



SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA



JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Memory IO Block Routing Optimization using Semi-Automation of Single Trunk Steiner Tree Routing

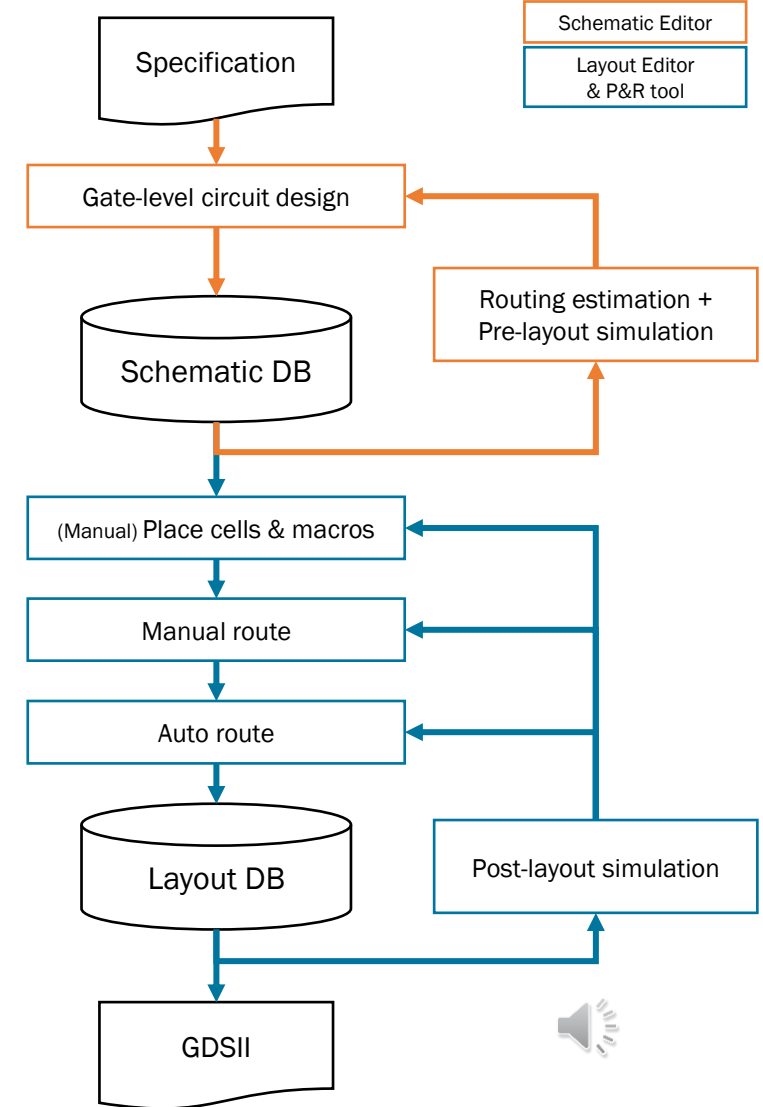
Seyong Ahn, Hyeyoun Kim, Kwangok Jeong and Jung Yun Choi
Samsung Electronics, Memory Business



Design Flow of Memory IO Blocks

- Overall design flows

- Design of gate-level circuits using schematic editor based on specification
- Simulation of circuit operations and performances by estimating routing patterns and parasitic RC
- Semi-automated placement of standard cells & macros based on schematic DB
- Manual routing of pre-simulated nets in the form of Single Trunk Steiner Tree
- Auto routing of the rest of the nets
- ECO based on post-layout simulation
- Tape out the layout after physical verifications



Design Flow of Memory IO Blocks (cont.)

- **Characteristics**

- Target specification and constraints are met using pre-layout simulation in the schematic design stage
- The purpose of P&R is to ensure the consistency between schematic and layout
 - Placement is determined during the schematic design
 - Critical net are simulated and routed in the form of Steiner tree with the same constraints used for schematic design

- **Limitations in terms of physical implementation**

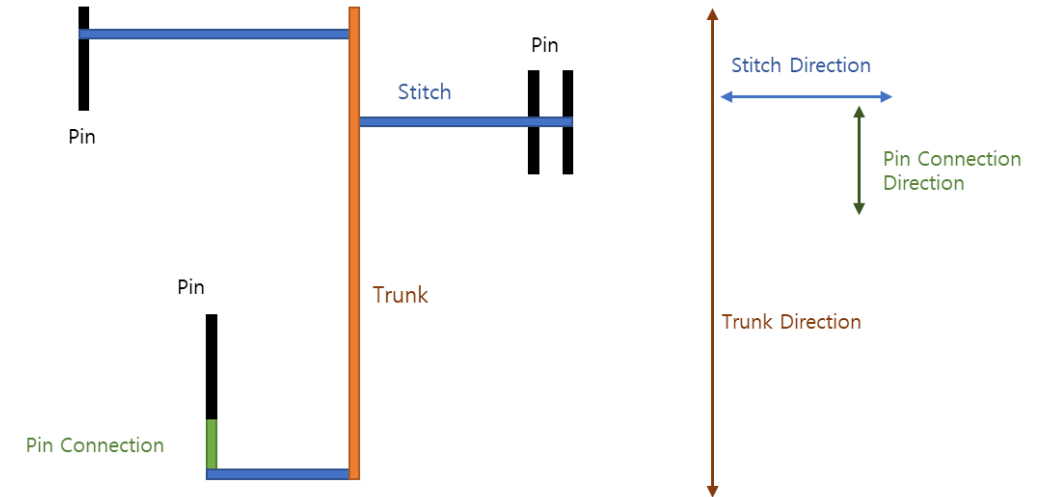
- A lot of manpower and turn-around time are required during manual routing
- Human errors occur during manual job
- Mismatches and deterioration of routing quality



Automation of Single-trunk Steiner Trees

- **Single-Trunk Steiner Trees (STST)**

- A special case of rectilinear Steiner minimum tree
- A net is routed with a single wire (trunk) and orthogonal wire segments (stitches) connecting each pin and the trunk.



- **Fishbone router**

- **Overview**

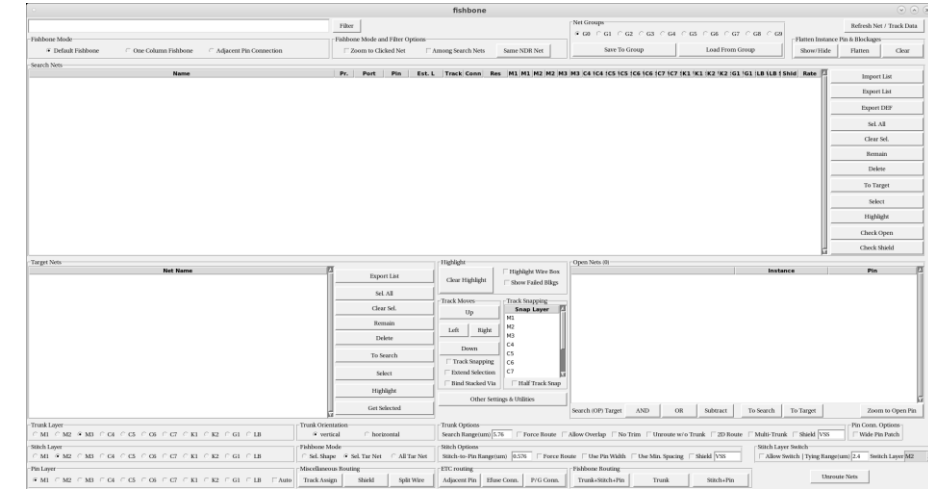
- An internally developed router using Text Command Language(Tcl) and Innovus text commands (We call it a “fishbone” as the STST geometry is similar to the shape of fish’s bone)
- The router is launched on Innovus and can be used interactively with its own GUI
- Automated assignment of each wire segment while minimizing length of each net and avoiding short and spacing violations



Automation of Single-trunk Steiner Trees (cont.)

- **Characteristics**

- Routed shapes follow given non-default rule
- Both on-grid and off-grid routing is supported
- The router makes a net be open rather than making shorts
- Three routing hierarchies to enhance routability (pin connection)



Fishbone router GUI

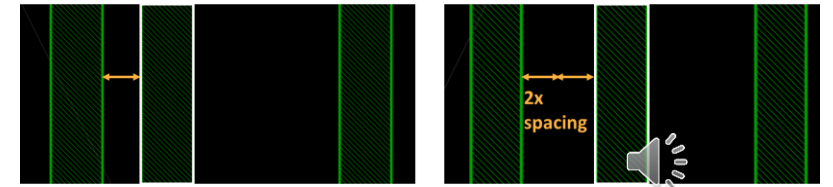
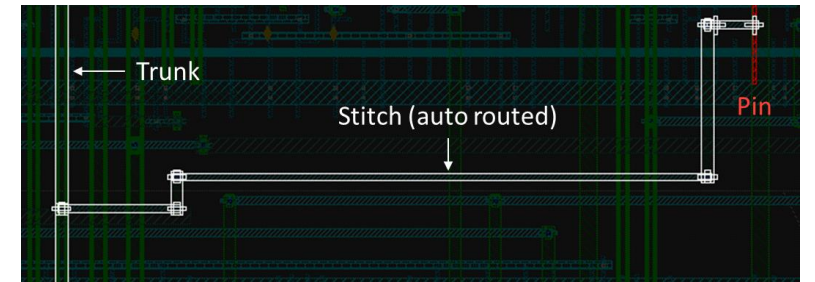
- **Limitations**

- As stitches cannot have jogs and each segment is routed sequentially, nets on the congested area might be left open
- Long parallel segments are routed closely, which leads to high cross-coupled capacitance



Routing Optimization Techniques

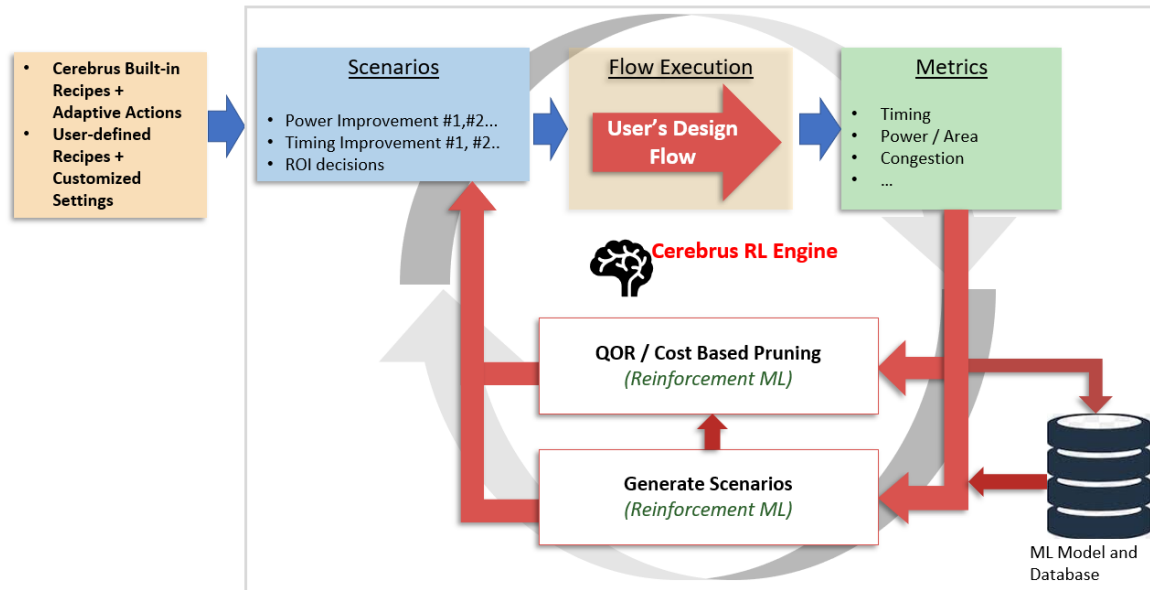
- **Stitch routing using P&R tool's auto router (Innovus nanoRoute)**
 - Trunk segments are converted from “wire” to “special wire”
 - Using “setAttribute” command, the following attributes are set on each net (-sub_net, -max_fanout, -top/bottom_preferred_routing_layer, -stripe_layer_range ...)
- **Reduction of net parasitic RC**
 - Each net is assigned to modified non-default rules having multiplied spacing compared to original one
 - Trunk segments are assigned automatically using the fishbone router



Routing Optimization Techniques (cont.)

- AI-driven router parameters optimization (Cadence Cerebrus)

- Routing quality is optimized using reinforcement learning
- Objective function & candidate parameters to be optimized are set by a designer
- Runtime and resources to explore huge design space are reduced



Cerebrus Reinforcement Learning Workflow, Cerebrus User Guide 22.14

User-defined routing parameters

- Skipping fixed object related violations
- Spreading routing at post route
- Using instance pin as feedthrough
- Reducing the total number of vias
- Using M1 layer as a pin extension only
- Enclosing via geometry inside std. cell pins
- Allowing non-preferred direction routing



Experimental Settings

- Tools

- Place & Route: Cadence Innovus 22.13_s094_1
- AI-driven Opt.: Cadence Cerebrus 22.14-s003_1

- Test design

- Flash IO block
 - Instance count: 4,765
 - Net count: 6,130

- Experiment cases

Cases		Baseline (Taped-out)	Case1	Case2	Case3
All nets (6,130)	Critical nets (1,342)	Fishbone routing + manual routing (for open nets)	Automated fishbone routing	Automated fishbone routing (2x trunk spacing)	Automated fishbone routing
	Auto nets (4,788)	Auto routing	Auto routing	Auto routing	Auto routing with Cerebrus Opt.

Experimental Results

- Key routing quality factors

- Total wirelength is reduced by 0.27% to 0.89% compared to Baseline (Taped-out)
- Routed net length overhead compared to FLUTE estimation is reduced by 0.3%p to 1%p from Baseline to Case1
- Total net resistance is decreased by 10.5% and total net capacitance is increased by 2.1% from Baseline to Case2
- Total physical #DRC and #shorts are reduced by 80.0% and 39.6%, respectively

- Turn-around time

- Total routing time has reduced from 1 week to 1 day

		Case1	Case2
Total wire length (compared to baseline)	All nets	-0.59%	-0.29%
	Critical nets	-0.89%	-0.27%
	Auto nets	-0.49%	-0.29%

		Baseline	Case1	Case2
Wire length overhead (compared to FLUTE*)	All nets	6.5%	6.1%	6.4%
	Critical nets	5.3%	4.3%	5.0%
	Auto nets	7.0%	6.7%	6.9%
Detoured net count (compared to FLUTE)	All nets	4191	4326	4357
	Critical nets	1240	1323	1324
	Auto nets	2951	3003	3033

		Case1	Case2
Wire resistance (compared to baseline)	All nets	4.24%	-10.54%
	Critical nets	5.81%	2.87%
	Auto nets	3.69%	-15.22%
Wire capacitance (compared to baseline)	All nets	1.82%	2.14%
	Critical nets	6.91%	1.02%
	Auto nets	-0.01%	2.54%



*FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design, TCAD, 2007

Experimental Results (cont.)

- Additional AI-driven routing results optimization (Case 3) based on the human engineer's optimized result (Case 1)
 - Total nets and auto routed nets' wirelengths are reduced by 0.80% and 1.03%, respectively
 - Total net resistance and capacitance are decreased by 2.93% and 1.92%, respectively
 - Total physical #DRC and #shorts are reduced by 72.5% and 57.1%, respectively
 - Total Cerebrus runtime is 5 hours (Each route runtime is about 2 hours)

(compared to Case1)		Case3
Total wire length	All nets	-0.80%
	Critical nets	-0.18%
	Auto nets	-1.03%
		Case3
Wire resistance	All nets	-2.93%
	Critical nets	-2.73%
	Auto nets	-3.00%
Wire capacitance	All nets	-1.92%
	Critical nets	-0.60%
	Auto nets	-2.45%
		Case3
Total physical DRC	All nets	-72.5%
Short	All nets	-57.1%



Summary

- **Characteristics and limitations of existing memory design methodology**
 - Target specification and constraints are met using pre-layout simulation in the schematic design stage
 - The purpose of P&R is to ensure the consistency between schematic and layout
 - Due to manual jobs, routing requires a lot of manpower, time and might result in human errors
- **Our contributions**
 - We devise a fishbone router automating single-trunk Steiner tree based routing, which reduces turn-around time of routing
 - We set environments and options of P&R tool's auto router to assist routing of fishbone router's open nets, which leads to routability enhancement and physical DRC violations reduction by considering comprehensive design rules
 - We modify critical nets' non-default rule spacing and assign their trunk segments to lower resistance layers, which minimizes coupling capacitance overheads as well as resistance
 - We further optimize routing results by finding optimal parameters of the auto router with reinforcement learning
- **Our proposed methodology can reduce #DRC violations and design turn-around time significantly, while maintaining layout expert's manual routing quality**

