DDR3 Case Study – High Speed Design Services, LLC

John Medina
PCB Layout Engineer
High Speed Design Services, LLC

Kim Owen
Signal Integrity Engineer
High Speed Design Services, LLC

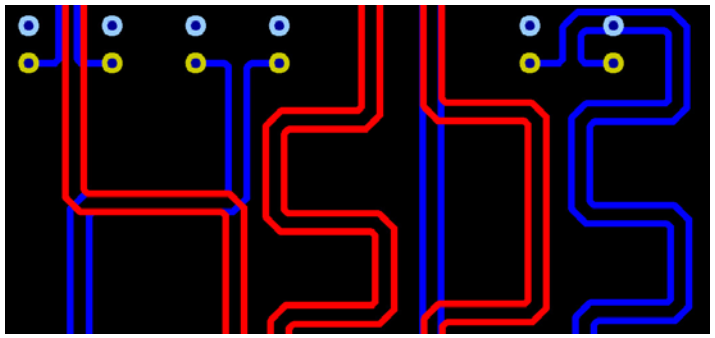
Audience and why you may be interested in this presentation

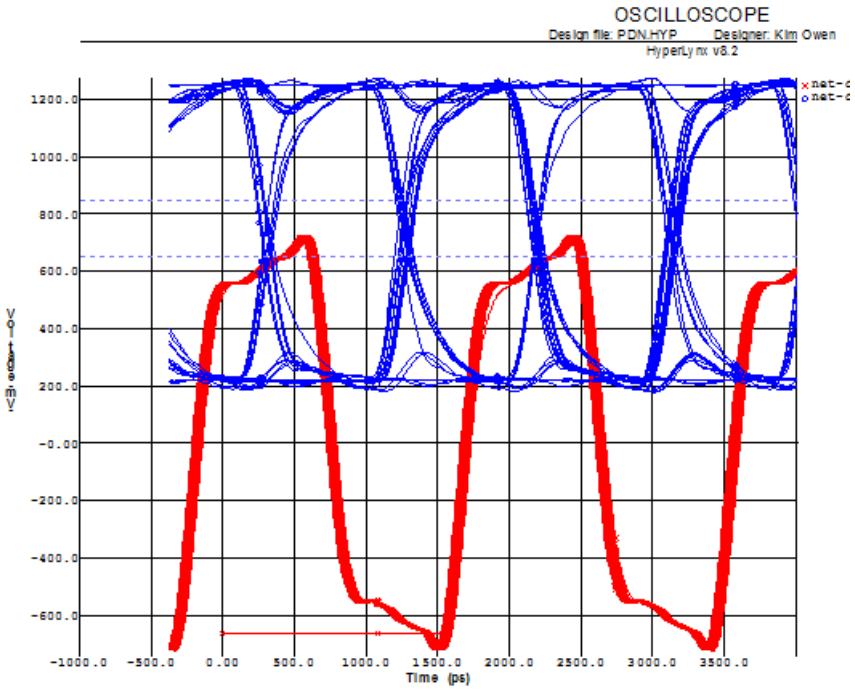
- PCB Designers
  - CES experience?
  - DDR3 routing and rule definition experience?
  - LVDS Differential routing experience?
  - High Pin count BGA breakout experience?
- Electrical Engineers
  - Do you define your specific design case timing and noise budgets based off simulations?
  - Knowledge of High Speed PCB Process?

HSDS Offers:

- *High Speed PCB Layout Class*
- *Signal Integrity Class*

DDR3 Case Study - Layout and Signal Integrity practices should be combined to deliver high-performance PCB designs





Objectives

- Review best practices used in high speed PCB design Layout
- To show how the collaboration of these best practices of layout and SI can lead to successful PCB designs
- Highlights of actual working DDR3/LVDS SERDES design

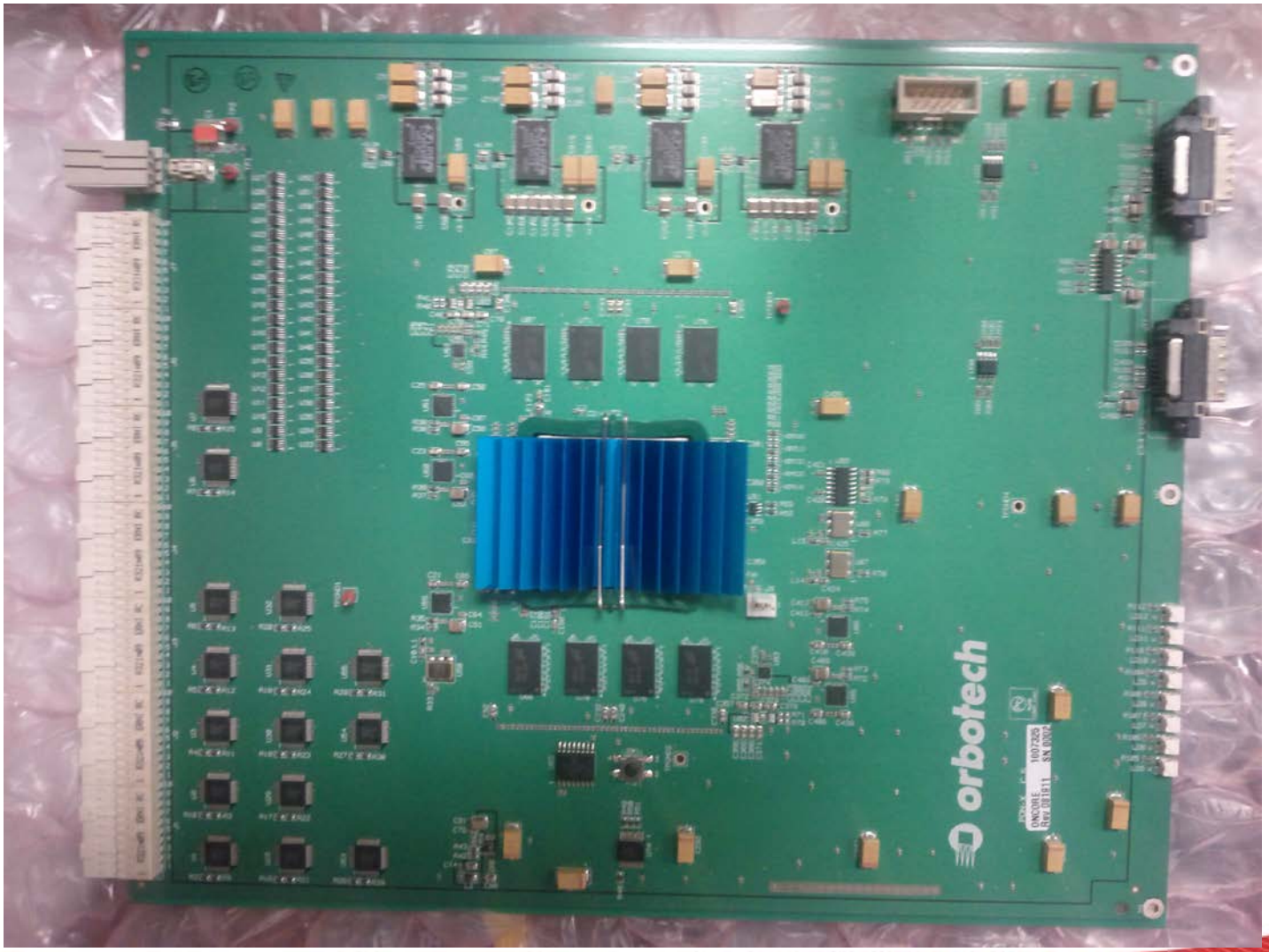
Results

- HyperLynx simulations and PCB design lessons learned
- Proven High Speed Process benefits are identified and featured

DDR3 Case Study Agenda

- Review of HDI High Speed Stackups and benefits.
- How to breakout high pin count BGA's. Power Planes and examples reviewed.
- IOD techniques and best practices
- CES - Constraint rules for DDR3 and LVDS SERDES.
- Routing best practices for DDR3 and LVDS SERDES.
- HyperLynx performance analysis by simulation (Eye Diagrams).
- Class information

DDR3 Design



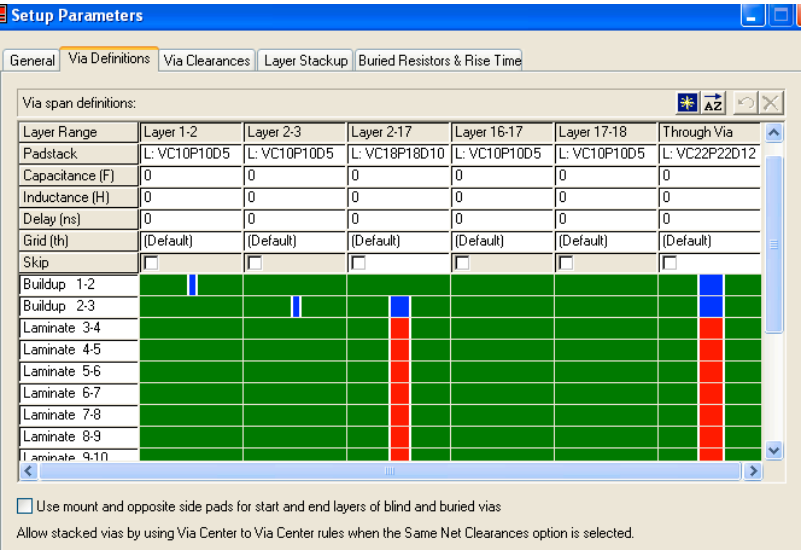
“Without saying too much, PDN is a very good example of success story”

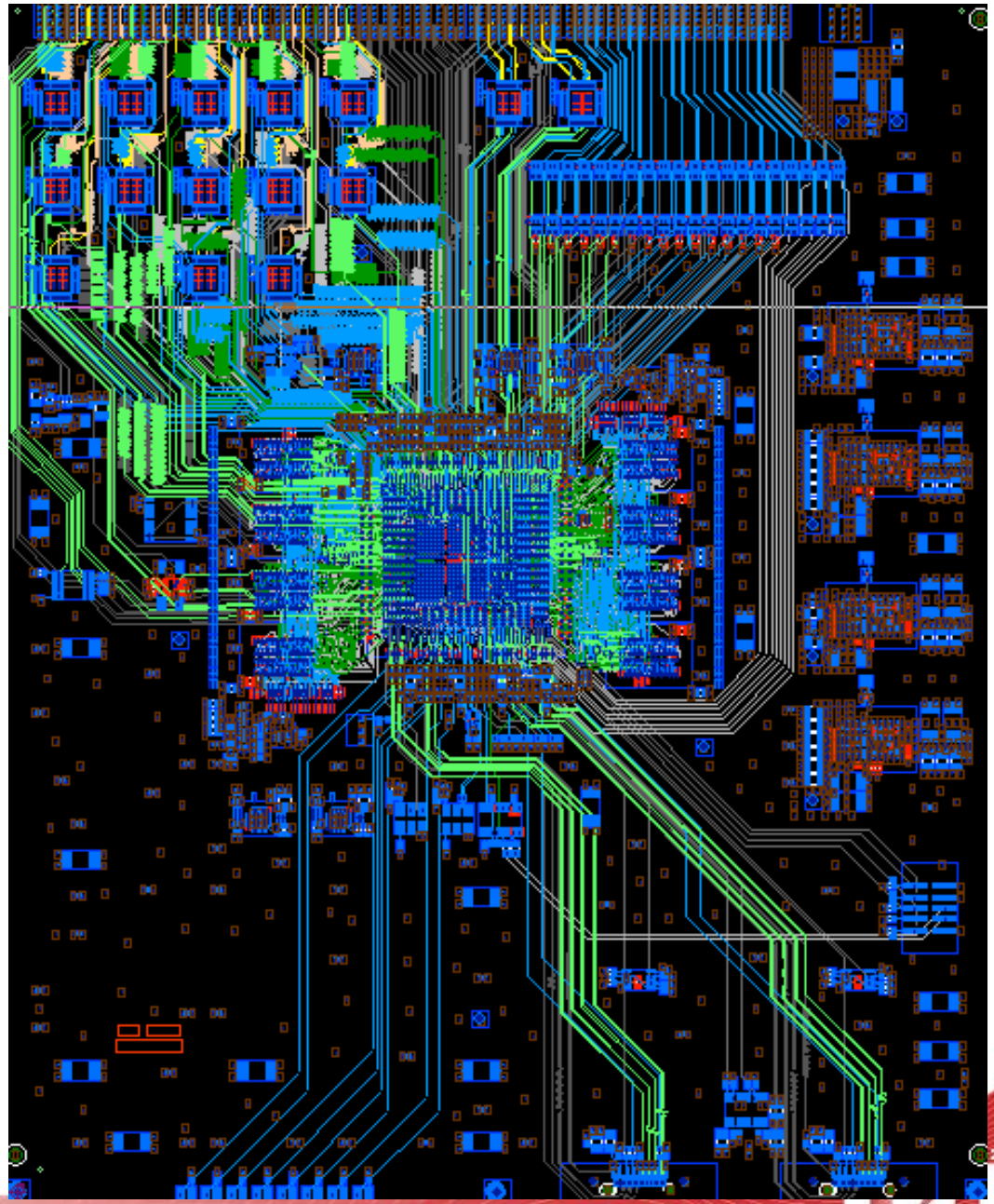
“Meticulous electrical design, perfect PCB design according to strict design rules and accurately tuned high speed traces gave us a perfect board.

Talking about DDR-3: as you probably recall, the architecture of PDN includes two banks of DDR3 memories, 64-bits each. My goal was operating the memories at about 400MHz (800Mbps). I have tested the board at maximum supported speeds (FPGA limited) of 533MHz (1067 Mbps): both banks are perfectly working with no issues. In addition, two High-Speed transceiver buses are perfectly running at 5Gbps per channel (overall data rate is 40Gbps), while 56 LVDS SERDES channels are running at 640Mbps.And, for the end, the first spin of the board is fully functional with no any electrical changes, blue wiring etc.

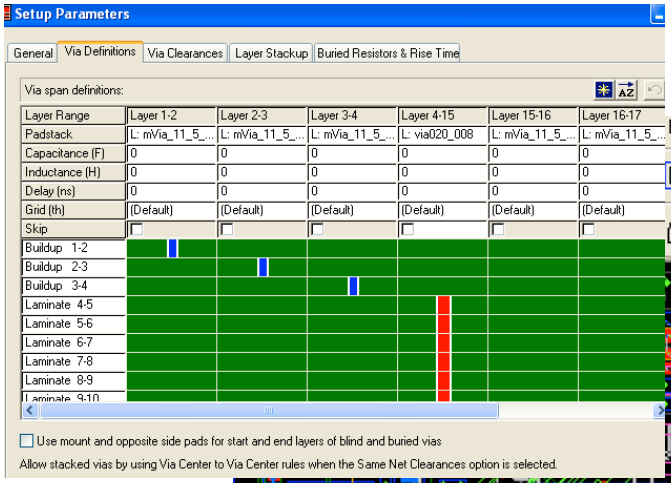
The first revision is probably going to be the last one. “

Altera EP4SGX180 KF40C2N



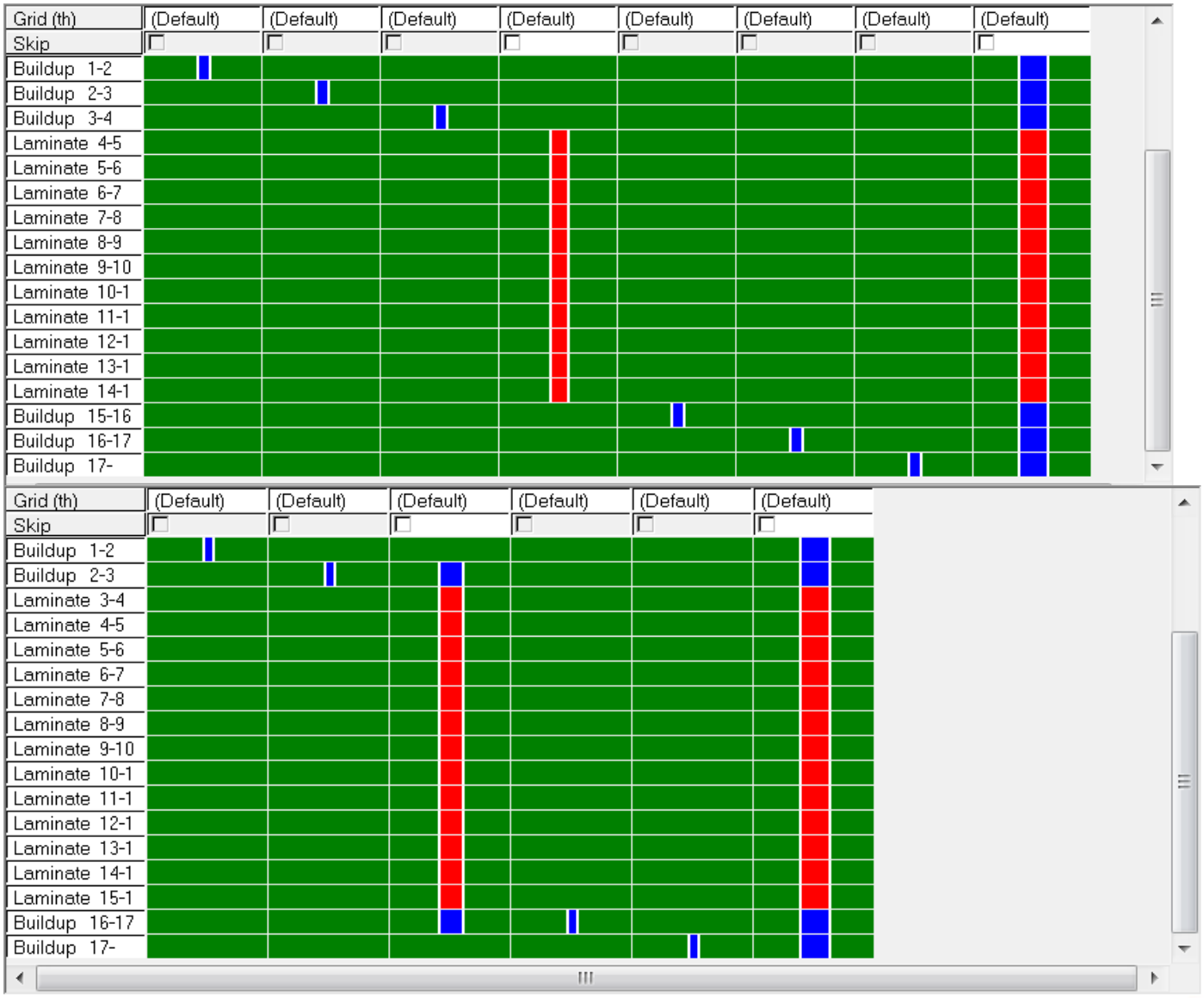


Xilinx U3 XC5VFX130T-1FF1738C|-SAT11255

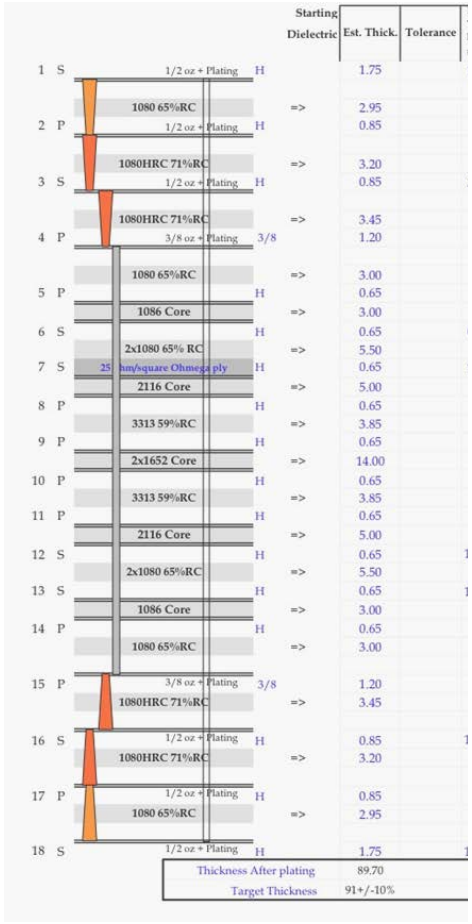


REVIEW OF HDI HIGH SPEED STACKUPS AND BENEFITS.

HDI Stackup Examples



DDR3 Design HDI Stackup



Layer	Cust Thickness	Calc Thickness	Primary Stack	Description
		0.0005		Taiyo 4000-BN
Layer - 1		0.0019		**3/8oz Sig (Std Plt)**
		0.0031	1080	FR408HR
Layer - 2		0.0019		**3/8oz P/G (Std Plt)**
		0.0046	106 106	FR408HR
Layer - 3		0.0006		**1/2oz Sig**
		0.0050	0.0050	FR408HR
Layer - 4		0.0012		**1oz P/G**
		0.0037	2113	FR408HR
Layer - 5		0.0006		**1/2oz Sig**
		0.0040	0.0040	FR408HR
Layer - 6		0.0006		**1/2oz Sig**
		0.0037	2113	FR408HR
Layer - 7		0.0012		**1oz P/G**
		0.0030	0.0030	FR408HR
Layer - 8		0.0006		**1/2oz Sig**
		0.0047	2116	FR408HR
Layer - 9		0.0012		**1oz P/G**
		0.0030	0.0030	FR408HR
Layer - 10		0.0012		**1oz P/G**
		0.0047	2116	FR408HR
Layer - 11		0.0006		**1/2oz Sig**
		0.0030	0.0030	FR408HR
Layer - 12		0.0012		**1oz P/G**
		0.0037	2113	FR408HR
Layer - 13		0.0006		**1/2oz Sig**
		0.0040	0.0040	FR408HR
Layer - 14		0.0006		**1/2oz Sig**
		0.0037	2113	FR408HR
Layer - 15		0.0012		**1oz P/G**
		0.0050	0.0050	FR408HR
Layer - 16		0.0006		**1/2oz Sig**
		0.0046	106 106	FR408HR
Layer - 17		0.0019		**3/8oz P/G (Std Plt)**
		0.0031	1080	FR408HR
Layer - 18		0.0019		**3/8oz Sig (Std Plt)**
		0.0005		Taiyo 4000-BN

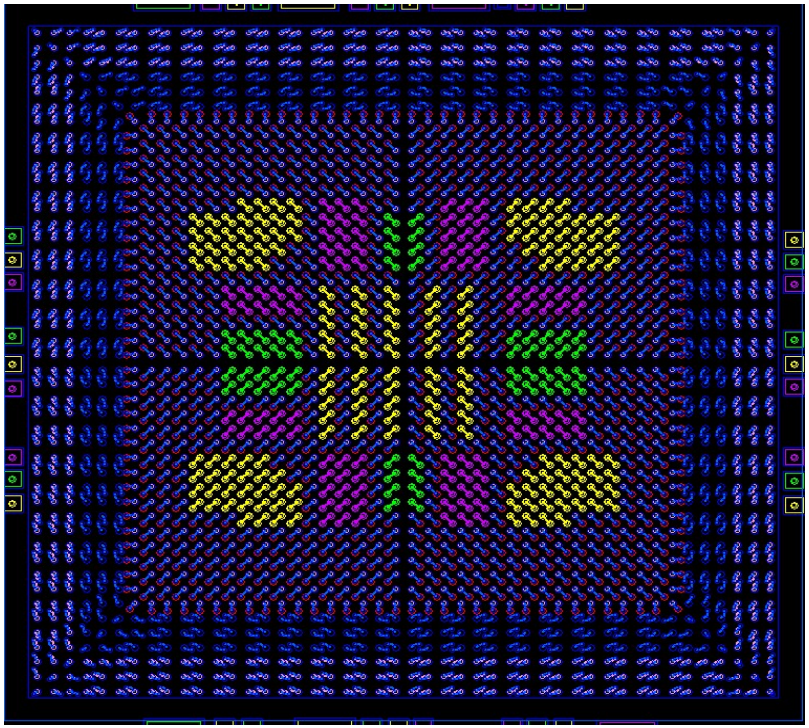
HDI Benefits

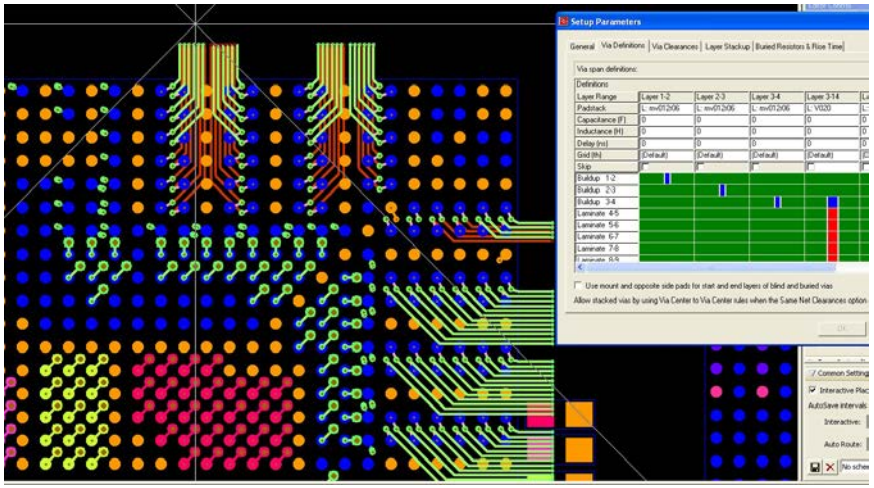
- The use of a custom microvia and core via structured breakout strategy for BGA's regions will assist in escape routing and increase power and ground planes
- Help eliminate via stubs when routing high speed nets
- Optimize discrete's placement, allowing room for breakouts and consider using microvia in pad for discrete parts with buried via combinations
- Create via grids to allow multi-channel routing in bga areas and over complete design wherever possible
- Create a stackup that allows the use of planes on outer 2 layers to pick up gnd and pwr connections

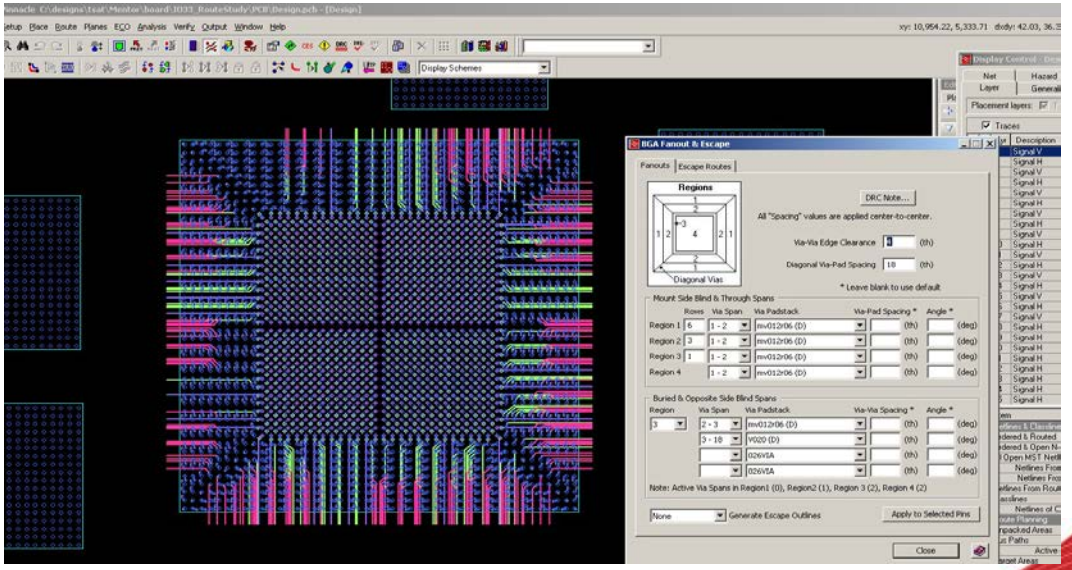
With these combinations of routing and placement best practices and optimization techniques we will reduce layer count. PC board's could be routed with less signal layers if these routing and stackup strategies are implemented.

HOW TO BREAKOUT HIGH PIN COUNT BGA'S

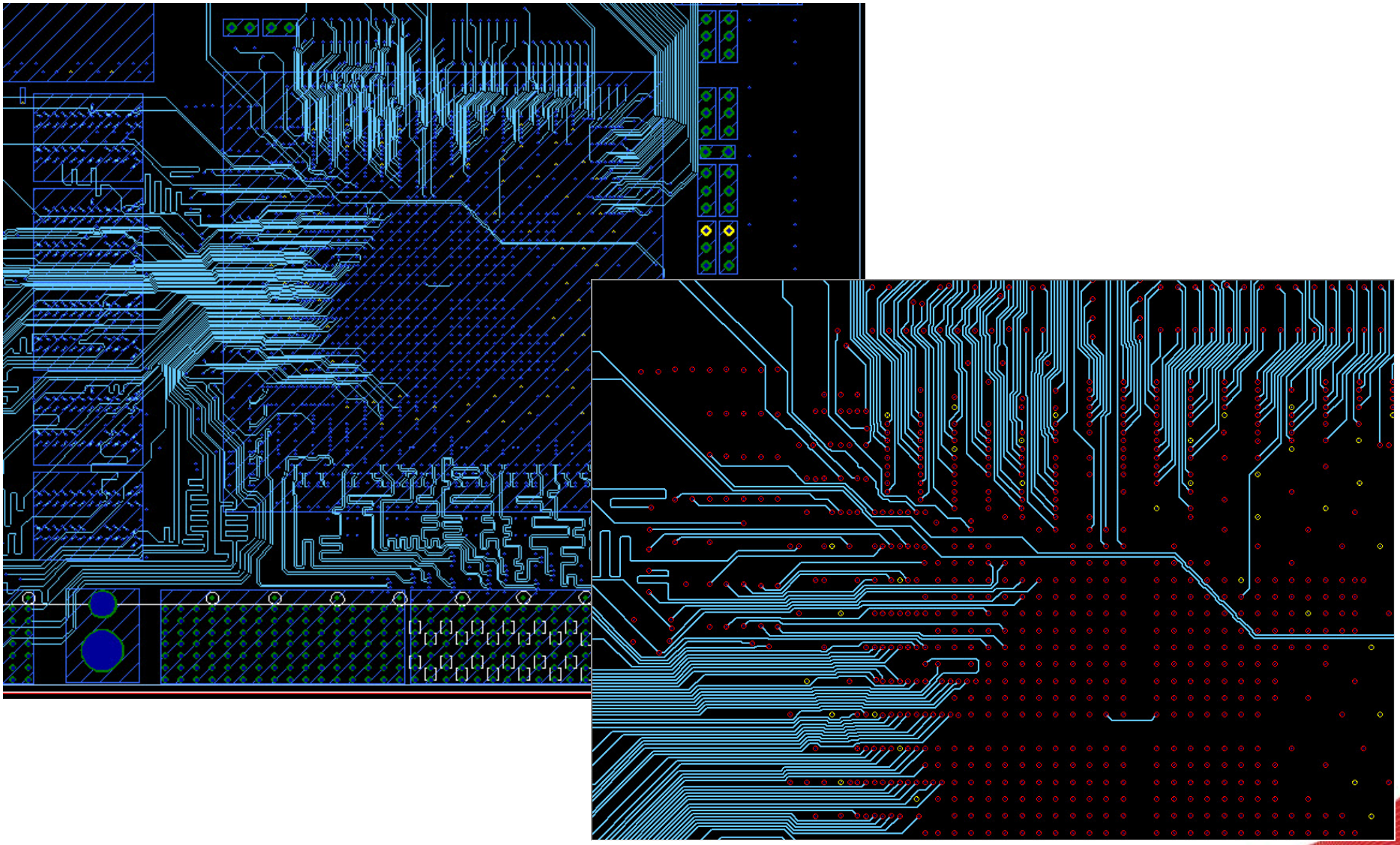
Example Breakout and Escape Patterns



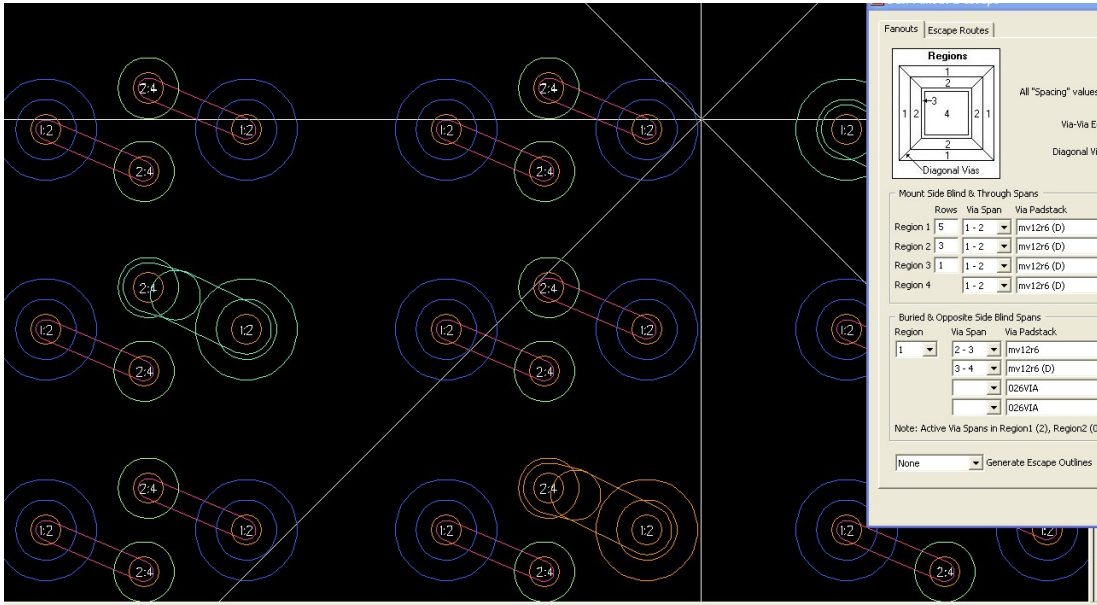


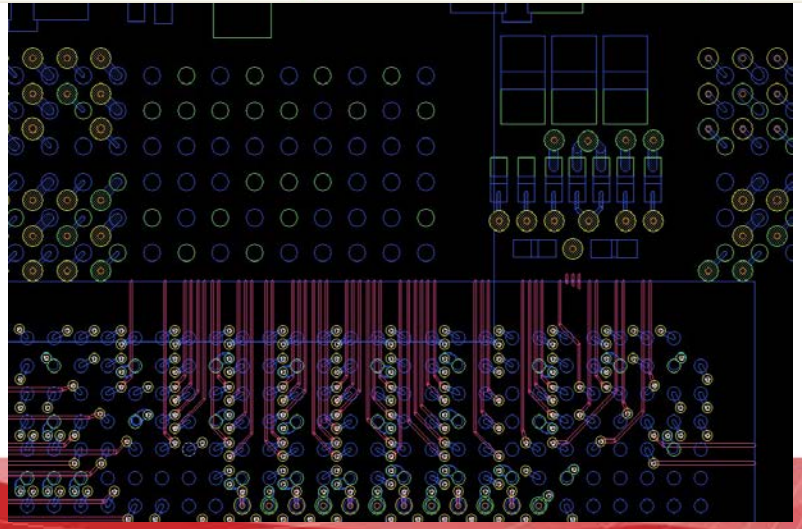


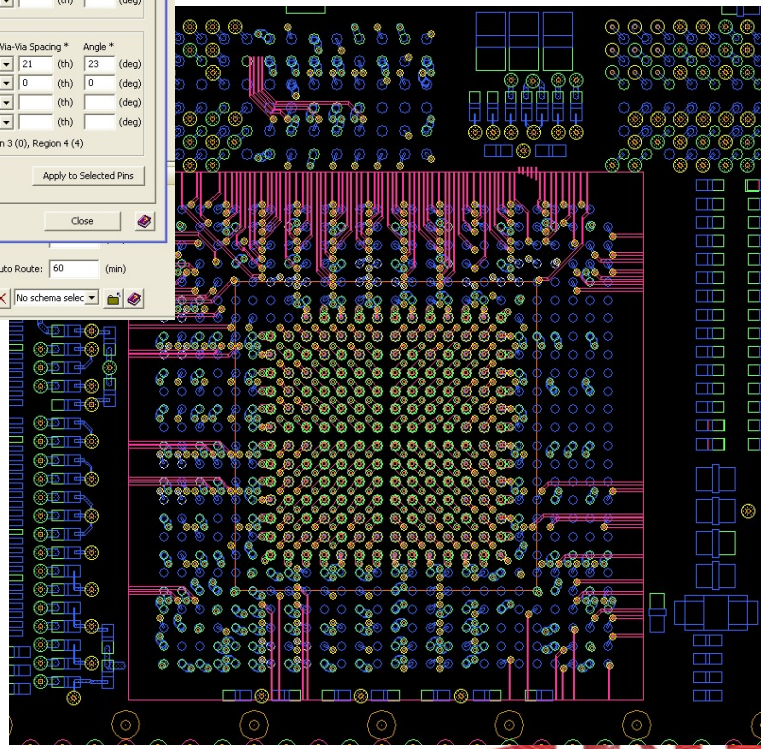
HDI Routing Layer 3



Microvia escape routing strategies will greatly enhance layer utilization





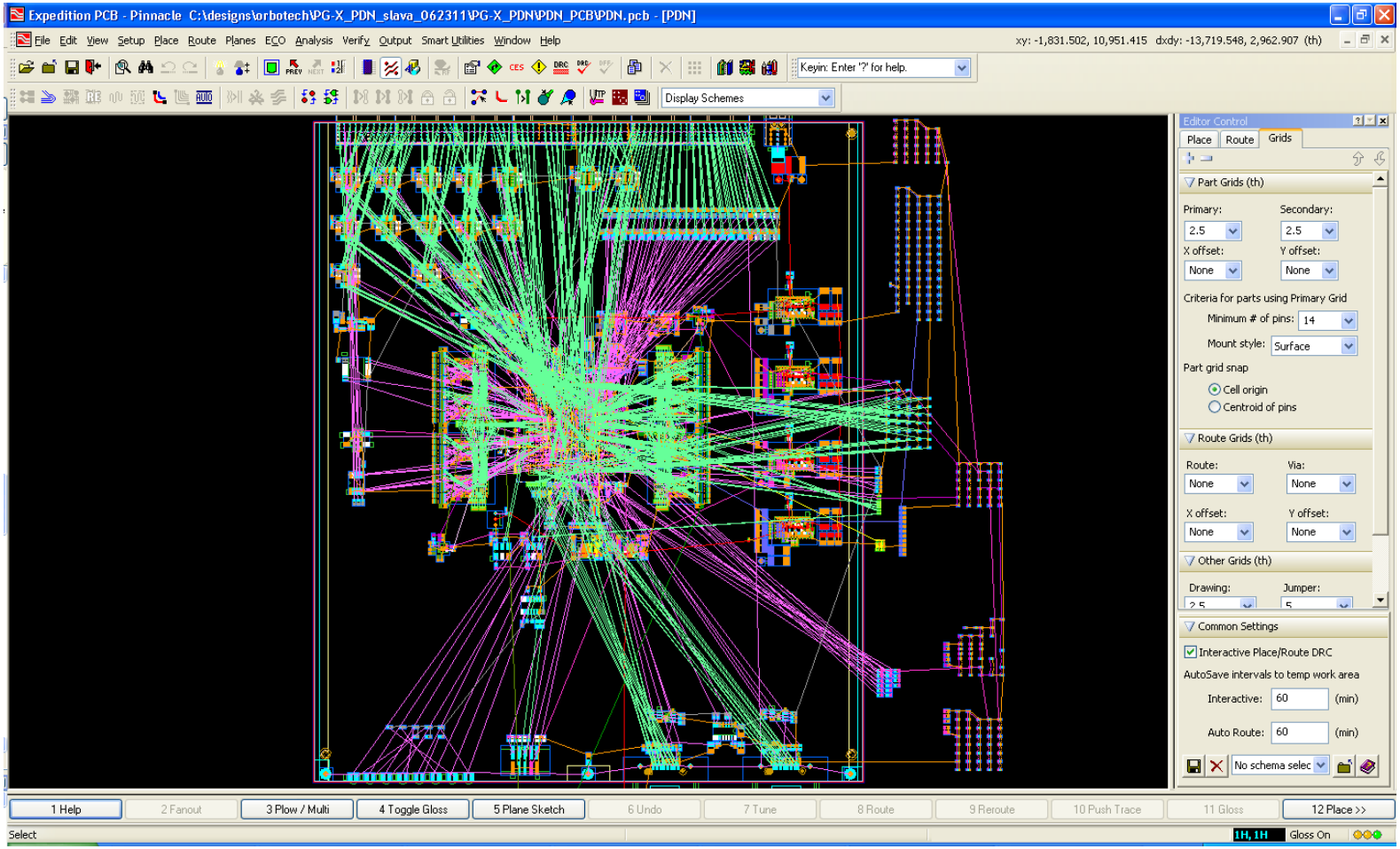


IOD TECHNIQUES AND BEST PRACTICES

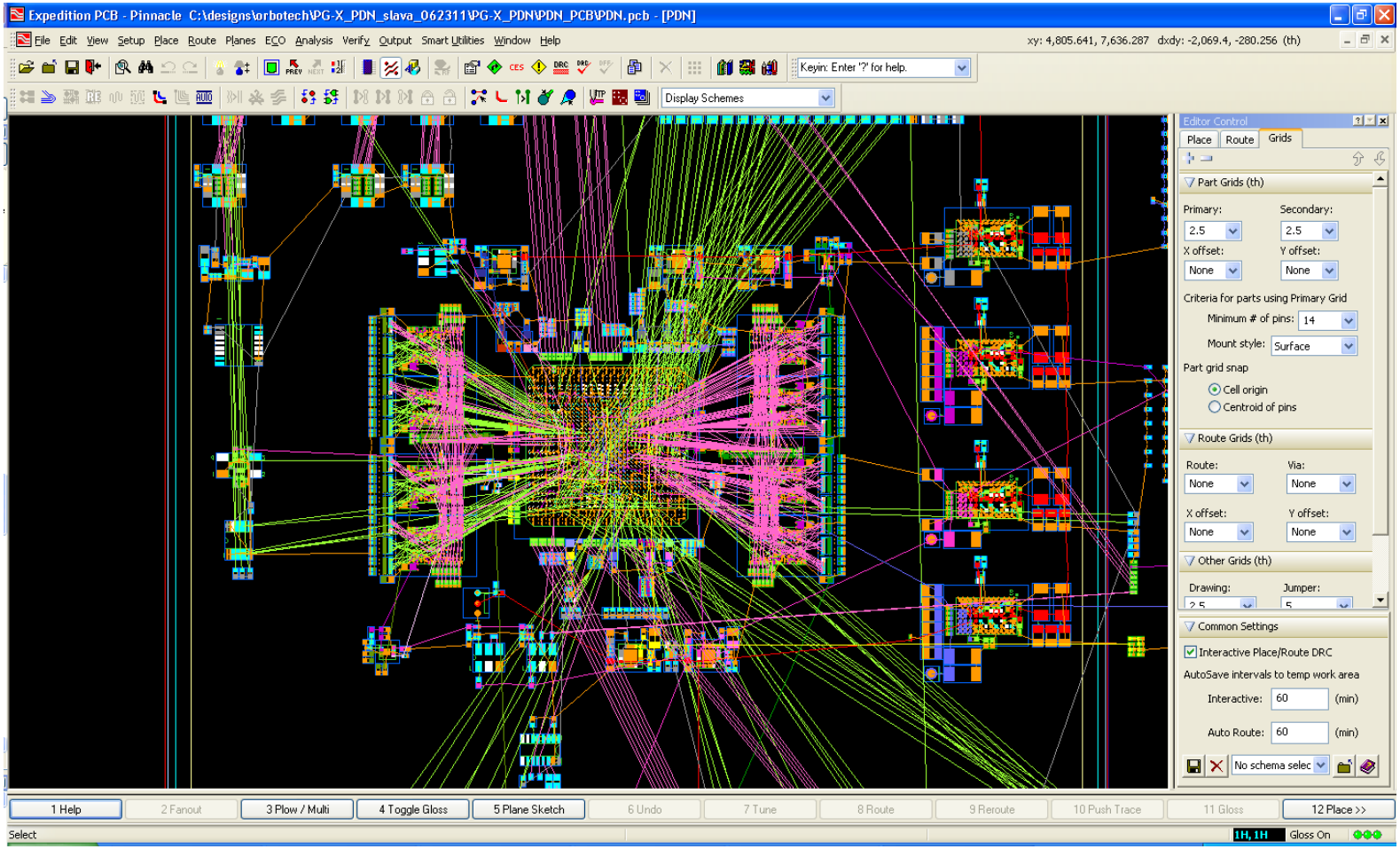
IO Designer Benefits

- IO Designer can be used early in the PCB process and schematic creation; and during initial placement to unravel nets on high pin count BGA's
- Then again after breakout and escapes are created
- The use of IO Designer is a good way to accurately predict optimal layer count and stackup when considering high pin count BGA's
- IO Designer tool can be used to optimally route traces out of BGA fields.
- "Unraveling" done in banks automatically rather than manually one by one. **A huge timesaver**
- If the IOD strategy is implemented – PCB design processes can be sped up significantly. A clear advantage for high density designs.
- To use IO Designer efficiently you must plan for it at the beginning and the IO Designer process must be followed

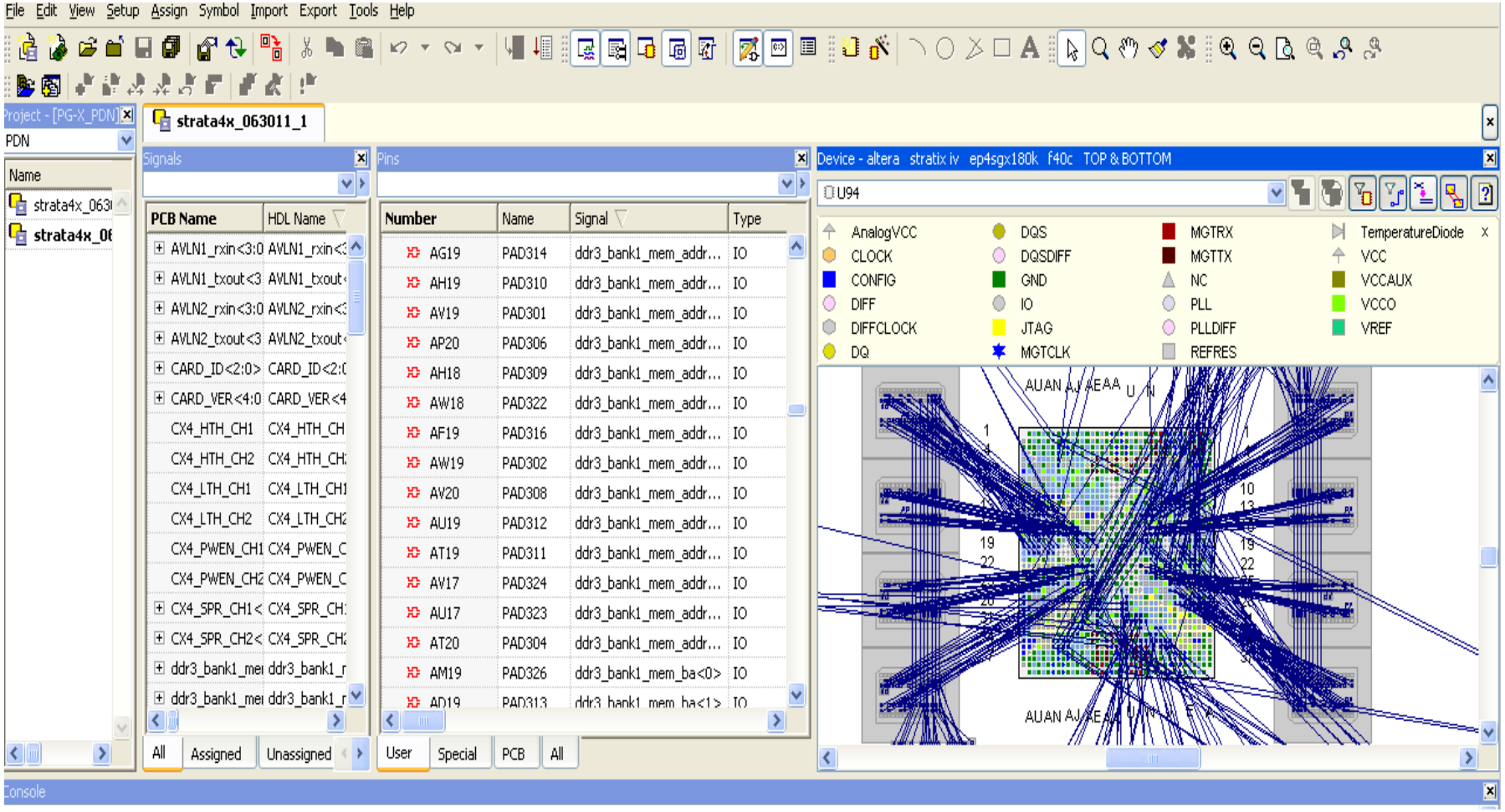
Before Unravel

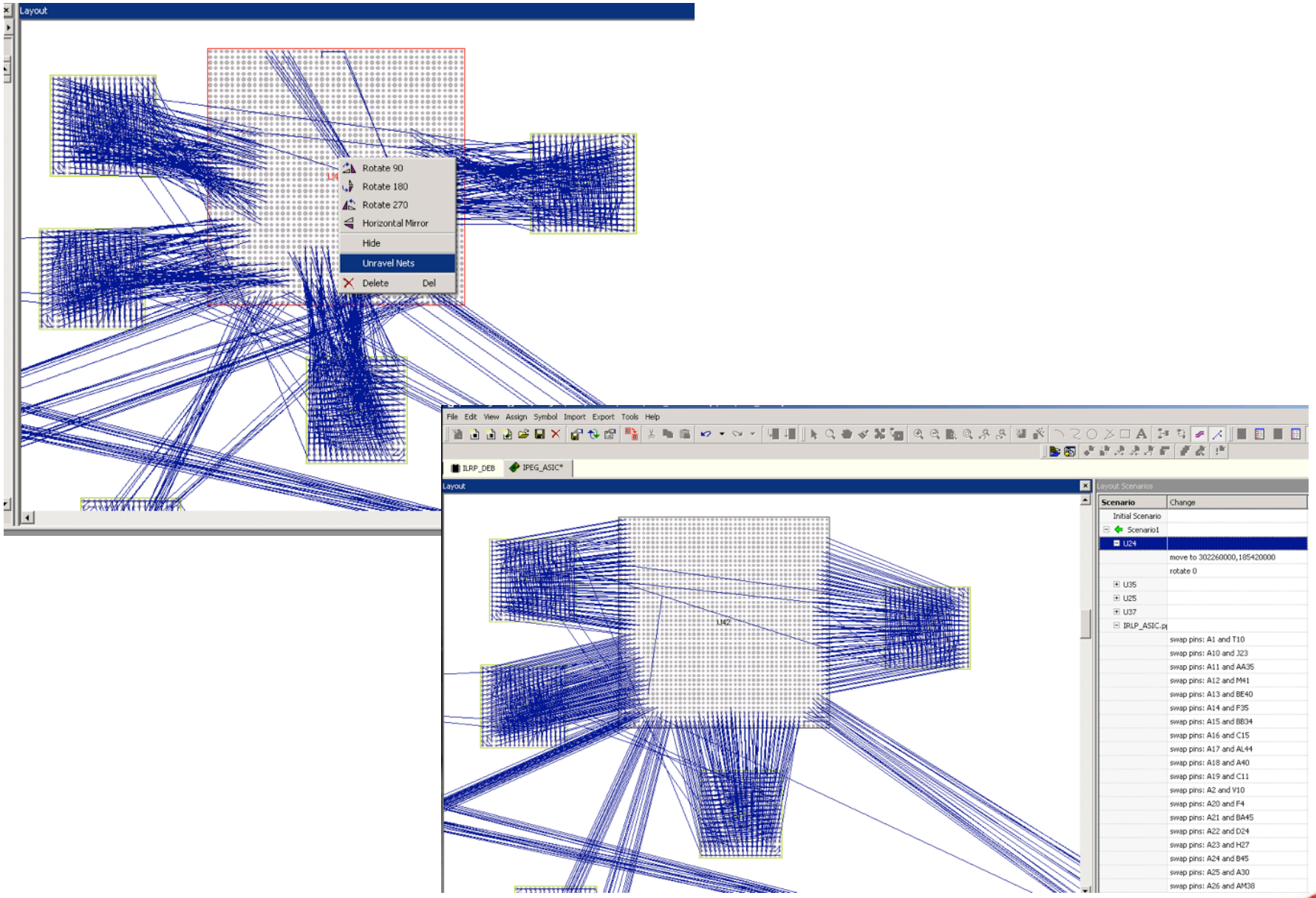


DDR3 Design Unraveled Using IOD



IOD Screen Shot




Layout
Rotate 90
Rotate 180
Rotate 270
Horizontal Mirror
Hide
Unravel Nets
Delete Del
File Edit View Assign Symbol Import Export Tools Help
IRLP_DEB
IPEG_ASIC*
Layout
Layout Scenarios
Scenario
Change
Initial Scenario
Scenario1
U24
move to 302260000,185420000
rotate 0
U35
U25
U37
swap pins: A1 and T10
swap pins: A10 and J23
swap pins: A11 and AA35
swap pins: A12 and M41
swap pins: A13 and BE40
swap pins: A14 and F35
swap pins: A15 and BB34
swap pins: A16 and C15
swap pins: A17 and AL44
swap pins: A18 and A40
swap pins: A19 and C11
swap pins: A2 and V10
swap pins: A20 and F4
swap pins: A21 and BA45
swap pins: A22 and D24
swap pins: A23 and H27
swap pins: A24 and B45
swap pins: A25 and A30
swap pins: A26 and AM38

CES - CONSTRAINT RULES FOR DDR3 AND LVDS SERDES.

How CES Works?

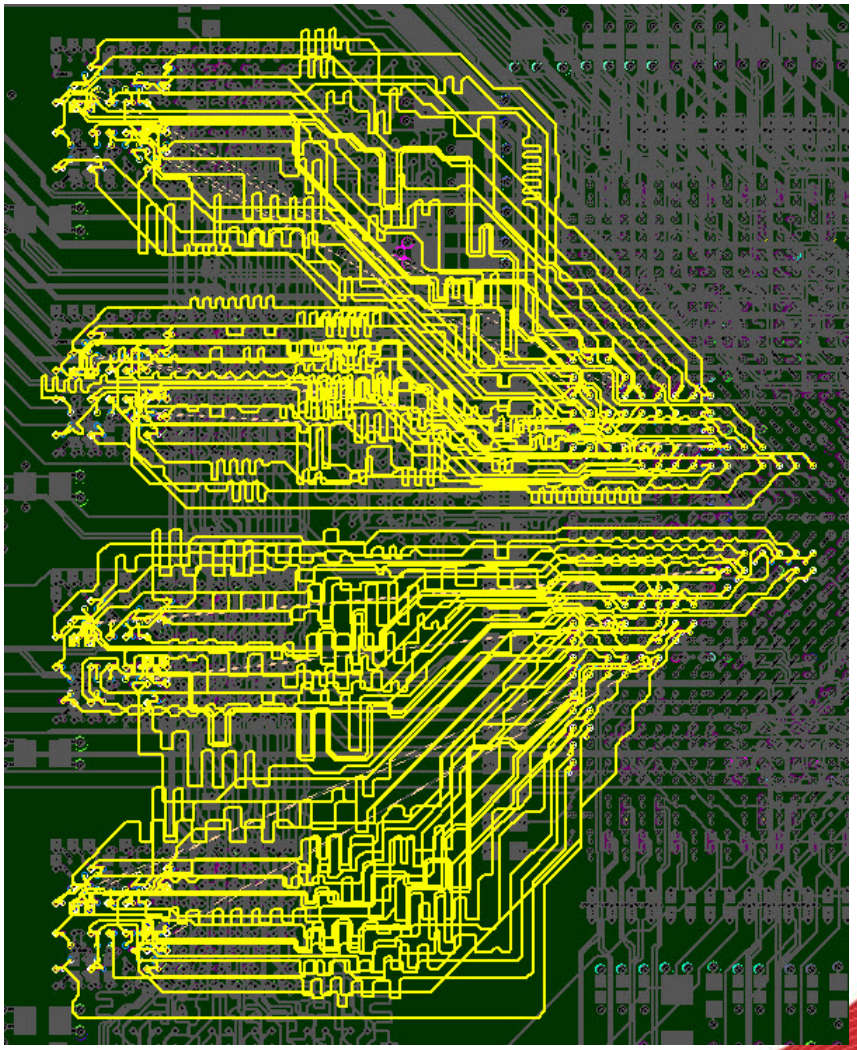
- CES works to organize all nets and enets into net classes and constraint classes while defining busses, clk nets and groups.
- Organizing overall netlist into interface based grouping and rule creation.
- This makes it easier for rule classifications of DDR3, SERDES, etc.. Into Electrical, physical, manufacturing constraint based rule sets.
- All Constraints will be accepted in the CES and can be reused by board or interface.

CES – Key Features

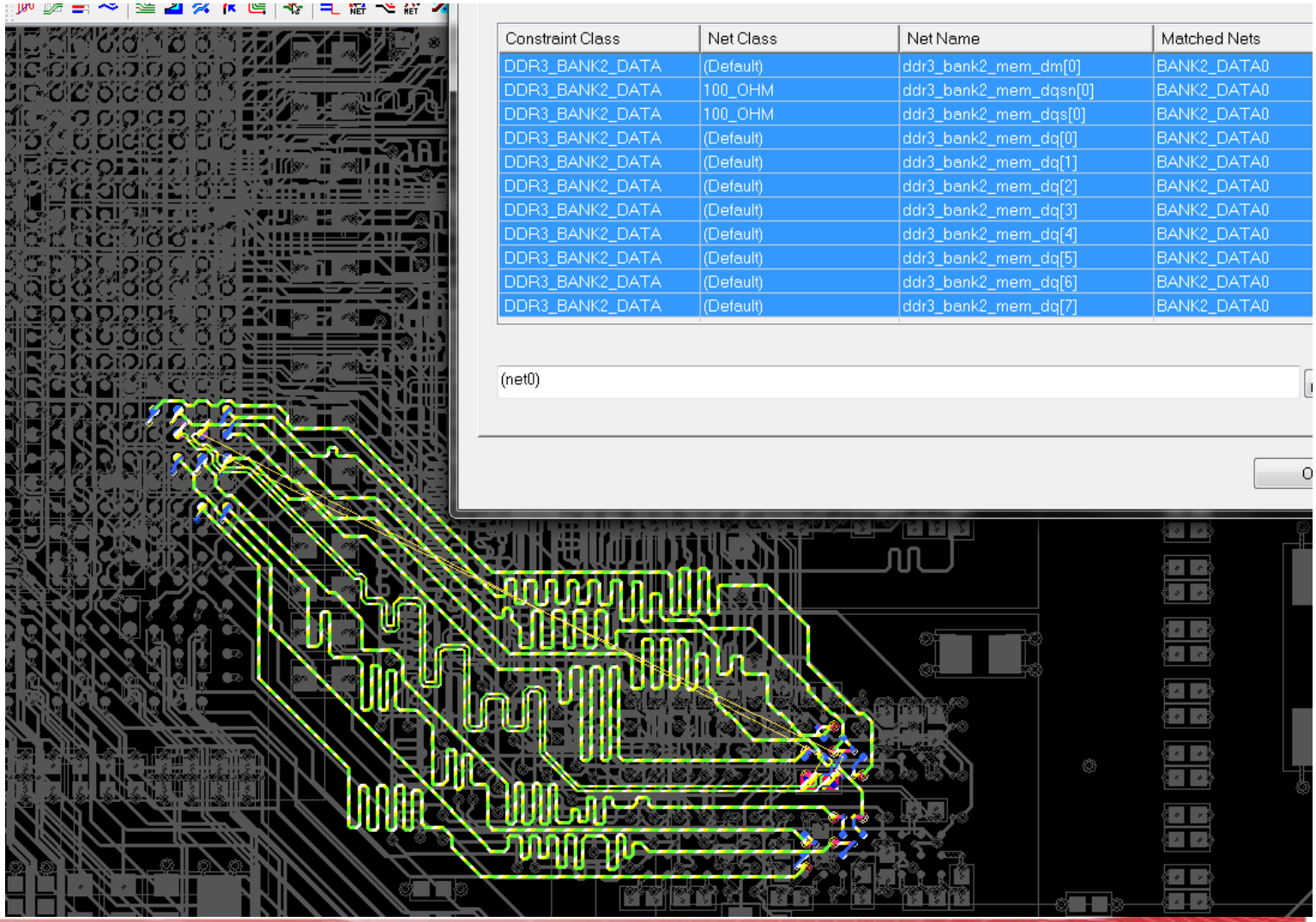
- CES is cross platform, worksheet and workbook based
- Able to set the constraint in schematics level and PCB level
- Common platform to define, view, edit and verify the layout constraints
- Supports Hierarchical management of constraints
- Analyze feature updates the spread sheet interactively
- Highlights design rule violation in real time

DDR3 DATA CES Rules

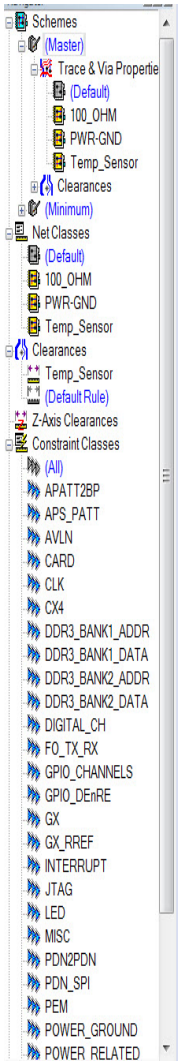
Constraint Class/Net	Match	Tol (th)/(pc)	Delta (th)/(pc)	Range (th):(th)/(pc):(p	Formulas: Formula	Formulas: Violation
ddr3_bank1_mem_dq[22]	data2	20	2.348	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.01th=1600.96:2400.96th
ddr3_bank1_mem_dq[21]	data2	20	2.079	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.28th=1600.96:2400.96th
ddr3_bank1_mem_dq[20]	data2	20	1.426	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.93th=1600.96:2400.96th
ddr3_bank1_mem_dq[19]	data2	20	2.134	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.22th=1600.96:2400.96th
ddr3_bank1_mem_dq[18]	data2	20	2.274	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.08th=1600.96:2400.96th
ddr3_bank1_mem_dq[17]	data2	20	2.247	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.11th=1600.96:2400.96th
ddr3_bank1_mem_dq[16]	data2	20	1.467	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.89th=1600.96:2400.96th
ddr3_bank1_mem_dq[15]	data1	20	0.009	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.96th=1600.96:2400.96th
ddr3_bank1_mem_dq[14]	data1	20	0.702	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.27th=1600.96:2400.96th
ddr3_bank1_mem_dq[13]	data1	20	0.898	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.08th=1600.96:2400.96th
ddr3_bank1_mem_dq[12]	data1	20	0.886	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.09th=1600.96:2400.96th
ddr3_bank1_mem_dq[11]	data1	20	0.771	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.2th=1600.96:2400.96th
ddr3_bank1_mem_dq[10]	data1	20	0.963	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.01th=1600.96:2400.96th
ddr3_bank1_mem_dq[9]	data1	20	0	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.97th=1600.96:2400.96th
ddr3_bank1_mem_dq[8]	data1	20	0.837	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.14th=1600.96:2400.96th
ddr3_bank1_mem_dq[7]	data0	20	0.835	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.5th=1600.96:2400.96th
ddr3_bank1_mem_dq[6]	data0	20	0.359	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.98th=1600.96:2400.96th
ddr3_bank1_mem_dq[5]	data0	20	0.969	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.37th=1600.96:2400.96th
ddr3_bank1_mem_dq[4]	data0	20	1.072	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.26th=1600.96:2400.96th
ddr3_bank1_mem_dq[3]	data0	20	1.043	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.29th=1600.96:2400.96th
ddr3_bank1_mem_dq[2]	data0	20	0.797	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.54th=1600.96:2400.96th
ddr3_bank1_mem_dq[1]	data0	20	1.09	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.25th=1600.96:2400.96th
ddr3_bank1_mem_dq[0]	data0	20	0.912	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.42th=1600.96:2400.96th
ddr3_bank1_mem_dm[7]	data7	20	1.602	1,999.65:2,00...	=(\ddr3_bank1_mem_dq[33]\)...	2000.16th=1600.96:2400.96th
ddr3_bank1_mem_dm[6]	data6	20	0.011	2,000.01:2,00...	=(\ddr3_bank1_mem_dq[33]\)...	2000.68th=1600.96:2400.96th
ddr3_bank1_mem_dm[5]	data5	20	0.07	2,000.135:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.41th=1600.96:2400.96th
ddr3_bank1_mem_dm[4]	data4	20	0.275	2,000.083:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.68th=1600.96:2400.96th
ddr3_bank1_mem_dm[3]	data3	20	0.856	2,000.035:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.07th=1600.96:2400.96th
ddr3_bank1_mem_dm[2]	data2	20	1.969	1,999.846:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.39th=1600.96:2400.96th
ddr3_bank1_mem_dm[1]	data1	20	0.373	2,000.009:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.6th=1600.96:2400.96th
ddr3_bank1_mem_dm[0]	data0	20	0.927	1,999.313:2,0...	=(\ddr3_bank1_mem_dq[33]\)...	2000.41th=1600.96:2400.96th



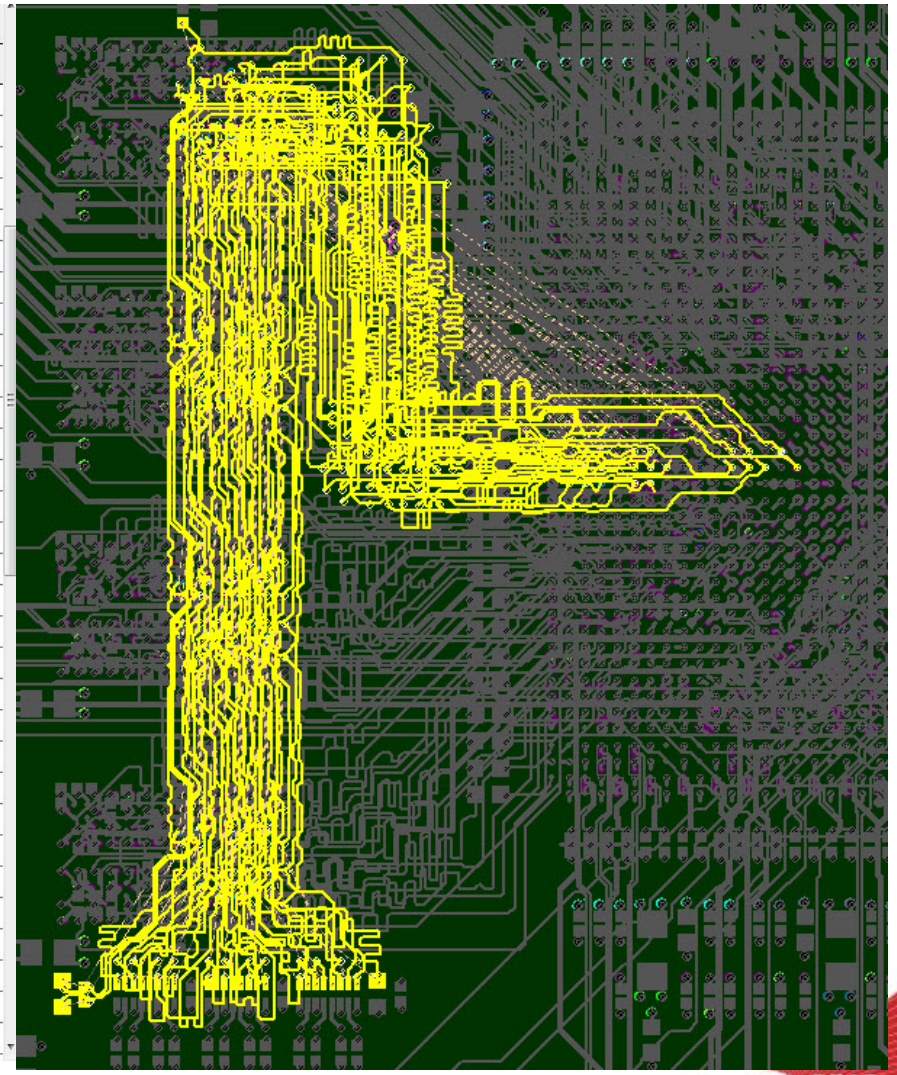
DDR3 0-7 and Clock; Constraint Rules



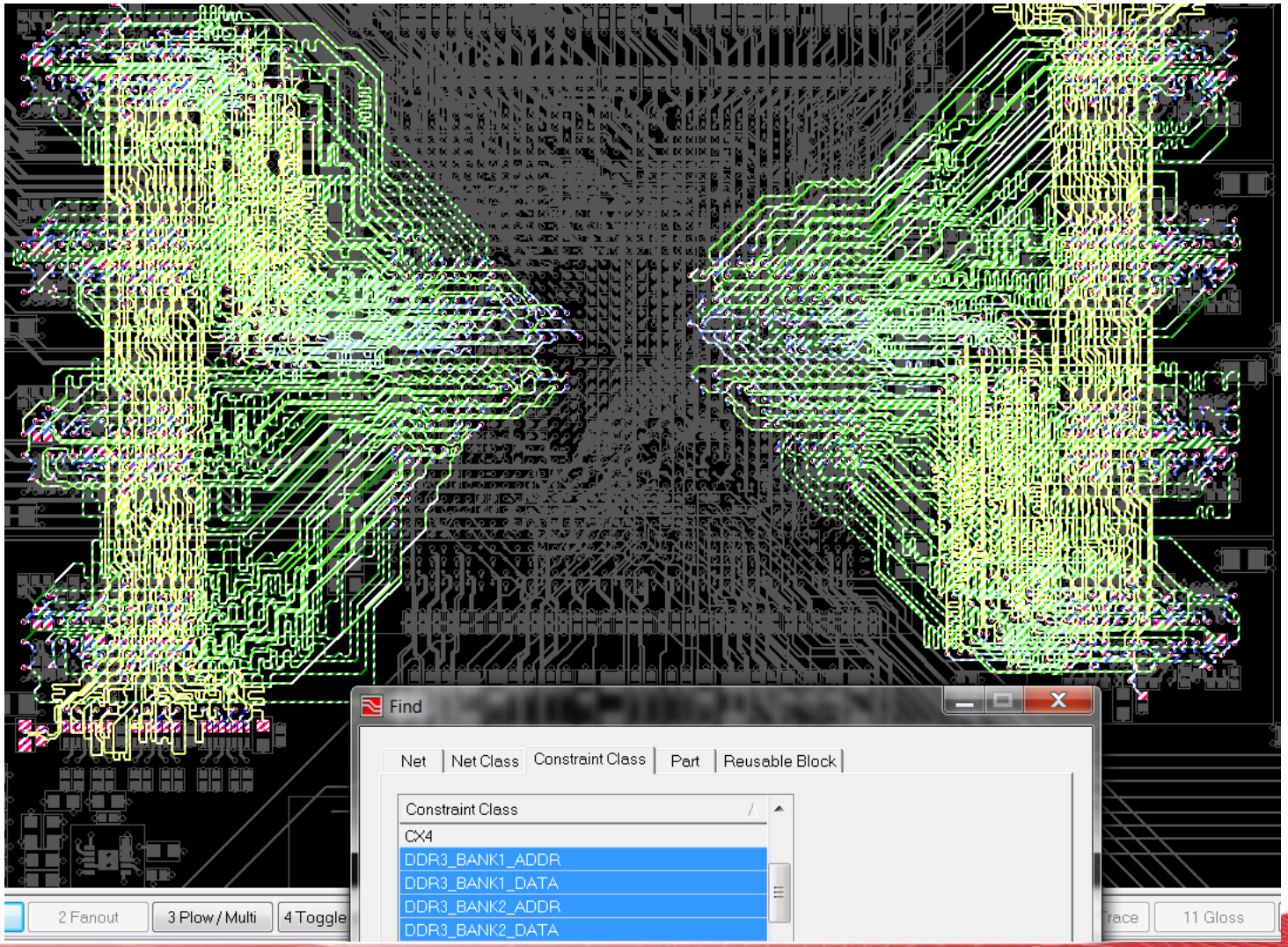
DDR3 CES - Address Rules



Constraint Class/Net	F Delay			
	Match	Tol (th)l(ns)	Delta	Range
ddr3_bank1_mem_clk_n				
ddr3_bank1_mem_clk_n				
S:U72-AV17,L:U66-K7	leg1	10	4.281	2,367.471:2,371.752
L:R275-2,L:C1284-2				
L:U75-K7,L:U78-K7	leg4	10	0	745.026:746.876
L:U78-K7,L:R275-2	leg5	10	0	603.028:605.056
L:U66-K7,L:U70-K7	leg2	10	0.942	742.664:745.966
L:U70-K7,L:U75-K7	leg3	10	3.3	744.065:747.6
ddr3_bank1_mem_addr[14]				
ddr3_bank1_mem_addr[14]				
L:U75-T7,L:U78-T7	leg4	10	1.725	745.026:746.876
L:U78-T7,S:RN4-7	leg5	10	1.539	603.028:605.056
L:U66-T7,L:U70-T7	leg2	10	0.134	742.664:745.966
L:U70-T7,L:U75-T7	leg3	10	2.075	744.065:747.6
S:U72-AT20,L:U66-T7	leg1	10	2.699	2,367.471:2,371.752
ddr3_bank1_mem_addr[13]				
ddr3_bank1_mem_addr[13]				
L:U75-T3,L:U78-T3	leg4	10	1.683	745.026:746.876
L:U78-T3,S:RN4-8	leg5	10	1.973	603.028:605.056
L:U66-T3,L:U70-T3	leg2	10	0.041	742.664:745.966
L:U70-T3,L:U75-T3	leg3	10	1.675	744.065:747.6
S:U72-AN18,L:U66-T3	leg1	10	1.772	2,367.471:2,371.752
ddr3_bank1_mem_addr[12]				
ddr3_bank1_mem_addr[12]				
L:U75-N7,L:U78-N7	leg4	10	1.549	745.026:746.876
L:U78-N7,S:RN5-3	leg5	10	1.896	603.028:605.056
L:U66-N7,L:U70-N7	leg2	10	0.722	742.664:745.966
L:U70-N7,L:U75-N7	leg3	10	2.356	744.065:747.6
S:U72-AP18,L:U66-N7	leg1	10	1.893	2,367.471:2,371.752
ddr3_bank1_mem_addr[11]				
ddr3_bank1_mem_addr[11]				

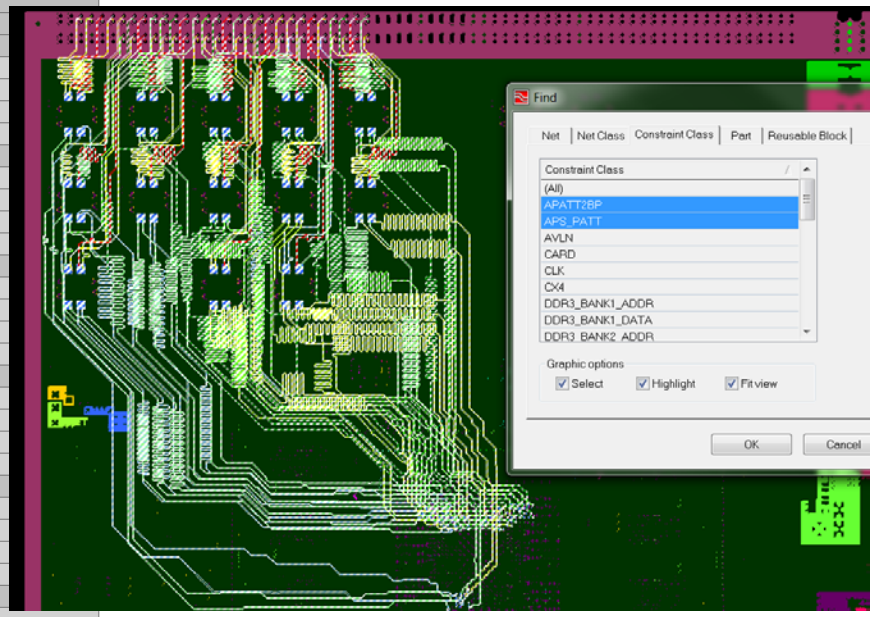


DDR3 Address and Data Routing



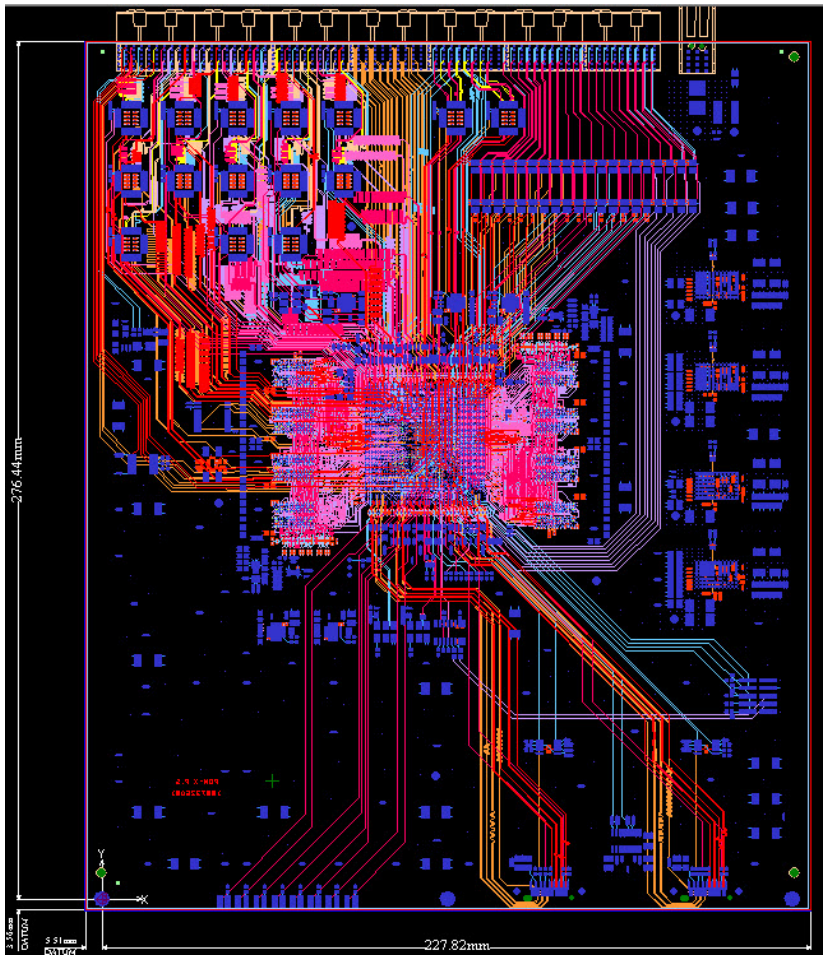
LVDS SERDES Rules and Routing

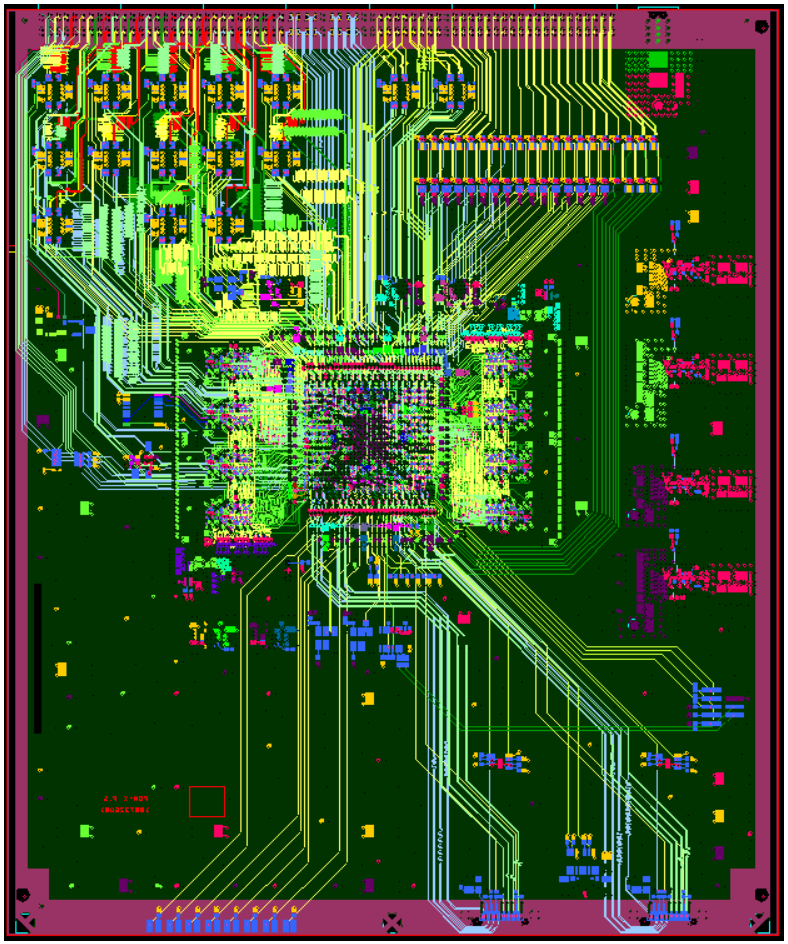
Constraint Class/Net	Formulas	
	Formula	*Violation*
⊞ (All)		
⊟ APS_PATT		
⊟ APS1_PATT_IN_CH0_n,APS1_PATT_IN_CH0_p	={\MCB_PATT_IN_CH1_n\}+/-5th	
⊟ APS1_PATT_IN_CH0_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.48th=7645.97:7655.97th*
APS1_PATT_IN_CH0_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.48th=7645.97:7655.97th*
⊟ APS1_PATT_IN_CH0_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
APS1_PATT_IN_CH0_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
⊟ APS1_PATT_IN_CH1_n,APS1_PATT_IN_CH1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	
⊟ APS1_PATT_IN_CH1_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.63th=7645.97:7655.97th*
APS1_PATT_IN_CH1_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.63th=7645.97:7655.97th*
⊟ APS1_PATT_IN_CH1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
APS1_PATT_IN_CH1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
⊟ APS1_PATT_IN_CH2_n,APS1_PATT_IN_CH2_p	={\MCB_PATT_IN_CH1_n\}+/-5th	
⊟ APS1_PATT_IN_CH2_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.23th=7645.97:7655.97th*
APS1_PATT_IN_CH2_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.23th=7645.97:7655.97th*
⊟ APS1_PATT_IN_CH2_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.54th=7645.97:7655.97th*
APS1_PATT_IN_CH2_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.54th=7645.97:7655.97th*
⊟ APS1_PATT_IN_CLK1_n,APS1_PATT_IN_CLK1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	
⊟ APS1_PATT_IN_CLK1_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.48th=7645.97:7655.97th*
APS1_PATT_IN_CLK1_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.48th=7645.97:7655.97th*
⊟ APS1_PATT_IN_CLK1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
APS1_PATT_IN_CLK1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
⊟ APS2_PATT_IN_CH0_n,APS2_PATT_IN_CH0_p	={\MCB_PATT_IN_CH1_n\}+/-5th	
⊟ APS2_PATT_IN_CH0_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.19th=7645.97:7655.97th*
APS2_PATT_IN_CH0_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.19th=7645.97:7655.97th*
⊟ APS2_PATT_IN_CH0_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.32th=7645.97:7655.97th*
APS2_PATT_IN_CH0_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.32th=7645.97:7655.97th*
⊟ APS2_PATT_IN_CH1_n,APS2_PATT_IN_CH1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	
⊟ APS2_PATT_IN_CH1_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
APS2_PATT_IN_CH1_n	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.26th=7645.97:7655.97th*
⊟ APS2_PATT_IN_CH1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.71th=7645.97:7655.97th*
APS2_PATT_IN_CH1_p	={\MCB_PATT_IN_CH1_n\}+/-5th	*7650.71th=7645.97:7655.97th*



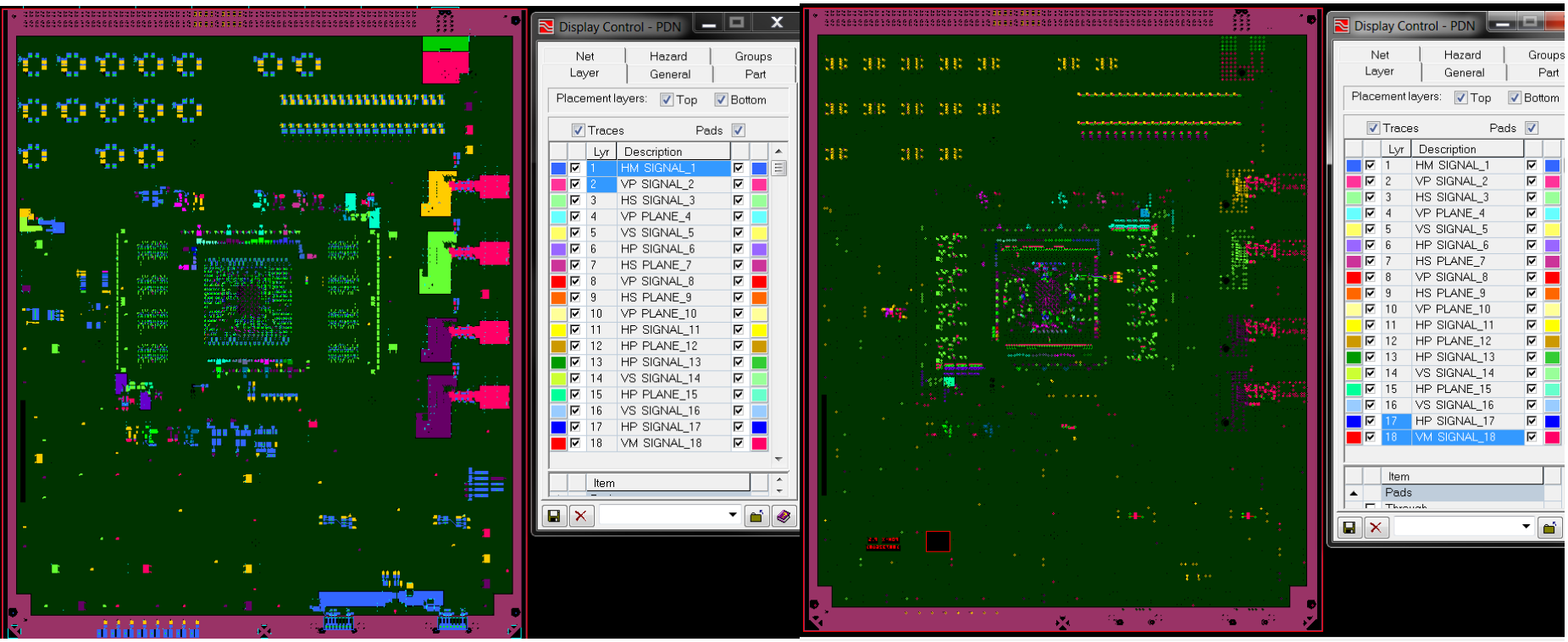
ROUTING BEST PRACTICES FOR DDR3 AND LVDS SERDES

DDR3, LVDS Design View

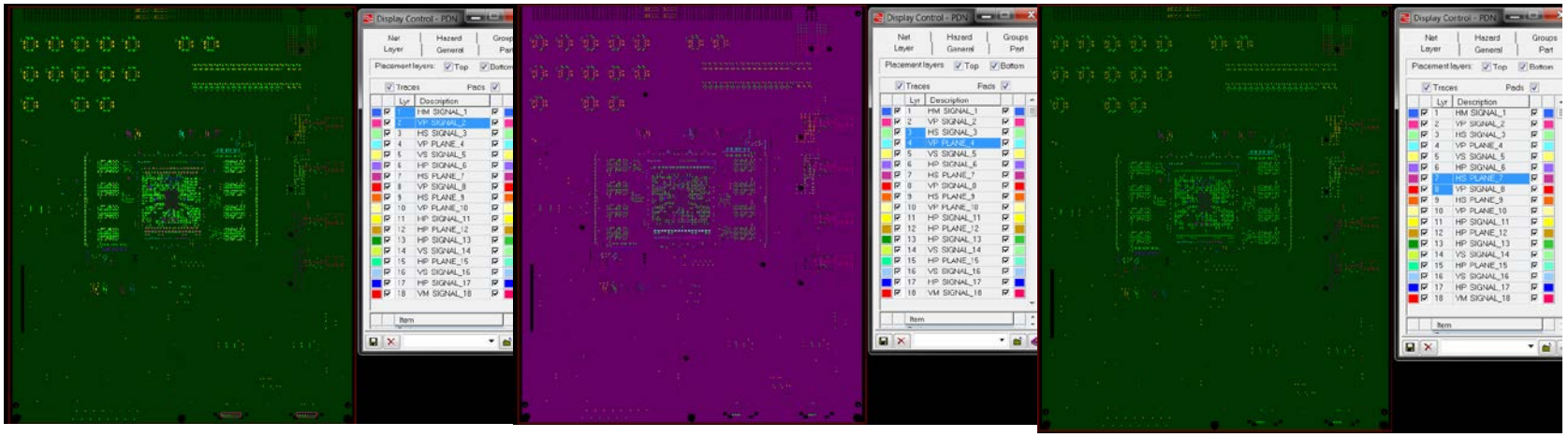


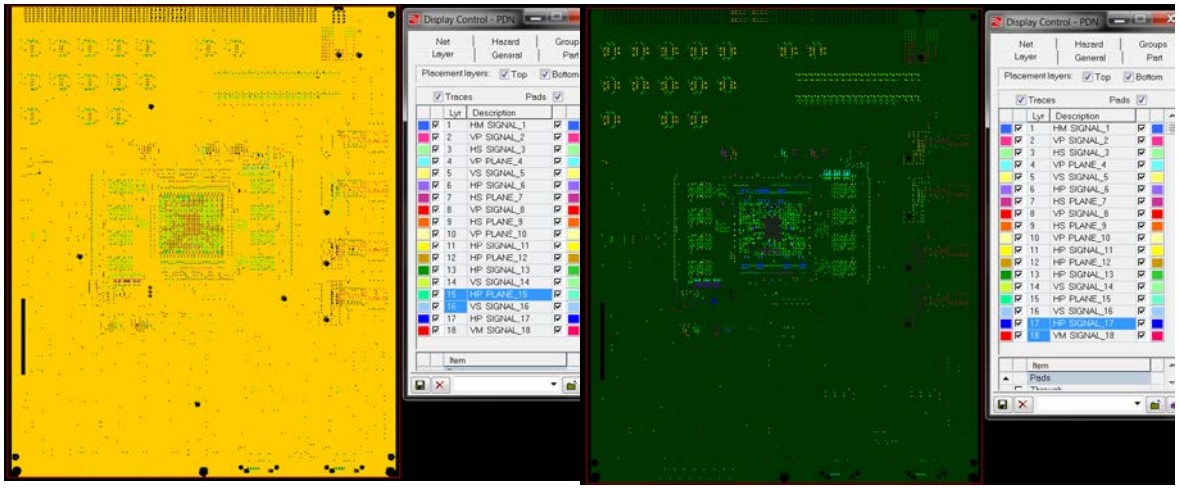


Top and Bottom Layers

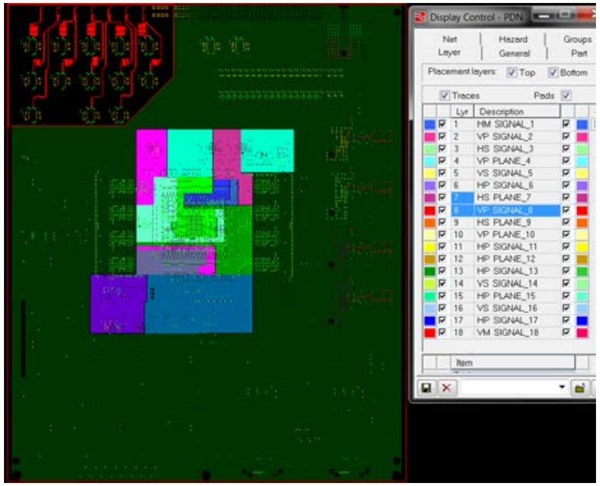


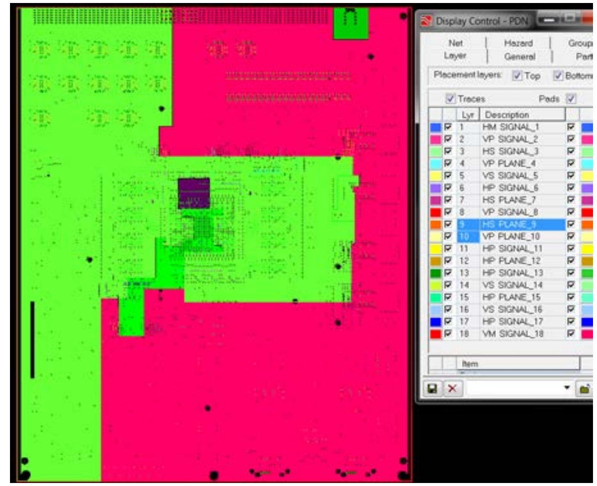
DDR3 Plane Layer 2,4,7,15,17 Examples

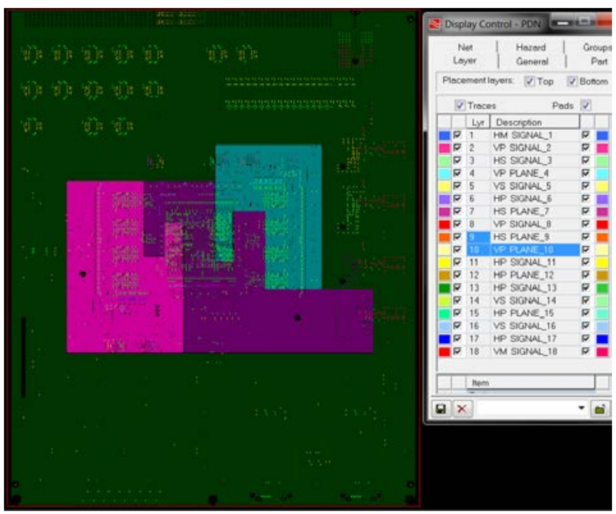


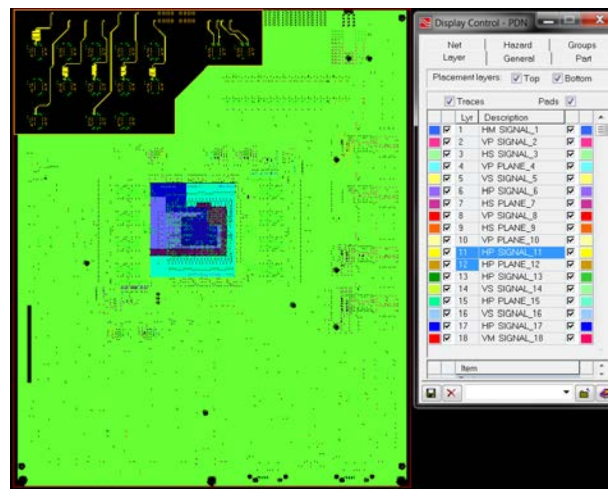


DDR3 Plane Layers 8,9,10, 11

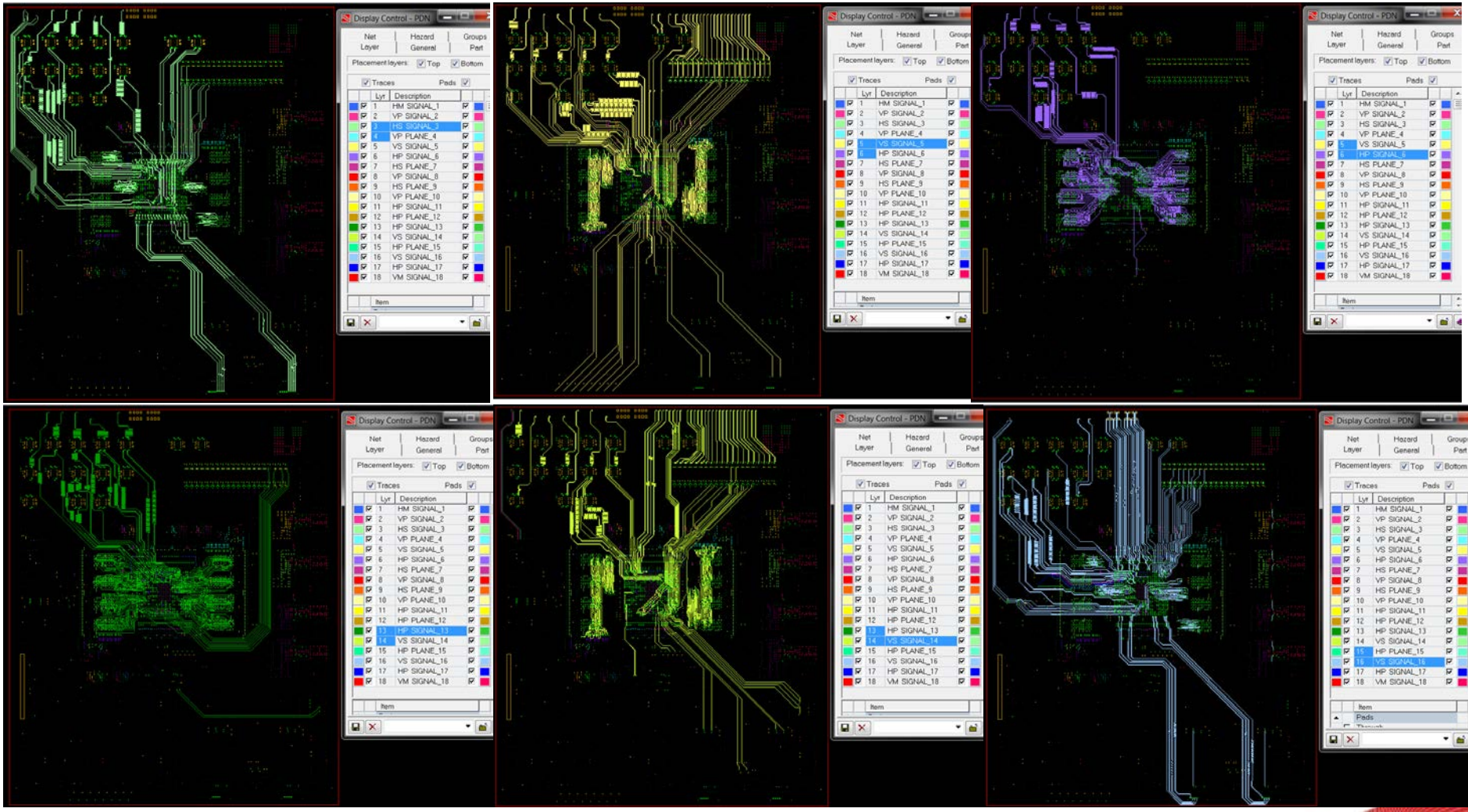




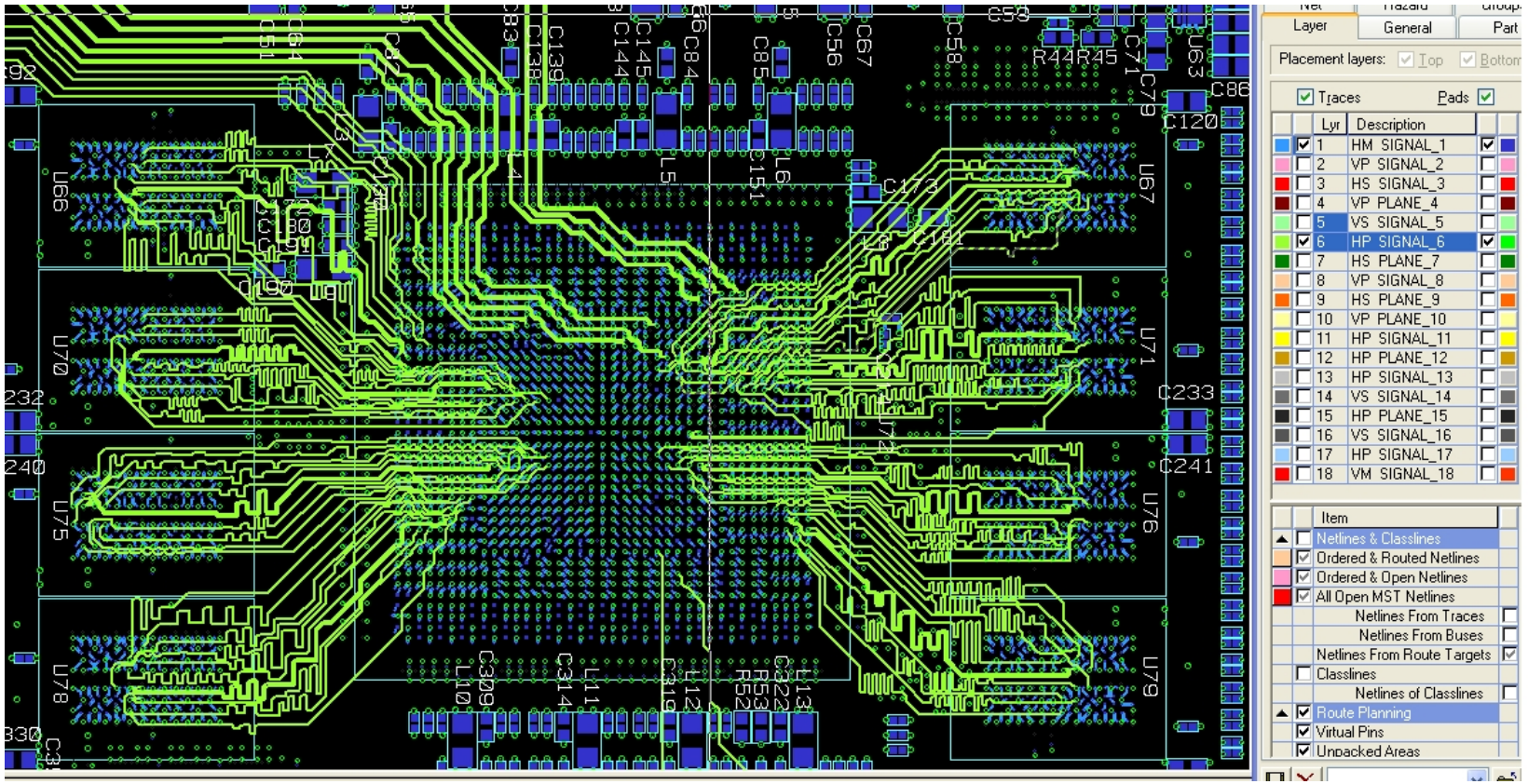




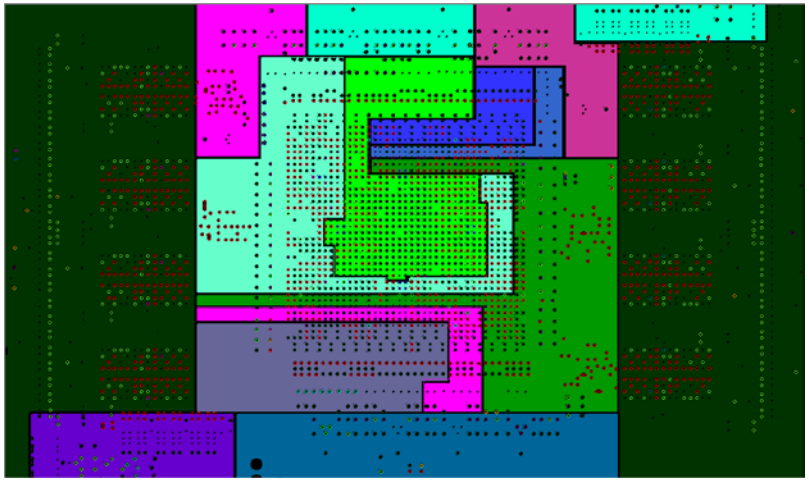
Signal Layers 3,5,6,13,14

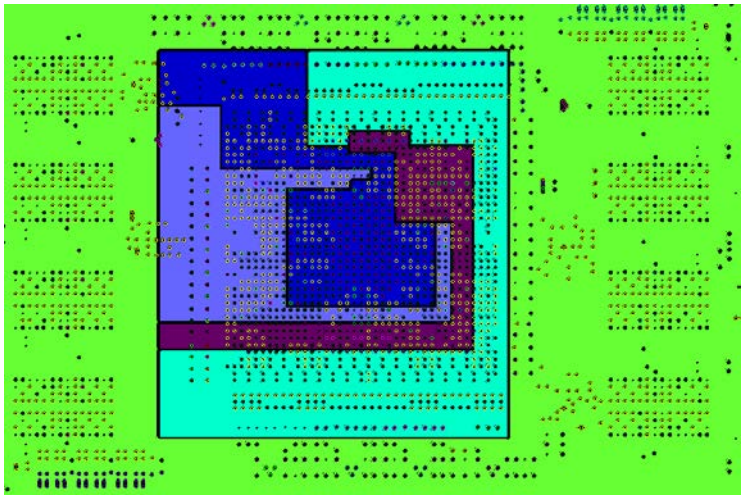


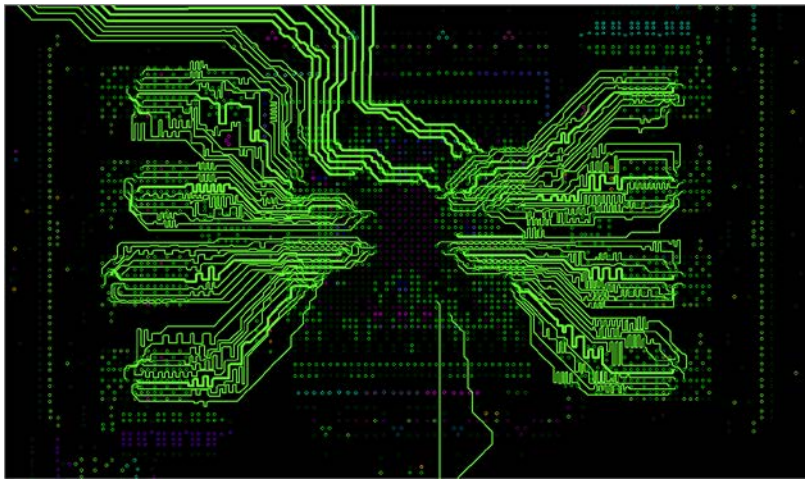
DDR3 Match Routing

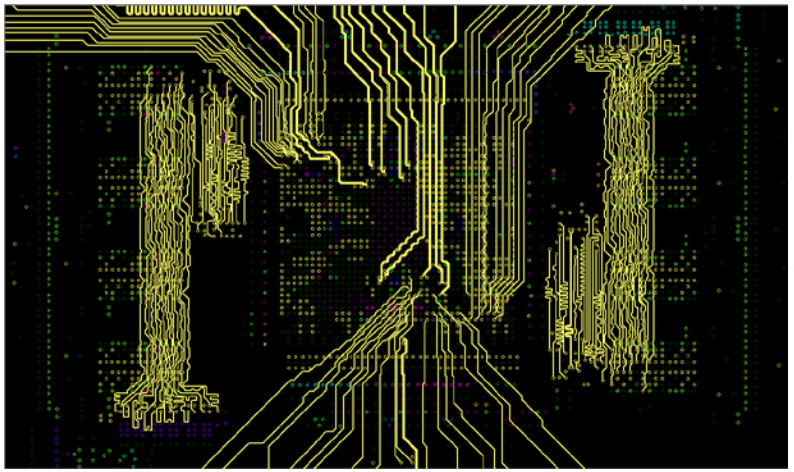


Close-ups Under BGA

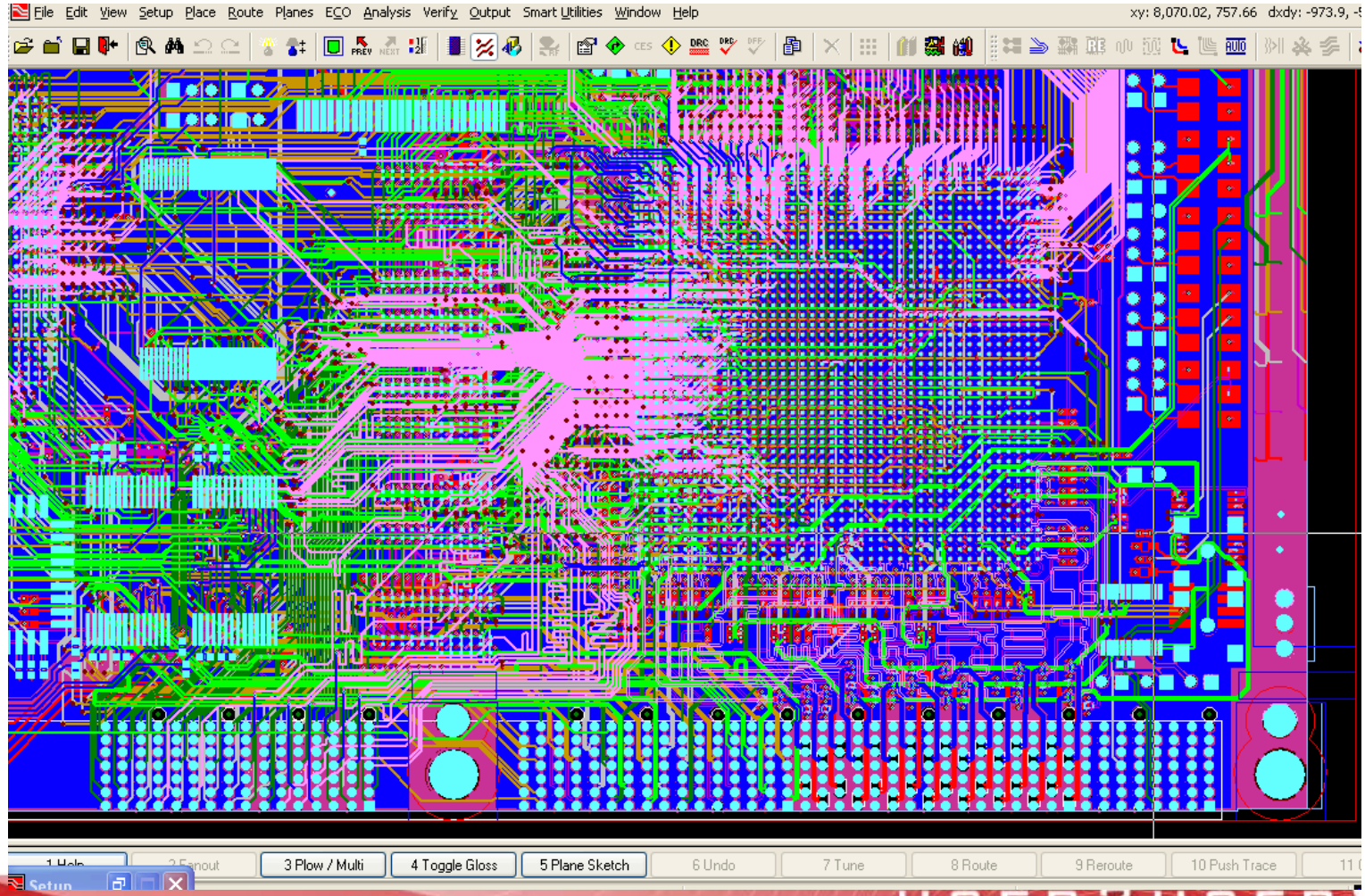




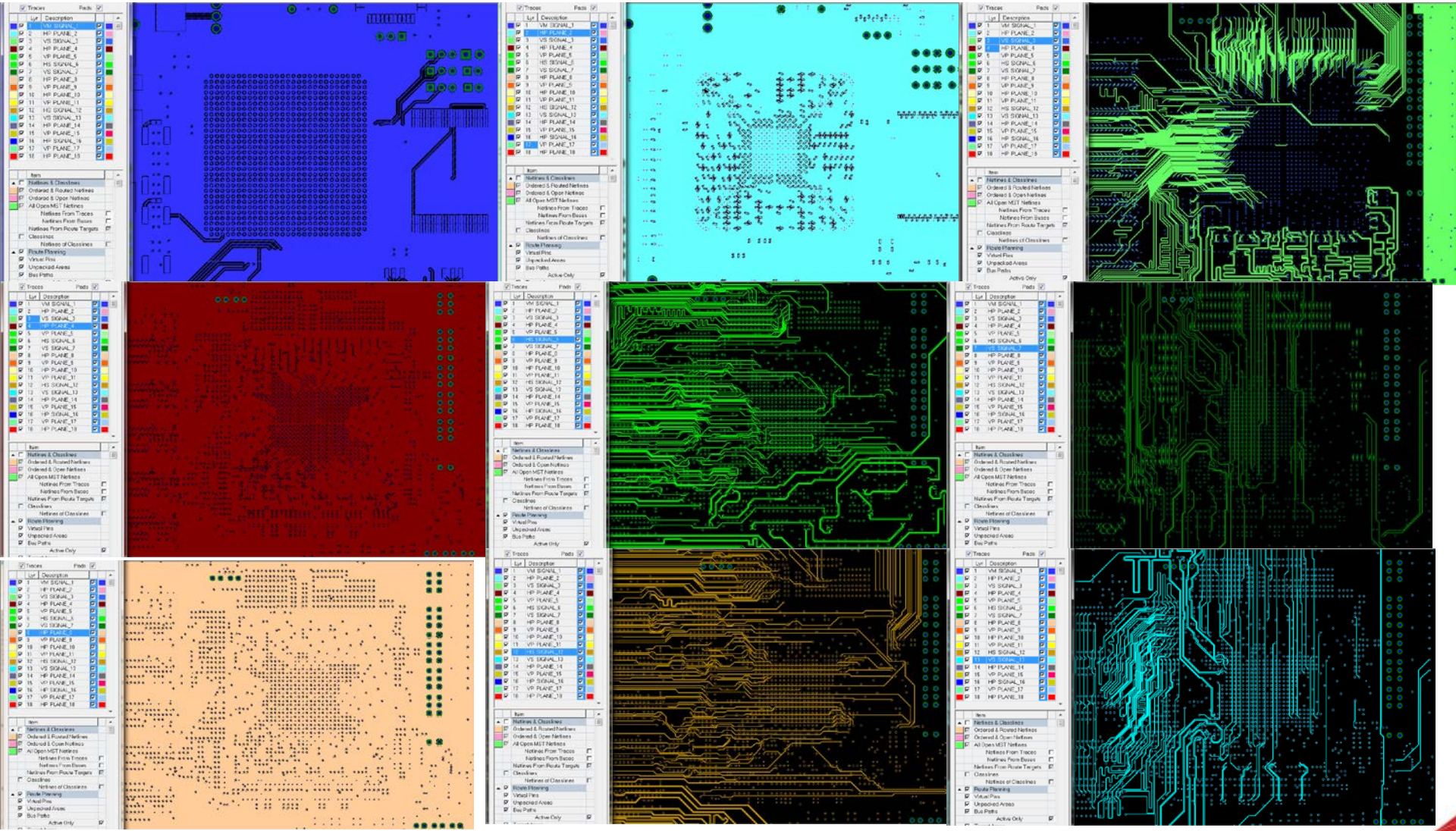




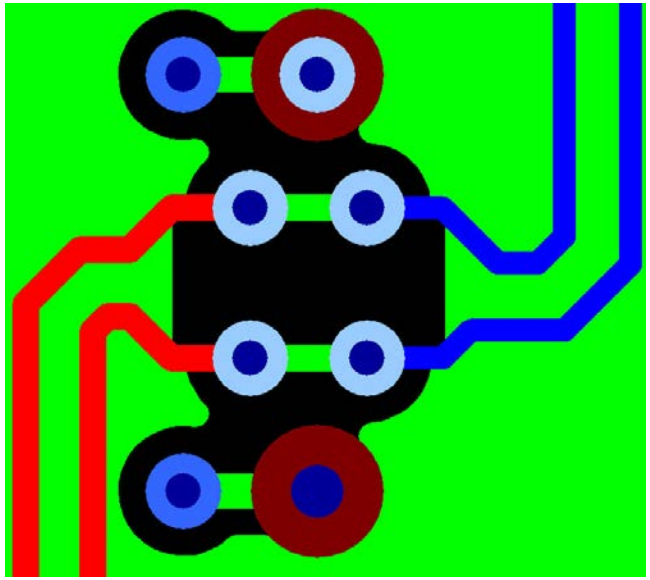
DDR3 and SERDES Example Design

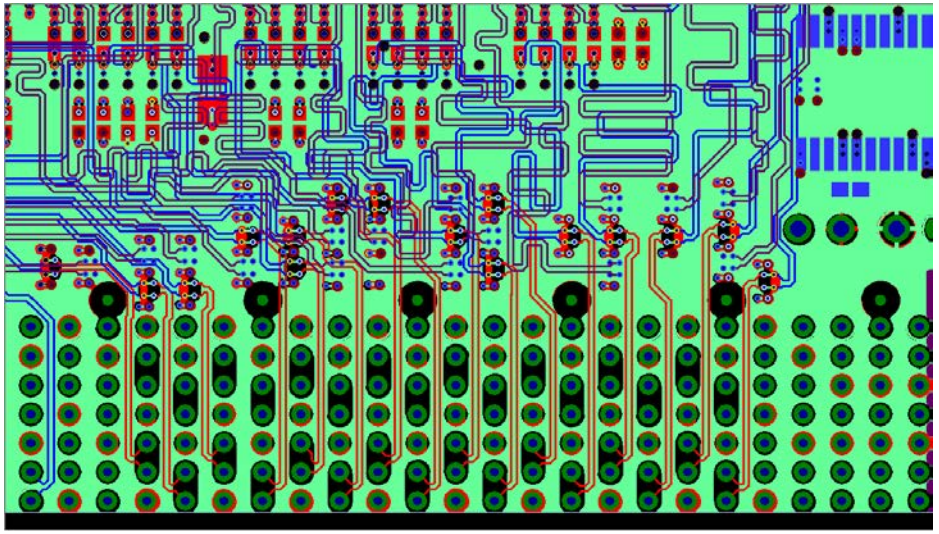


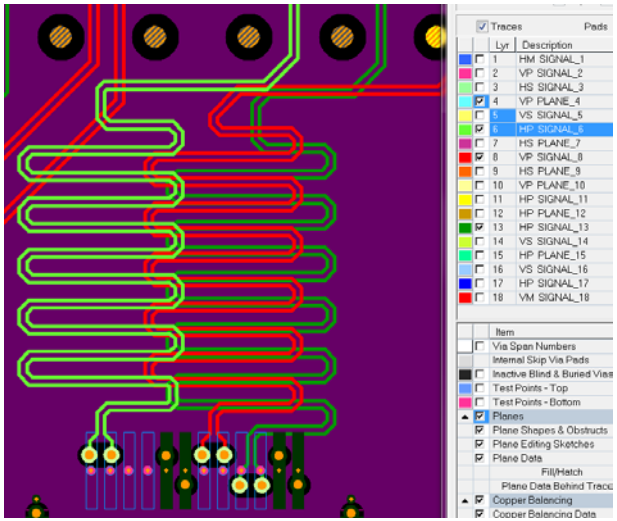
Breakout Examples

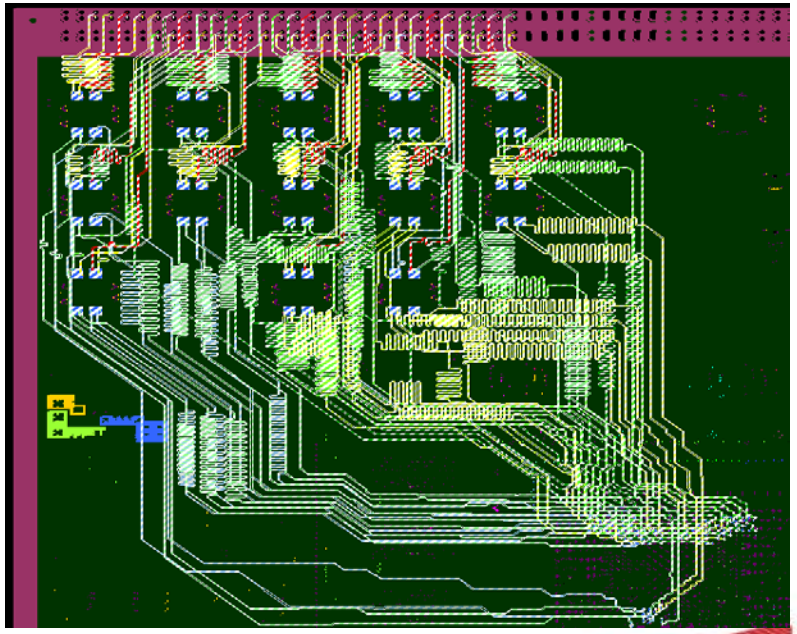


SERDES Routing Examples

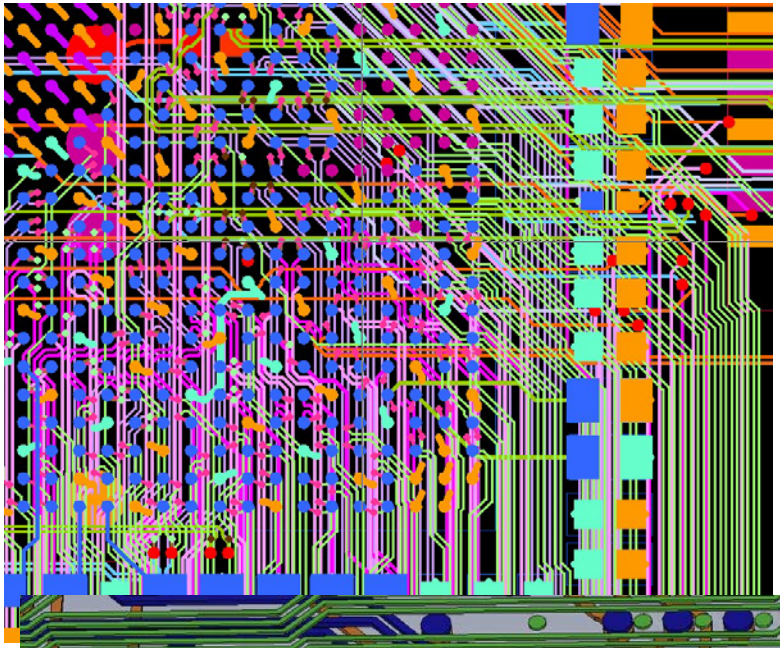


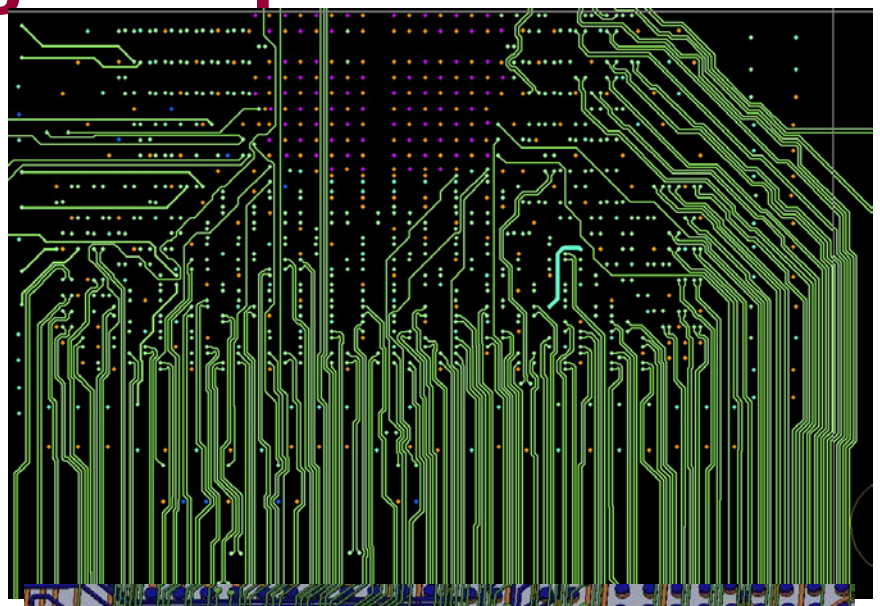


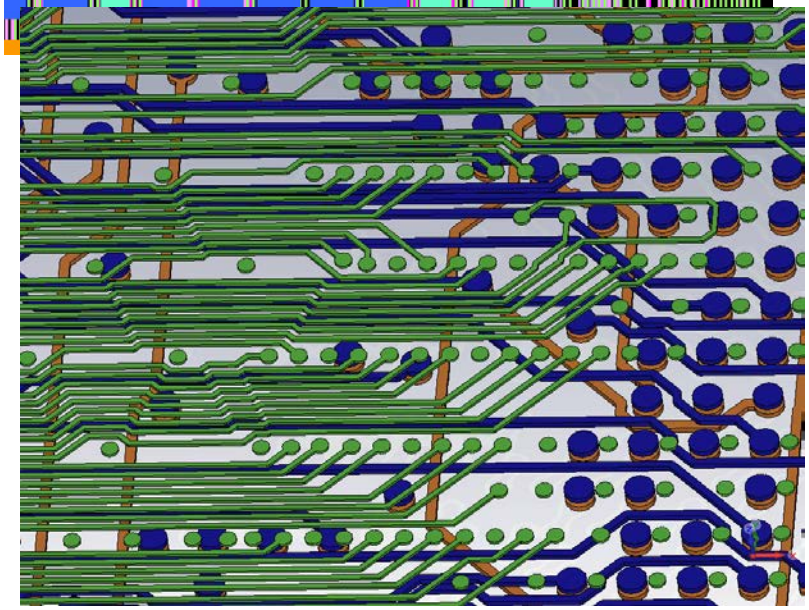


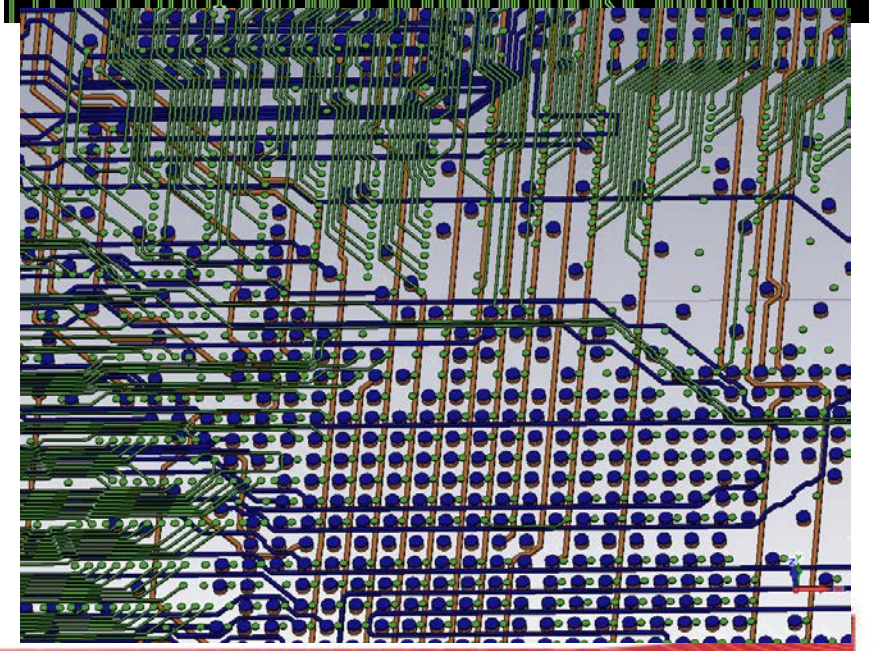


Handcrafted Routing Examples

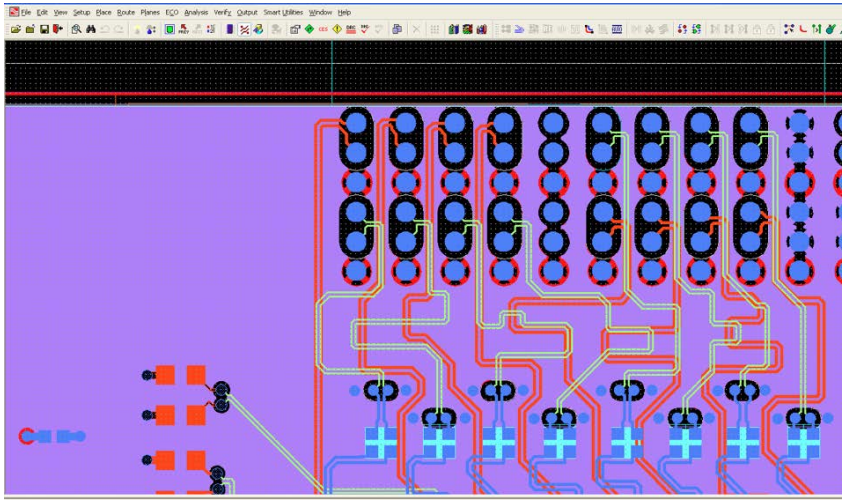


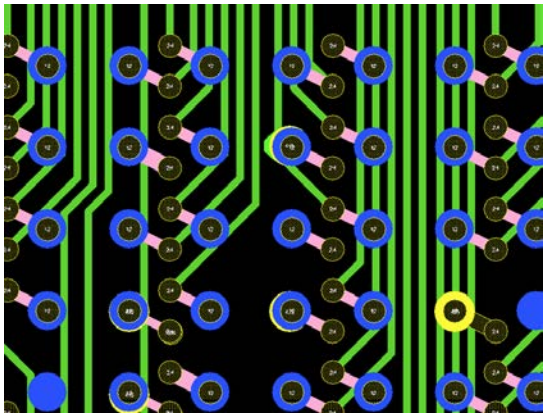


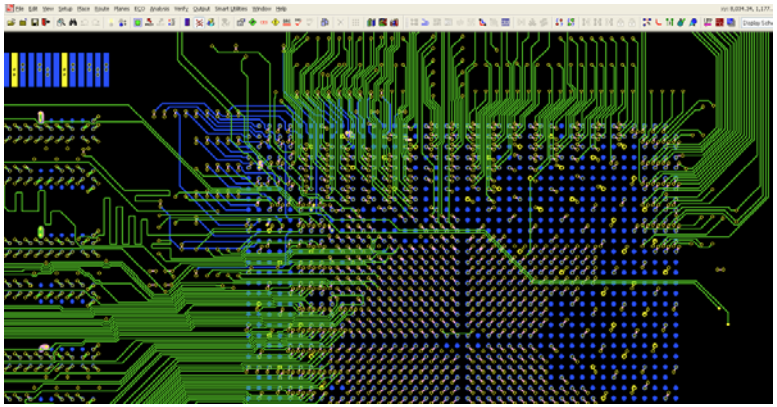


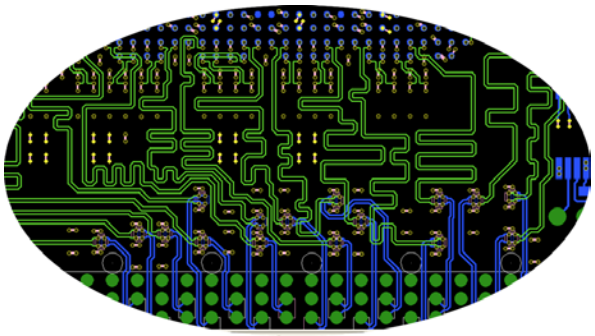


Routing Examples



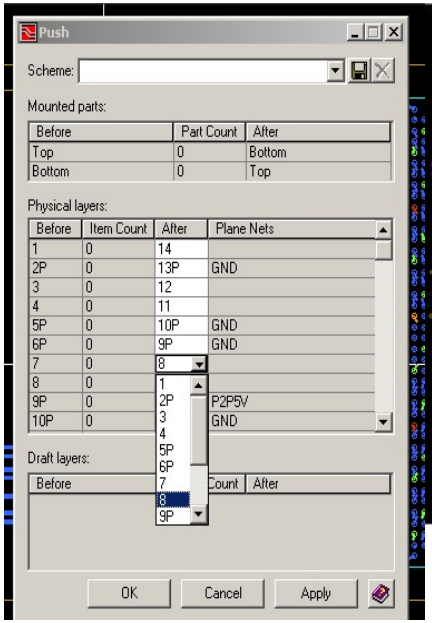


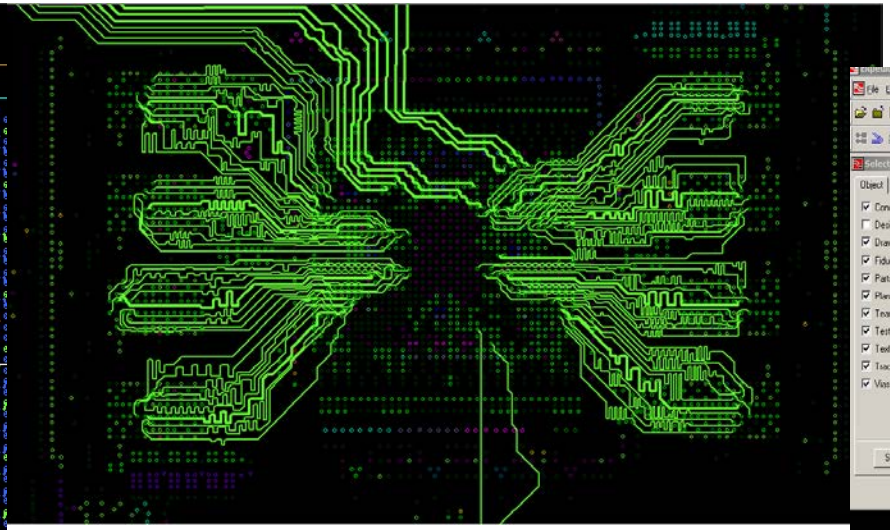


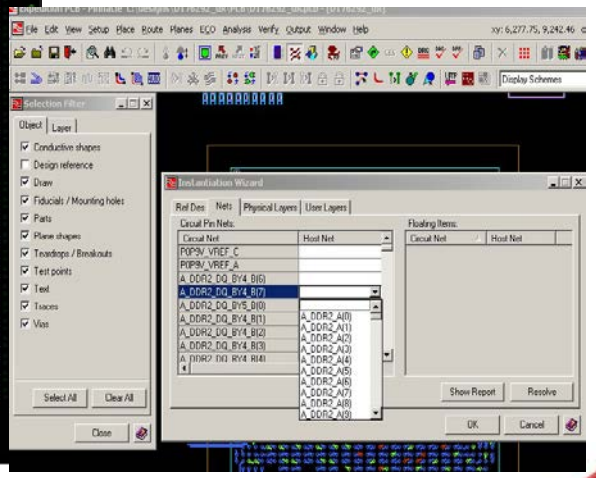


Copy Circuit Functionality

- Benefits
  - This can be very helpful and extreme time saver
  - Reuse and scalability functionality can be optimal for breakouts and general routing re-use
  - For our designs this is an essential capability coupled with ICX and CES this can and has expedited designs and changes







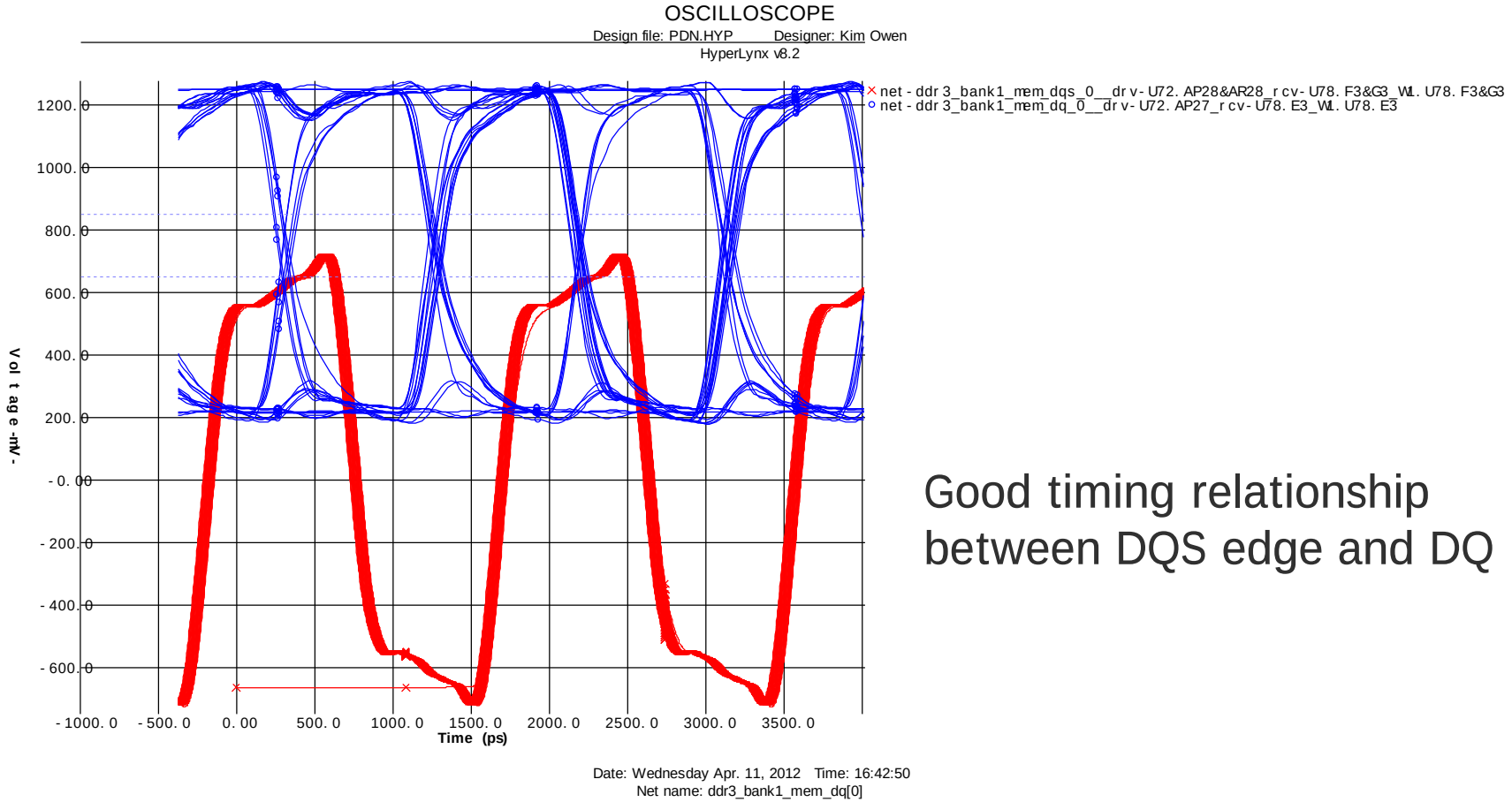
HyperLynx BoardSim / LineSim DDR3 Simulation Highlights

Kim Owen Signal Integrity Engineer

USER2USER

HyperLynx BoardSim DDR3 Simulation Results

Bank 1 DQS(0) (Red) vs DQ(0) PRBS Data Stream (Blue) Write @ SDRAM IV Die



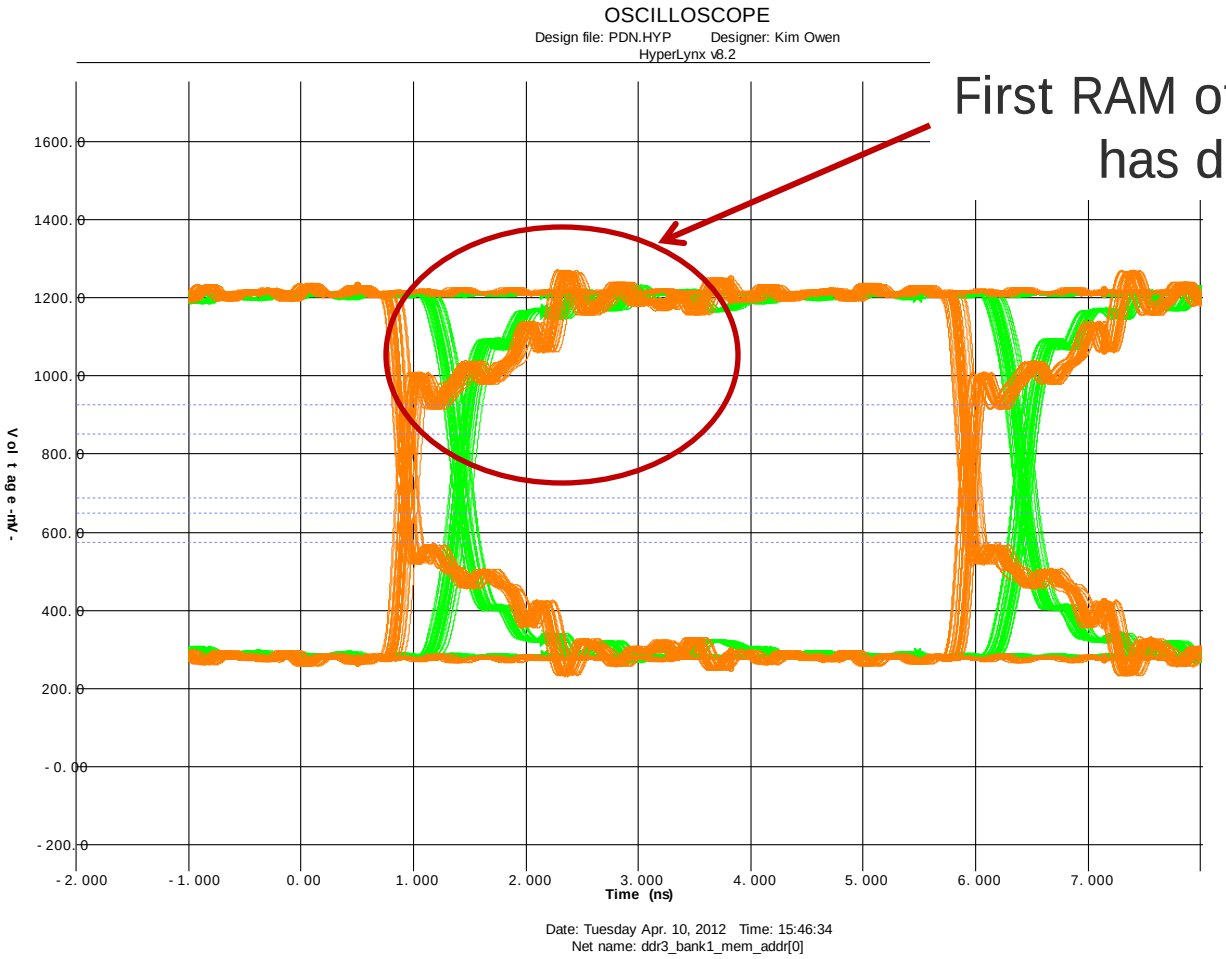
Good timing relationship between DQS edge and DQ Eye

Tx-Stratix sstl15c1_rio_r50

Rx-40 ohm Micron Driver, 60 ohm odt

DDR3 Bank 1 Memory Address @ First and Last SDRAM Die

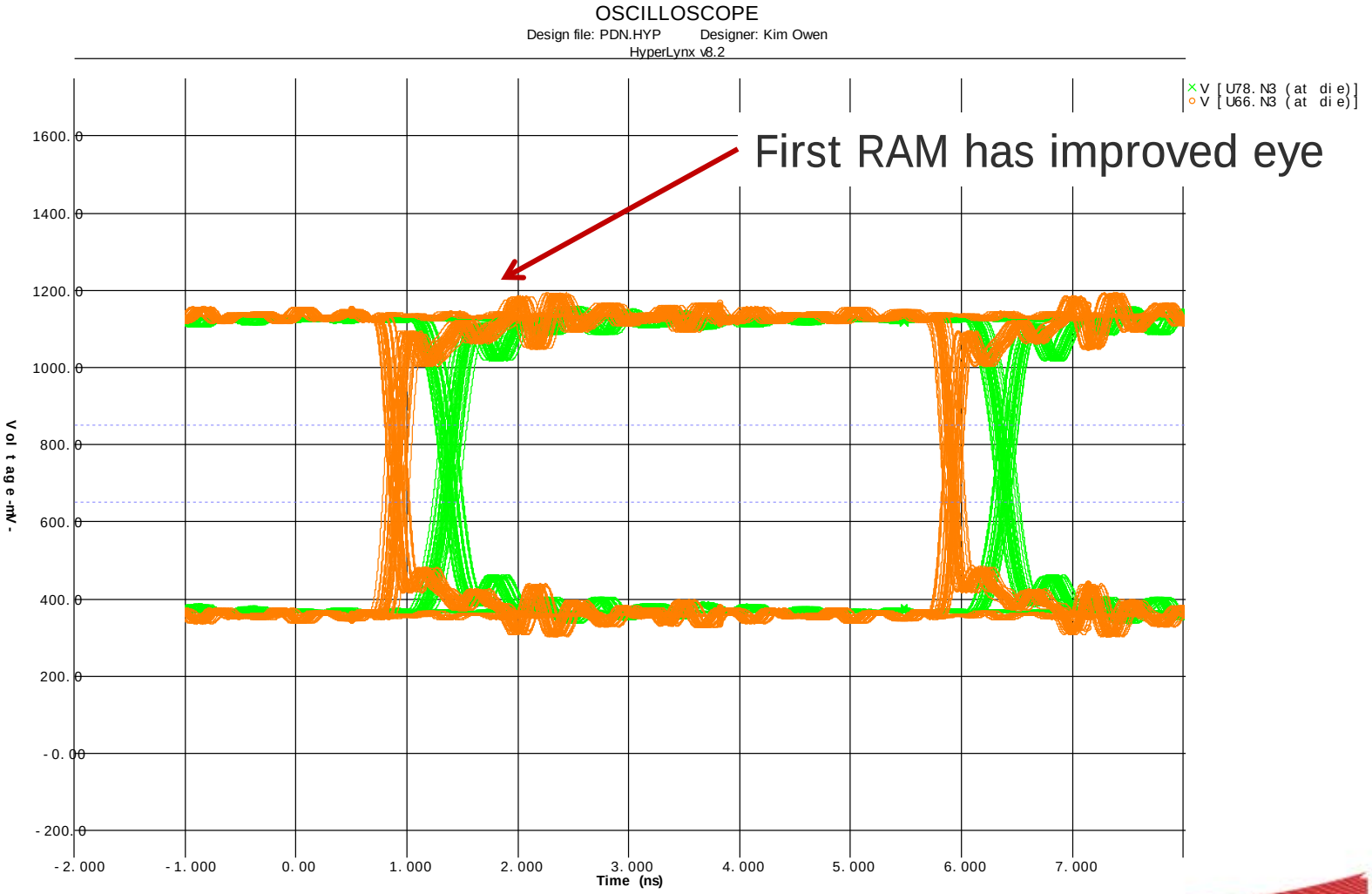
Using 60 ohm Default Termination



First RAM of “Fly By” topology has degraded eye

Bank 1 Memory Address @ First and Last SDRAM Die

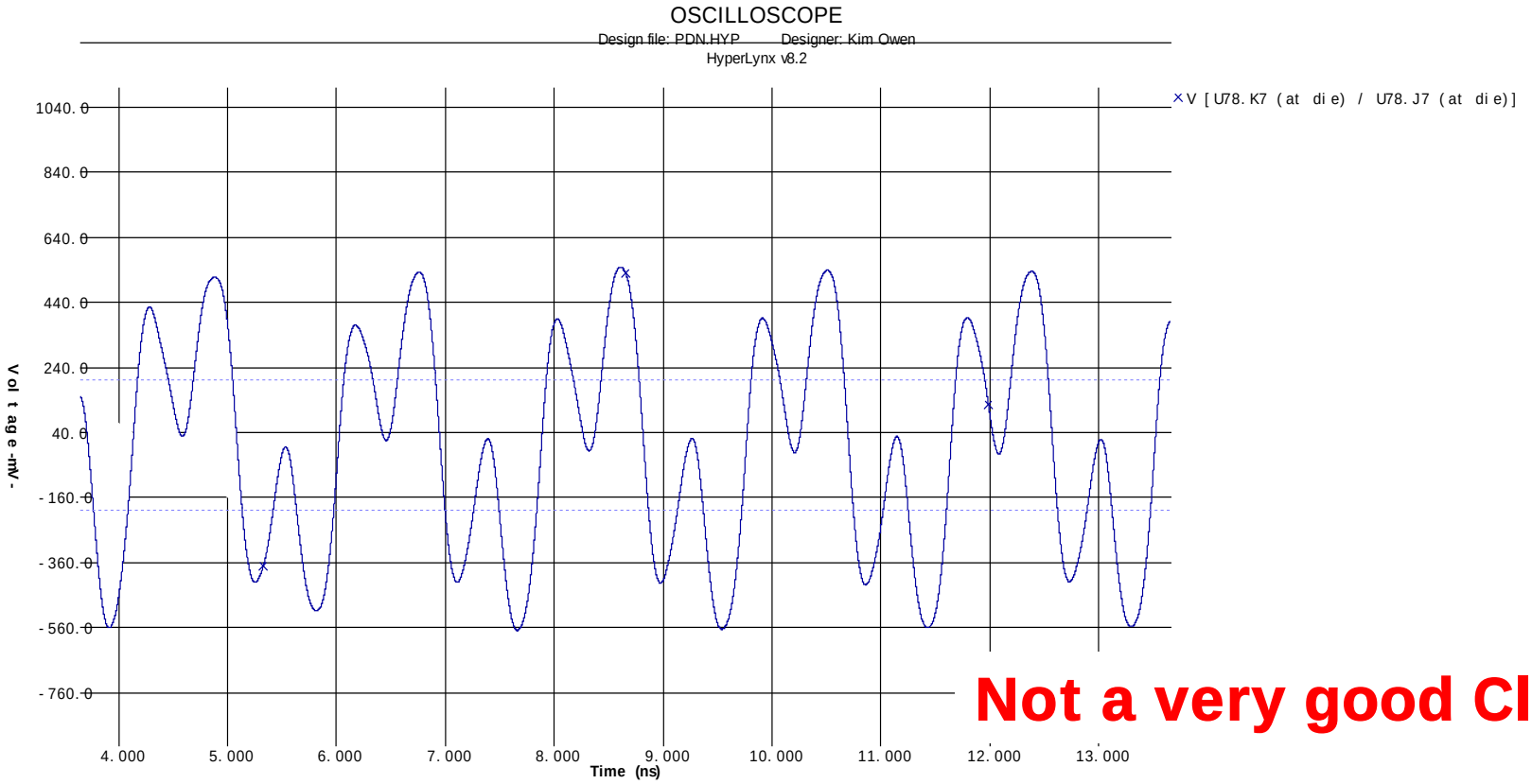
39 ohm Termination per Simulation



HyperLynx BoardSim Simulation Results

Initial Placement of Compensation Cap

DDR3_CLK(0) @ SDRAM Die



Not a very good Clock

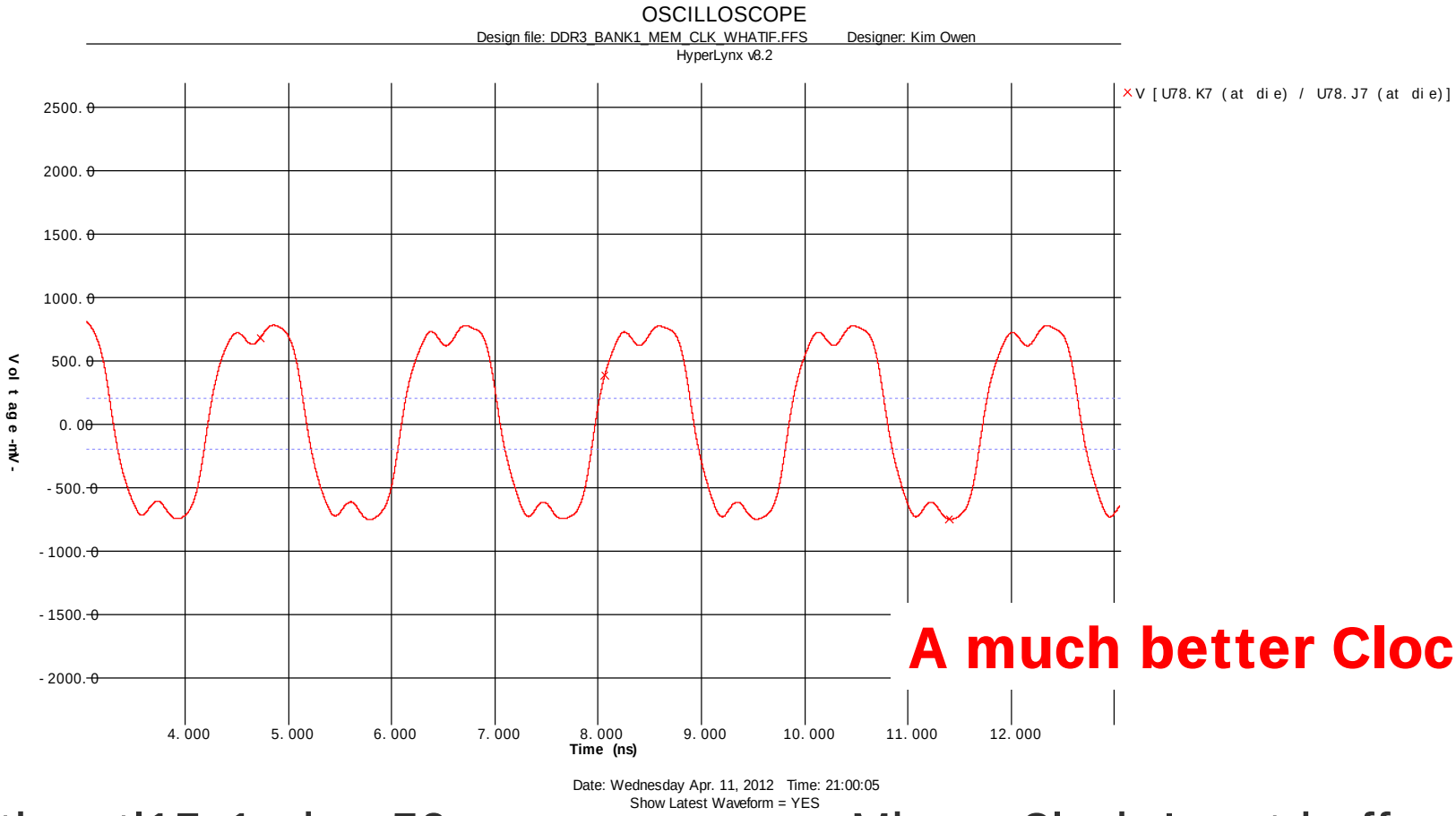
Stratix sstl15c1_rio_r50

Micron Clock Input buffer

HyperLynx BoardSim Simulation Results

Better Placement of Compensation Cap

DDR3_CLK(0) @ SDRAM Die



A much better Clock!

Stratix sstl15c1_rio_r50

Micron Clock Input buffer

Summary

- PCB Design process should be well defined and supported by internal and external management
- Fabrication shop and stackup must be included from the beginning
- EE need to define Interconnect delays and realistic noise parameters, not rule of thumb
- High pin count bga's need special consideration
- Pre and Post layout simulations need to be in design cycle for successful 1st time design to work
- This is not trivial work anymore and clock rise time are below 500ps, and getting faster. The PCB board is now a electrical challenge and adopted layout best practices and SI Simulations can help achieve high performance designs

Bio and HSDS Info

- With over 30 years of extensive experience in PCB design, training and analysis using state of the art tool sets, including Expedition, CES (Constraint Editor System), ICX, HyperLynx, Power SI. I currently own and operate High Speed Design Services, LLC.
- My background is completing complex PCB designs, and delivering training of PCB design tools and methodologies. I utilize my background to complete complex, HDI, high speed, RF, mixed technology designs, while combining signal integrity verification and analysis.
- I have worked at Apple, Northrop Grumman Radio Systems and also at Space Park delivering flight designs for Aerospace and Military applications. I previously worked for Apple, HP, Agilent, Nokia and Cisco as a working technical manger and built teams that could deliver complex HDI CPU designs. I have implemented a PCB Process and tool flow, which includes SI Tools for verification.
- **High Speed Design Services** *Provides concept to creation, with confidence*
- **Our Goal:** To provide world class PCB design and Signal Integrity services. Focusing on completing complex high-speed designs driven by performance, with electrical constraints derived from simulations and including years of experience of PCB and SI best practices.
- Experienced senior level PCB and SI Engineer staff is posed to problem solving and providing solutions others cannot achieve. We also have the industry knowledge required to capture the power of today's powerful CAD tools to optimize design and deliver high end products on or before schedule, through the use of automation techniques and re-use capabilities.
- At High Speed Design Services, LLC we provide more than just an extra pair of hands for PCB design and SI Analysis and verification. We offer leading edge PCB/SI design techniques, methodologies, and training. High Speed Design Services possess senior level skill sets to deliver top-notch world-class results in a timely manner. At High Speed Design Services we operate as an "A team" to assist your internal engineer resources or HSDS can be your engineering resource.


2012
U2U
USER2USER