

The logo for Mentor Graphics, featuring the word "Mentor" in a bold, red, sans-serif font above the word "Graphics" in a similar font, with a registered trademark symbol (®) to the right.The logo for xpedition, featuring a red stylized 'x' icon followed by the word "xpedition" in a black, sans-serif font, with a trademark symbol (™) to the right.

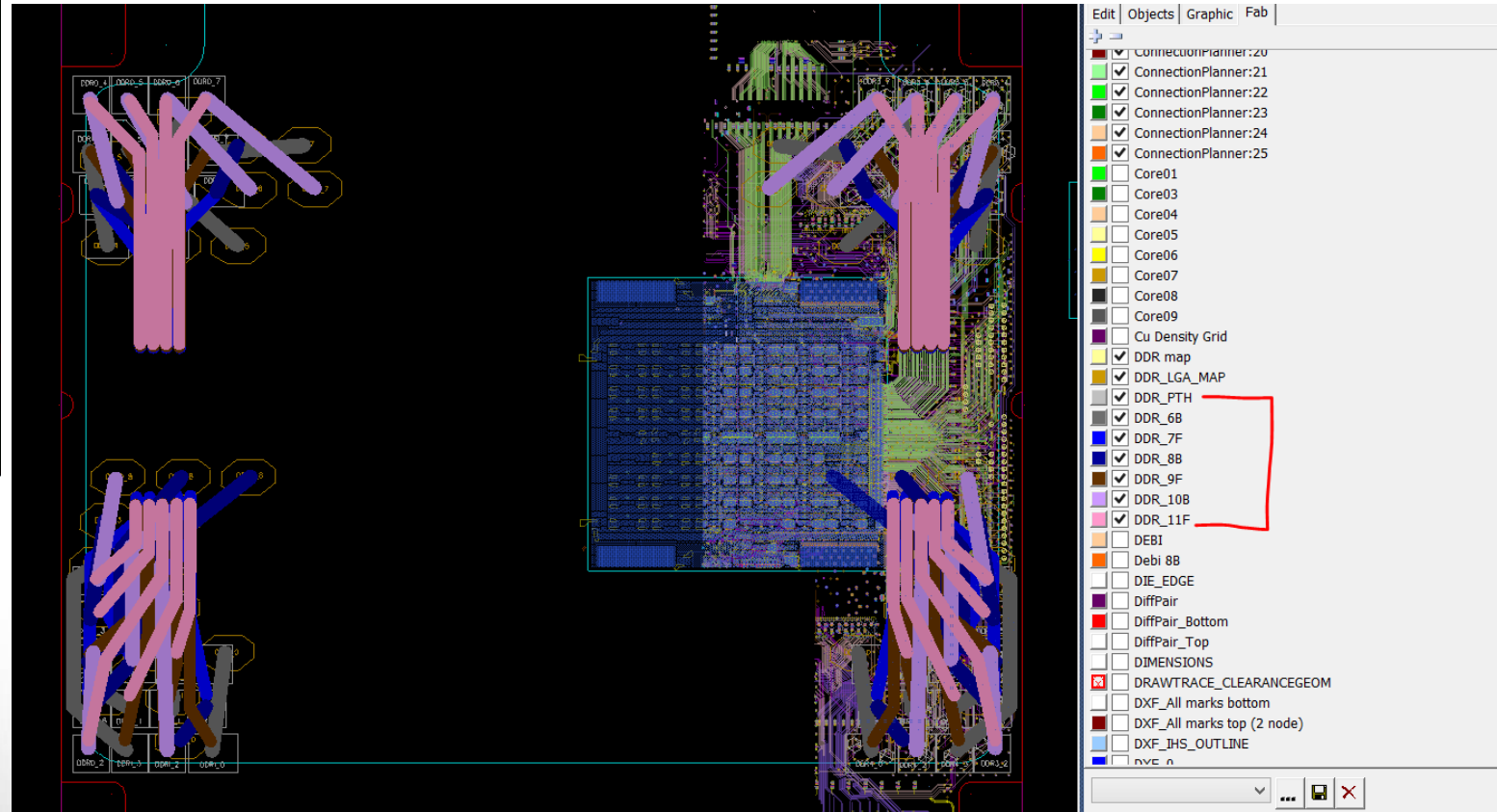
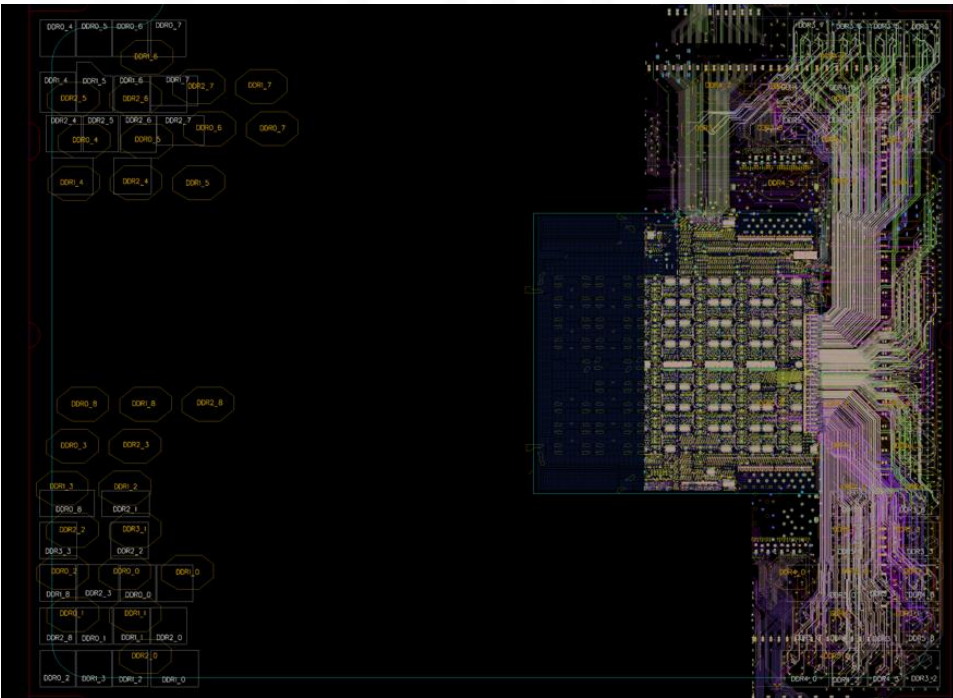
- Package Integrator & Sketch Router/Planner Board Centric Process

■ John Medina  
09/23/2016

# What are the problems with planning of DDR routing

- Planning of DDR routing from Package die to PCB Dimms is very time consuming.
- Many iterations may happen during pathfinding and beyond
- Routing of nets can be a challenge if placement of via farms and LGA pins is not optimal
- Routing proximity of byte lanes to each other is not preferred on back side (after core layers) routing (HLDRC)

# Current manual method of DDR Route planning using user layer graphics to determine layer and byte lane planning used on KNH



# Unravel of via farms north before and after unravel

Via Farms combined or individual view?

The screenshot displays a PCB design tool interface. The main window shows a floorplan with a dense grid of vias and signal traces. A component labeled 'U2' is visible. A red box highlights a specific area of the via farm. The 'Unravel nets' dialog box is open, showing performance settings and options for concurrent unraveling. A console window at the bottom displays the results of the unraveling process.

Console output:

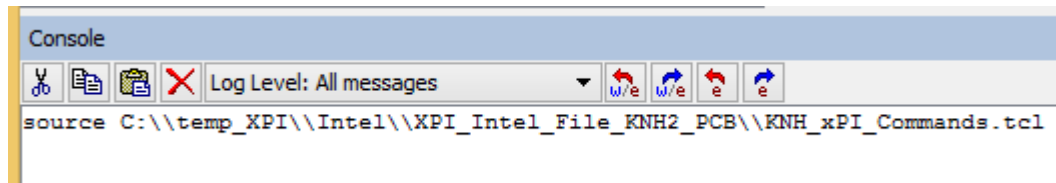
```
Unraveling finished successfully.  
Total unravel elapsed time 00:00:01  
Total nets length is 120.83 in, was 121.19 in. Total nets length reduction achieved: 0.30%.  
Total count of crossings is 157, was 3267. Total crossings reduction achieved: 95.19%.
```

# Process for creating and configuring the PCB and Package

- Generate symbol pin list from native CAD tool to create parts
  - Simplifies traditional board translation
- Import text file into XPI when adding part(s)
  - One spreadsheet with more data is good
  - Automation is useful when creating parts
  - Create via farm with multiple ILNs
  - Set swap regions
  - Setting pin groups and colors is helpful

# TCL Script for adding PCB database using symbol pin list generated from PCB Layout

This set of files need to be unzipped in the directory C:\temp\_XPI\Intel. Then you can open XPI and run the following command in the console (close the initial create/open dialog form).  
source C:\\temp\_XPI\\Intel\\XPI\_Intel\_File\_KNH2\_PCB\\KNH\_xPI\_Commands.tcl



This will open a new KNH project. **Adding the Design PCB.** Read the KNH BGA and Conn288 devices and signal files. It finishes by doing Sync FE Design. At this point the script is done. These are the steps I did to complete the design and sync the layout.

- Select Device KNH\_BGA and run Edit >Buses > Autodetect Buses
- Select Device CONN288\_J1 and run Edit >Buses > Autodetect BusesSelect Design PCB and run Export > Connectivity to Layout (this will invoke the layout tool)
  - 1.Select the PCB.prj file. Form Next
  - 2.Design Technology: Package, Template: Package Design 8 Layer Template. Form Finish then Close
  - 3.Xpedition will Invoke
  - 4.Copy the C:\\temp\_XPI\\Intel\\XPI\_Intel\_File\_KNH2\_PCB \\place.txt file into directory C:\\temp\_XPI\\Intel\\KNH\\PCB\\PCB
  - 5.Run FA
  - 6.Run keyin "pr -file=place.txt"
  - 7.Grow the Board Outline and Route Border
  - 8.Save design and exit Xpedition
- In xPI run Import > Layout



# Results of TCL Script to build PCB Database

**Project - [KNH\_XPI\_VX2\_DDR\_0516\_PM] [BOARD\_ALLEGRO]**

Instance	Definition	Ref Des	Pattern
board			
KNH2_M...			
ViaFar...	ViaFarm_DDR...	DDR3_0	Contact
VF_KN...	VF_KNH_DDR_2	DDR3_2	Contact
TEST2	TEST		Contact
TEST1	TEST		Contact
TEST	TEST		Contact
P76x7...	P76x78_3647L...	A1	Contact
KNIGH...	KNIGHTS_HILL...	U2	Contact
DDR_N...	DDR_NORTH	DDR_NORTH	Contact
DDR3_3	DDR3_3	DDR3_3	Contact
DDR3_1	DDR3_1	DDR3_1	Contact
BOARD_...			
LGA_U...	LGA_U110	U110	Contact
CONN	CONN	J1	Contact

```
insole
Log Level: All messages
rangefilter -connList 1 {[M-P]} {} true
renview KNH2_MASTER_0503/P76x78_3647L_01A
view for document KNH2_MASTER_0503/P76x78_3647L_01A has been successfully opened.
_in_group_import Z:\Mentor_Designs\XPI\KNH_XPI_VX2_DDR_0505\PINGROUP.xml
:_j_set_current_board BOARD_ALLEGRO
renview BOARD_ALLEGRO
```

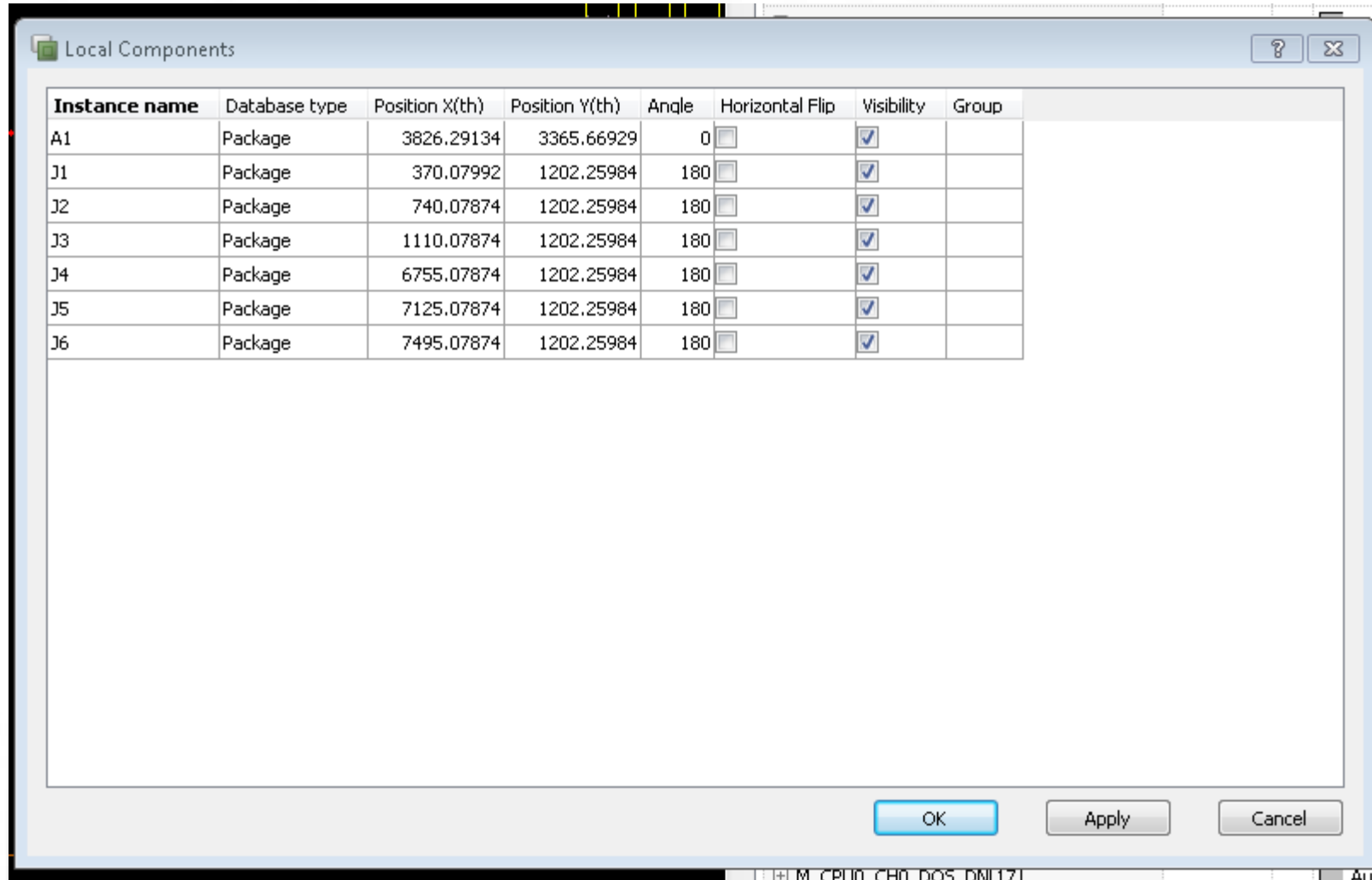
Instance	Definition	Ref Des	Pattern	Interface	Layer Site
board					
KNH2_M...					
ViaFar...	ViaFarm_DDR...	DDR3_0	Contact		
VF_KN...	VF_KNH_DDR_2	DDR3_2	Contact		
TEST2	TEST		Contact		
TEST1	TEST		Contact		
TEST	TEST		Contact		
P76x7...	P76x78_3647L...	A1	Contact		
KNIGH...	KNIGHTS_HILL...	U2	Contact		
DDR_N...	DDR_NORTH	DDR_NORTH	Contact		
DDR3_3	DDR3_3	DDR3_3	Contact		
DDR3_1	DDR3_1	DDR3_1	Contact		
BOARD_...					
LGA_U...	LGA_U110	U110	Contact		
CONN	CONN	J1	Contact		

Component	Pin Number	Color	Test Group
U110	1	Automatic	
U110	2	Automatic	
U110	3	Automatic	
U110	4	Automatic	
U110	5	Automatic	
U110	6	Automatic	
U110	7	Automatic	
U110	8	Automatic	
U110	9	Automatic	
U110	10	Automatic	
U110	11	Automatic	
U110	12	Automatic	
U110	13	Automatic	
U110	14	Automatic	
U110	15	Automatic	
U110	16	Automatic	
U110	17	Automatic	
U110	18	Automatic	
U110	19	Automatic	
U110	20	Automatic	
U110	21	Automatic	
U110	22	Automatic	
U110	23	Automatic	
U110	24	Automatic	
U110	25	Automatic	
U110	26	Automatic	
U110	27	Automatic	
U110	28	Automatic	
U110	29	Automatic	
U110	30	Automatic	
U110	31	Automatic	
U110	32	Automatic	
U110	33	Automatic	
U110	34	Automatic	
U110	35	Automatic	
U110	36	Automatic	
U110	37	Automatic	
U110	38	Automatic	
U110	39	Automatic	
U110	40	Automatic	
U110	41	Automatic	
U110	42	Automatic	
U110	43	Automatic	
U110	44	Automatic	
U110	45	Automatic	
U110	46	Automatic	
U110	47	Automatic	
U110	48	Automatic	
U110	49	Automatic	
U110	50	Automatic	
U110	51	Automatic	
U110	52	Automatic	
U110	53	Automatic	
U110	54	Automatic	
U110	55	Automatic	
U110	56	Automatic	
U110	57	Automatic	
U110	58	Automatic	
U110	59	Automatic	
U110	60	Automatic	
U110	61	Automatic	
U110	62	Automatic	
U110	63	Automatic	
U110	64	Automatic	
U110	65	Automatic	
U110	66	Automatic	
U110	67	Automatic	
U110	68	Automatic	
U110	69	Automatic	
U110	70	Automatic	
U110	71	Automatic	
U110	72	Automatic	
U110	73	Automatic	
U110	74	Automatic	
U110	75	Automatic	
U110	76	Automatic	
U110	77	Automatic	
U110	78	Automatic	
U110	79	Automatic	
U110	80	Automatic	
U110	81	Automatic	
U110	82	Automatic	
U110	83	Automatic	
U110	84	Automatic	
U110	85	Automatic	
U110	86	Automatic	
U110	87	Automatic	
U110	88	Automatic	
U110	89	Automatic	
U110	90	Automatic	
U110	91	Automatic	
U110	92	Automatic	
U110	93	Automatic	
U110	94	Automatic	
U110	95	Automatic	
U110	96	Automatic	
U110	97	Automatic	
U110	98	Automatic	
U110	99	Automatic	
U110	100	Automatic	

```
Console
Log Level: All messages
* definition BOARD_ALLEGRO/conn/1 successfully loaded
* definition BOARD_ALLEGRO/conn/2 successfully loaded
* definition BOARD_ALLEGRO/conn/3 successfully loaded
* definition BOARD_ALLEGRO/conn/4 successfully loaded
* definition BOARD_ALLEGRO/conn/5 successfully loaded
* definition BOARD_ALLEGRO/conn/6 successfully loaded
* definition BOARD_ALLEGRO/conn/7 successfully loaded
* definition BOARD_ALLEGRO/conn/8 successfully loaded
* definition BOARD_ALLEGRO/conn/9 successfully loaded
* definition BOARD_ALLEGRO/conn/10 successfully loaded
* definition BOARD_ALLEGRO/conn/11 successfully loaded
* definition BOARD_ALLEGRO/conn/12 successfully loaded
* definition BOARD_ALLEGRO/conn/13 successfully loaded
* definition BOARD_ALLEGRO/conn/14 successfully loaded
* definition BOARD_ALLEGRO/conn/15 successfully loaded
* definition BOARD_ALLEGRO/conn/16 successfully loaded
* definition BOARD_ALLEGRO/conn/17 successfully loaded
* definition BOARD_ALLEGRO/conn/18 successfully loaded
* definition BOARD_ALLEGRO/conn/19 successfully loaded
* definition BOARD_ALLEGRO/conn/20 successfully loaded
* definition BOARD_ALLEGRO/conn/21 successfully loaded
* definition BOARD_ALLEGRO/conn/22 successfully loaded
* definition BOARD_ALLEGRO/conn/23 successfully loaded
* definition BOARD_ALLEGRO/conn/24 successfully loaded
* definition BOARD_ALLEGRO/conn/25 successfully loaded
* definition BOARD_ALLEGRO/conn/26 successfully loaded
* definition BOARD_ALLEGRO/conn/27 successfully loaded
* definition BOARD_ALLEGRO/conn/28 successfully loaded
* definition BOARD_ALLEGRO/conn/29 successfully loaded
* definition BOARD_ALLEGRO/conn/30 successfully loaded
* definition BOARD_ALLEGRO/conn/31 successfully loaded
* definition BOARD_ALLEGRO/conn/32 successfully loaded
* definition BOARD_ALLEGRO/conn/33 successfully loaded
* definition BOARD_ALLEGRO/conn/34 successfully loaded
* definition BOARD_ALLEGRO/conn/35 successfully loaded
* definition BOARD_ALLEGRO/conn/36 successfully loaded
* definition BOARD_ALLEGRO/conn/37 successfully loaded
* definition BOARD_ALLEGRO/conn/38 successfully loaded
* definition BOARD_ALLEGRO/conn/39 successfully loaded
* definition BOARD_ALLEGRO/conn/40 successfully loaded
* definition BOARD_ALLEGRO/conn/41 successfully loaded
* definition BOARD_ALLEGRO/conn/42 successfully loaded
* definition BOARD_ALLEGRO/conn/43 successfully loaded
* definition BOARD_ALLEGRO/conn/44 successfully loaded
* definition BOARD_ALLEGRO/conn/45 successfully loaded
* definition BOARD_ALLEGRO/conn/46 successfully loaded
* definition BOARD_ALLEGRO/conn/47 successfully loaded
* definition BOARD_ALLEGRO/conn/48 successfully loaded
* definition BOARD_ALLEGRO/conn/49 successfully loaded
* definition BOARD_ALLEGRO/conn/50 successfully loaded
* definition BOARD_ALLEGRO/conn/51 successfully loaded
* definition BOARD_ALLEGRO/conn/52 successfully loaded
* definition BOARD_ALLEGRO/conn/53 successfully loaded
* definition BOARD_ALLEGRO/conn/54 successfully loaded
* definition BOARD_ALLEGRO/conn/55 successfully loaded
* definition BOARD_ALLEGRO/conn/56 successfully loaded
* definition BOARD_ALLEGRO/conn/57 successfully loaded
* definition BOARD_ALLEGRO/conn/58 successfully loaded
* definition BOARD_ALLEGRO/conn/59 successfully loaded
* definition BOARD_ALLEGRO/conn/60 successfully loaded
* definition BOARD_ALLEGRO/conn/61 successfully loaded
* definition BOARD_ALLEGRO/conn/62 successfully loaded
* definition BOARD_ALLEGRO/conn/63 successfully loaded
* definition BOARD_ALLEGRO/conn/64 successfully loaded
* definition BOARD_ALLEGRO/conn/65 successfully loaded
* definition BOARD_ALLEGRO/conn/66 successfully loaded
* definition BOARD_ALLEGRO/conn/67 successfully loaded
* definition BOARD_ALLEGRO/conn/68 successfully loaded
* definition BOARD_ALLEGRO/conn/69 successfully loaded
* definition BOARD_ALLEGRO/conn/70 successfully loaded
* definition BOARD_ALLEGRO/conn/71 successfully loaded
* definition BOARD_ALLEGRO/conn/72 successfully loaded
* definition BOARD_ALLEGRO/conn/73 successfully loaded
* definition BOARD_ALLEGRO/conn/74 successfully loaded
* definition BOARD_ALLEGRO/conn/75 successfully loaded
* definition BOARD_ALLEGRO/conn/76 successfully loaded
* definition BOARD_ALLEGRO/conn/77 successfully loaded
* definition BOARD_ALLEGRO/conn/78 successfully loaded
* definition BOARD_ALLEGRO/conn/79 successfully loaded
* definition BOARD_ALLEGRO/conn/80 successfully loaded
* definition BOARD_ALLEGRO/conn/81 successfully loaded
* definition BOARD_ALLEGRO/conn/82 successfully loaded
* definition BOARD_ALLEGRO/conn/83 successfully loaded
* definition BOARD_ALLEGRO/conn/84 successfully loaded
* definition BOARD_ALLEGRO/conn/85 successfully loaded
* definition BOARD_ALLEGRO/conn/86 successfully loaded
* definition BOARD_ALLEGRO/conn/87 successfully loaded
* definition BOARD_ALLEGRO/conn/88 successfully loaded
* definition BOARD_ALLEGRO/conn/89 successfully loaded
* definition BOARD_ALLEGRO/conn/90 successfully loaded
* definition BOARD_ALLEGRO/conn/91 successfully loaded
* definition BOARD_ALLEGRO/conn/92 successfully loaded
* definition BOARD_ALLEGRO/conn/93 successfully loaded
* definition BOARD_ALLEGRO/conn/94 successfully loaded
* definition BOARD_ALLEGRO/conn/95 successfully loaded
* definition BOARD_ALLEGRO/conn/96 successfully loaded
* definition BOARD_ALLEGRO/conn/97 successfully loaded
* definition BOARD_ALLEGRO/conn/98 successfully loaded
* definition BOARD_ALLEGRO/conn/99 successfully loaded
* definition BOARD_ALLEGRO/conn/100 successfully loaded
```



# PCB Parts



Instance name	Database type	Position X(th)	Position Y(th)	Angle	Horizontal Flip	Visibility	Group
A1	Package	3826.29134	3365.66929	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
J1	Package	370.07992	1202.25984	180	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
J2	Package	740.07874	1202.25984	180	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
J3	Package	1110.07874	1202.25984	180	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
J4	Package	6755.07874	1202.25984	180	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
J5	Package	7125.07874	1202.25984	180	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
J6	Package	7495.07874	1202.25984	180	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

# Core Via Farm(s) with ILN's attached to multiple Instances

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Dia_175	IO	DQ1	0	0	DQ345	M_CPU1_CH3_DQ<5>	M_CPU1_CH3_DQ<13>	M_CPU1_CH3_DQ<21>	M_CPU1_CH3_DQ<29>				
2	Dia_175	IO	DQ2	0	588	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
3	Dia_175	IO	DQ3	0	1764	DQ345	M_CPU1_CH3_DQ<4>	M_CPU1_CH3_DQ<12>	M_CPU1_CH3_DQ<20>	M_CPU1_CH3_DQ<28>				
4	Dia_175	IO	DQ4	0	2352	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
5	Dia_175	IO	DQ5	508	-294	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
6	Dia_175	IO	DQ6	508	882	DQ345	M_CPU1_CH3_DQ<0>	M_CPU1_CH3_DQ<8>	M_CPU1_CH3_DQ<16>	M_CPU1_CH3_DQ<24>				
7	Dia_175	IO	DQ7	508	1470	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
8	Dia_175	IO	DQ8	608.278	2471.896	DQ345	M_CPU1_CH3_DQS_DP<9>	M_CPU1_CH3_DQS_DP<10>	M_CPU1_CH3_DQS_DP<11>	M_CPU1_CH3_DQS_DP<12>				
9	Dia_175	IO	DQ9	915.723	1938.104	DQ345	M_CPU1_CH3_DQS_DN<9>	M_CPU1_CH3_DQS_DN<10>	M_CPU1_CH3_DQS_DN<11>	M_CPU1_CH3_DQS_DN<12>				
10	Dia_175	IO	DQ10	1016	0	DQ345	M_CPU1_CH3_DQ<1>	M_CPU1_CH3_DQ<9>	M_CPU1_CH3_DQ<17>	M_CPU1_CH3_DQ<25>				
11	Dia_175	IO	DQ11	1016	588	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
12	Dia_175	IO	DQ12	1524	-294	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
13	Dia_175	IO	DQ13	1524	882	DQ345	M_CPU1_CH3_DQ<7>	M_CPU1_CH3_DQ<15>	M_CPU1_CH3_DQ<23>	M_CPU1_CH3_DQ<31>				
14	Dia_175	IO	DQ14	1524	1470	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
15	Dia_175	IO	DQ15	1624.278	2471.896	DQ345	M_CPU1_CH3_DQS_DP<0>	M_CPU1_CH3_DQS_DP<1>	M_CPU1_CH3_DQS_DP<2>	M_CPU1_CH3_DQS_DP<3>				
16	Dia_175	IO	DQ16	1931.723	1938.104	DQ345	M_CPU1_CH3_DQS_DN<0>	M_CPU1_CH3_DQS_DN<1>	M_CPU1_CH3_DQS_DN<2>	M_CPU1_CH3_DQS_DN<3>				
17	Dia_175	IO	DQ17	2032	0	DQ345	M_CPU1_CH3_DQ<6>	M_CPU1_CH3_DQ<14>	M_CPU1_CH3_DQ<22>	M_CPU1_CH3_DQ<30>				
18	Dia_175	IO	DQ18	2032	588	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
19	Dia_175	IO	DQ19	2540	-294	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
20	Dia_175	IO	DQ20	2540	882	DQ345	M_CPU1_CH3_DQ<2>	M_CPU1_CH3_DQ<10>	M_CPU1_CH3_DQ<18>	M_CPU1_CH3_DQ<26>				
21	Dia_175	IO	DQ21	2540	1470	DQ345	(Net0)	(Net0)	(Net0)	(Net0)				
22	Dia_175	IO	DQ22	2540	2646	DQ345	M_CPU1_CH3_DQ<3>	M_CPU1_CH3_DQ<11>	M_CPU1_CH3_DQ<19>	M_CPU1_CH3_DQ<27>				

Could change IO to diff pair parent name, and include color, *possible swap code ILNs\** in this spreadsheet. Using one spreadsheet for both part creation and/or signal source would be useful. Also, the ability to use multiple ILN would be good when building part if possible.

# Building Via Farms

The screenshot shows a PCB design software interface with a 'Device properties' dialog box open. The dialog is titled 'Device properties' and has a 'Spreadsheet pins data' page selected. The main area of the dialog is for defining parameters for spreadsheet import. The 'Spreadsheet file' is set to 'Z:\Mentor\_Designs\XPI\KNH\_XPL\_VX2\_DDR\_0505\KNH2\_MASTER\_0516\Output\ViaFarm\_DDR\_DQ0-7\_0\_9\_CH3.csv'. The 'Delimiter' is set to a comma, and the 'Unit' is set to 'UM'. The 'Pad size' is set to '175'. There are checkboxes for 'Ignore header' and 'Mirror', both of which are currently unchecked. An 'Attributes' table is visible on the right side of the dialog, listing properties like Pin Number, X Coord, Y Coord, Signal Type, Functional Signal, Signal Direction, and Instance Level Number. Below the main configuration area is a 'File preview' section showing a list of pins and their coordinates. At the bottom of the dialog, there are two text boxes for 'Power net regular expression' and 'Ground net regular expression', with a note explaining their use. The 'OK' and 'Cancel' buttons are at the bottom right of the dialog.

Property	Column order
Pin Number	1
X Coord	5
Y Coord	6
Signal Type	3
Functional Signal	4
Signal Direction	
Instance Level Number	8

File preview:

```

1, Dia_111, IO, M_CPU1_CH3_DQ<5>,0,0,DQ0-7_0_9, M_CPU1_CH3_DQ<5>
2, Dia_111, IO, (Net0),0,588,DQ0-7_0_9, (Net0)
3, Dia_111, IO, M_CPU1_CH3_DQ<4>,0,1764,DQ0-7_0_9, M_CPU1_CH3_DQ<4>
4, Dia_111, IO, (Net0),0,2352,DQ0-7_0_9, (Net0)
5, Dia_111, IO, (Net0),508,-294,DQ0-7_0_9, (Net0)
6, Dia_111, IO, M_CPU1_CH3_DQ<0>,508,882,DQ0-7_0_9, M_CPU1_CH3_DQ<0>
7, Dia_111, IO, (Net0),508,1470,DQ0-7_0_9, (Net0)
8, Dia_111, IO, M_CPU1_CH3_DQS_DP<9>,608,278,2471,896,DQ0-7_0_9, M_CPU1_CH3_DQS_DP<9>
    
```

The following two fields can be used to define regular expressions allowing to find Power and Ground nets in file. Such nets will be converted to signals of the appropriate type. If no regular expressions are defined only multipin nets will be recognized and Vdd type will be assumed.

Power net regular expression:

Ground net regular expression:

# Multiple ILN's for multiple instances

The screenshot displays the Xpedition Package Integrator interface. The main window shows a board layout with several blue circular markers representing DDR3\_0 instances. A 'Device properties' dialog box is open, showing the 'Spreadsheet signals data' tab. The dialog includes a table with columns for Instance Name, I2N Column, and I2N Row. The table lists 14 instances of DDR3\_0, each with a unique instance name and coordinates.

Instance Name	I2N Column	I2N Row
KNH2_MASTER_0503/DDR3_0	4	8
KNH2_MASTER_0503/DDR3_1	1	9
KNH2_MASTER_0503/DDR3_2	7	10
KNH2_MASTER_0503/DDR3_3	11	11

The 'Data preview' section of the dialog shows a grid of data for each instance, including signal names and values. The data is organized into columns corresponding to the instance names listed in the table above.

# Defining DDR BYTE Lane pin groups

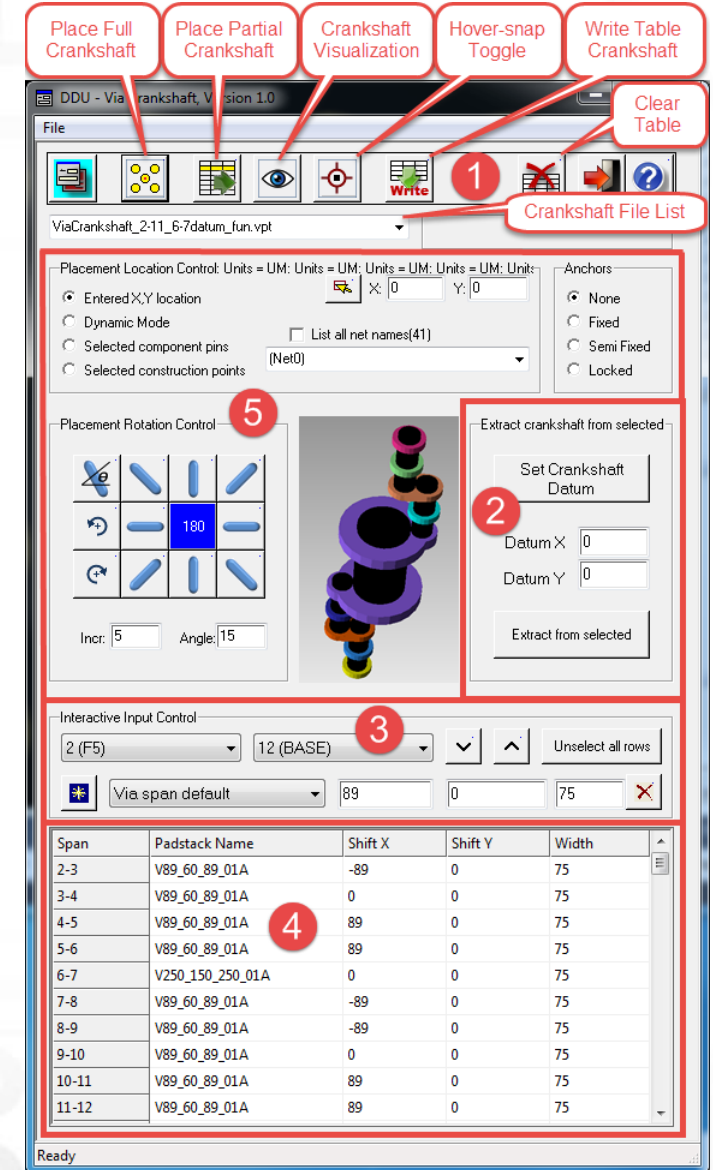
The screenshot displays the Xpedition Package Integrator interface for a DDR controller. The main window is divided into several panes:

- Region Properties:** Shows a list of signals and their values.
- Project:** Lists project components like `KNH2_MASTER_0503`.
- Pin List Table:** A large table with columns for Instance Name, UCL Name, Instance Level Net, Instance Lev., Dir, Pin, Number, Name, Functional Signal, and Instance Level Net. It lists numerous DDR signals such as `M_CPU1_CH0_DQ<0>` through `M_CPU1_CH0_DQ<51>`.
- Pin Grid:** A physical pin grid layout showing the placement of pins. Red circles highlight specific pin groups, likely representing DDR BYTE lanes.
- PreGroups:** A small table on the right showing group names and types.

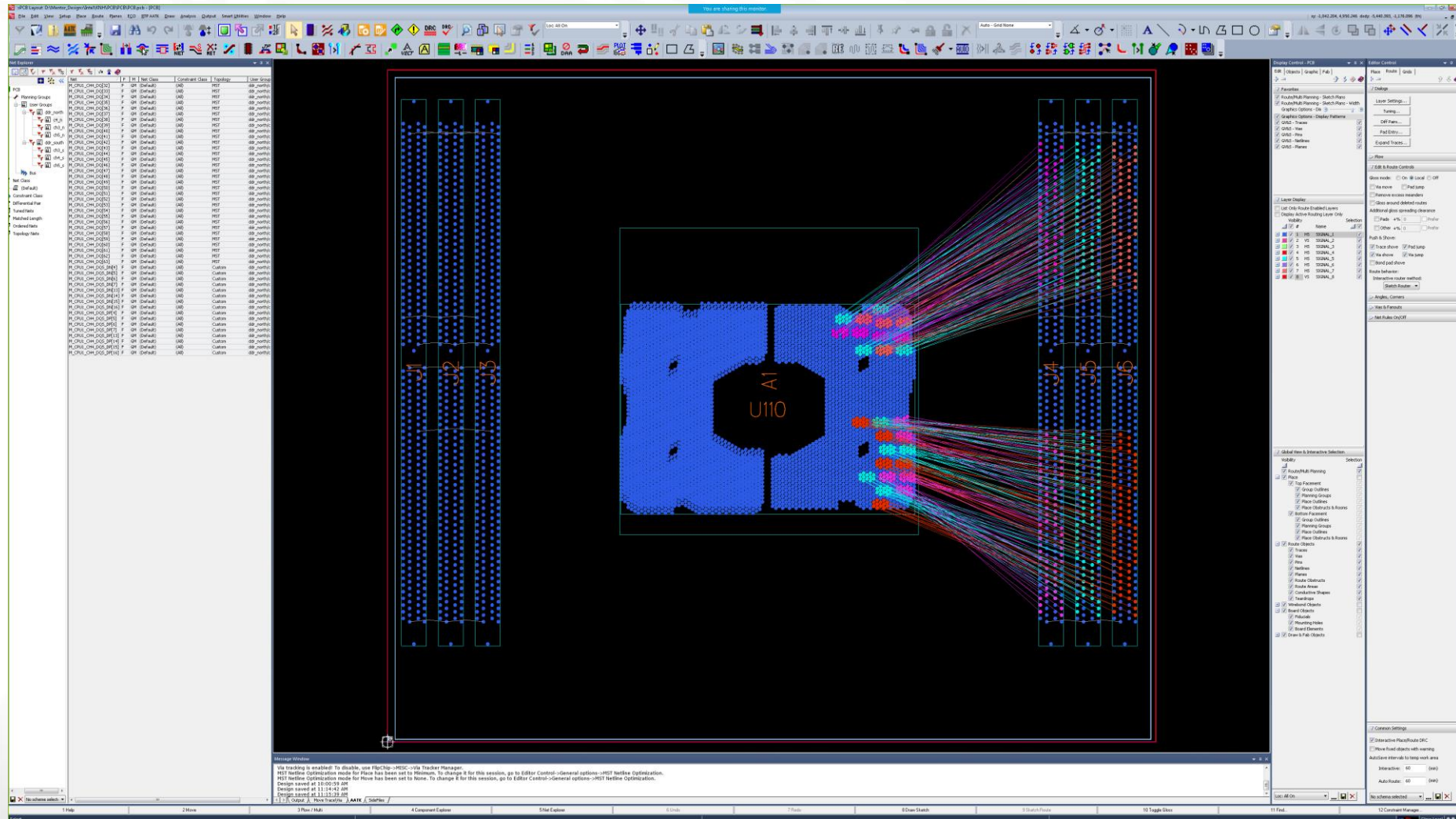
# Assign Crankshaft Vias on Via Farms

Create and assign via crankshafts with net adoption from selected pins or existing net objects in added crankshaft via location.

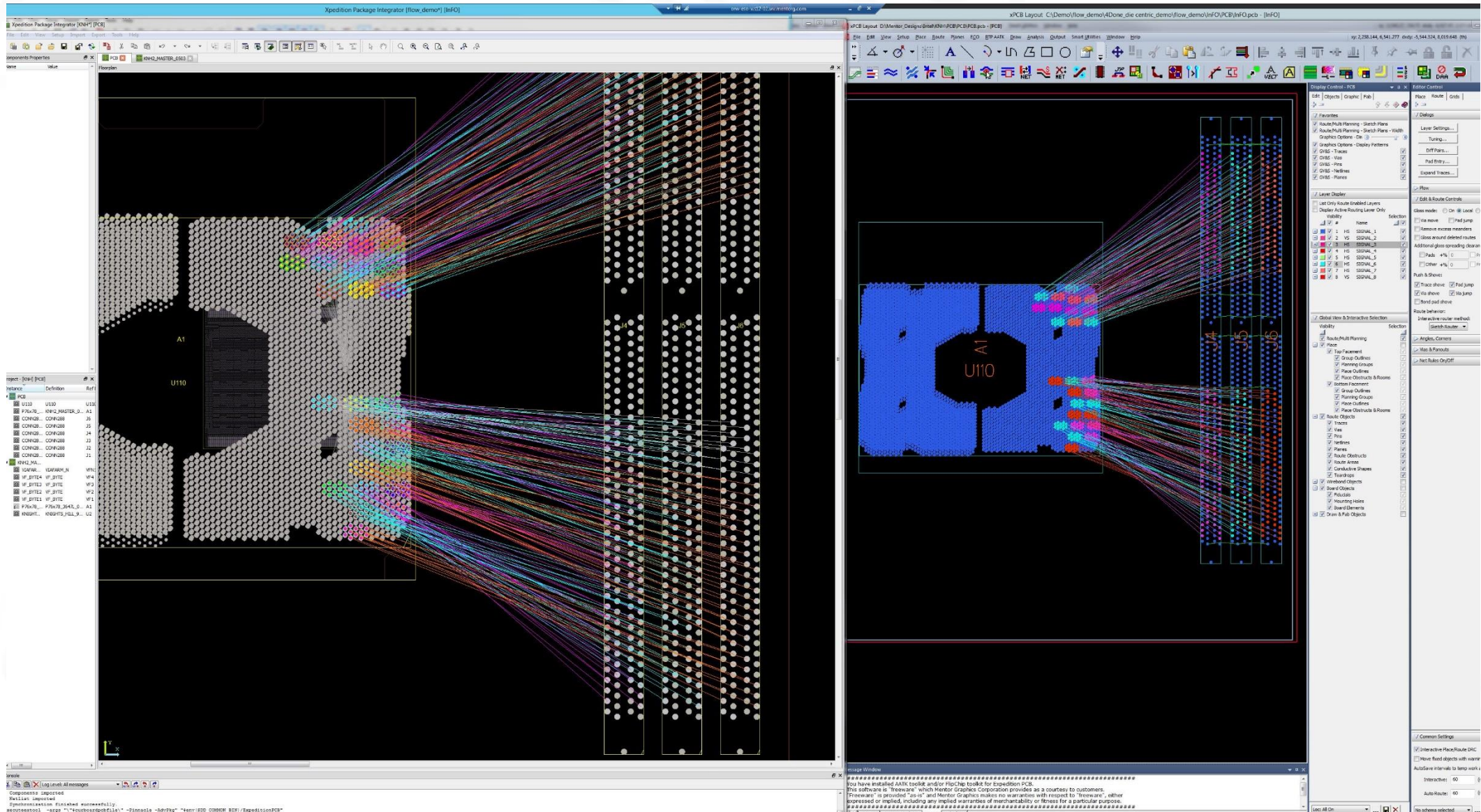
1. Via crankshaft toolbar and file control
2. Extract via crankshaft from selected
3. Interactive build via crankshaft
4. Via crankshaft Table
5. Dynamic and menu placement control



# PCB with package LGA and Board LGA View per Byte Lane

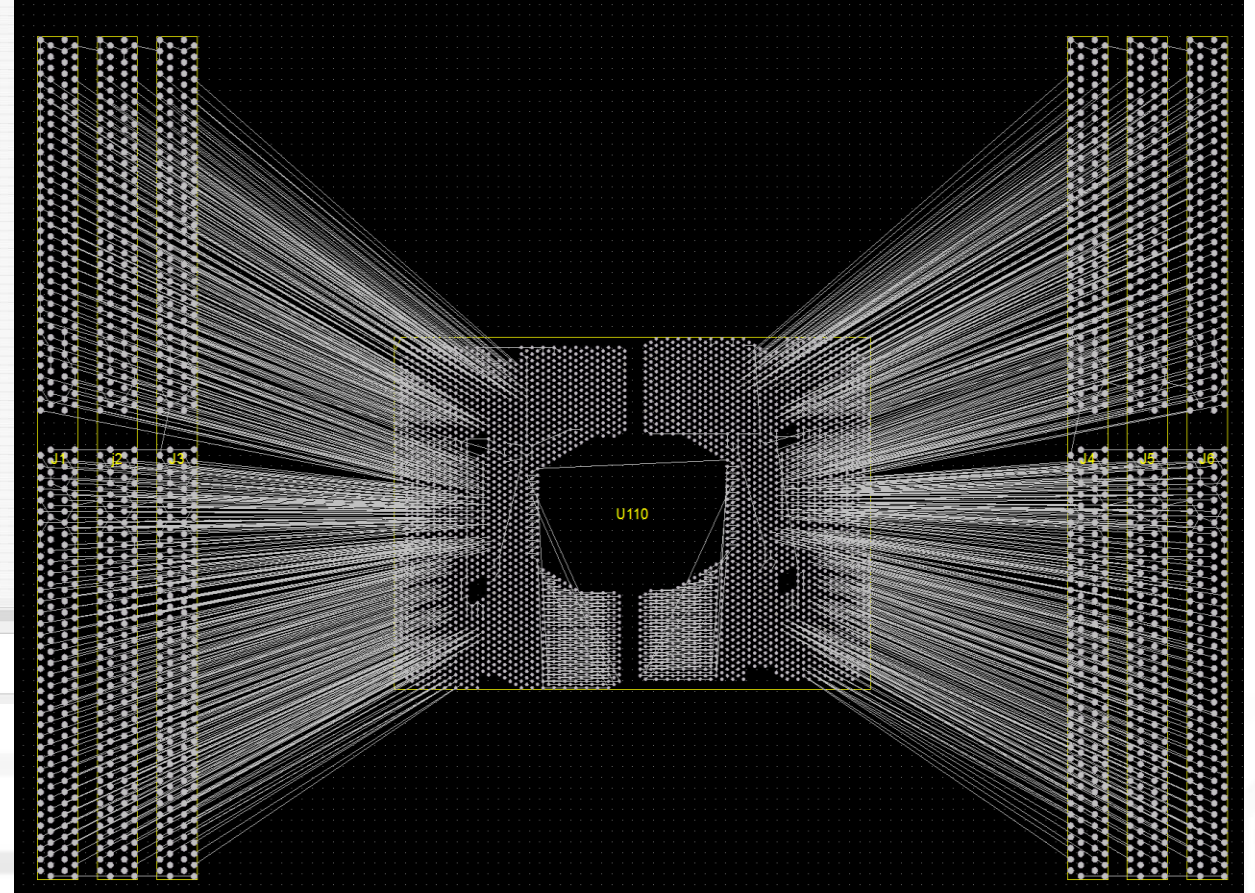
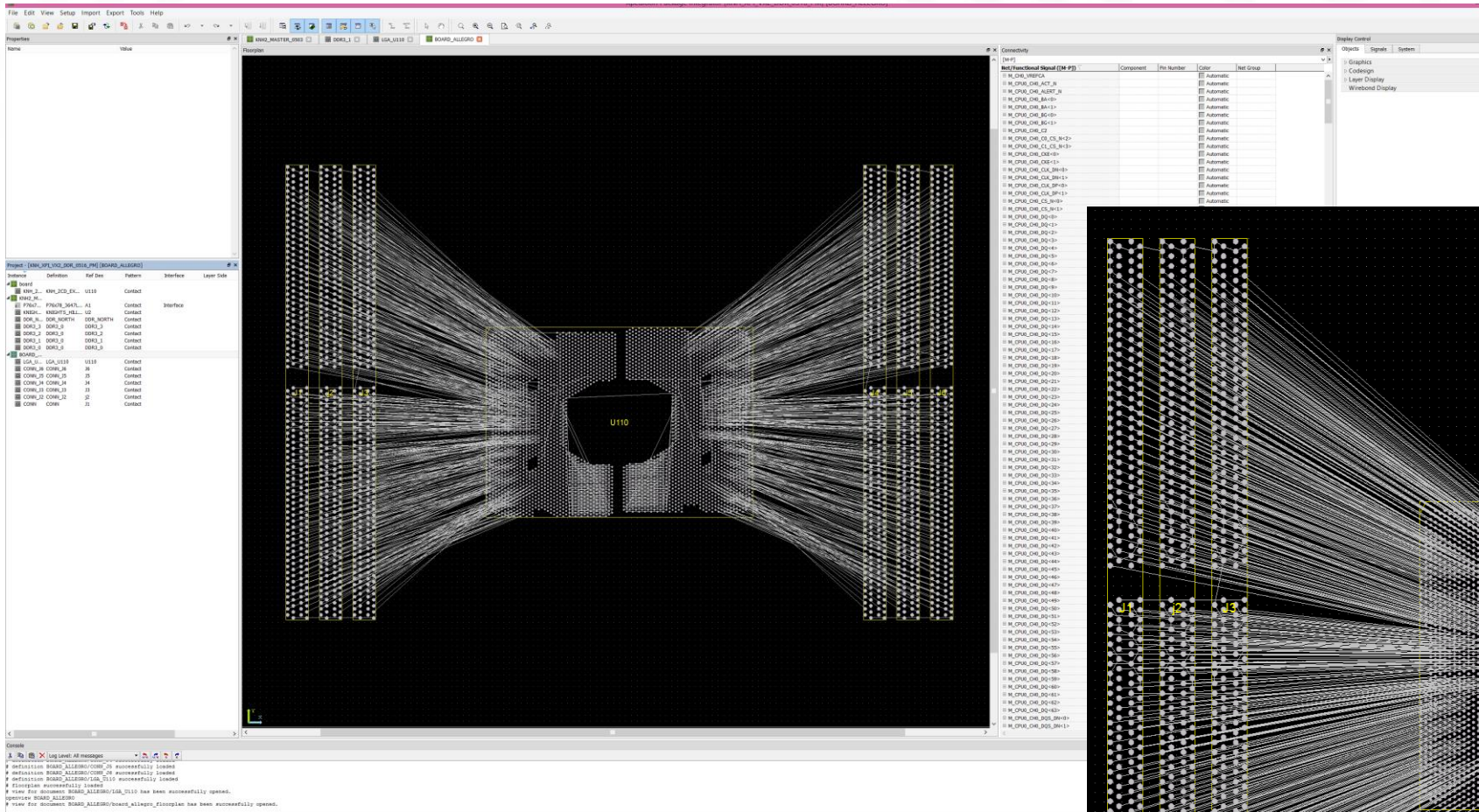


# Die and PCB Floorplan in Floorplan XPI view and PCB View

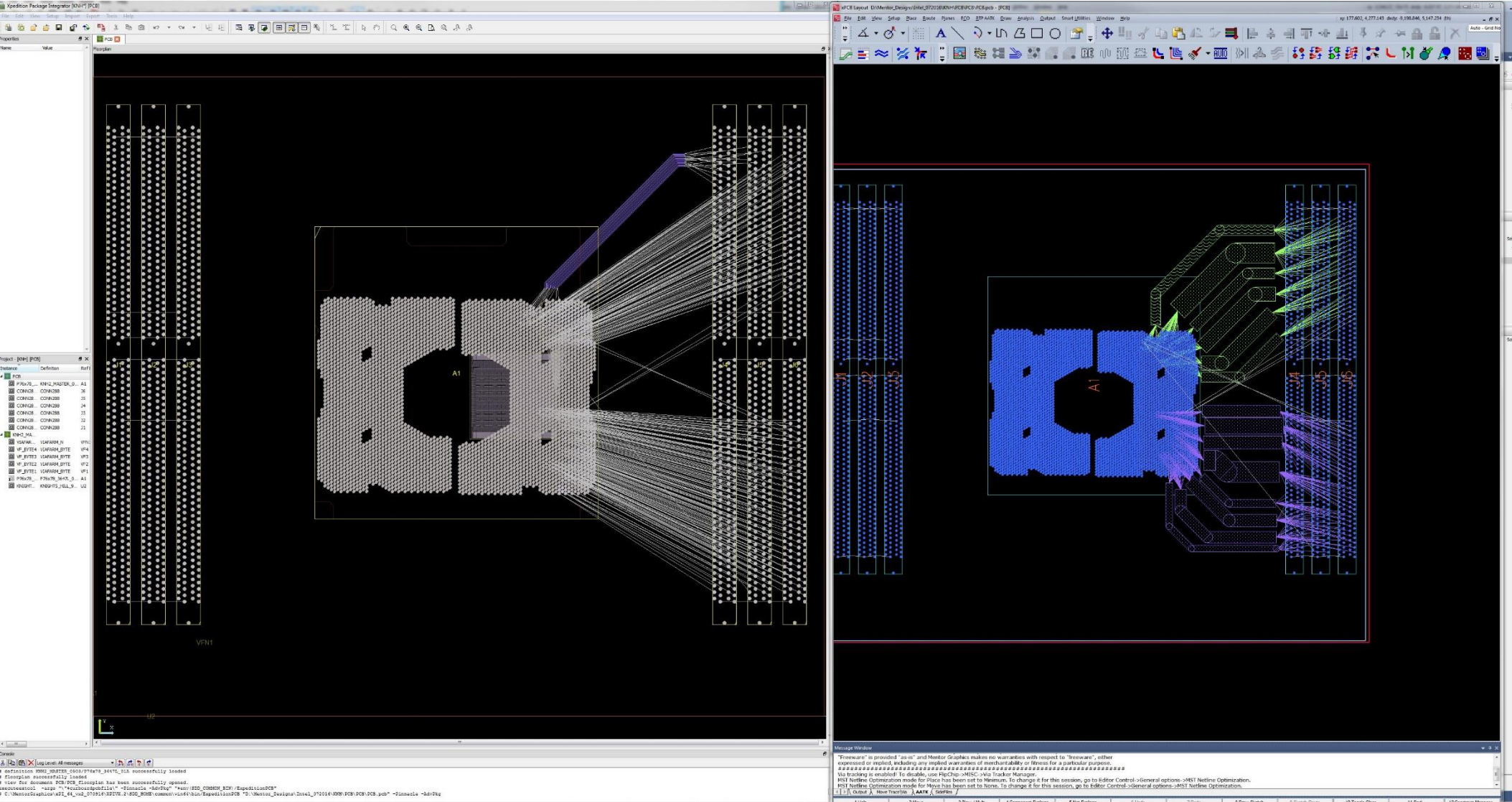




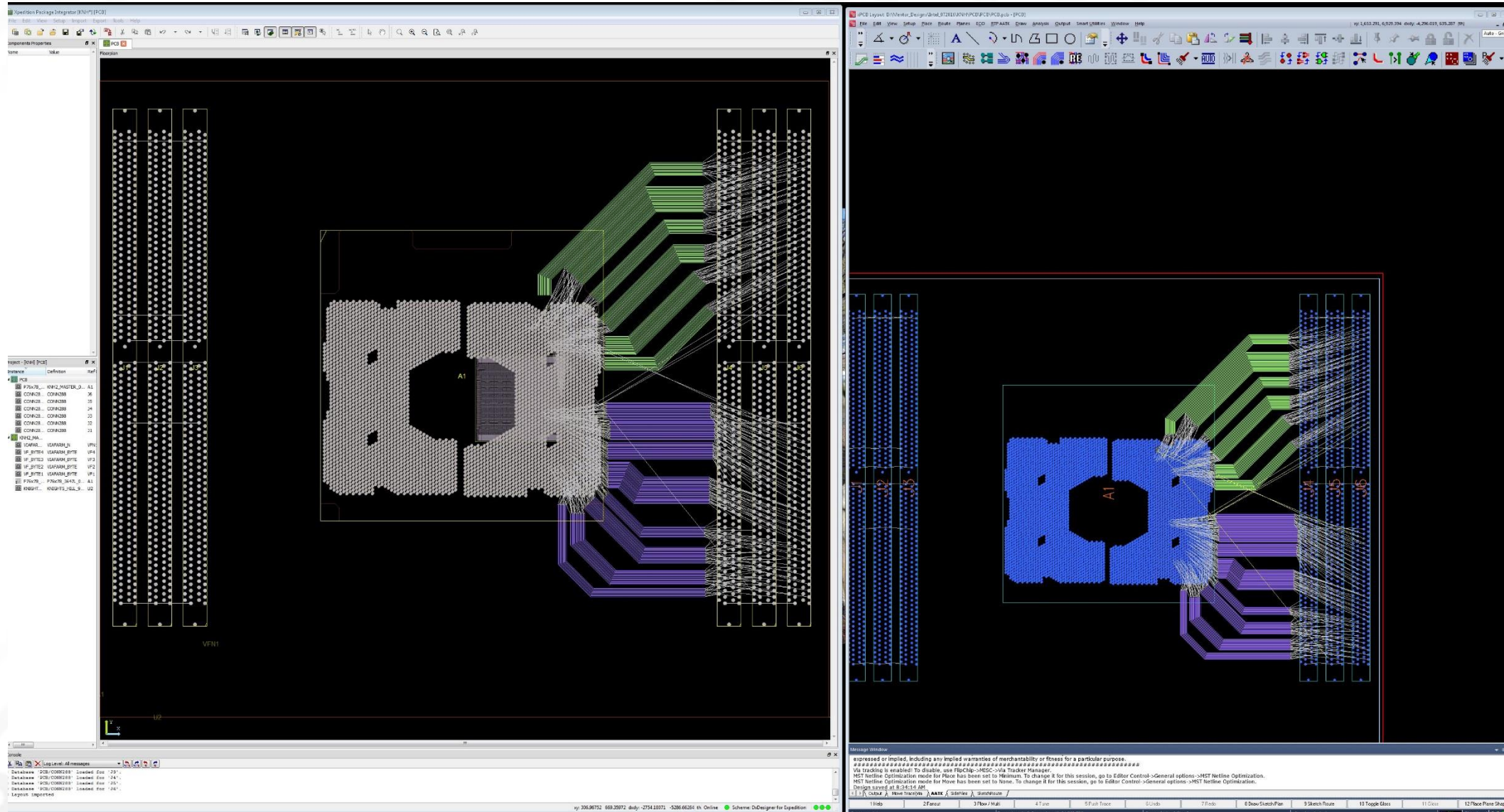
# PCB Before and After Unravel



# xPI and xPCB Sketch Planner view



# PCB view with Package Floor plan with trunk routing



# Package Integrator Demo

## Sketch Router/Planner/Trunk routing demo