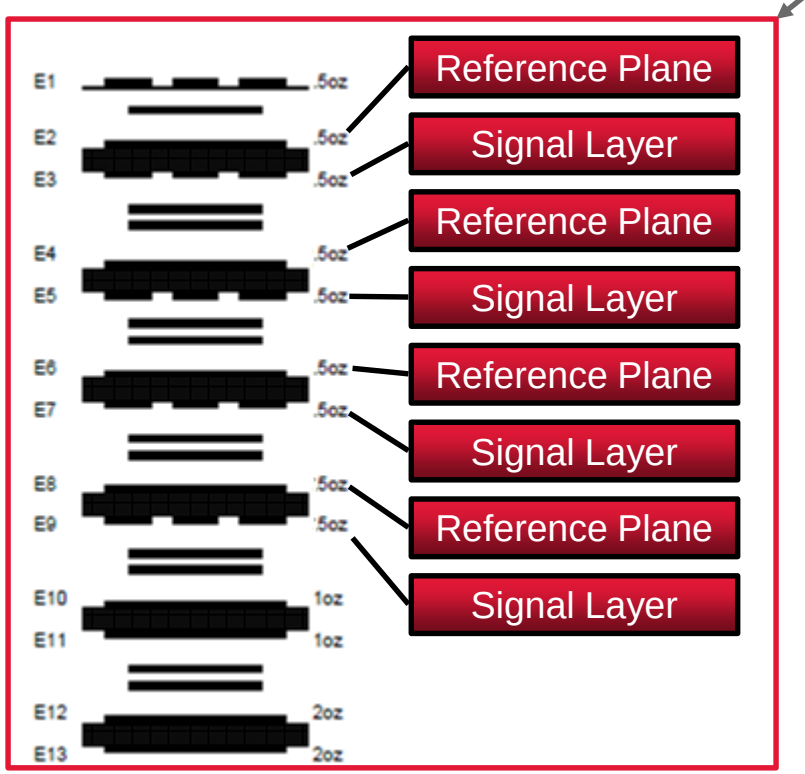
- Impedance requirements
  - Single-ended target = 50 – 60Ω
  - Differential-pair target = 100 – 120Ω
- Design stack-up considerations

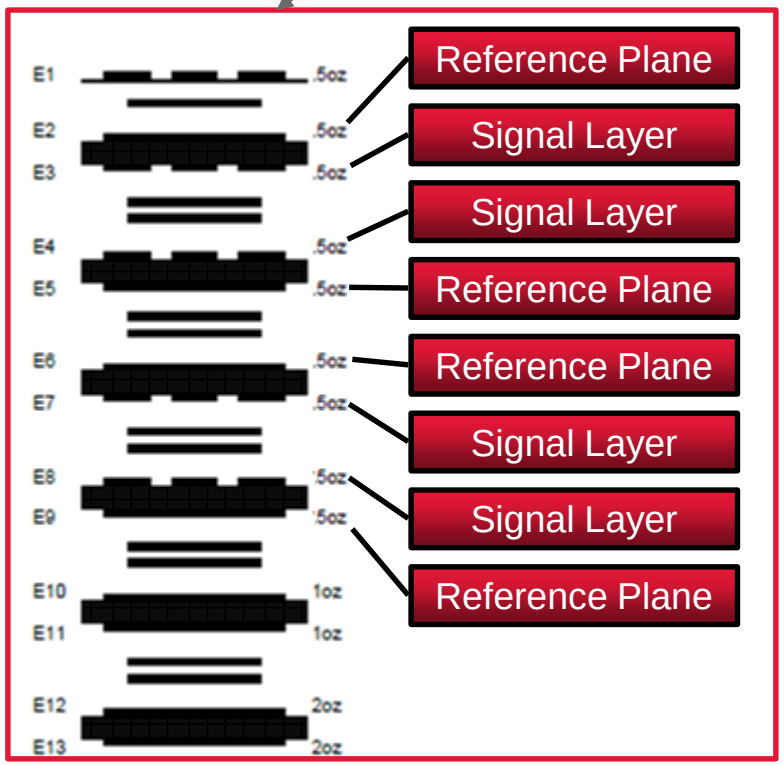


Stripline



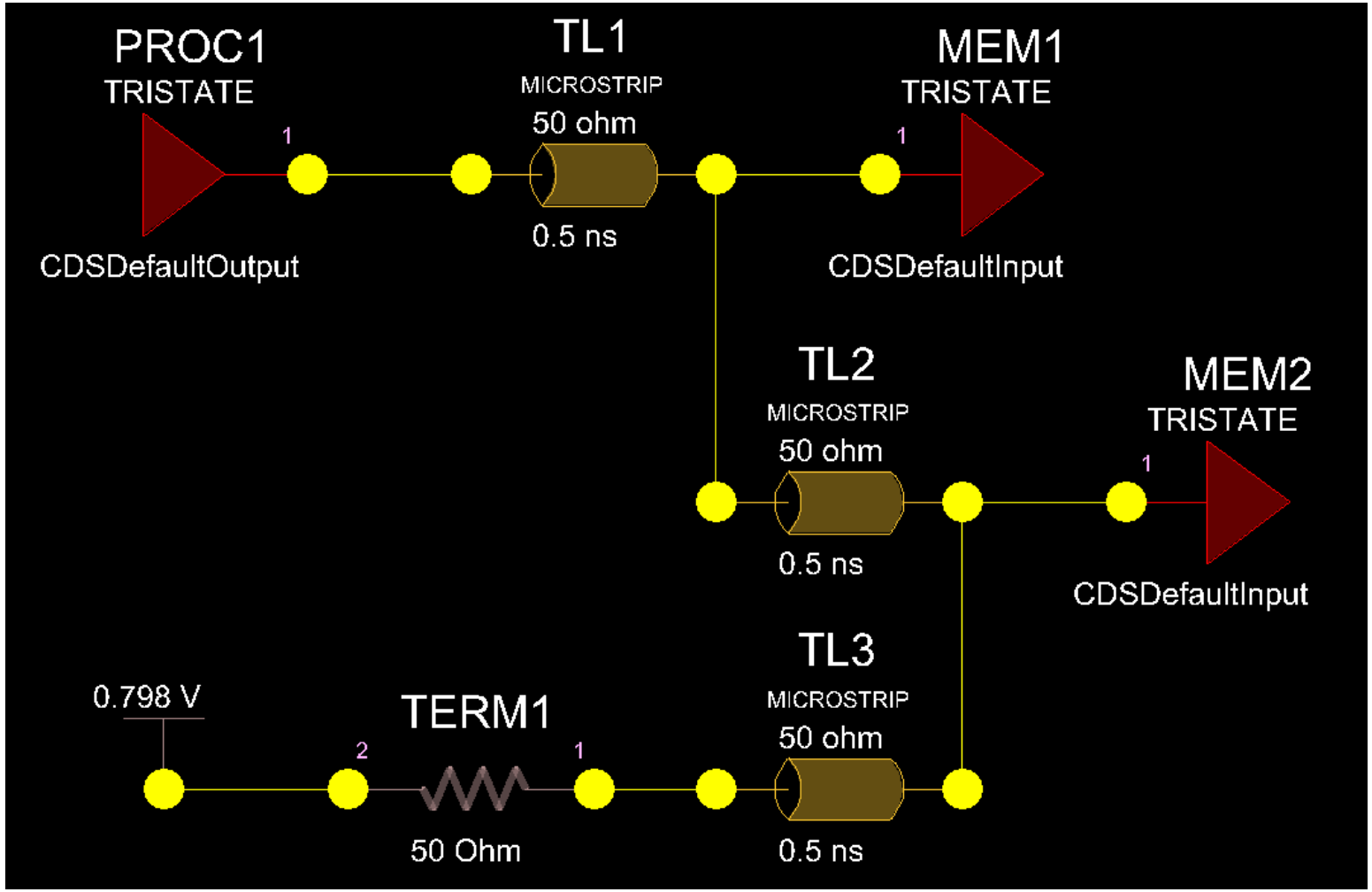
Dual Striplines





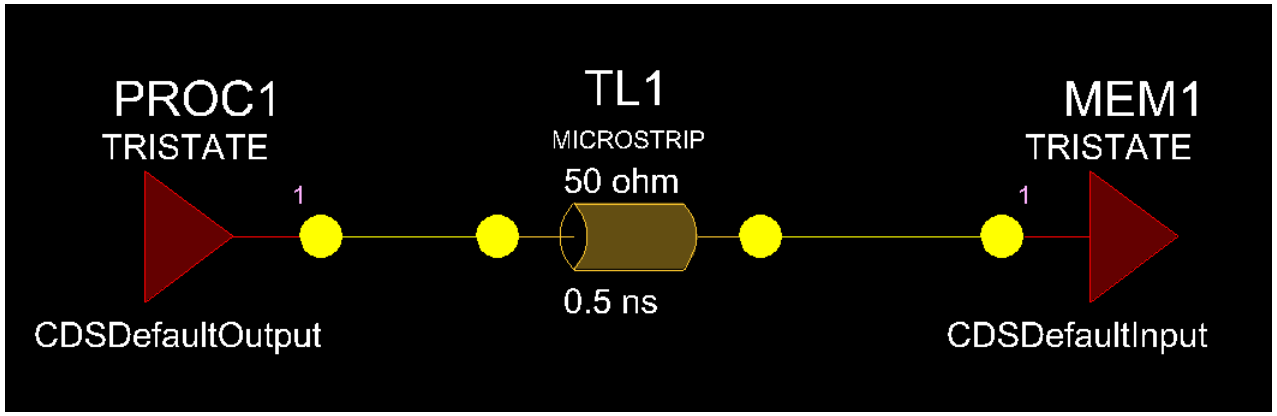
Interconnect Topologies—Daisy-Chain Routing

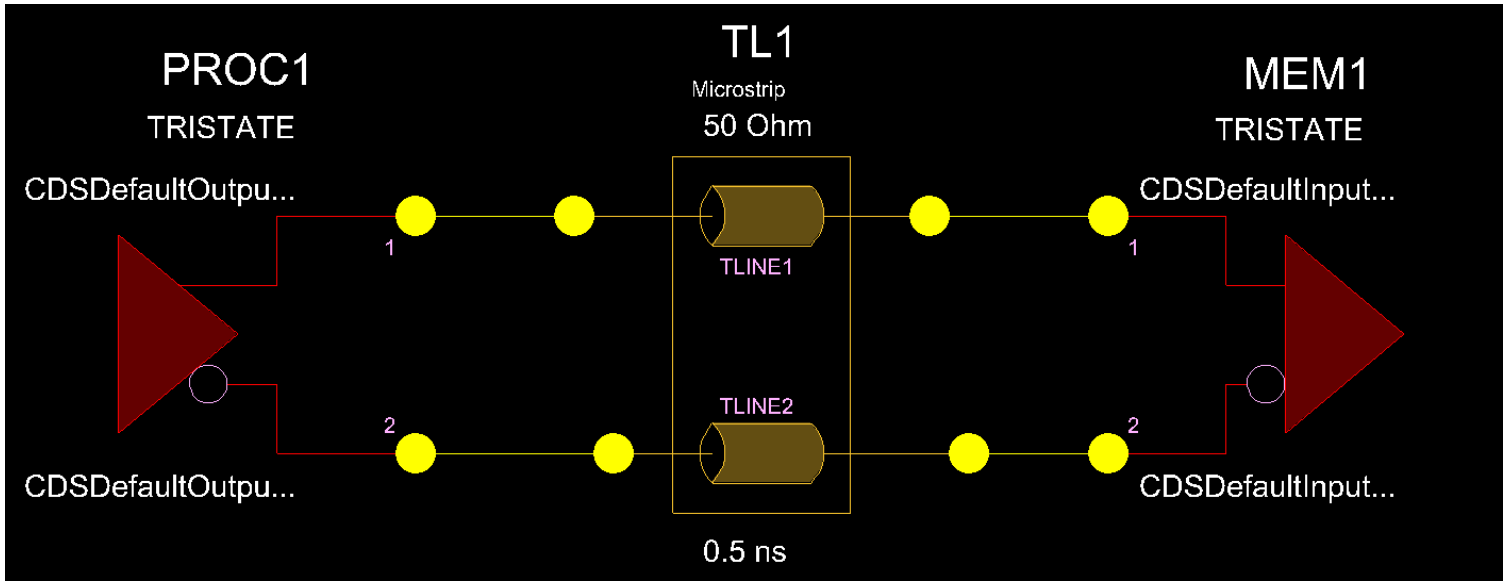
Address/command/control routing



Interconnect Topologies—Point to Point

Data bytelane routing/differential pair routing



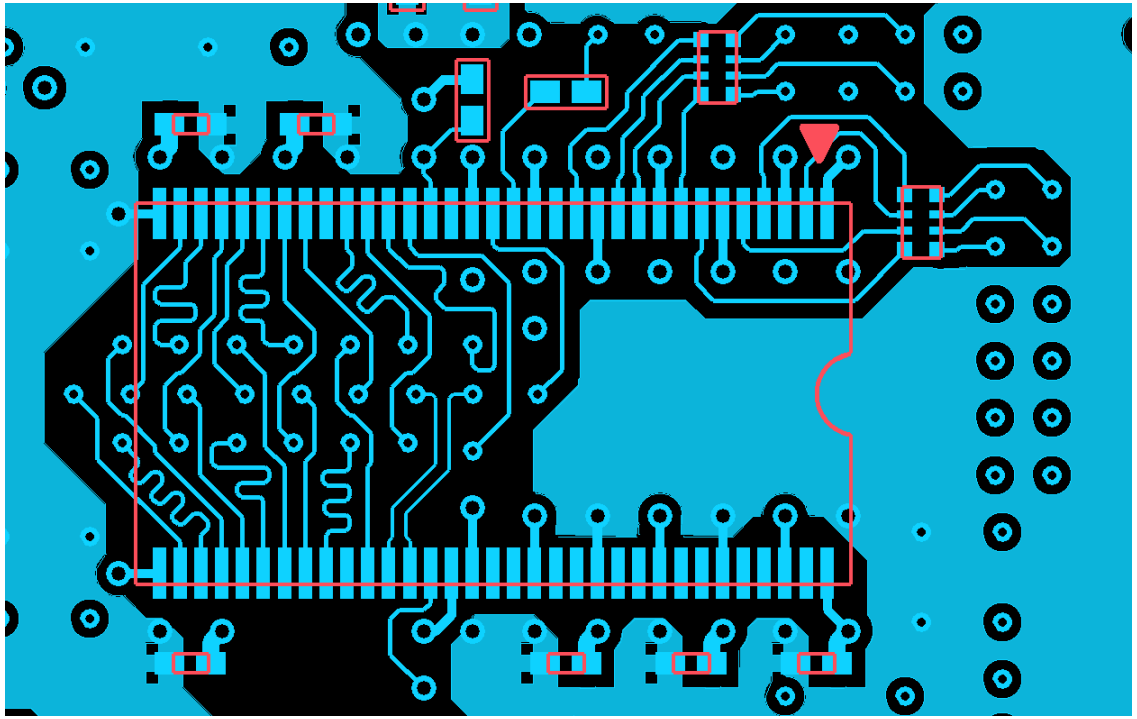


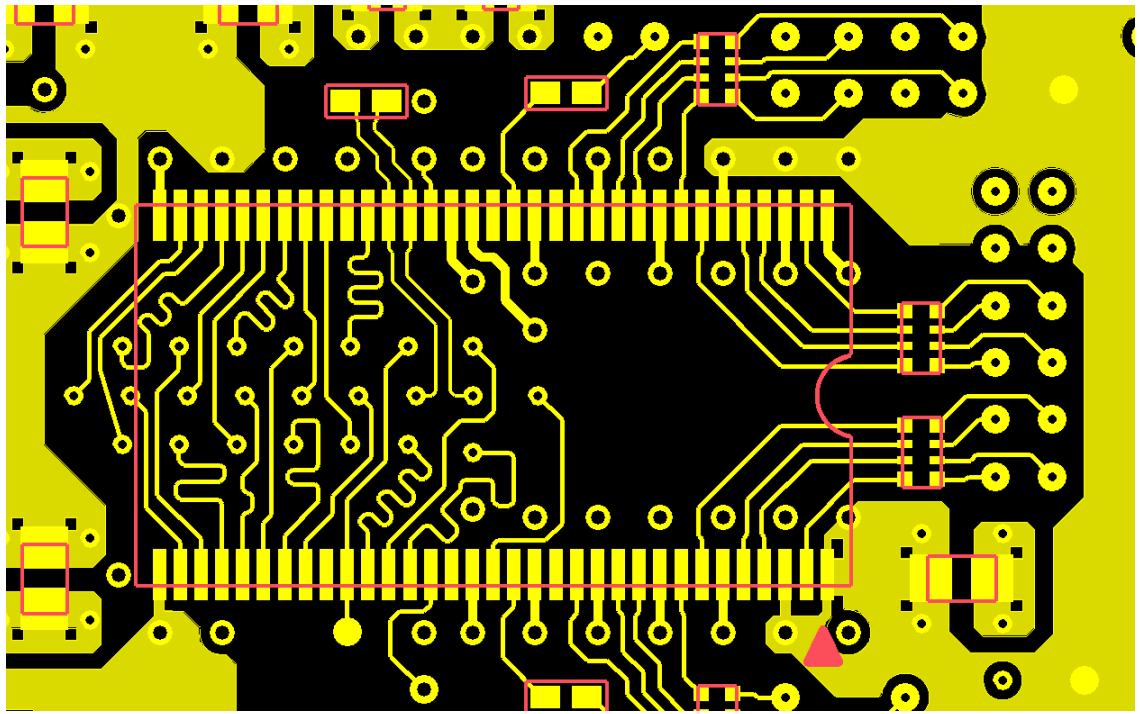
Placement Techniques
Component placement

- Careful planning of memory chips or DIMM connectors placement to allow best possible path for routing
- Reserve space for pin escape (fanout), termination resistors as well as termination power supplies
- Locate memory chips to allow address/command/control/differential clock daisy-chain (fly-by) routing, starting at controller, then connecting to lowest data bit chip first (Bytelane0), progressing up bytelane numbers, and ending at highest data bit chip
- Approximate spacing between memory chips should be no less than 200mils to allow matching outside of via/BGA field of devices

Placement Techniques
Pin escaping (fanout)

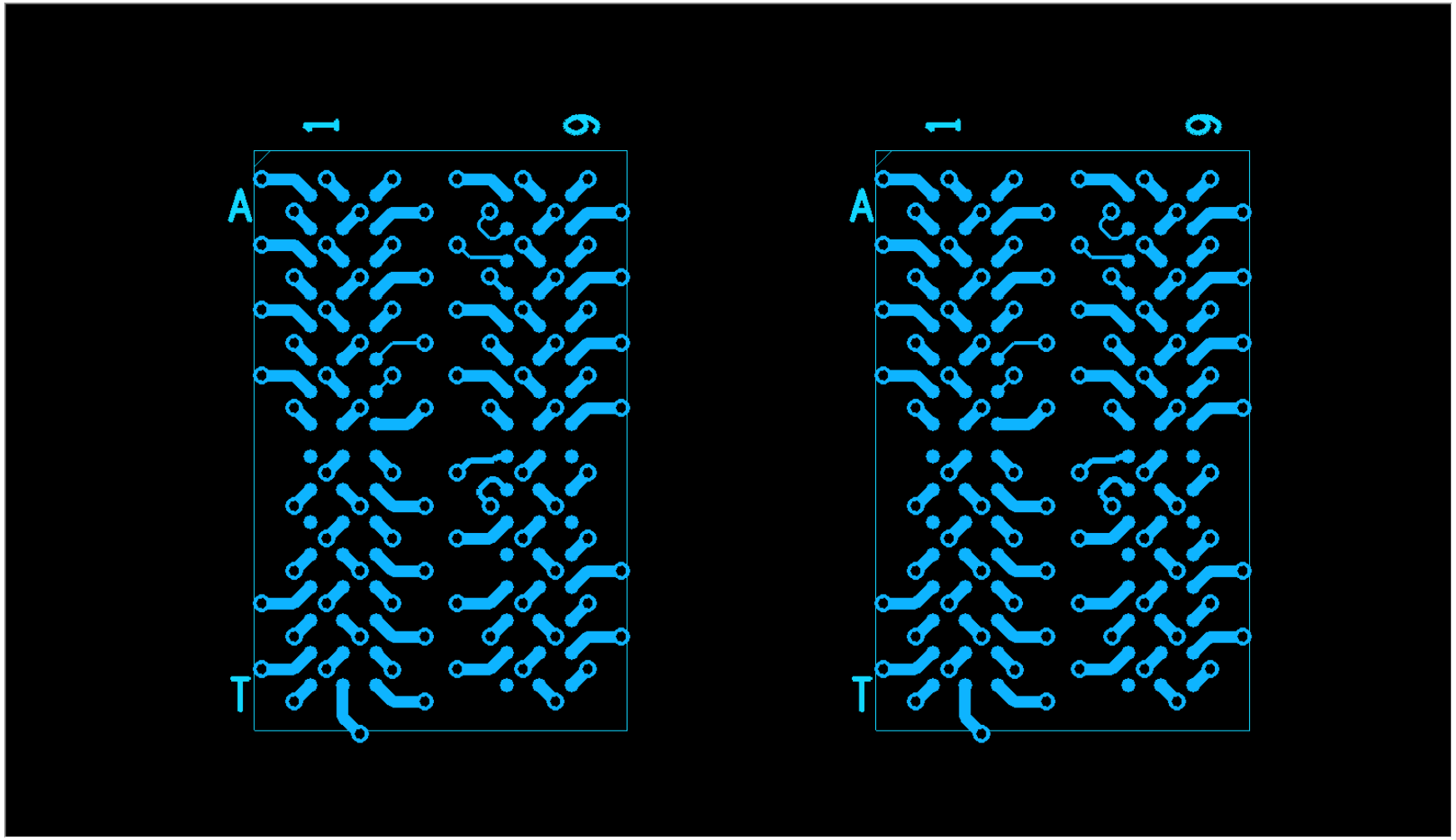
- Spread vias out to allow routing of at least two traces between vias, where possible, while maintaining reference to adjacent plane layers (avoid routing thru via voids in the plane)
- Keep in mind interconnect topologies of pins that you are escaping
  - Share vias to form a t-point for the DDR2 address bus, remembering to have room to match them on the surface of the board





Placement Techniques—DDR4: 4 Bytelanes

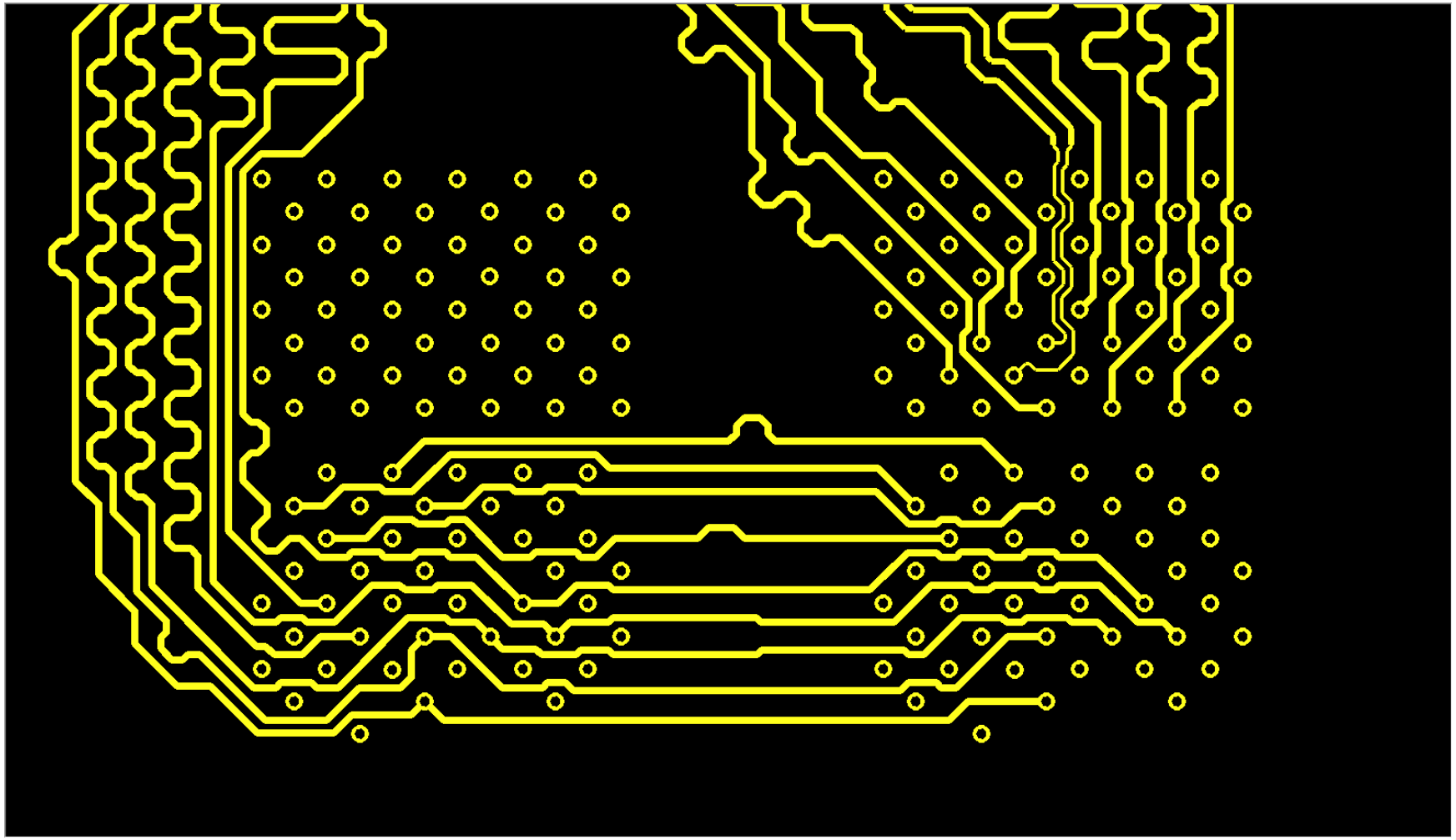
0.8mm ICs on top—Notice spreading of fanout to increase routing channels



Vias = 10mil Drill / 18mil Pad / 28mil Plane Void (AntiPad)
Spacing = ~31mil Diagonal Via Centers

Placement Techniques—DDR4: 4 Bytelanes

Partial address bus and one data bytelane routed on surface

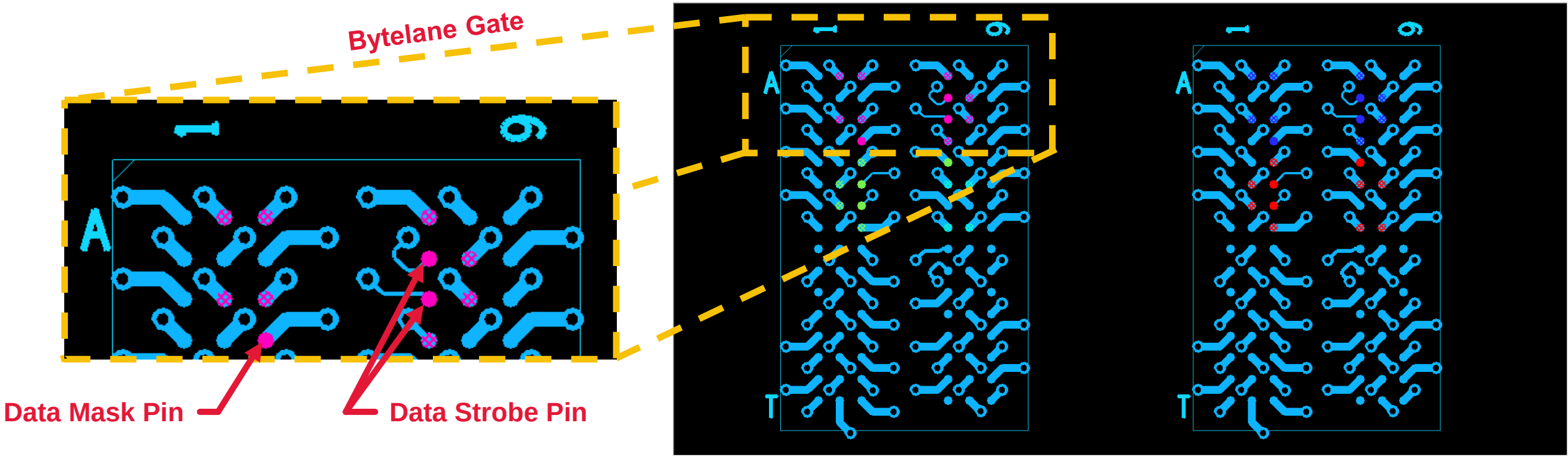


Traces = 7.5mil Single Ended / 3.5mil with 8mil Space Diff Pairs
Trace to Via = 9.5mil Single Ended / 7.5mil Diff Pairs

Placement Techniques
Pre-route planning

- **Pin swapping:** Data bits (DQ[63:0]) can be swapped within a bytelane to improve routing
  - Excludes data mask (DM[7:0]) and data strobe (DQS[7:0])

- **Gate swapping:** All members of one bytelane can be swapped with all members of another bytelane



Routing Technique on High-Speed Interfaces

Routing Challenge

Older strategies have run out of steam

- Simply jumping into routing or turning on auto-router after completing placement was never an efficient way of getting a design completed
- Designs today have some type of high-speed interfaces (XFI, XGMII, XAUI, DDRx, PCIe, etc.) with increasingly sensitive signals and complex electrical/physical constraints to ensure interface will function as expected
- Very important to build a stable foundation, starting at PCB placement using “forward-thinking” and continuously planning for next stages in design
- Plan overall design strategy as early in design process as possible and allow electrical design rules to guide a successful placement and design

Routing Challenge

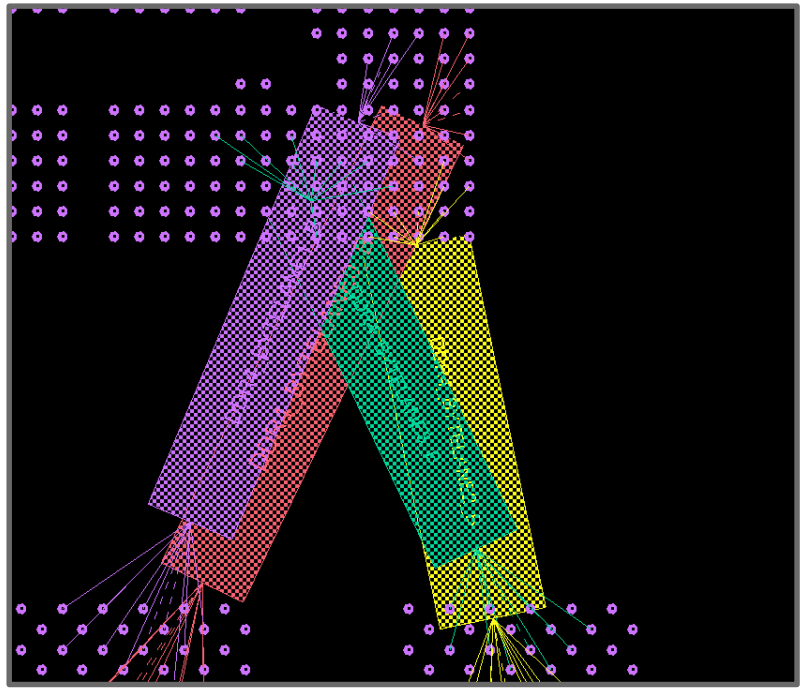
There must be a better way

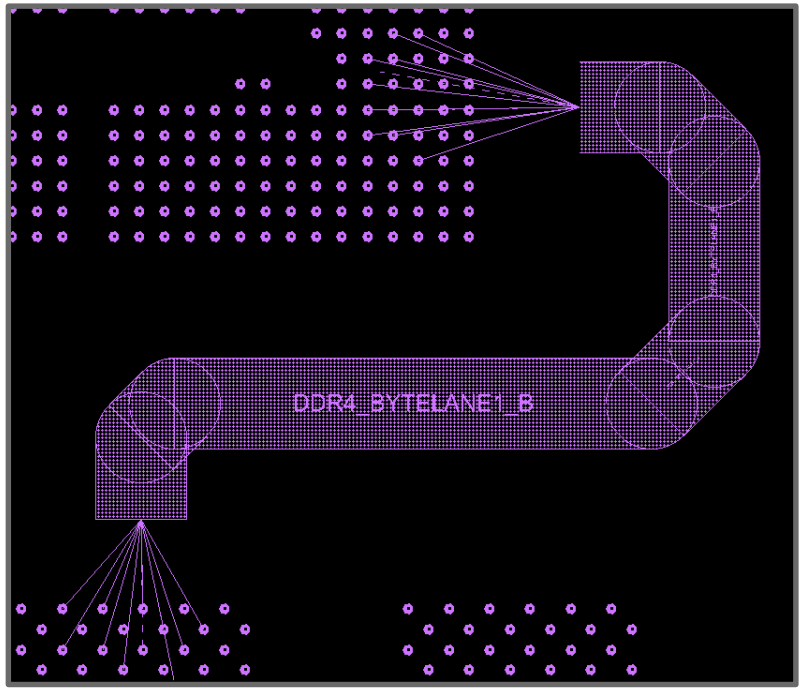
- Sketching routes from component A to component B without reviewing interconnect sequence first is no longer an effective approach
  - Sequence and pin-swapping tradeoffs must be exercised and reviewed on one or both sides of a interface to develop highest quality and properly ordered breakout solution
- Package density continues to increase and there are only so many routing channels to escape from a device without adding manufacturing expense of extra routing layers
- Following slides will demonstrate techniques of bundling bus routes so you can focus on finding an interconnect solution for each side of the bus

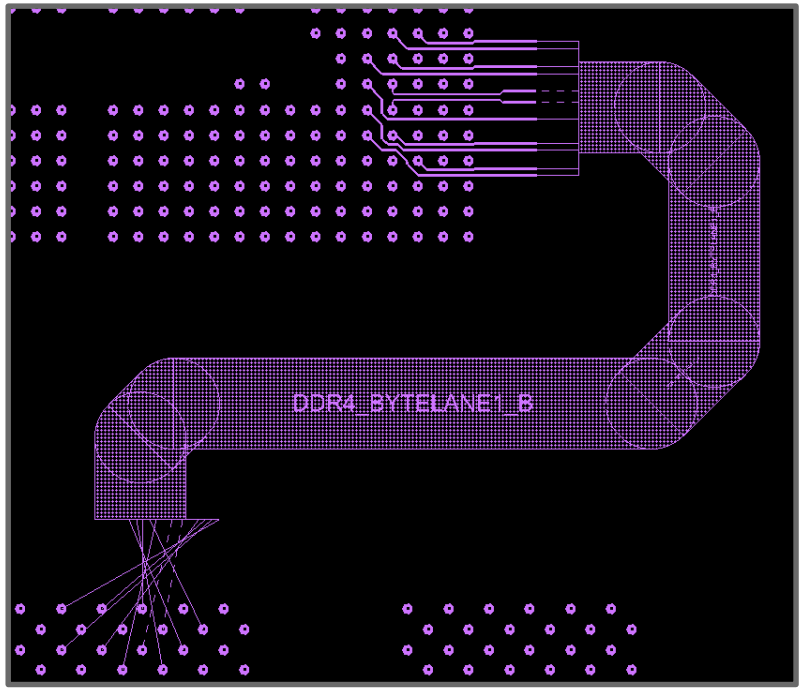
Flow Planning Interface Routing

Bundle routing by group for a hierarchal view of route path

- Group each data bytelane into bundles for a simplified view of routing groups
  - Provides more defined focus on the interconnect solution
- Adjust bundle (flow) to represent expected path the route to take
  - One data bytelane shown for clarity
- Breakout (route escape) starting from more complex side of bundle
  - Ensure cleanest possible breakout



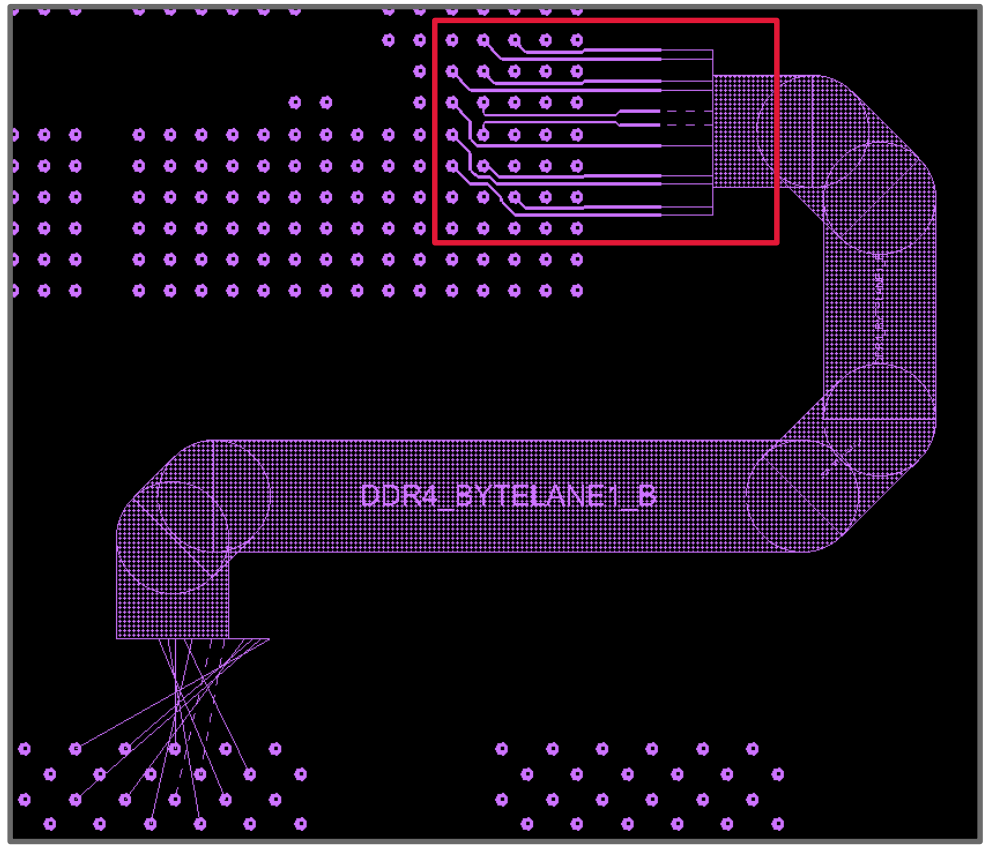


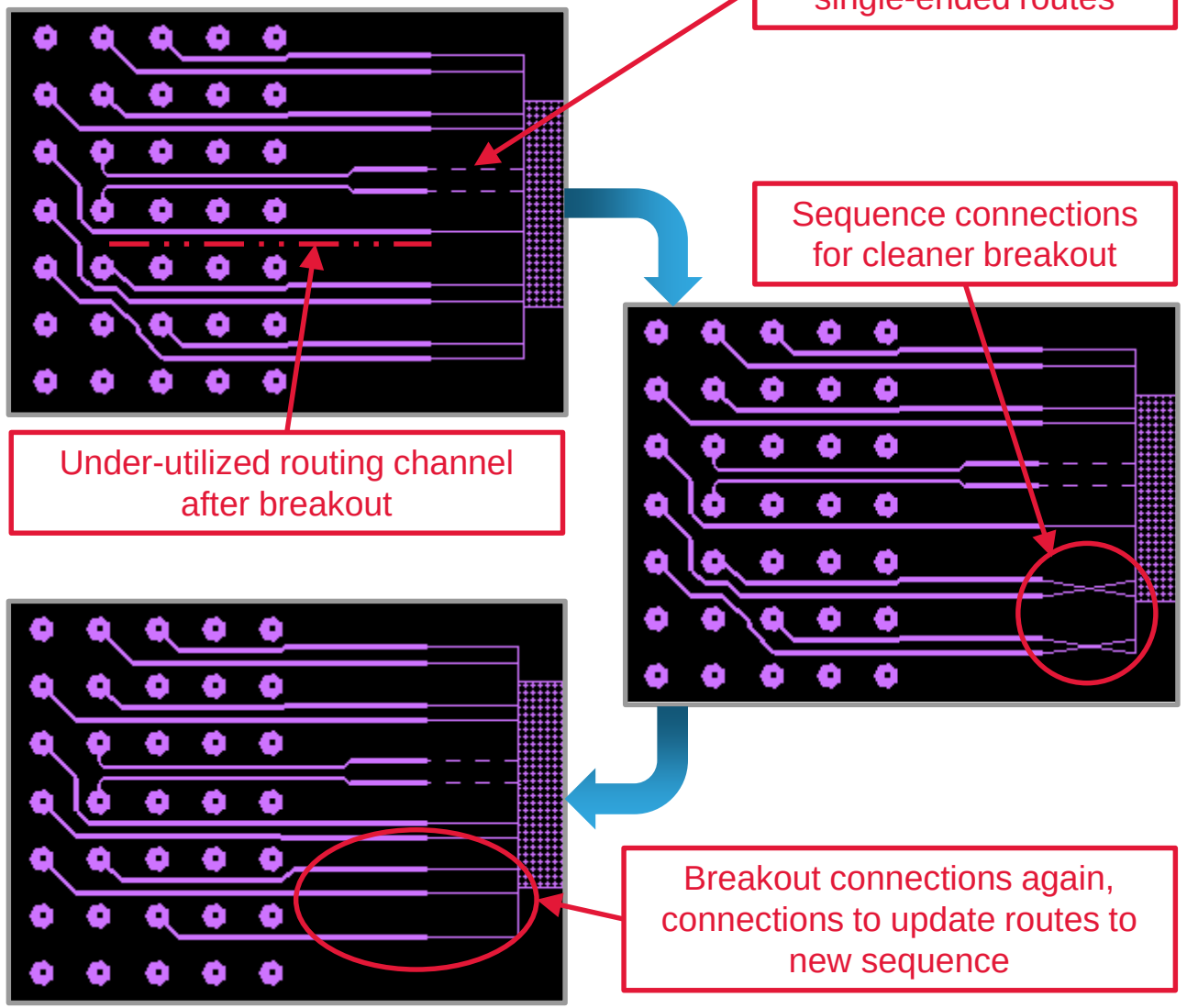


Flow Planning and Sequence Interface Routing

Route connections outside of package boundary/pin field

- Review breakout result and adjust sequence as required with focus on utilizing all routing channels.

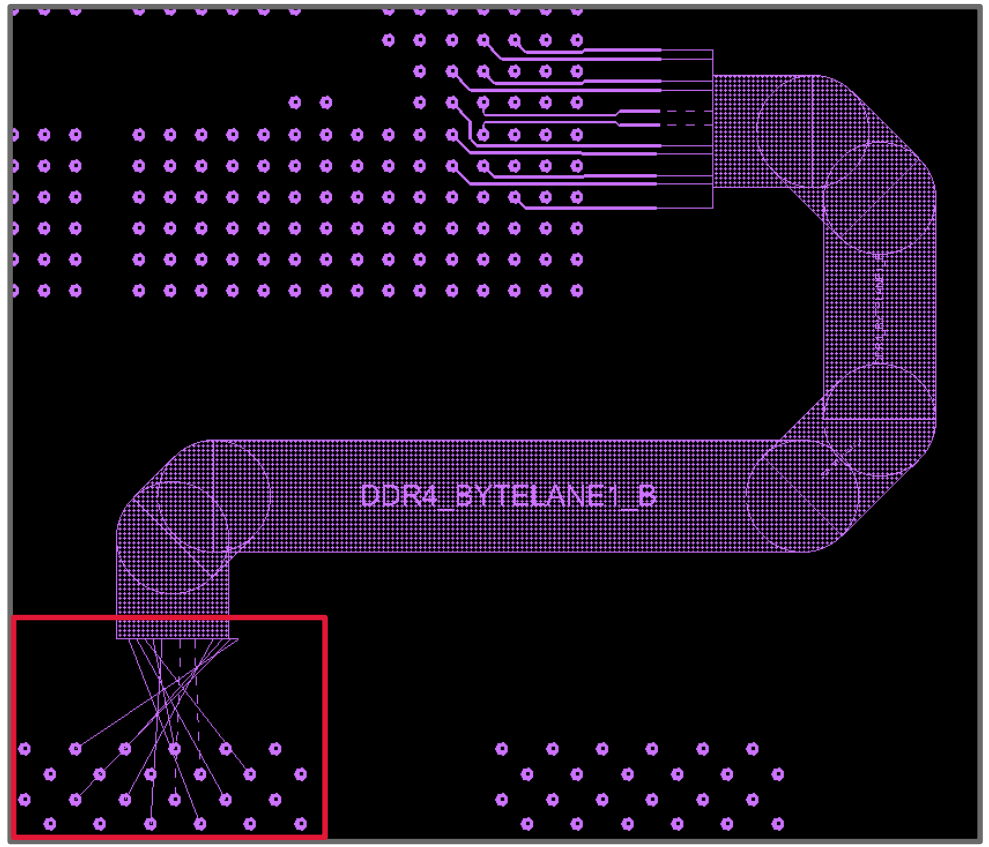


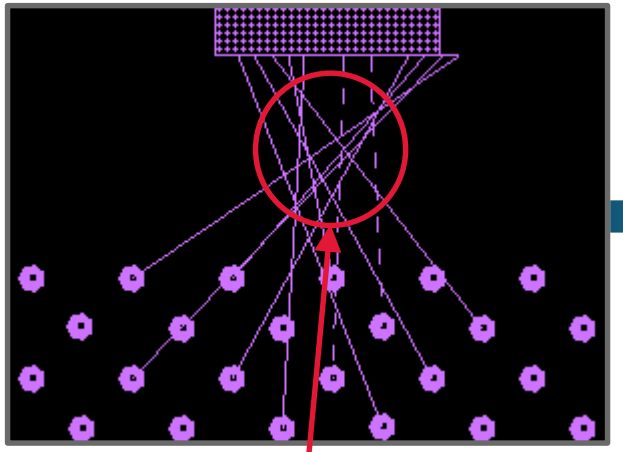


Pin Swapping and Escape Interface Routing

Resolving routing escape solution

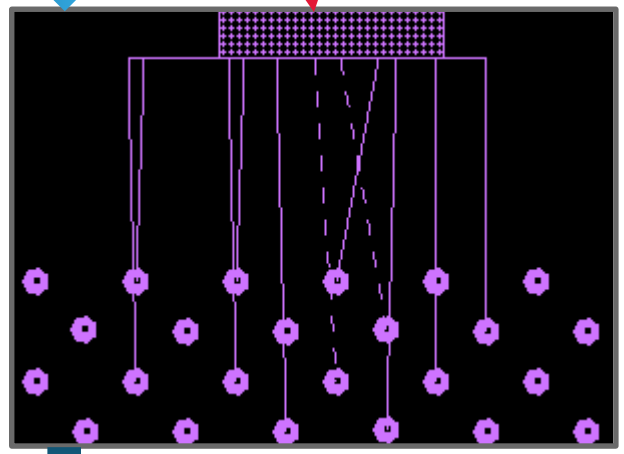
- With clean breakout on processor side, you can turn your focus on memory side, where pin swapping can resolve crosses

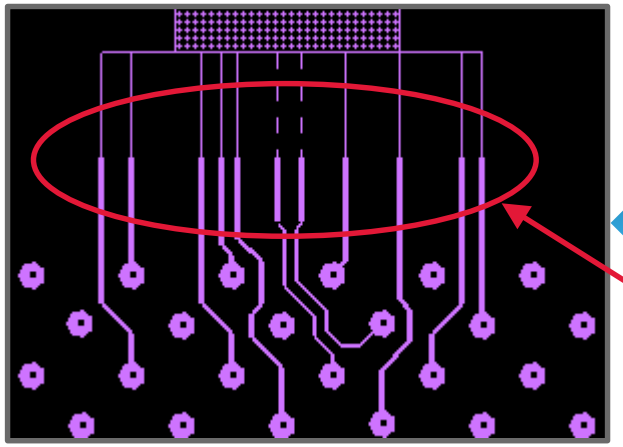




Crossing routes makes it close to impossible to escape cleanly

Pin swapping on memory side, resolves crosses for a clean breakout without affecting sequence



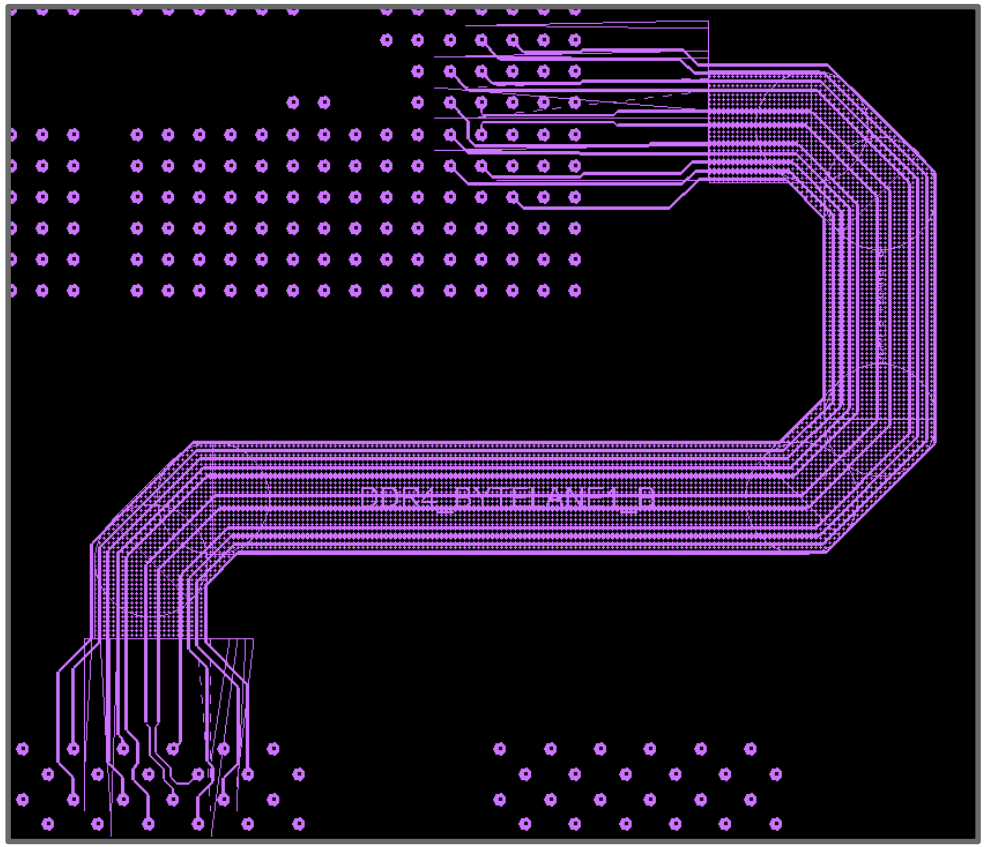


Breakout is completed without crossing on both ends of bundle and now can be successfully routed

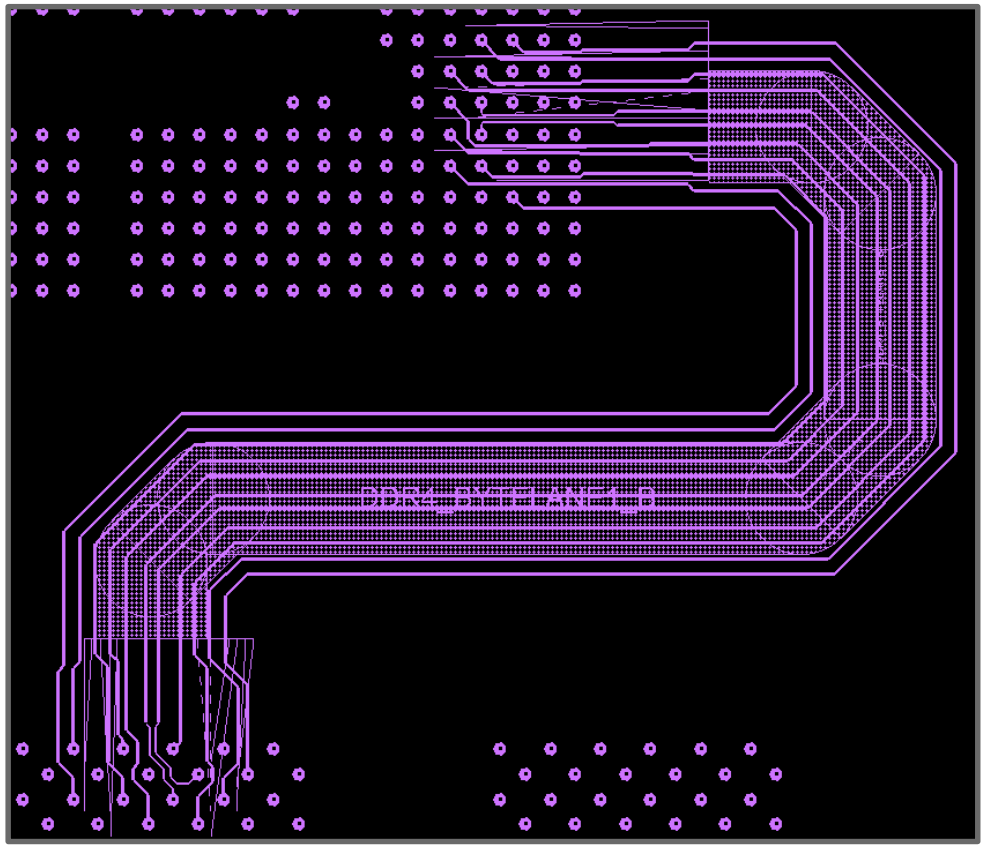
Trunk Route of Escape Routing

Complete following recommended routing flow

- Breakout has been completed on both ends of bundle so a direct route “trunk route” can be made to complete connections

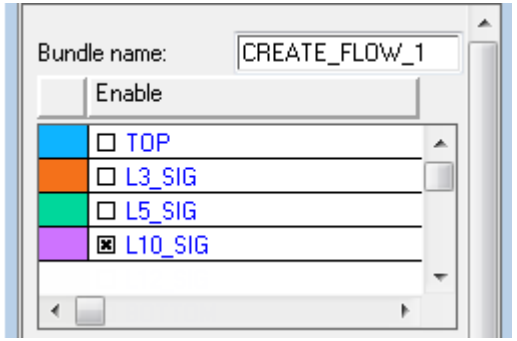


- Trunk route uses minimum DRC spacing and can be easily re-spaced using a user-defined value for a larger spacing margin

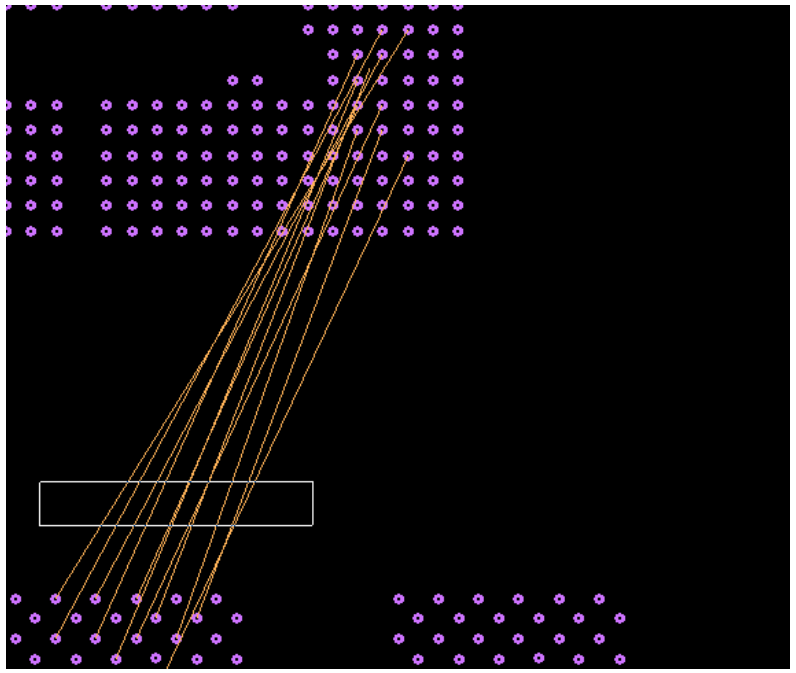


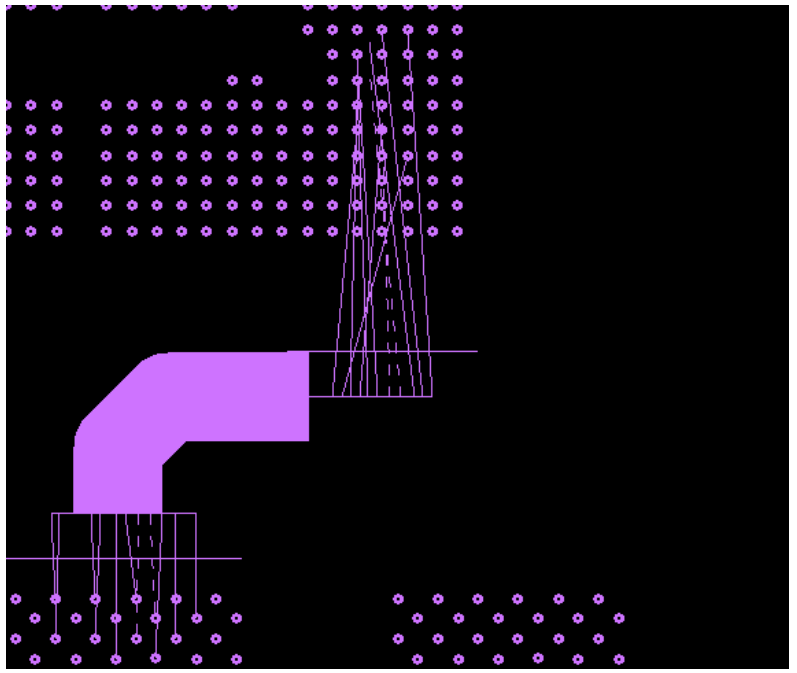
Dynamic Flow Planning Interface Routing

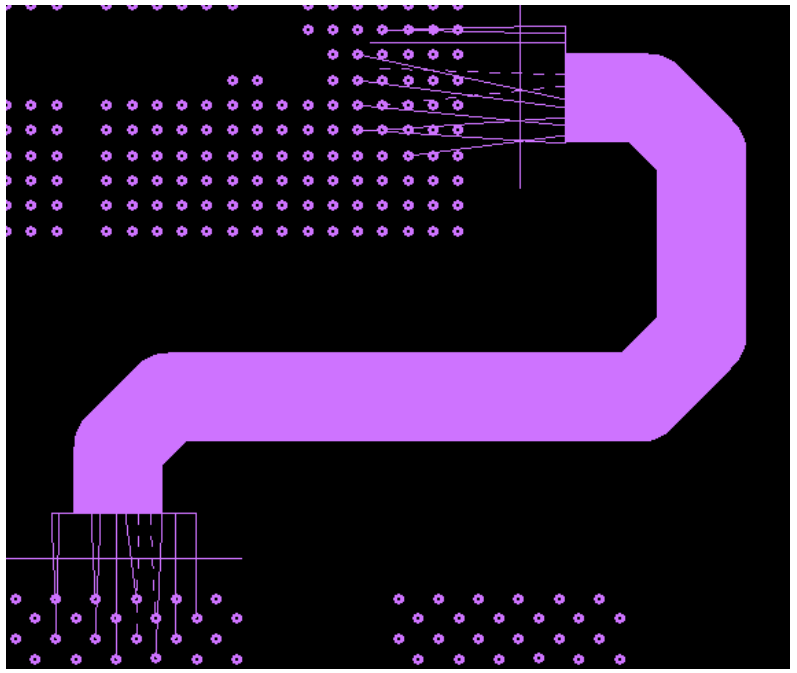
Create flow by net selection



- Specify layer(s) for routing and select data bytelane connections to begin flow
- Define bundle (flow) to represent expected path for the routing to take
- Once routing flow is complete, it is time to complete routing

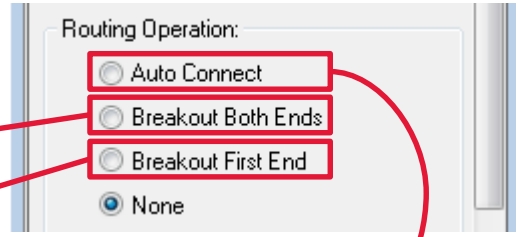




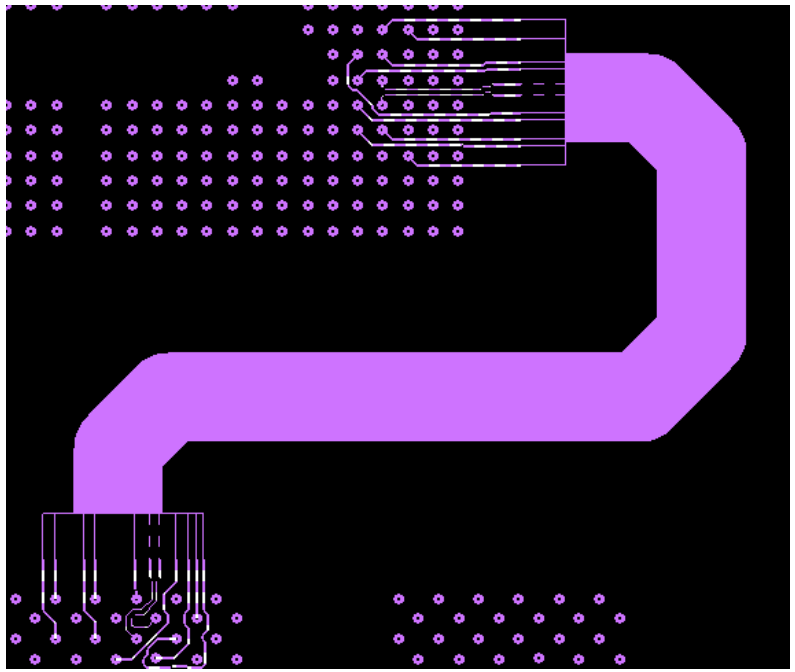


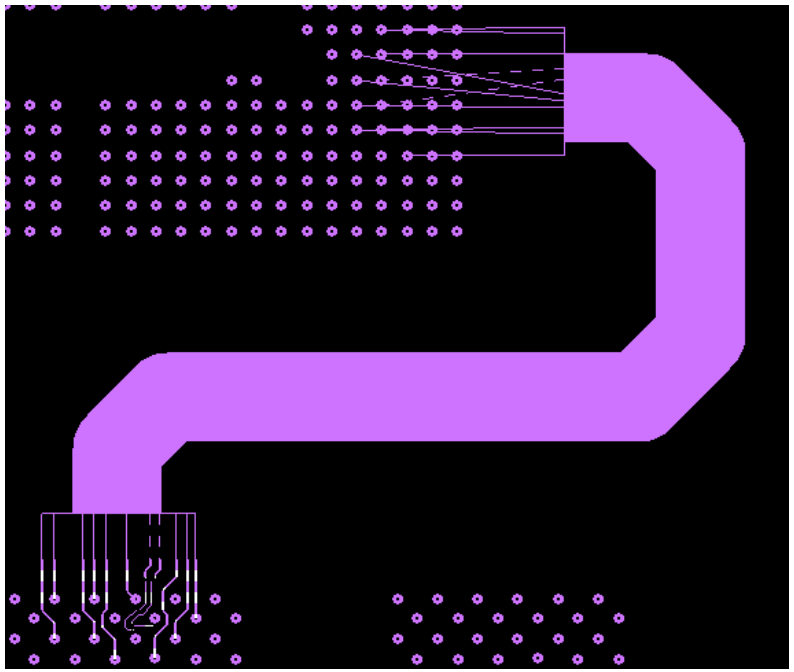
Dynamic Flow Planning Interface Routing

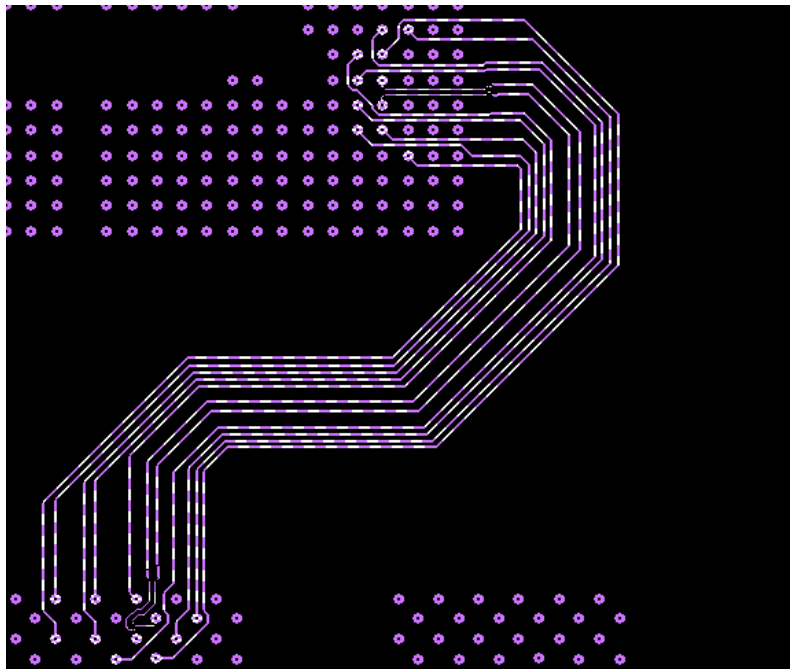
Complete routing operations



- Breakout Both Ends
  - Automatic sequence and breakout both ends for optimization review in preparation for trunk route
- Breakout First End
  - Automatic sequence and breakout starting end for optimization prior to breakout of far end, followed by trunk route
- Auto Connect
  - Runs Breakout Both Ends and trunk route all in one step
  - Other routing operations allow optimization prior to trunk route



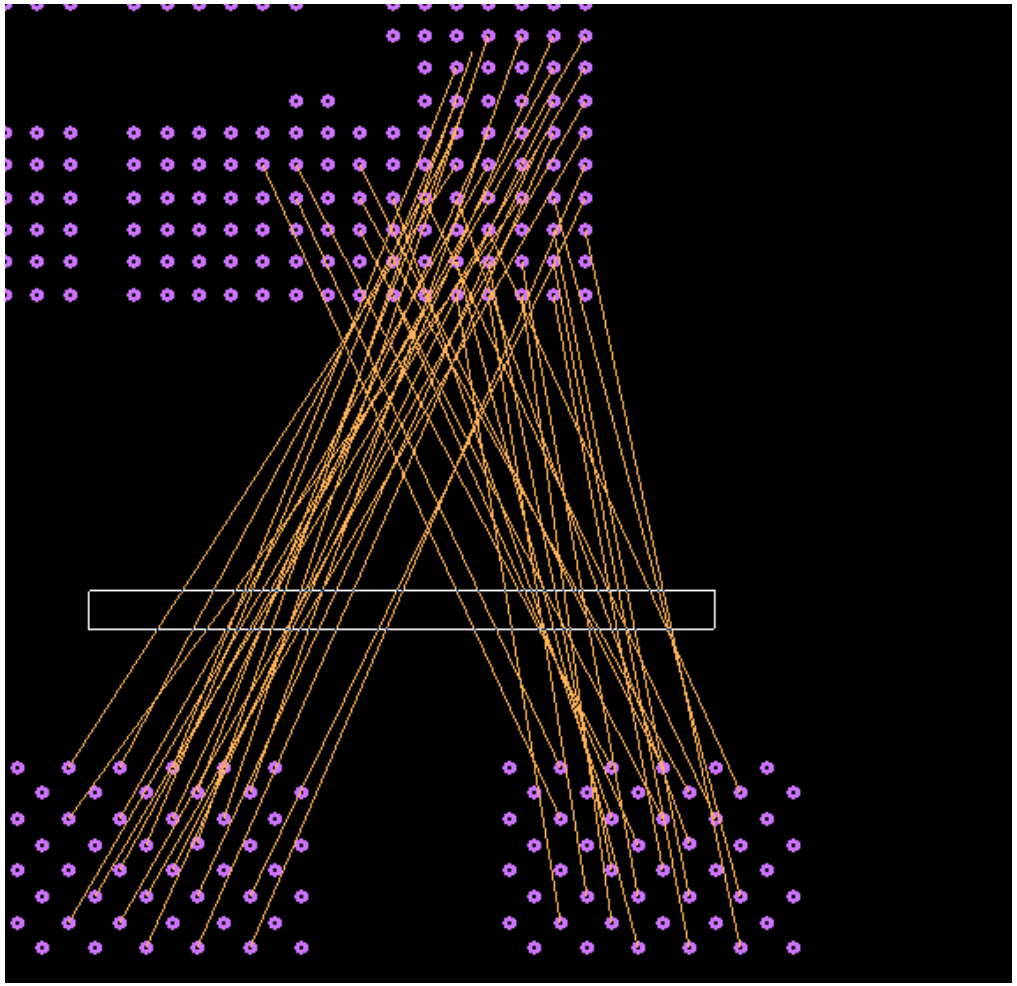




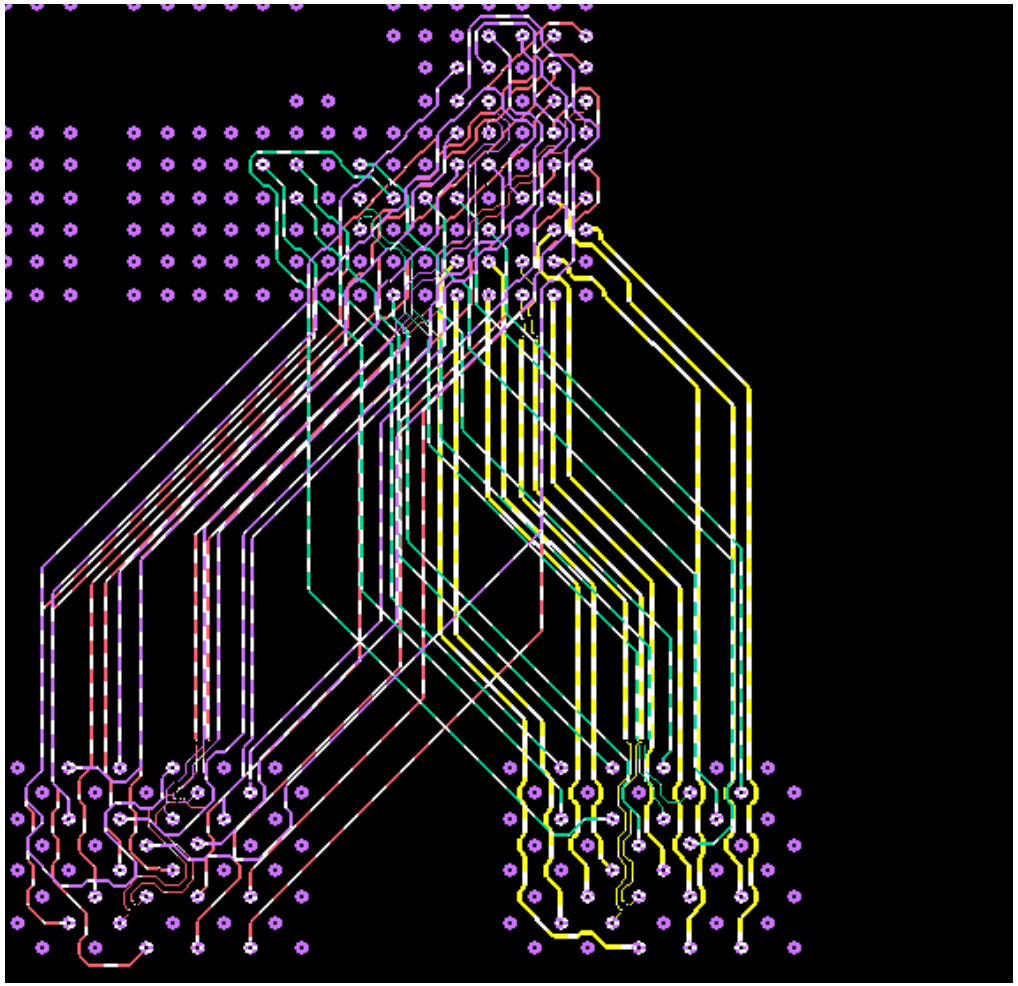
Selective Interface Routing

Quick and easy point-to-point routing without flow-planning bundles

- Select un-routed connections to route



- Connections completed on assigned layers



Timing Closure on High-Speed Interfaces

Timing Challenge

Older strategies have run out of steam

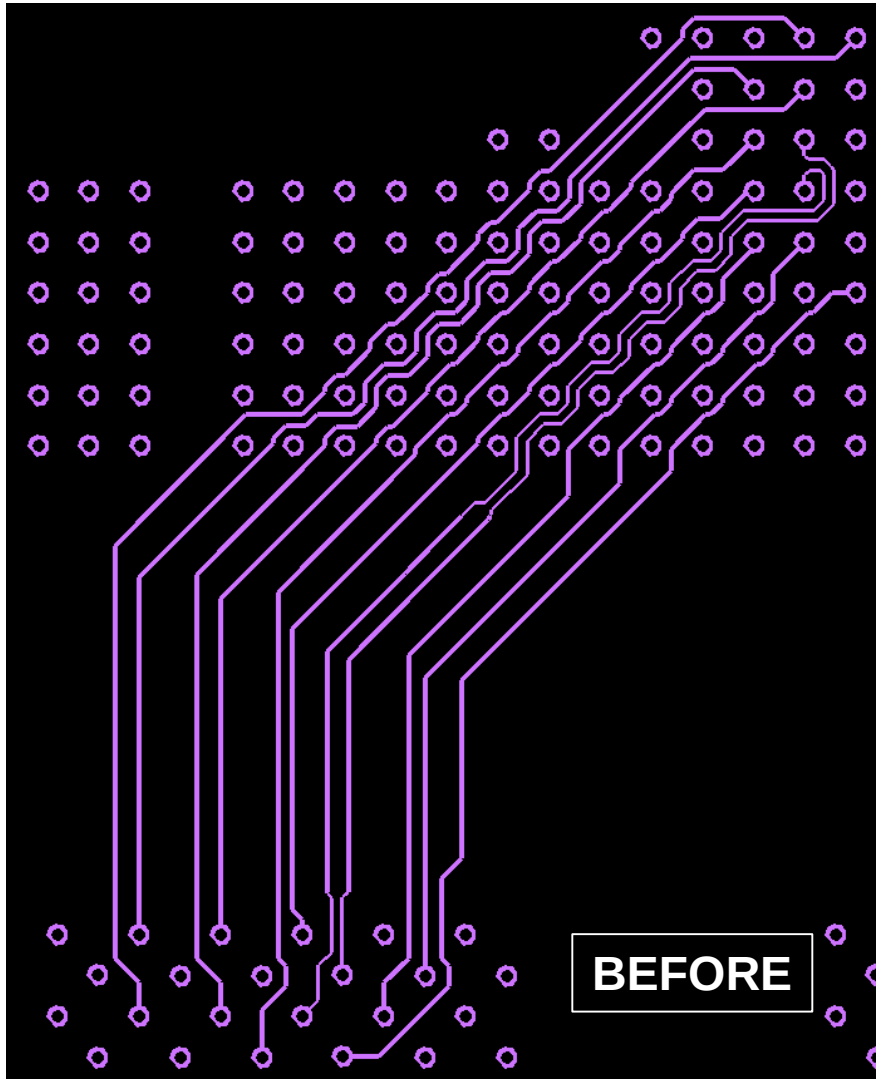
- As design groups are being asked to do more with less, PCB designers need to find more efficient ways of successfully tuning interfaces—Manually tuning each group can be painfully slow, not efficient and error prone
- Manual delay tuning can also lead to adding extra length to entire group even though all members match within tolerance
- Careful tuning must be done to avoid “tug-of-war” with different levels of electrical rules that give appearance of being within tolerance

  For example:

  – Match all members to tolerance within group while differential pairs are out of phase
  – Match all members to tolerance within group while exceeding overall length

Timing Challenge

Route tuning examples—Excess length added during matching

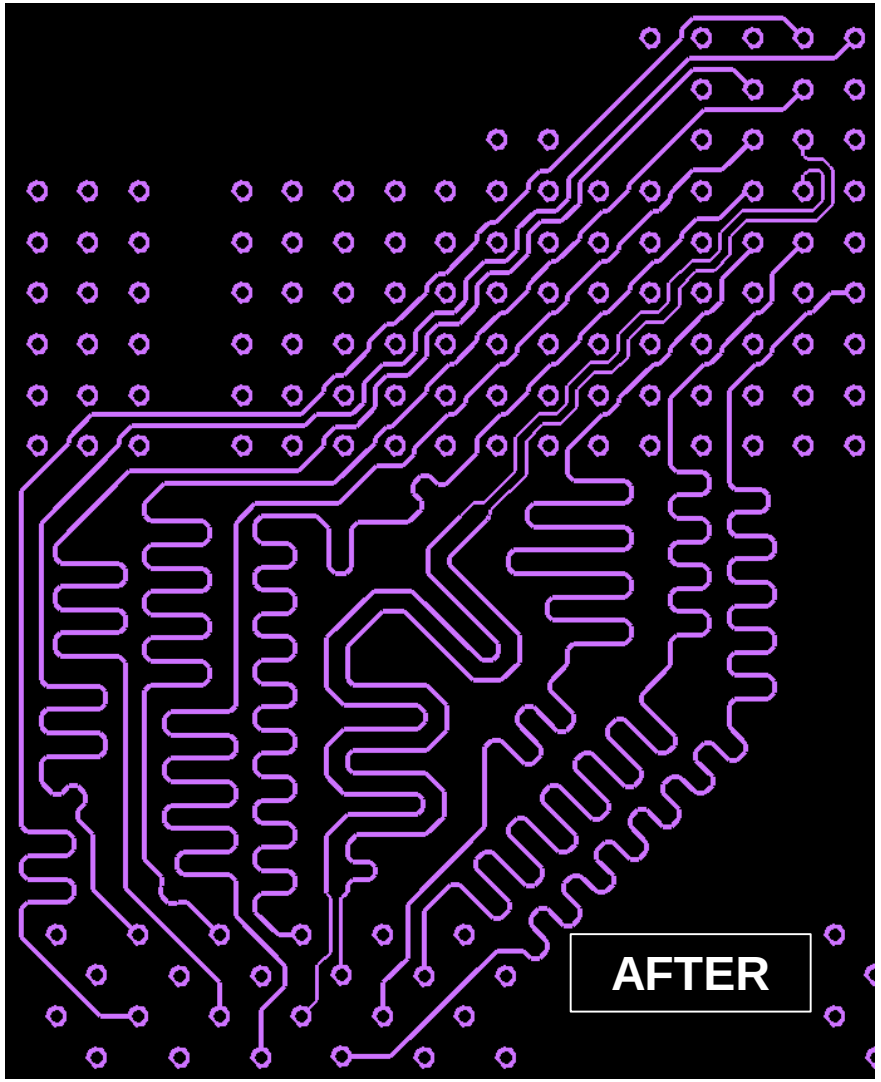


Match group
Tuned to *5mils* of each other

Overall length
1435mils

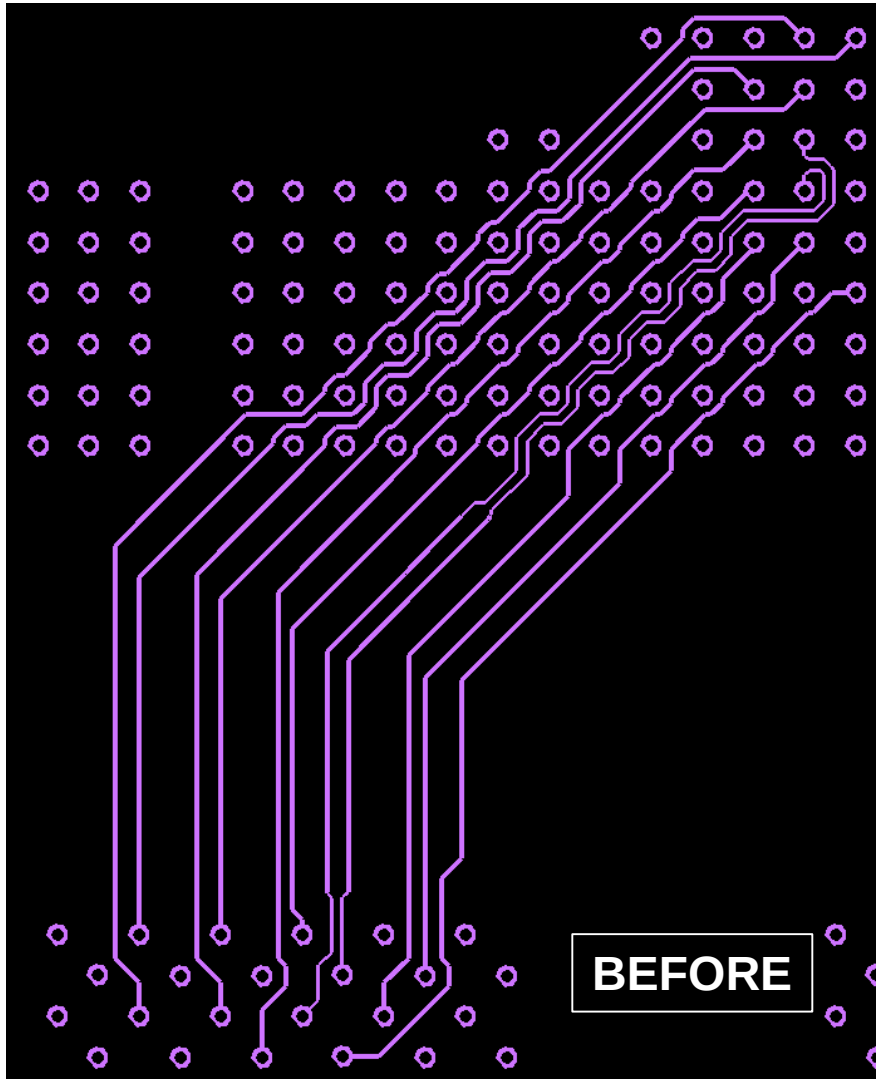
Board area
0.3128 sq. in

Blocked channels after tuning
~10 route paths



Timing Challenge

Route tuning examples—Clean tuning maintaining an overall minimum length

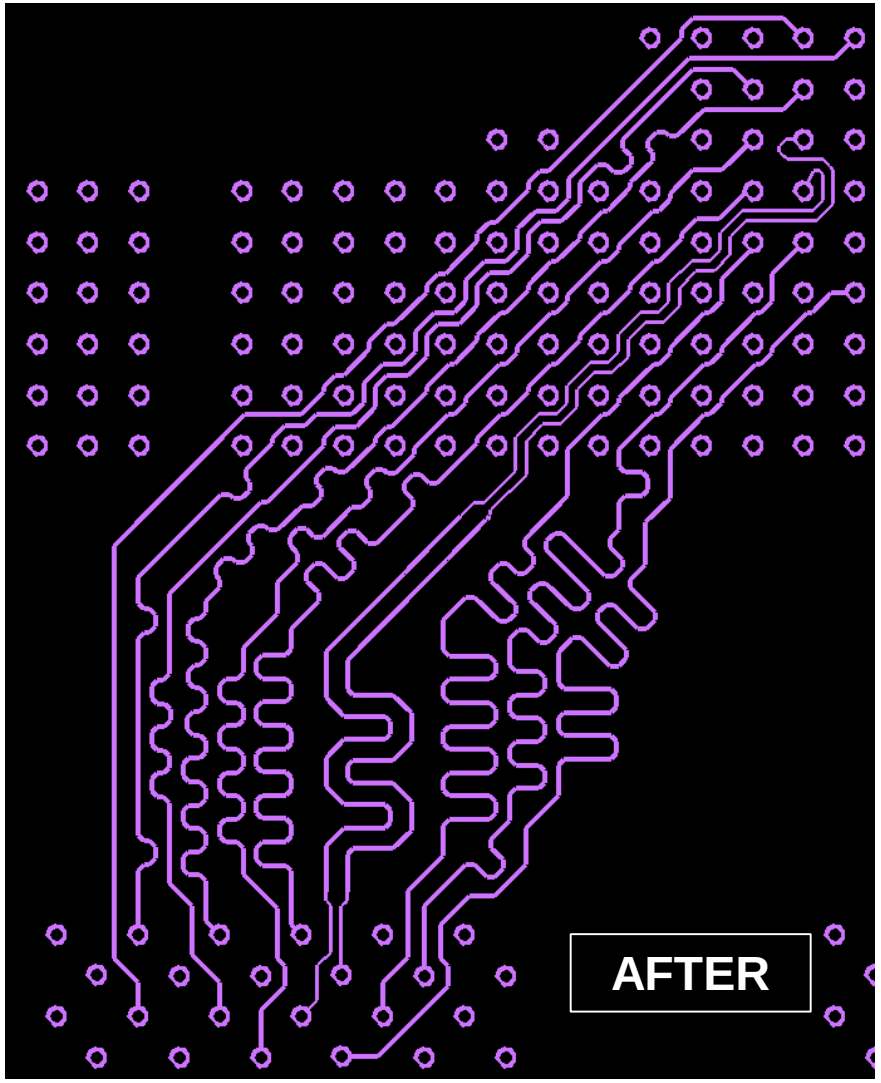


Match group
Tuned to *5mils* of each other

Overall length
1148mils
(-287mils)

Board area
0.2402 sq. in
(-.0726 sq. in)

Blocked channels after tuning
0 route paths



Timing Challenge
How to achieve better results

- Is it possible to meet timing without excess length while keeping a close eye on all different types of electrical timing constraints for high-speed interfaces today?

- Short answer is “yes” but it is a very time-consuming process:
  - Diff pairs must be first brought into phase with each other before any other tuning is done
  - Sort routed length of match group members to determine “target net”
  - Color nets in three different categories (short nets, long nets, target nets)
  - Continuously check lengths of all match group members during tuning to ensure you are not creeping overall length up higher than it needs to be
  - Working with delay units in time (NS) can be a challenge

- Process normally requires an experienced PCB designer to complete, and setting up electrical constraints are only a small piece of overall effort
  - Even with an experienced PCB designer the process can be very time consuming and can make or break a project schedule

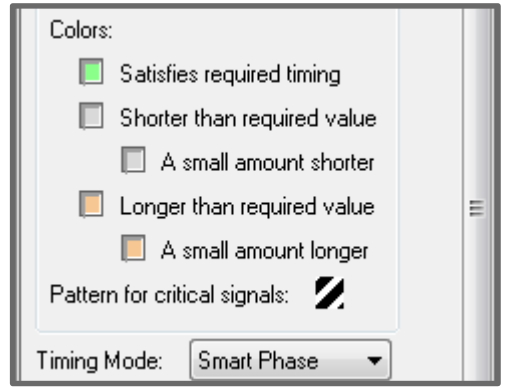
Timing Challenge

There has to be a better way

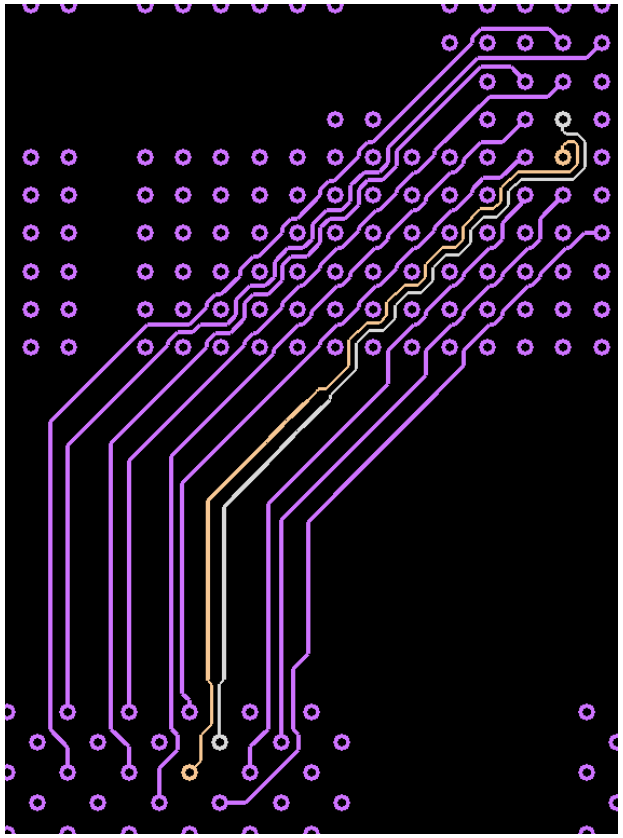
- Analyze timing requirements for design, and calculate individual length goals to guide tuning effort using a “real” target net
- Real-time feedback while tuning directly on canvas without need of looking at timing meters or opening constraint management tools to keep track of match group members
- Guided or automatic way to quickly and easily tune these complex interfaces while having ability to interact with functionality to achieve timing closure result that meets your needs as well as design requirements
- Following slides will demonstrate techniques of tuning routes efficiently using custom coloring and stippled patterns on routed connections for real-time feedback on canvas

Intelligent Tuning

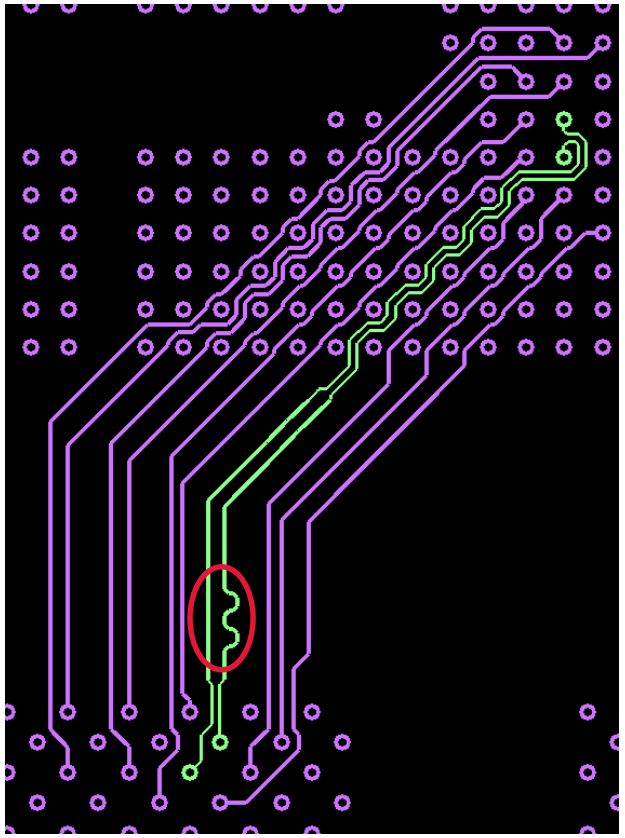
Manual phase tune based on calculated goals (Smart Phase)



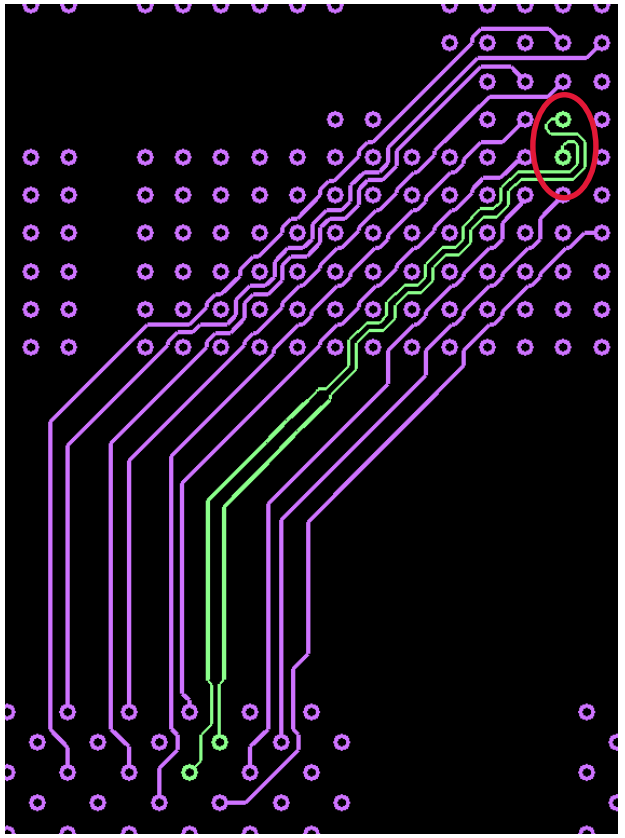
- Color coding to identify which member needs a phase adjustment



- Color change as route is brought into phase
- Loosely coupled

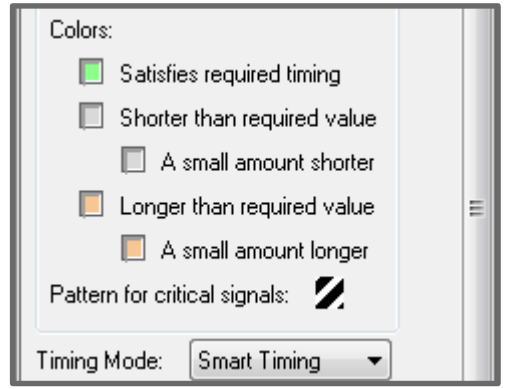


- Pad entry length adjust

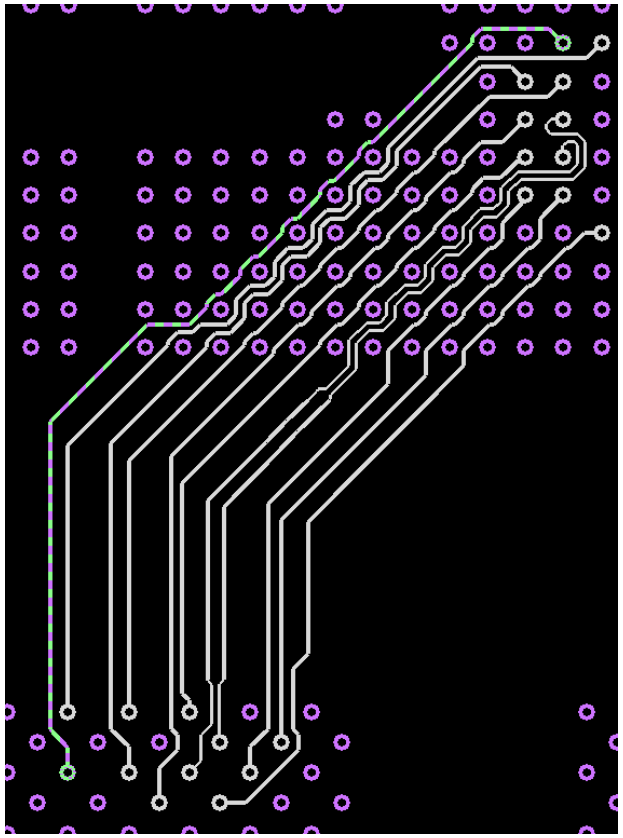


Intelligent Tuning

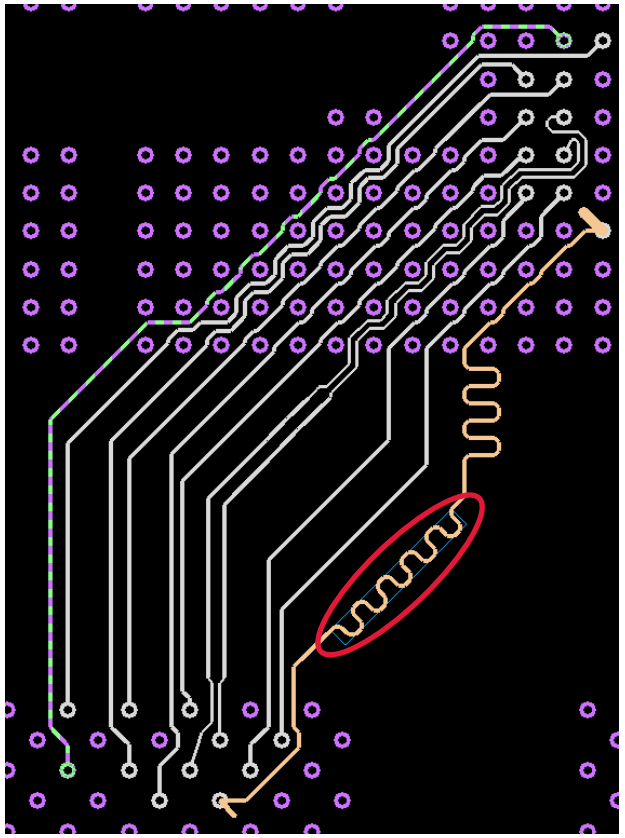
Manual tuning based on calculated goals (Smart Timing)



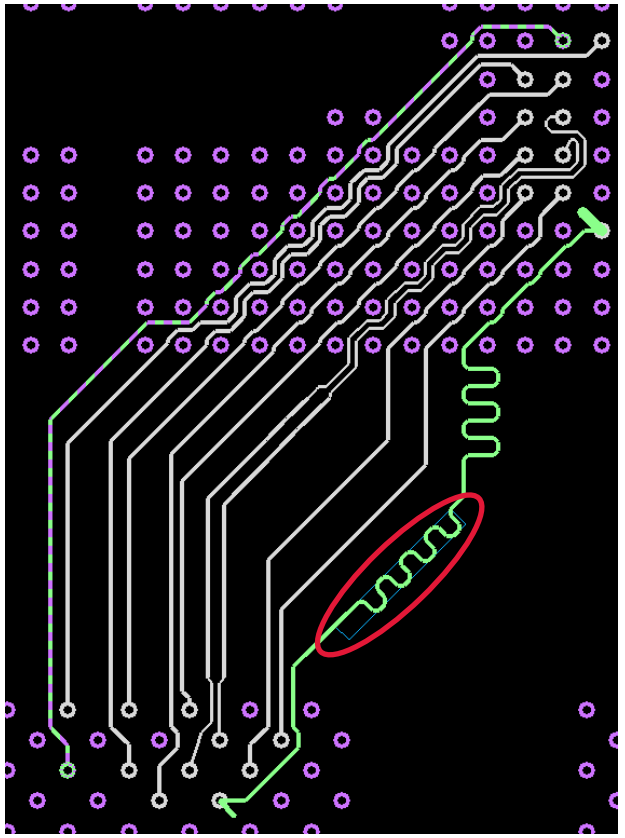
- Color coding to identify short, long, and target (critical) nets



- Color change as individual routes are manually tuned



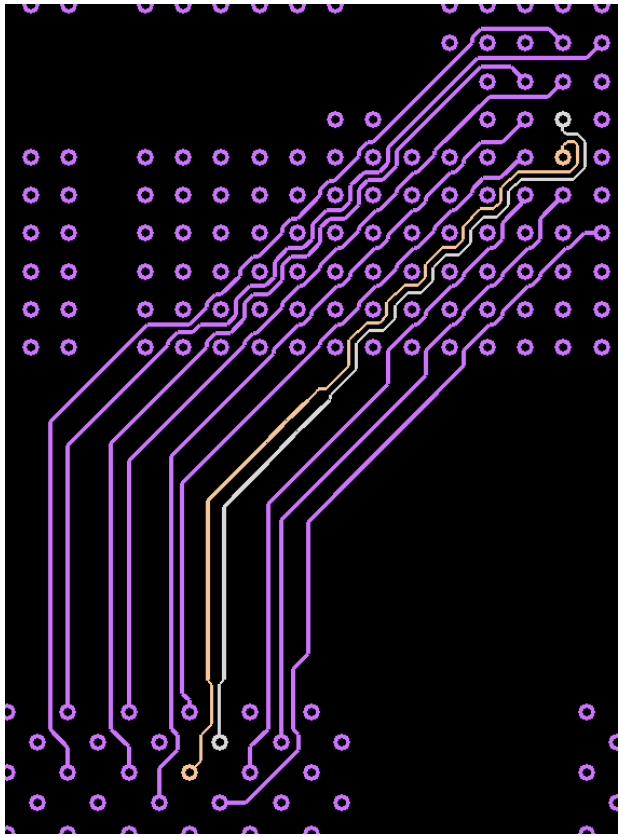
- Satisfies tuning requirement



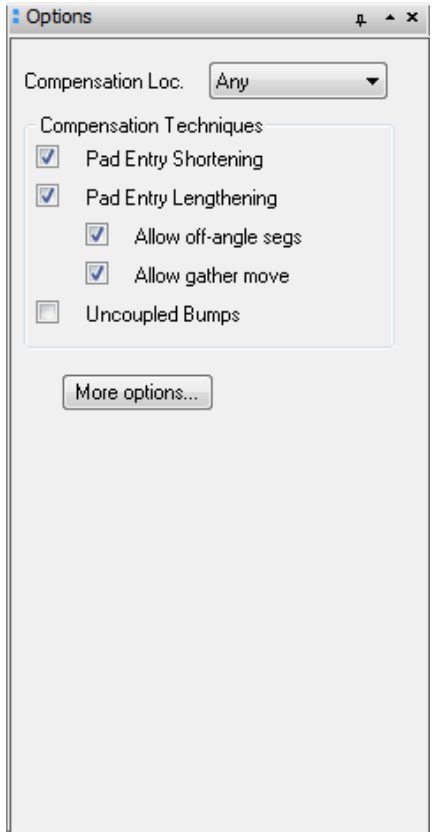
Intelligent Tuning—Allegro TimingVision Environment

Automatic delay tuning based on Smart Goals using Auto-Interactive Phase Tune

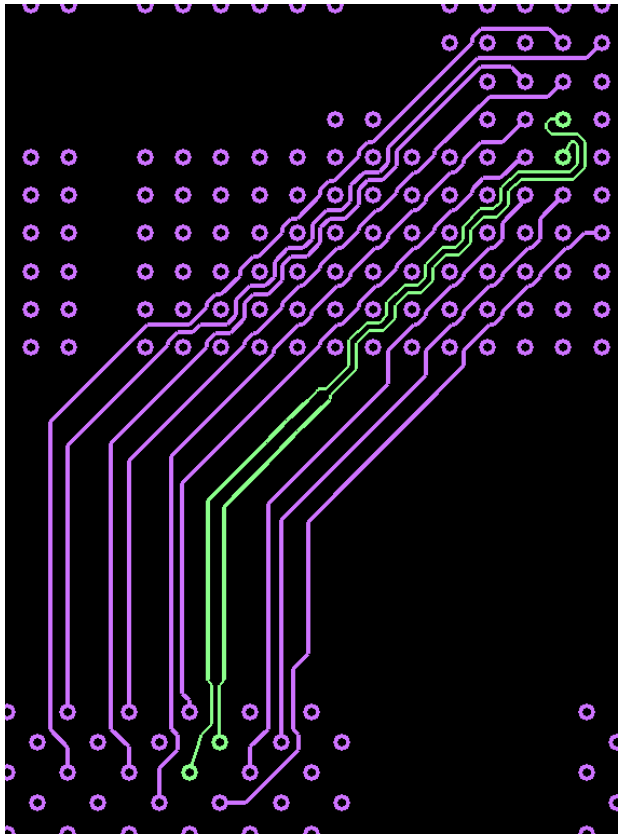
- Automatic color coding to identify which member needs a phase adjustment



- Auto-Interactive Phase Tune
- Set options and window select diff pair routes



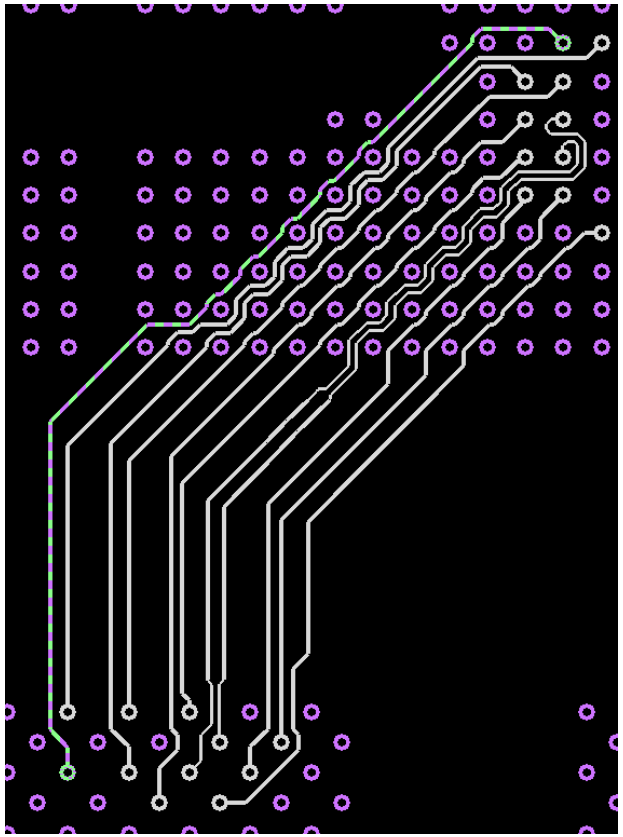
- Phase matching complete
- Color change indicates diff pair routing is in phase



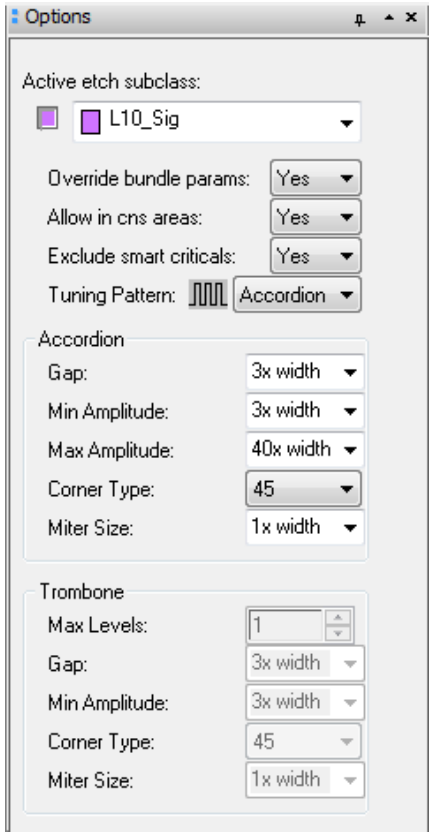
Intelligent Tuning—Allegro TimingVision Environment

Automatic delay tuning based on Smart Goals using Auto-Interactive Delay Tune

- Automatic color coding to identify short, long, and target (critical) nets



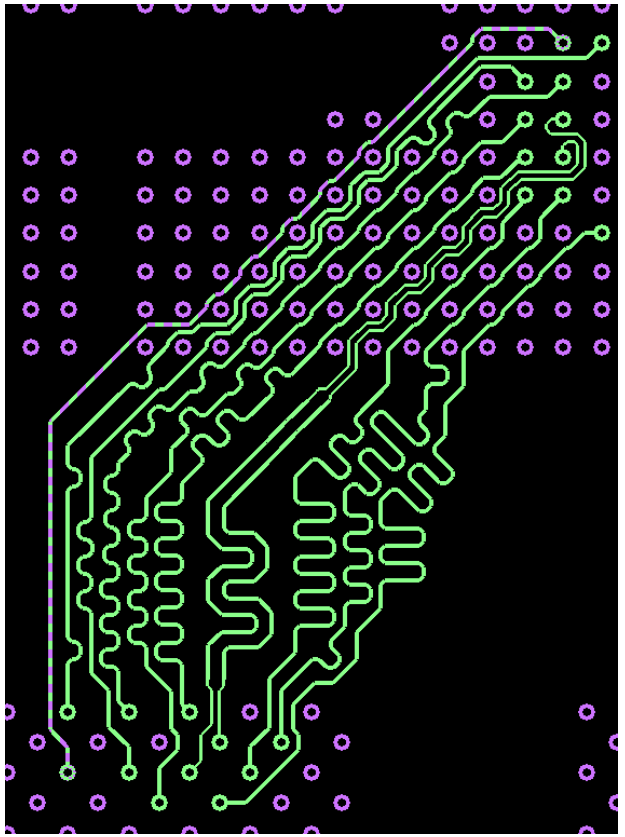
- Auto-Interactive Delay Tune
- Set options and window select routes



Auto-Interactive Delay Tune with Allegro® TimingVision™ runtime
10-12 sec

Manual delay without TimingVision runtime
8-10 min

- Delay tuning complete
- Color change indicates route tuning is just right ☺



Customer Reference Design

Route and tune DDR interface in 4 days vs. 4 weeks manually

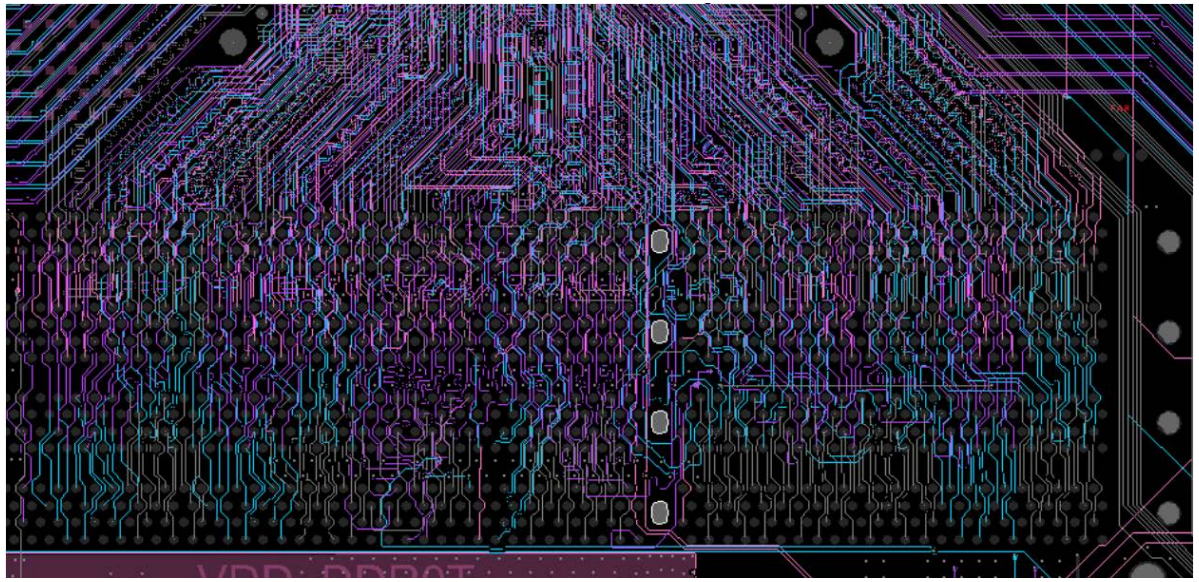
- Cavium Network Processor 78xx Evaluation Board
  - 1mm pitch BGA (2601 pins) / 4 DDR signal layers / 4 DDR3/DDR4 channels
  - Utilized a flow planning / Allegro TimingVision environment approach to DDR design which resulted in high-quality, tightly constrained DDR subsystem in less than one week!
- Customer quote:

  "With Allegro TimingVision environment, everything is right there in front of you—this simple fact allows the routing process to be sped up dramatically, from the manual routing efforts we have seen that can take up to four weeks, down to four days."

  Bill Munroe, Principal PCB Designer, Cavium
- Reference paper:
  - **Paper:** "Routing DDR3/DDR4 Channels to Specifications 4X Faster"
  - **Event:** CDNLive Boston 2014
  - **Speaker:** Bill Munroe
  - **Company:** Cavium

4 DDR Channels

2 Channels Across 4 DIMMs



Routing DDR4 Interfaces Quickly and Efficiently

Topics covered in this session

- DDR4 memory interfaces overview
  - Functional group to route group mapping, general design requirements, and bus topologies
- DDR4 electrical design rules overview
  - Electrical constraint targets, design stack-up, interconnect topologies, and placement techniques
- Routing technique on high-speed interfaces
  - Routing challenges, interface flow planning, route escape, and flow routing
- Timing closure on high-speed interfaces
  - Timing challenges, intelligent tuning (custom coloring, stippled patterns, real-time canvas feedback)

Routing DDR4 Interfaces Quickly and Efficiently

Topics we learned in this session

- Build a stable PCB design foundation
  - Careful planning of component placement and reserving space for pin escape (fanout)
  - Pin escape with interconnect topologies, routing, and any possible testing requirements in mind
  - Split plane planning during placement stage—too late once board is routed

- Route flow planning between devices is critical to the success of any design
  - Most direct route path to accommodate all routing including maintaining all design clearances
    - Automatic bundle creation and flow planning
    - Dynamic bundle and flow creation
    - Selective routing
  - Quick and precise component routing escape optimizing the interconnect order on both sides of the bus
    - Techniques of bundling bus routes to focus on finding an interconnect solution for each side of the bus
  - Effective timing closure with goal guidance using interactive and automatic means
    - Techniques of tuning routes efficiently using custom coloring and stippled patterns on routed connections for real-time feedback on canvas

YOUR FEEDBACK IS IMPORTANT! DON'T FORGET YOUR SPEAKER EVALUATION.

PLEASE REMEMBER TO RETURN THE EVALUATION FORMS TO THE PRESENTER, TO THE REGISTRATION DESK OR TO THE DROP BOX IN THE LOBBY.

THANK YOU,

PCB WEST SHOW MANAGEMENT