PCB Design With FGPAs and BGA Packages by Kyler Callahan

This tutorial assumes that the reader has a basic knowledge of how to use Mentor Graphics PADs, and DxDesigner. If you have not already please go through the PADs evaluation guide located in: C:\PADS_ES_Evaluation\Document\PADS ES Suite Evaluation Guide.pdf

Introduction:

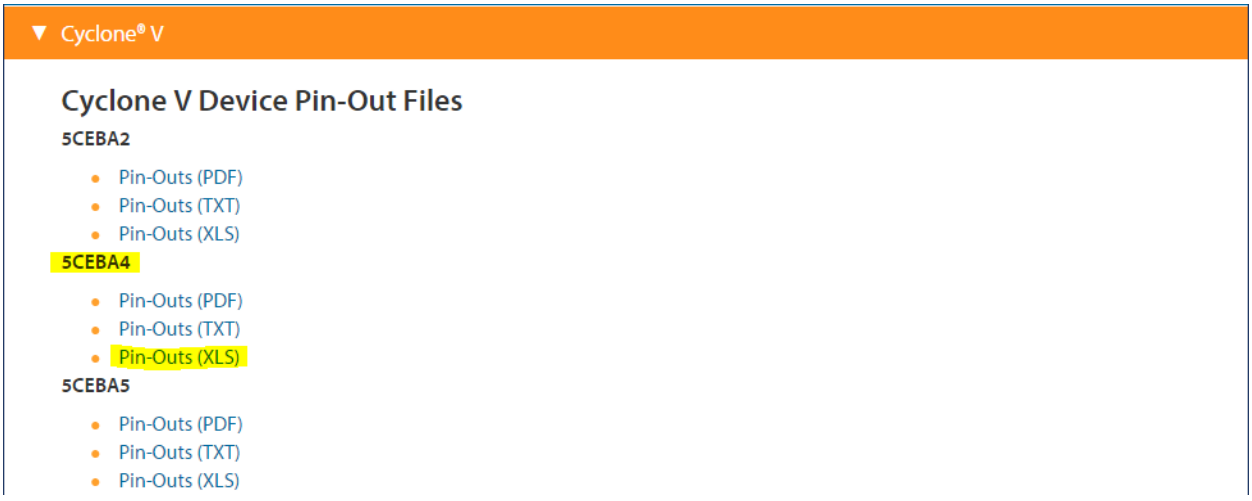
This tutorial will cover techniques of how to work with FPGAs and BGA packages using the *Mentor Graphics Suite*. The programs we will be focusing on using will be: PADS Layout, PADS Router, and DxDesigner. Topics covered will be: Acquiring necessary resources for using the specified chip, creating FPGA symbols in DxDesigner and PADS, basic break-out techniques, choosing decoupling capacitors, and further recommended reading.

This tutorial will be targeting the *5CEBA4U15C7N* Altera Cyclone V device. However many topics discussed in this tutorial will carry over to non-Altera devices and even non-FPGA devices.

Acquiring necessary resources:

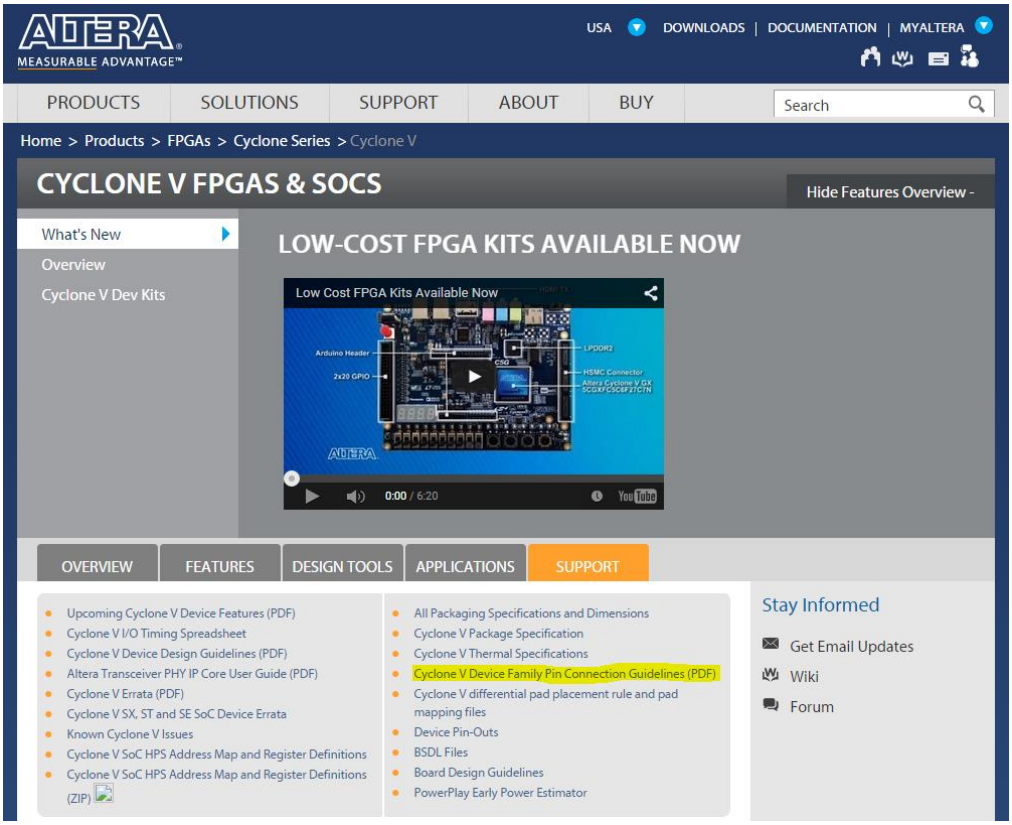
1) The first and foremost important things that need to be done before we do anything else, is finalizing a device and device package type. The requirements for doing this changes from project to project and will not be covered in this tutorial.

2) Once you have a *device* and *device package type* you need to find your pin-out files. Check with your device manufacturing website to find the pin-out file of your device. The pin-out may also be located in the device data sheet.



2a) If an XLS Pin-out is available it is recommended to use that as we will be using Excel to make our device symbols for DxDesigner.

3) Next acquire the *Device Connection Guidelines* (If available). This document will prove useful in determining how to hook up critical pins in the device.

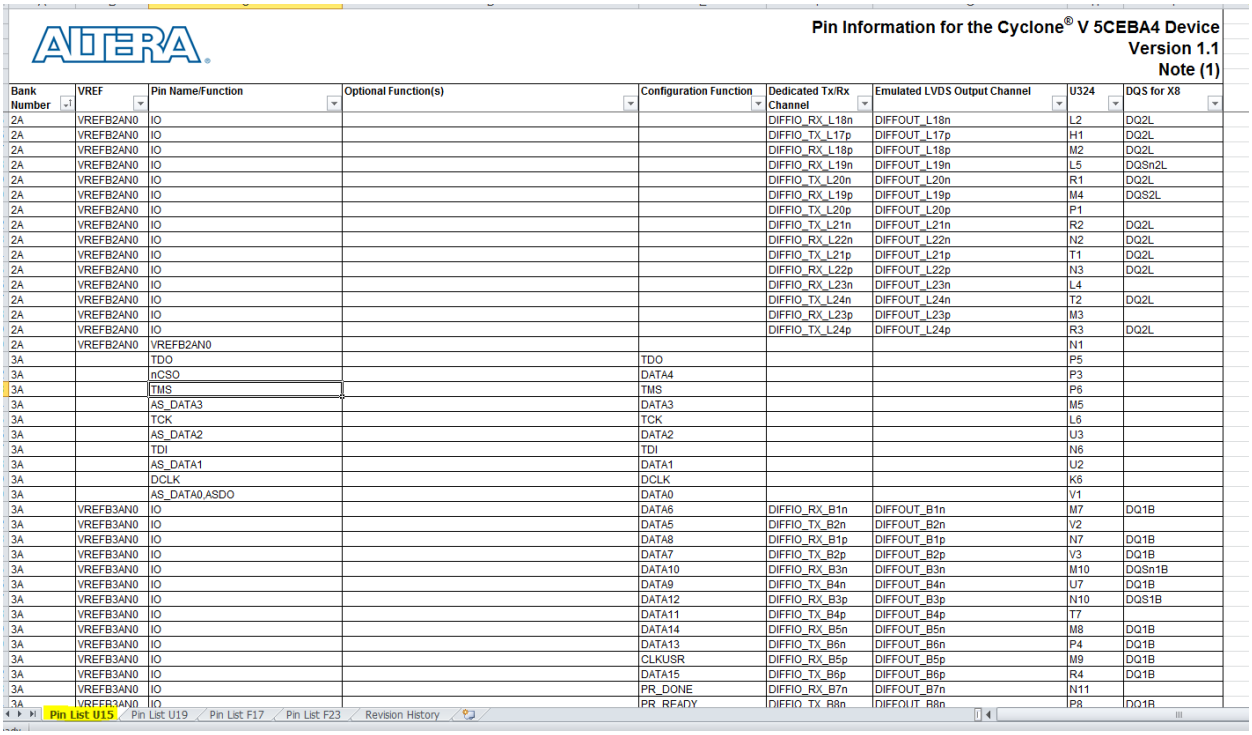


4) It is also recommended to look through other documentation provided by the device manufacturer to better familiarize yourself with how to hook up the device and whether there may be recommended designs or layouts.

Note: You may skip step 4 if you are following along in this tutorial and trying to replicate the example as all of that work has been done for you.

Creating FPGA Symbols in DxDesigner and PADS:

1) Open the pin-out file in Excel and browse to the package type that you have chosen. For our example we are using the *U15 (Stands for UltraFineBGA at .8mm pin pitch with a 15mmx15mm* package)



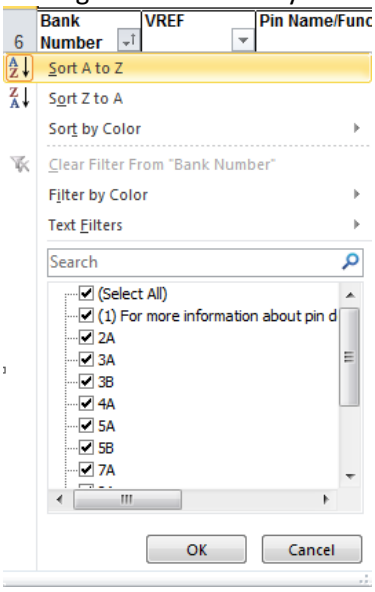
Pin Information for the Cyclone® V 5CEBA4 Device
Version 1.1
Note (1)

Bank Number	VREF	Pin Name/Function	Optional Function(s)	Configuration Function	Dedicated Tx/Rx Channel	Emulated LVDS Output Channel	U324	DQS for X8
2A	VREFB2AN0	IO			DIFFIO_RX_L18n	DIFFOUT_L18n	L2	DQ2L
2A	VREFB2AN0	IO			DIFFIO_TX_L17p	DIFFOUT_L17p	H1	DQ2L
2A	VREFB2AN0	IO			DIFFIO_RX_L18p	DIFFOUT_L18p	M2	DQ2L
2A	VREFB2AN0	IO			DIFFIO_RX_L19n	DIFFOUT_L19n	L5	DQSn2L
2A	VREFB2AN0	IO			DIFFIO_TX_L20n	DIFFOUT_L20n	R1	DQ2L
2A	VREFB2AN0	IO			DIFFIO_RX_L19p	DIFFOUT_L19p	M4	DQS2L
2A	VREFB2AN0	IO			DIFFIO_TX_L20p	DIFFOUT_L20p	P1	
2A	VREFB2AN0	IO			DIFFIO_TX_L21n	DIFFOUT_L21n	R2	DQ2L
2A	VREFB2AN0	IO			DIFFIO_RX_L22n	DIFFOUT_L22n	N2	DQ2L
2A	VREFB2AN0	IO			DIFFIO_TX_L21p	DIFFOUT_L21p	T1	DQ2L
2A	VREFB2AN0	IO			DIFFIO_RX_L22p	DIFFOUT_L22p	N3	DQ2L
2A	VREFB2AN0	IO			DIFFIO_RX_L23n	DIFFOUT_L23n	L4	
2A	VREFB2AN0	IO			DIFFIO_TX_L24n	DIFFOUT_L24n	T2	DQ2L
2A	VREFB2AN0	IO			DIFFIO_RX_L23p	DIFFOUT_L23p	M3	
2A	VREFB2AN0	IO			DIFFIO_TX_L24p	DIFFOUT_L24p	R3	DQ2L
2A	VREFB2AN0	VREFB2AN0					N1	
3A		TDO		TDO			P5	
3A		nCSO		DATA4			P3	
3A		TMS		TMS			P6	
3A		AS_DATA3		DATA3			M5	
3A		TCK		TCK			L6	
3A		AS_DATA2		DATA2			U3	
3A		TDI		TDI			N6	
3A		AS_DATA1		DATA1			U2	
3A		DCLK		DCLK			K6	
3A		AS_DATA0,ASDO		DATA0			V1	
3A	VREFB3AN0	IO		DATA6	DIFFIO_RX_B1n	DIFFOUT_B1n	M7	DQ1B
3A	VREFB3AN0	IO		DATA5	DIFFIO_TX_B2n	DIFFOUT_B2n	V2	
3A	VREFB3AN0	IO		DATA8	DIFFIO_RX_B1p	DIFFOUT_B1p	N7	DQ1B
3A	VREFB3AN0	IO		DATA7	DIFFIO_TX_B2p	DIFFOUT_B2p	V3	DQ1B
3A	VREFB3AN0	IO		DATA10	DIFFIO_RX_B3n	DIFFOUT_B3n	M10	DQSn1B
3A	VREFB3AN0	IO		DATA9	DIFFIO_TX_B4n	DIFFOUT_B4n	U7	DQ1B
3A	VREFB3AN0	IO		DATA12	DIFFIO_RX_B3p	DIFFOUT_B3p	N10	DQS1B
3A	VREFB3AN0	IO		DATA11	DIFFIO_TX_B4p	DIFFOUT_B4p	T7	
3A	VREFB3AN0	IO		DATA14	DIFFIO_RX_B5n	DIFFOUT_B5n	M8	DQ1B
3A	VREFB3AN0	IO		DATA13	DIFFIO_TX_B6n	DIFFOUT_B6n	P4	DQ1B
3A	VREFB3AN0	IO		CLKUSR	DIFFIO_RX_B5p	DIFFOUT_B5p	M9	DQ1B
3A	VREFB3AN0	IO		DATA15	DIFFIO_TX_B6p	DIFFOUT_B6p	R4	DQ1B
3A	VREFB3AN0	IO		PR_DONE	DIFFIO_RX_B7n	DIFFOUT_B7n	N11	
3A	VREFB3AN0	IO		PR_READY	DIFFIO_TX_B8n	DIFFOUT_B8n	P8	DQ1B

Pin List U15 | Pin List U19 | Pin List F17 | Pin List F23 | Revision History

The pin-out file should look something like this for Altera devices

1a) We are going to break up the pin-outs into banks so we can create manageable symbols. Start by sorting the document by bank number.



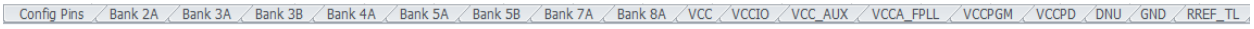
2) Now that you have the pins sorted by bank number you will need to decide how to split up the banks. It helps if you have a single Excel file with separate sheets for each symbol you are going to make. For each sheet make note of at least:

- Pin Name/ Function
- Pin Number

You may find it useful to also include

- Bank Number
- Page number in the Connection Guideline and any notes about connection guidelines

In this particular example we will have 18 different symbols:



- Config Pins
  - These are pins used for configuring the device. They may come from other banks, if they do leave them off of that bank when you make it

	A	B	C	D	F
1	**Config Function**	**Pin Number**	**Bank Number**	**Page in Connection Guidelines/Notes**	**Pin Name**
2	CONF_DONE	C6	9A	Pg5: Connect 10k pullup to VCCPGM	CONF_DONE
3	DATA0	V1	3A	Pg3: See guide	AS_DATA0,ASDO
4	DATA1	U2	3A	Pg3: See guide	AS_DATA1
5	DATA2	U3	3A	Pg3: See guide	AS_DATA2
6	DATA3	M5	3A	Pg3: See guide	AS_DATA3
7	DATA4	P3	3A	Pg4: See guide	nCSO
8	DCLK	K6	3A	Pg8: See guid, do not leave floating	DCLK
9	MSEL0	J6	9A	Pg3: Tie to GND for JTAG	MSEL0
10	MSEL1	H5	9A	Pg3: Tie to GND for JTAG	MSEL1
11	MSEL2	D3	9A	Pg3: Tie to GND for JTAG	MSEL2
12	MSEL3	G5	9A	Pg3: Tie to GND for JTAG	MSEL3
13	MSEL4	G4	9A	Pg3: Tie to GND for JTAG	MSEL4
14	nCE	E4	9A	Pg4: Active low enable	nCE
15	nCONFIG	D4	9A	Pg4: Reset during configuration	nCONFIG
16	nSTATUS	D6	9A	Pg6: Connect 10k pull-up	nSTATUS
17	TCK	L6	3A	Pg6: JTAG test clock input	TCK
18	TDI	N6	3A	Pg7: JTAG test data input	TDI
19	TDO	P5	3A	Pg7: JTAG test data output	TDO
20	TMS	P6	3A	Pg7: JTAG test mode select	TMS

- Banks 2A, 3A, 3B, 4A, 5A, 5B, 7A, and 8A
  - These are all of the I/O pins with included VREF pins. If an I/O pin has an additional function it is indicated. Example below.

	A	B	C	D
1	**Pin Name/Function**	**Pin Number**	**Bank Number**	**Page in Connection Guidelines/Notes**
2	I/O	L16	5A	
3	I/O	G13	5A	
4	I/O	H13	5A	
5	I/O	J13	5A	
6	I/O	J14	5A	
7	I/O	J15	5A	
8	I/O	J16	5A	
9	I/O	K18	5A	
10	I/O	L17	5A	
11	I/O (CRC_ERROR)	K13	5A	Pg8: Detects errors in configuration
12	I/O (CvP_CONFDONE)	N18	5A	Pg10: Indicates CvP is done
13	I/O (DEV_CLRn)	K16	5A	Pg8: Clears all registers
14	I/O (DEV_OE)	K17	5A	Pg9: Tristates all I/O pins
15	I/O (INIT_DONE)	L14	5A	Pg9: Indicates an enter into user mode
16	I/O (nCEO)	M18	5A	Pg5: Use I/O or leave floating
17	I/O (PR_REQUEST)	L15	5A	Pg11: Partial reconfiguration request
18	VREFB5AN0	M17	5A	Pg20: Input reference voltage

- VCC pins
    - These are the core voltages for the device

	A	B	C
1	**Pin Name/Function**	**Pin Number**	**Page in Connection Guidelines/Notes**
2	VCC	M11	Pg17: Connect to 1.1V
3	VCC	L12	Pg17: Connect to 1.1V
4	VCC	L10	Pg17: Connect to 1.1V
5	VCC	L8	Pg17: Connect to 1.1V
6	VCC	K11	Pg17: Connect to 1.1V
7	VCC	K9	Pg17: Connect to 1.1V
8	VCC	K7	Pg17: Connect to 1.1V
9	VCC	J12	Pg17: Connect to 1.1V
10	VCC	J10	Pg17: Connect to 1.1V
11	VCC	J8	Pg17: Connect to 1.1V
12	VCC	H11	Pg17: Connect to 1.1V
13	VCC	H9	Pg17: Connect to 1.1V
14	VCC	H7	Pg17: Connect to 1.1V
15	VCC	G12	Pg17: Connect to 1.1V
16	VCC	G10	Pg17: Connect to 1.1V
17	VCC	G8	Pg17: Connect to 1.1V

- VCCIO pins
    - These are the pins that determine the voltage that each bank is at.

	A	B	C
1	**Pin Name/Function**	**Pin Number**	**Page in Connection Guidelines/Notes**
2	VCCIO2A	H4	Pg19: Powers the I/O banks
3	VCCIO2A	P2	Pg19: Powers the I/O banks
4	VCCIO2A	L3	Pg19: Powers the I/O banks
5	VCCIO2A	J2	Pg19: Powers the I/O banks
6	VCCIO3A	R5	Pg19: Powers the I/O banks
7	VCCIO3A	U6	Pg19: Powers the I/O banks
8	VCCIO3B	T8	Pg19: Powers the I/O banks
9	VCCIO3B	V9	Pg19: Powers the I/O banks
10	VCCIO4A	T18	Pg19: Powers the I/O banks
11	VCCIO4A	U16	Pg19: Powers the I/O banks
12	VCCIO4A	U11	Pg19: Powers the I/O banks
13	VCCIO4A	R15	Pg19: Powers the I/O banks
14	VCCIO5A	K15	Pg19: Powers the I/O banks
15	VCCIO5A	M16	Pg19: Powers the I/O banks
16	VCCIO5B	F18	Pg19: Powers the I/O banks
17	VCCIO5B	G16	Pg19: Powers the I/O banks
18	VCCIO7A	B16	Pg19: Powers the I/O banks
19	VCCIO7A	E15	Pg19: Powers the I/O banks
20	VCCIO7A	C14	Pg19: Powers the I/O banks
21	VCCIO7A	A13	Pg19: Powers the I/O banks
22	VCCIO8A	C9	Pg19: Powers the I/O banks
23	VCCIO8A	B6	Pg19: Powers the I/O banks

- VCC_AUX
    - These pins are the AUX power pins

	A	B	C
1	**Pin Name/Function**	**Pin Number**	**Page in Connection Guidelines/Notes**
2	VCC_AUX	E6	PG18: See Connection Guidelines
3	VCC_AUX	C10	PG18: See Connection Guidelines
4	VCC_AUX	D14	PG18: See Connection Guidelines
5	VCC_AUX	R14	PG18: See Connection Guidelines
6	VCC_AUX	T10	PG18: See Connection Guidelines
7	VCC_AUX	R6	PG18: See Connection Guidelines

- VCCA_FPLL
    - These are the power pins for the PLL

	A	B	C
1	**Pin Name/Function**	**Pin Number**	**Page in Connection Guidelines/Notes**
2	VCCA_FPLL	H3	Pg17: Connect to 2.5V
3	VCCA_FPLL	N5	Pg17: Connect to 2.5V
4	VCCA_FPLL	F5	Pg17: Connect to 2.5V
5	VCCA_FPLL	N15	Pg17: Connect to 2.5V
6	VCCA_FPLL	F15	Pg17: Connect to 2.5V

- VCCPGM
    - Pins related to the PGM function and extra VCC BAT pin

	A	B	C
1	**Pin Name/Function**	**Pin Number**	**Page in Connection Guidelines/Notes**
2	VCCPGM	T6	Pg19: Dedicated power pins
3	VCCPGM	M15	Pg19: Dedicated power pins
4	VCCPGM	E7	Pg19: Dedicated power pins
5	VCCBAT	C5	Pg19: Connect to 2.5V

- VCCPD
    - Power pins related to the pre-driver voltages

	A	B	C
1	**Pin Name/Function**	**Pin Number**	**Page in Connection Guidelines/Notes**
2	VCCPD1A2A	K4	Pg19: Dedicated power pins
3	VCCPD1A2A	J5	Pg19: Dedicated power pins
4	VCCPD3A	R7	Pg19: Dedicated power pins
5	VCCPD3B4A	R12	Pg19: Dedicated power pins
6	VCCPD3B4A	R8	Pg19: Dedicated power pins
7	VCCPD5A	K14	Pg19: Dedicated power pins
8	VCCPD5B	H15	Pg19: Dedicated power pins
9	VCCPD7A8A	D8	Pg19: Dedicated power pins
10	VCCPD7A8A	E13	Pg19: Dedicated power pins

- DNU
  - Do not use pins. While these pins are not used you will still need a symbol for them
  -

	A	B	C
1	**Pin Name/Function**	**Pin Number**	**Page in Connection Guidelines/Notes**
2	DNU	A2	Pg17: Leave floating
3	DNU	B2	Pg17: Leave floating
4	DNU	C15	Pg17: Leave floating
5	DNU	C8	Pg17: Leave floating

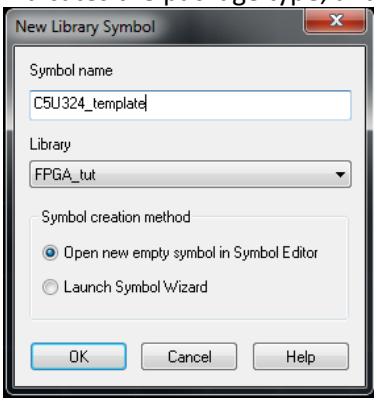
- GND
  - Ground pins. There are more pins than shown below, it's just been edited for space reasons

	A	B	C	D
1	**Pin Name/Function**	**Pin Number**	**Bank Number**	**Page in Connection Guidelines/Notes**
2	GND	D15	7A	Pg20: Connect to ground plane
3	GND	D5	9A	Pg20: Connect to ground plane
4	GND	D12		Pg20: Connect to ground plane
5	GND	V18		Pg20: Connect to ground plane
6	GND	V14		Pg20: Connect to ground plane
7	GND	V4		Pg20: Connect to ground plane
8	GND	U1		Pg20: Connect to ground plane
9	GND	T13		Pg20: Connect to ground plane
10	GND	T3		Pg20: Connect to ground plane
11	GND	R10		Pg20: Connect to ground plane

- RREF_TL
  - Special pin related to transceiver functions

	A	B	C	D
1	**Pin Name/Function**	**Pin Number**	**Bank Number**	**Page in Connection Guidelines/Notes**
2	RREF_TL	A1		Pg16: Connect to GND 2.0kΩ 1% res
3				

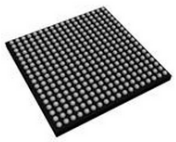
3) Next we create the symbols in DxDesigner. Open DxDesigner and give your symbol name something descriptive. This symbol will be our template for making the 18 other symbols. For this particular example we will be using the name C5U324_template. The C5 lets us know it's a Cyclone 5, U324 indicates the package type, and template shows that it is a template.



4) Add the following property fields to your PCB properties and fill out

- DEVICE
  - This should be something short but descriptive of the device. For this example we will use C5_U324 for the same reasons as above.
- HETERO and HETERO_SYMNAME
  - HETERO is a field of *all* the HETERO_SYMNAMES separated by commas.
    - We are going to split our FPGA up into multiple symbols, this field is necessary for DxDesigner to associate all the separate symbols with each other.
    - This field can only accommodate up to 255 character so make your HETERO_SYMNAMES short
    - The field for this example looks like this: C5U324_Config, C5U324_B2A, C5U324_B3A, C5U324_B3B, C5U324_B4A, C5U324_B5A, C5U324_B5B, C5U324_B7A, C5U324_B8A, C5U324_VCC, C5U324_VCCIO, C5U324_VCC_AUX, C5U324_VCCA_FPLL, C5U324_VCCPGM, C5U324_VCCPD, C5U324_DNU, C5U324_GND, C5U324_RREF_TL
  - HETERO_SYMNAME is the field that is the symbol name of each individual symbols. This field should match the Name
    - Leave this field blank for now. We will fill it out with each individual component. This is our template for now.
    - Make these names descriptive but short. For example all of the names in this example start with "C5U324_" and have a descriptive name after the underscore such as "Config" or "B2A" to represent the Config symbol or the Bank 2A symbol.
- P/D_NUM
  - This is the Part Distributer number. This is the number used by the distributer (IE: DigiKey or Mouser) in their catalog.
- PART_LABEL
  - This is a quick part description
- PART_NUMBER
  - The is the Part Manufacturer's number
- PKG_GRP
  - This is the group of packages this part belongs to. EG: BGA, 16-UFQFN, 100-QFP… etc
- PKG_TYPE
  - This is the name of the footprint in PADS that this component will use
- REFDES
  - This is the reference designator as to what logic group this part belongs to. For more info on this visit this page http://en.wikipedia.org/wiki/Reference_designator

Much of this information can be found on the part's page of the distributer's website.



Enlarge

Mouser Part #:	989-5CEBA4U15C7N	P/D_NUM
Manufacturer Part #:	5CEBA4U15C7N	PART_NUMBER
Manufacturer:	Altera Corporation	
Description:	FPGA - Field Programmable Gate Array FPGA - Cyclone V E 1848 LABs 176 IOs	PART_LABEL
Lifecycle:	NEW **New Technology:** Cutting edge technology for the newest designs.	

Learn more about Altera Corporation 5CEBA4U15C7N

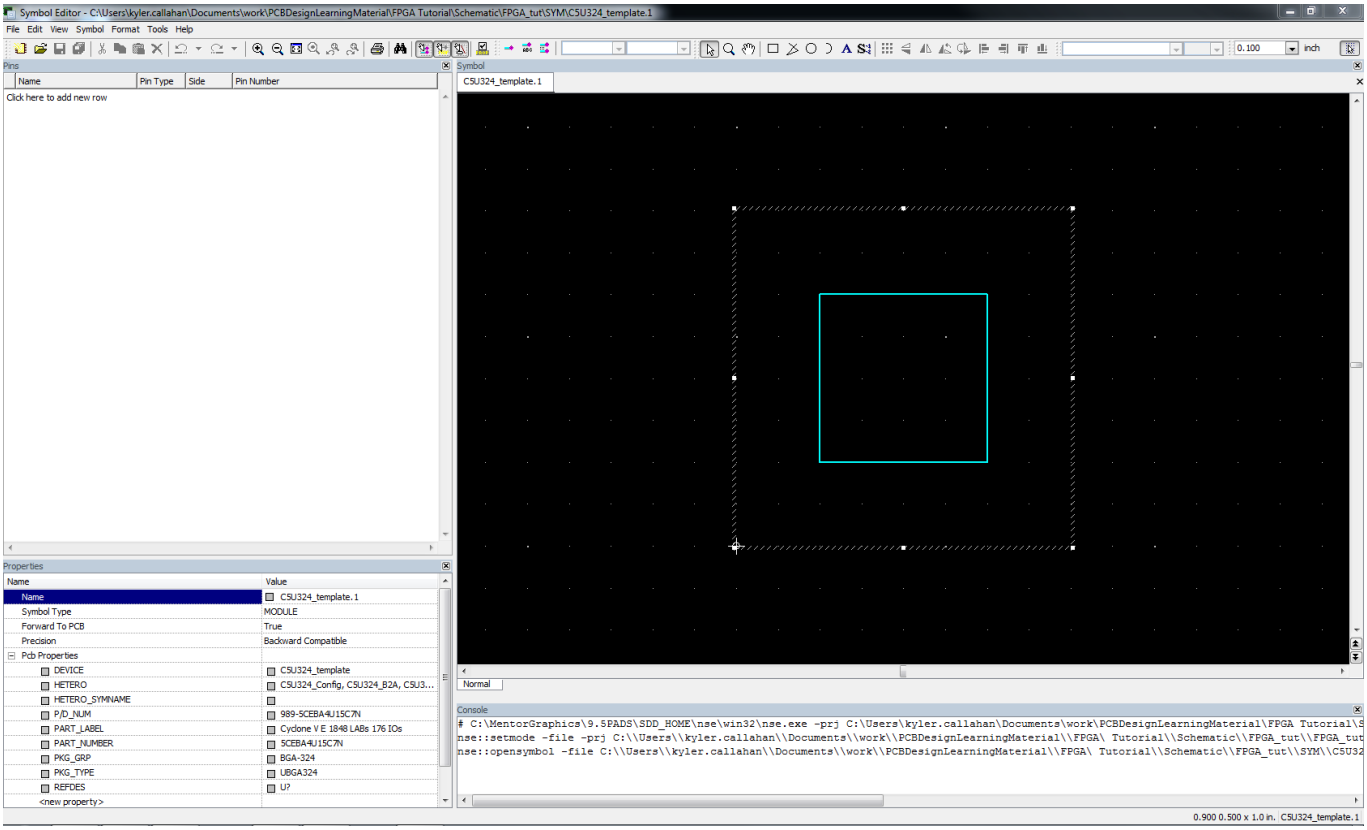
Page 312, Mouser Online Catalog
Page 312, PDF Catalog Page
Data Sheet

Images are for reference only
See Product Specifications

Specifications | Features | Documents (2) | My Notes

Manufacturer:	Altera	☑
Product Category:	FPGA - Field Programmable Gate Array	☑
RoHS:	RoHS Details	
Product:	Cyclone V E	☐
Number of Logic Elements:	49000	☐
Number of Logic Array Blocks - LABs:	18480	☐
Total Memory:	3383 kbit	☐
Number of I/Os:	176	☐
Operating Supply Voltage:	1.1 V	☐
Maximum Operating Temperature:	+ 70 C	☐
Mounting Style:	SMD/SMT	☐
Package / Case:	BGA-324 PKG_GRP	☐
Brand:	Altera Corporation	
Embedded Block RAM - EBR:	303 kbit	
M10K Memory:	3080 kbit	
Maximum Operating Frequency:	800 MHz	
Minimum Operating Temperature:	0 C	
Packaging:	Tray	
Series:	Cyclone V E	
Factory Pack Quantity:	119	

Show Similar

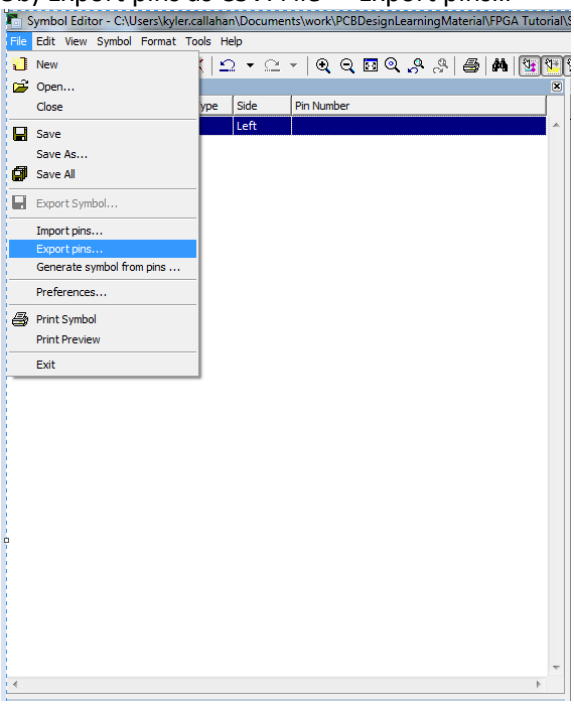


When you're finished your project should look something like this

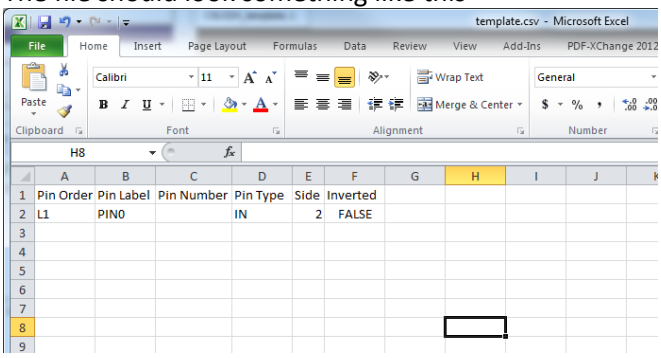
5) Create a CSV template that you will use for all of your symbols

5a) Place a single pin

5b) Export pins as CSV: File -> Export pins…

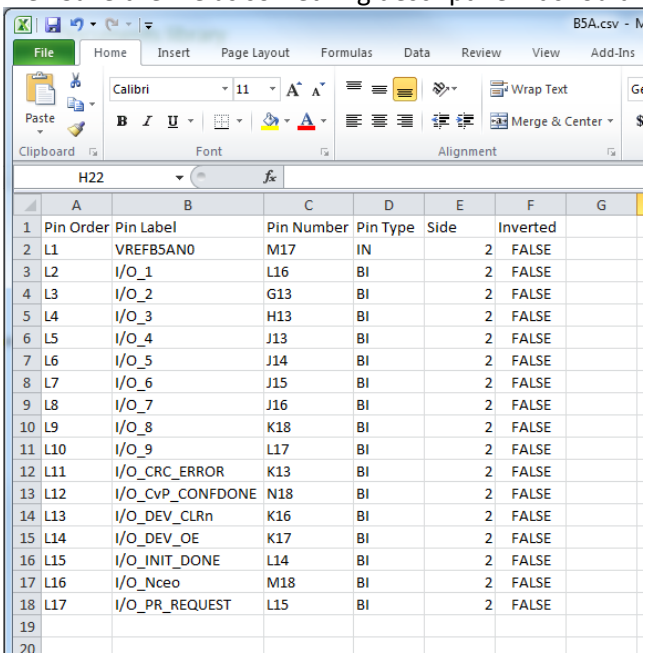


The file should look something like this



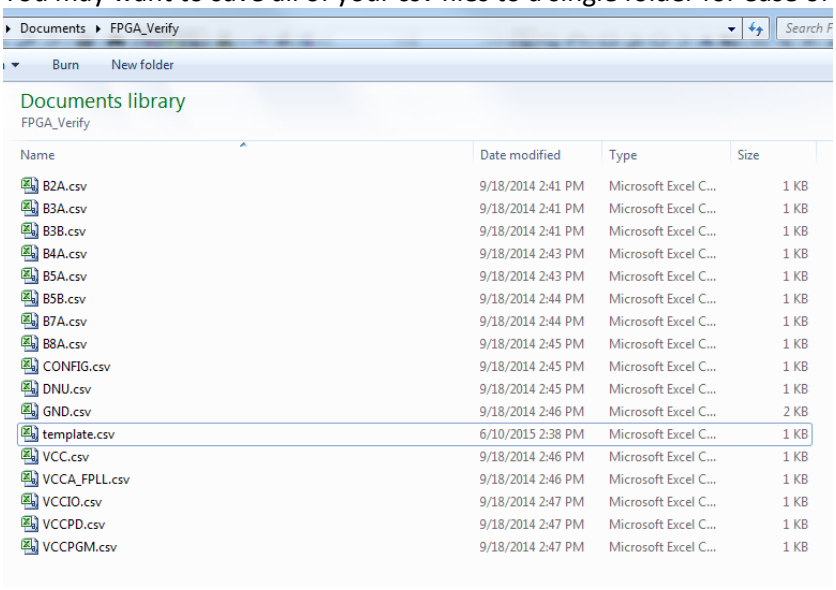
- Pin Order (Accepted inputs: T#,B#,L#,R#)
  - The order that the pins will be in in the *pins* field with respect to it's orientation
- Pin Label
  - The label on the pin, make this descriptive. No two labels can match.
- Pin Number
  - The pin number. This is the number used by PADS to assign pins
- Pin Type (Accepted inputs: IN,OUT,BI,ANALOG,OCL,OEM,TRI,POWER,GROUND,TERMINAL)
  - This field is functionally insignificant to what we are doing right now but you may set the values if you wish, otherwise just leave them as "BI"
- Side (Accepted inputs: 0 = top, 1 = bottom, 2 = left, 3 = right)
  - This is the side that the pin is located on. It should match the corresponding Pin Order
- Inverted
  - Leave FALSE

6) Next edit the file to match the name and pin number from the sheets you made in your other Excel file. Save the file as something descriptive. It should look something like this when you're done.



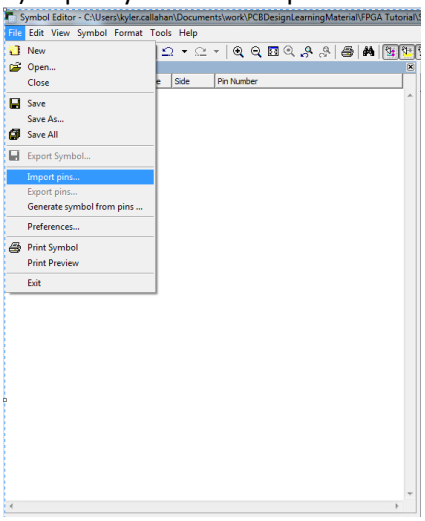
	A	B	C	D	E	F	G
1	Pin Order	Pin Label	Pin Number	Pin Type	Side	Inverted	
2	L1	VREFB5AN0	M17	IN	2	FALSE	
3	L2	I/O_1	L16	BI	2	FALSE	
4	L3	I/O_2	G13	BI	2	FALSE	
5	L4	I/O_3	H13	BI	2	FALSE	
6	L5	I/O_4	J13	BI	2	FALSE	
7	L6	I/O_5	J14	BI	2	FALSE	
8	L7	I/O_6	J15	BI	2	FALSE	
9	L8	I/O_7	J16	BI	2	FALSE	
10	L9	I/O_8	K18	BI	2	FALSE	
11	L10	I/O_9	L17	BI	2	FALSE	
12	L11	I/O_CRC_ERROR	K13	BI	2	FALSE	
13	L12	I/O_CvP_CONFDONE	N18	BI	2	FALSE	
14	L13	I/O_DEV_CLRn	K16	BI	2	FALSE	
15	L14	I/O_DEV_OE	K17	BI	2	FALSE	
16	L15	I/O_INIT_DONE	L14	BI	2	FALSE	
17	L16	I/O_Nceo	M18	BI	2	FALSE	
18	L17	I/O_PR_REQUEST	L15	BI	2	FALSE	
19							
20							

You may want to save all of your csv files to a single folder for ease of use and re-use.



Be sure to triple check that each pin number matches its appropriate bank and pin name/function. Making a mistake here be very costly in the future.

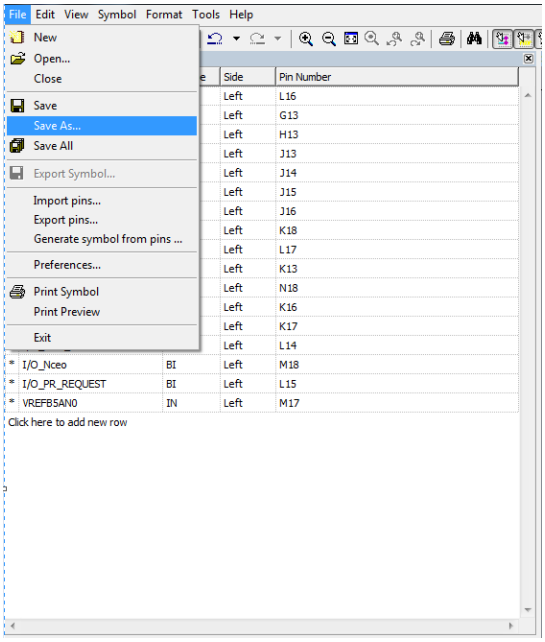
7) Import your CSV's as pins. File -> Import Pins

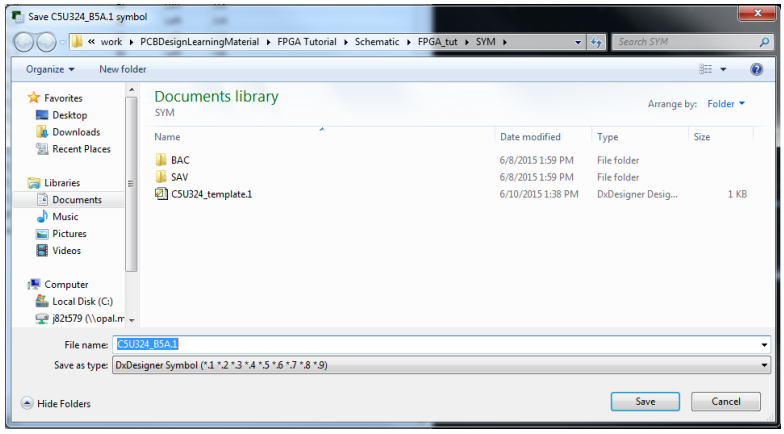


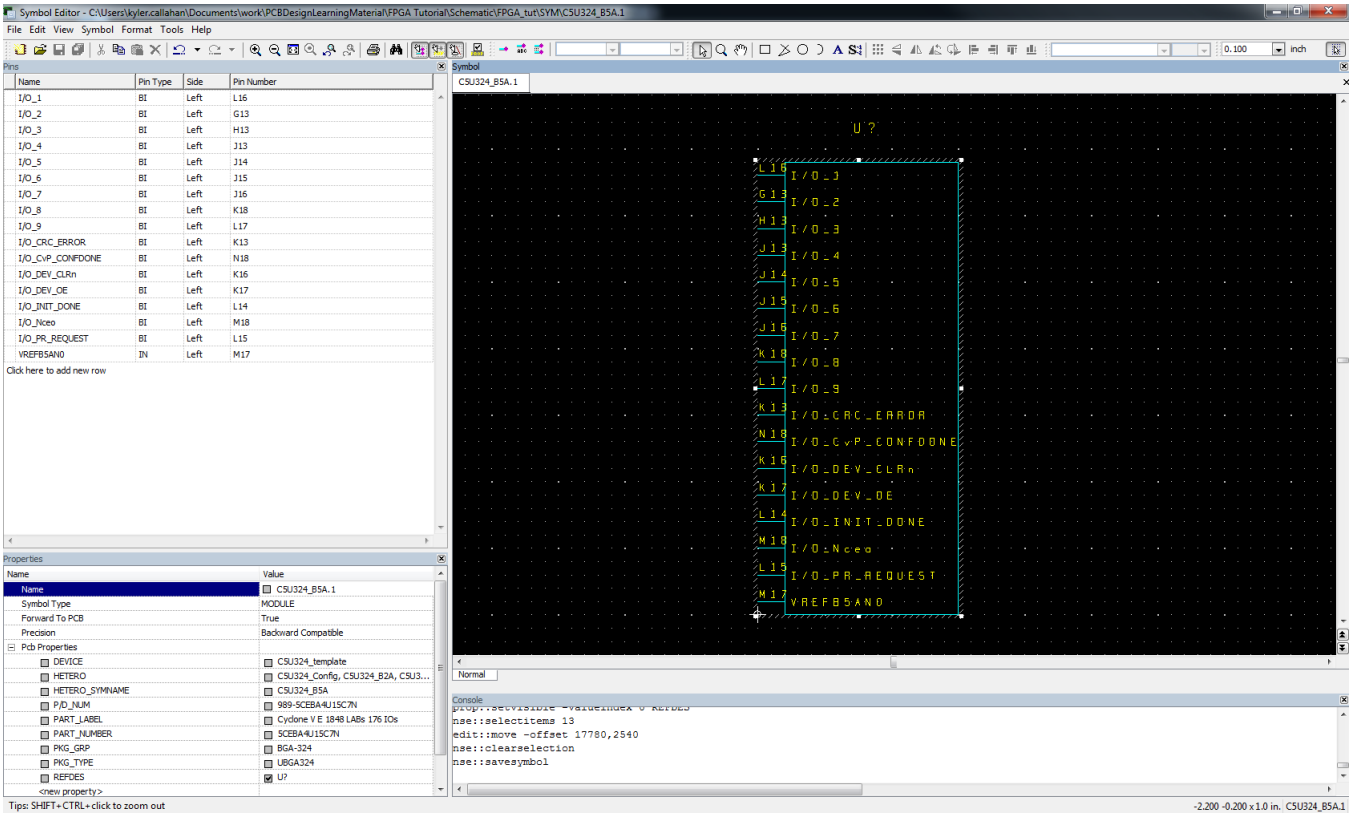
8) Edit Symbol Name, and HETERO_SYMNAME to match the symbol this is associated with and place the pins. Then **SAVE AS A NEW PART!**

Important note!

Do *not* press "save" or ctrl+s when editing from your template. DxDesigner will save the wrong name to the wrong place and mess up your template and whatever symbol you were trying to make. Make sure to press "Save as" instead.



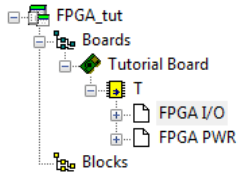


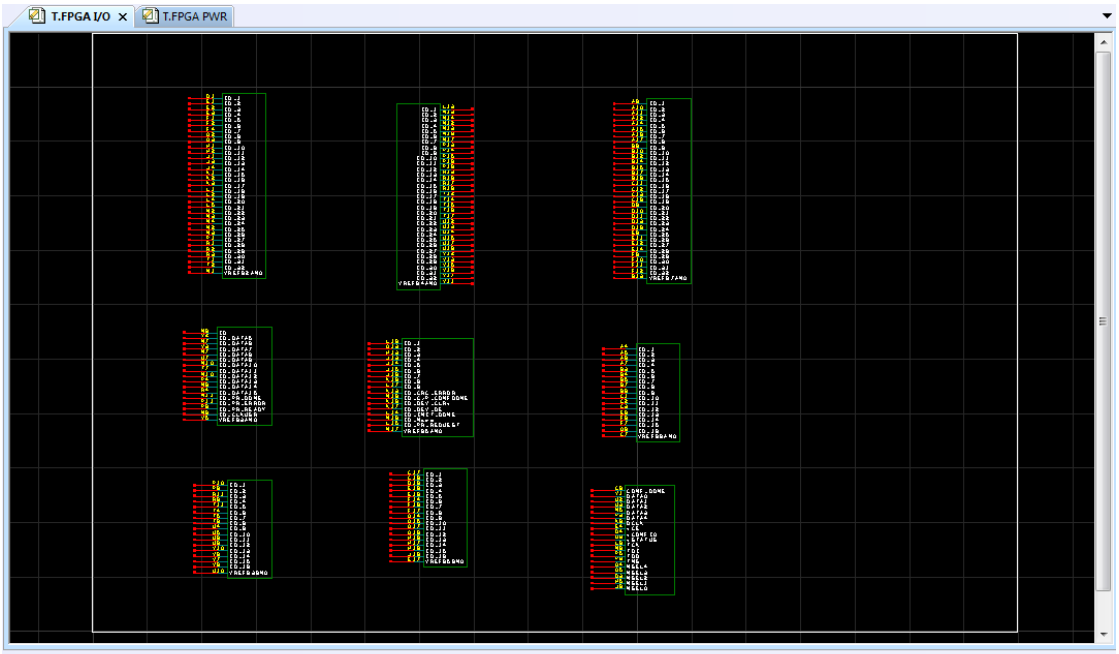


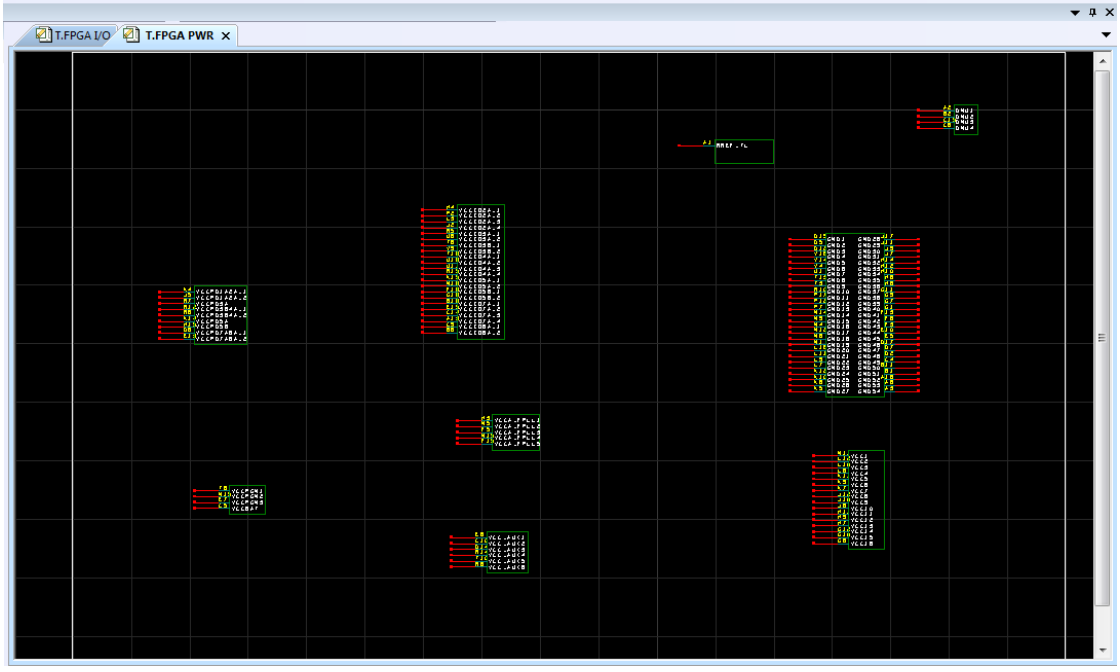
Your symbol should look something like that when you're finished.

8) Reload your template file and repeat step 7 as needed.

9) After you have created all of your symbols you need to place them in a schematic in DxDesigner. If you have a lot of symbols it is recommended to split them across multiple sheets by function.

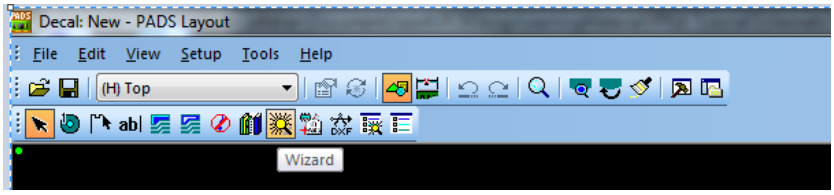






10) Next open PADS Layout and make a new library part: File -> Library -> Decal -> New

And launch the decal wizard



11) Go to the tab that matches your package type and fill out the necessary information. For this example we are making a BGA/PGA. If the info is not available in the data sheet there may be a separate document containing package information.

Package Information	
Description	**Specification**
Ordering Code Reference	U
Package Acronym	UBGA
Substrate Material	BT
Solder ball composition	Regular: 63Sn:37Pb (Typ.) Pb-free: Sn:3Ag:0.5Cu (Typ.)
JEDEC Outline Reference	MO-216 Variation: BAK-1
Lead Coplanarity	0.005 inch (0.12 mm)
Weight	0.8 g (Typ.)
Moisture Sensitivity Level	Printed on moisture barrier bag

Package Outline Dimension Table			
Symbol	**Millimeters**		
	Min.	**Nom.**	**Max.**
A	1.20	1.35	1.50
A1	0.35	0.40	0.45
A2	0.75	0.95	1.15
A3	0.65	0.70	0.75
D	15.00 BSC		
E	15.00 BSC		
b	0.40	0.50	0.60
e	0.80 BSC		

Package Outline

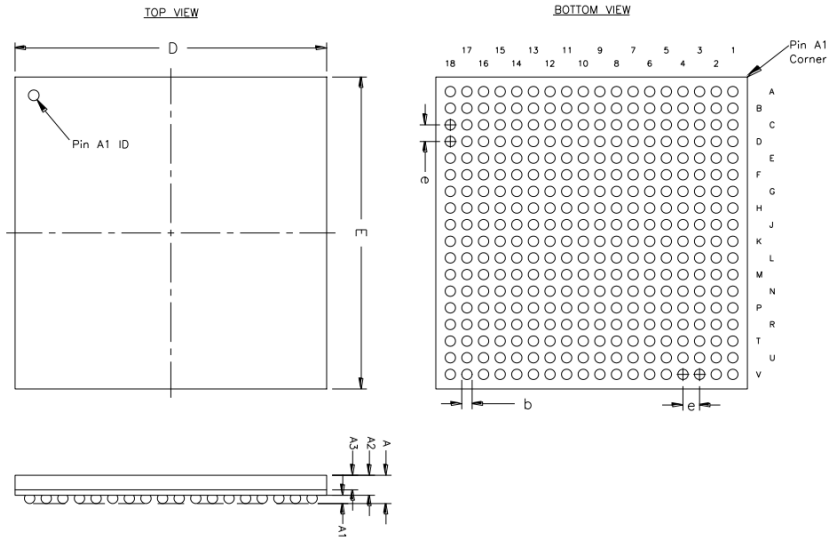
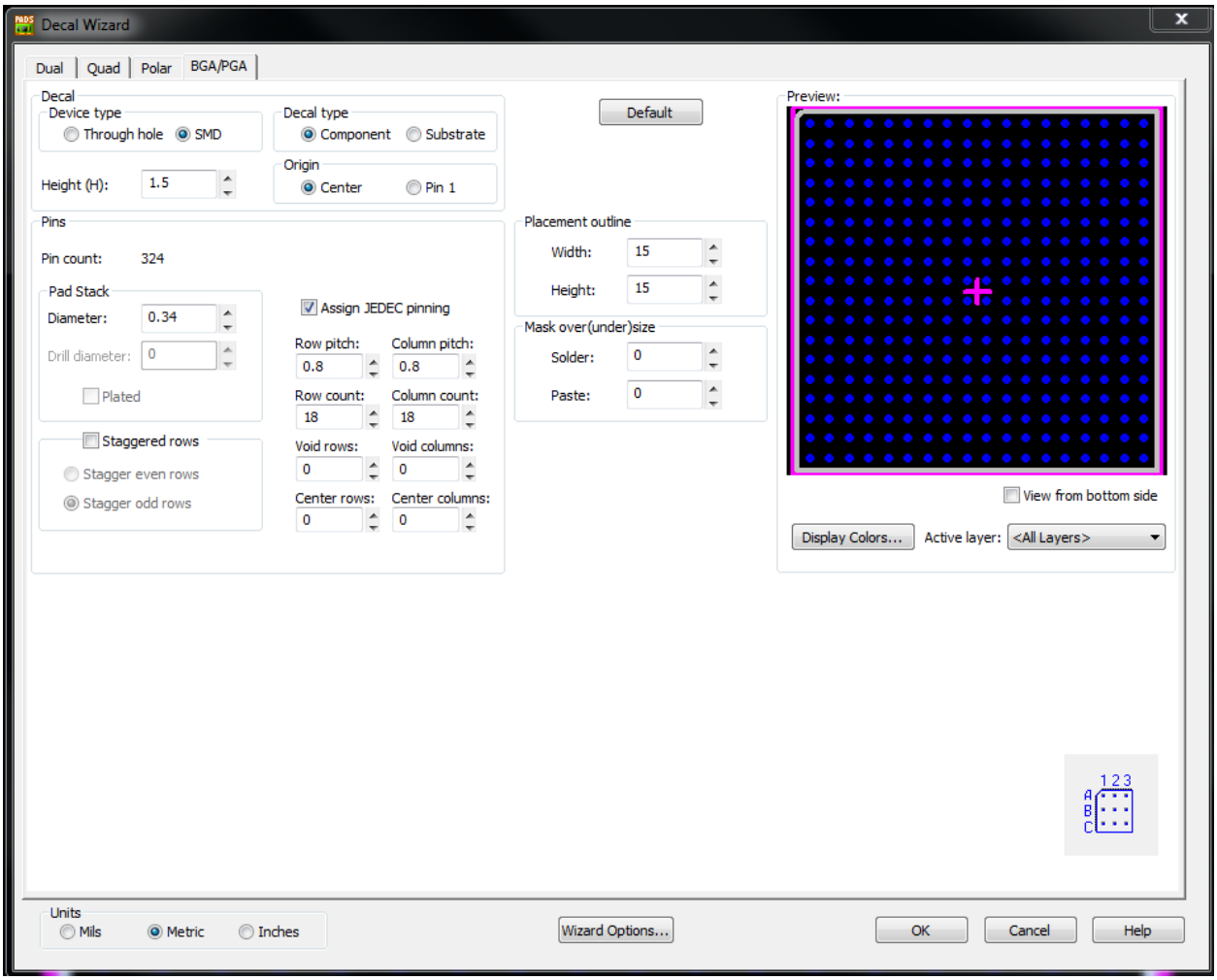
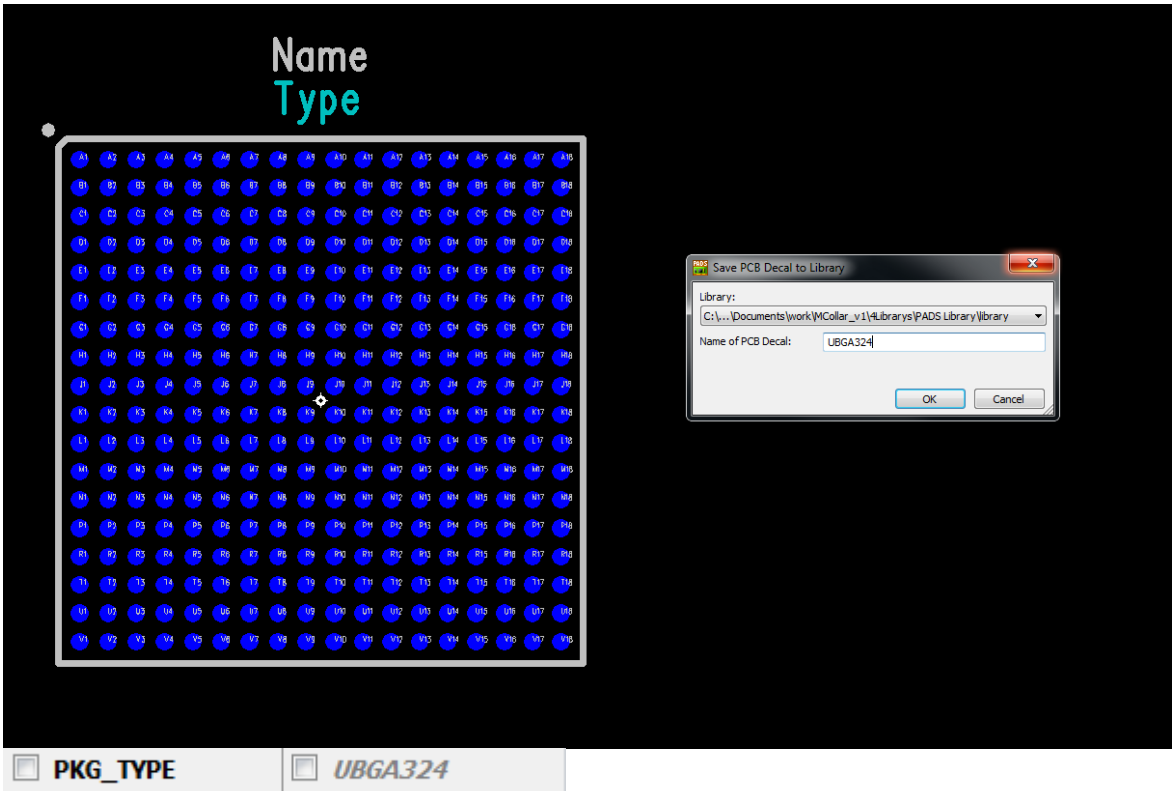


Table 3. Recommended Pad Sizes for SMD and NSMD Pads				
BGA Pad Pitch	**BGA Pad Opening (A) (mm)**	**Solder Ball Diameter (B) (mm)**	**Recommended SMD Pad Size (mm)**	**Recommended NSMD Pad Size (mm)**
1.27 mm (Plastic Ball Grid Array (PBGA))	0.60	0.75	0.60	0.51
1.27 mm (Super Ball Grid Array (SBGA))	0.60	0.75	0.60	0.51
1.27 mm (Tape Ball Grid Array (TBGA))	0.60	0.75	0.60	0.51
1.27 mm (flip-chip) *(1)*	0.65	0.75	0.65	0.55
1.00 mm (wirebond) *(1)*	0.45	0.63	0.45	0.38
1.00 mm (flip-chip) *(1)*	0.55	0.63	0.55	0.47
1.00 mm (flip-chip) *(1)* APEX 20KE	0.60	0.65	0.60	0.51
0.80 mm UBGA (BT Substrate)	0.4	0.55	0.4	0.34
0.80 mm UBGA (EPC16U88)	0.4	0.45	0.4	0.34
0.50 mm MBGA	0.3	0.3	0.27	0.26

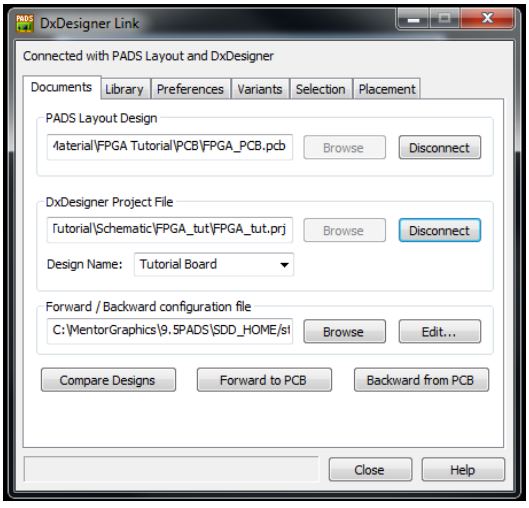
Using the table taken from *Designing With High-Density BGA Packages for Altera Devices* we can determine what the landing pad sizes are going to be. For this example we are using a 0.8 UBGA following the Recommended NSMD Pad Size. For further explanation please refer to the app note.



12) Make sure the pin numbering matches that of the data sheet, add a pin 1 indicator, and save the decal exactly what you put in the PKG_TYPE field of the DxDesigner decal.



13) Forward your design to your PADS project. If you did everything correctly then your part should show up!

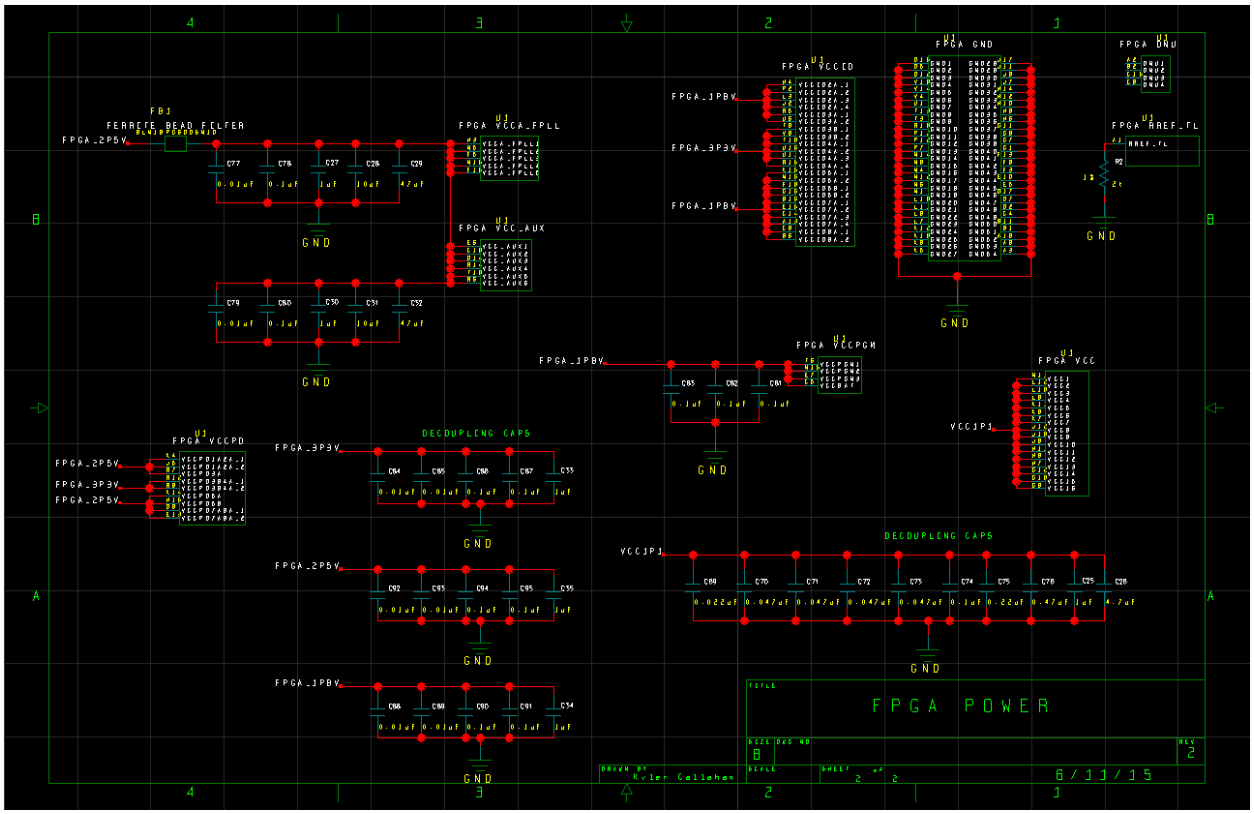


If you get an error forwarding your part common issues are forgetting to add one of the symbols in the hetero field. Check and make sure all parts are placed and that all your HETERO_SYMNAMES matches those in the hetero field.

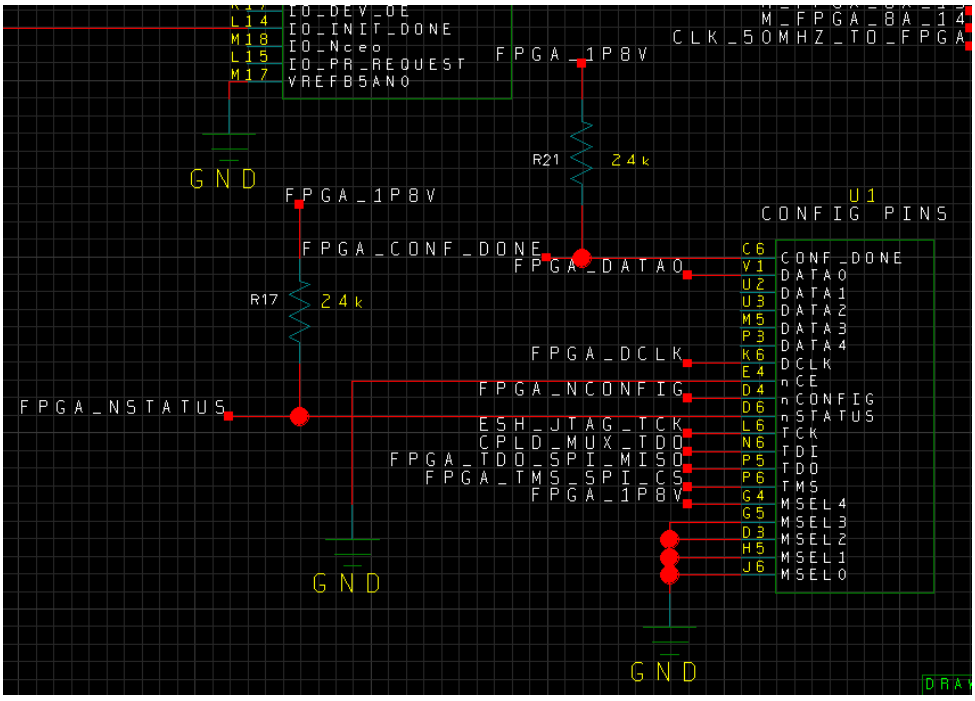
Basic Break-Out Techniques In PADS

This is just a quick guide on getting started routing BGA packages in PADS. Other techniques exist but will not be covered in this guide. For more information on different BGA package routing techniques please refer to the *further reading* section at the end of this tutorial. Additionally, this *PADS Specific* design flow is what I, the author, have found to be best use for quickly achieving high density routing. Other design flows exist and may change based upon your intended application. This design flow is designed to keep your schematic in sync with your layout while accommodating limited routing space, and assumes that the schematic creator and layout designer are working closely together (or the same person).

1) Your first step will be to hook up all your power circuitry according to your data sheet specification.

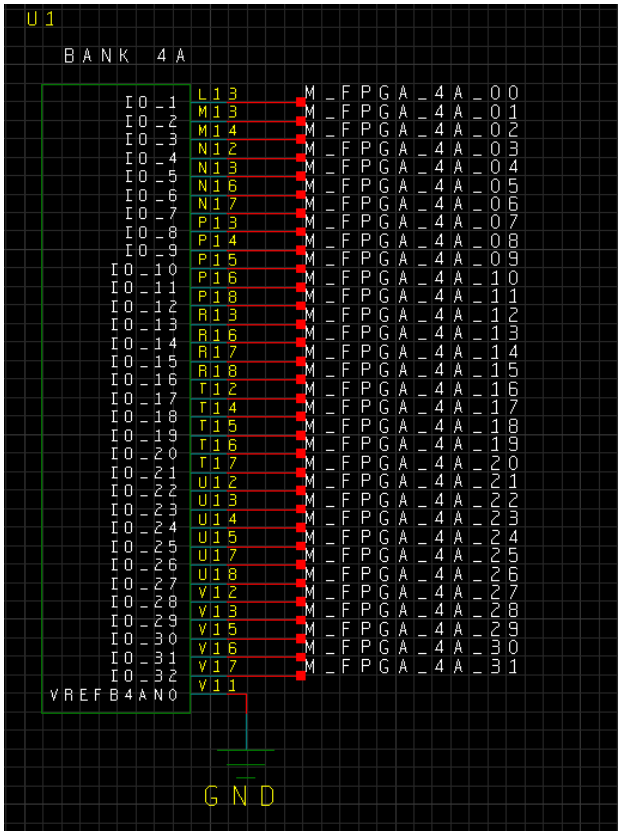


2) Your next step is to hook up all the needed config pins as recommended by the user's manual.

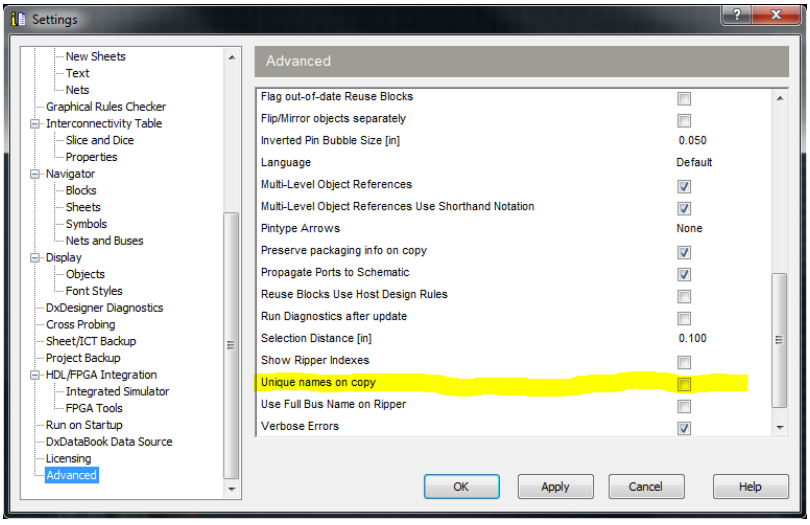


Note: These steps are important for actual design work, but for the sake of the tutorial they may be skipped if you are trying to follow along.

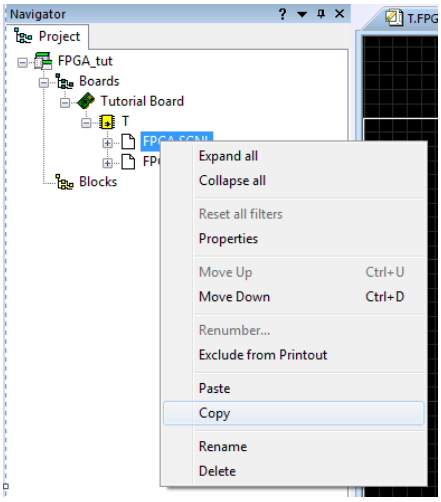
3) The next step is to hook up *all* (even if you don't need all of them) of the I/O pins with a general signal name. Use a name that relates to the Bank that the pin belongs to. Example below. These names will be changed later but will be useful when breaking out the FPGA and hooking up general I/O signals.



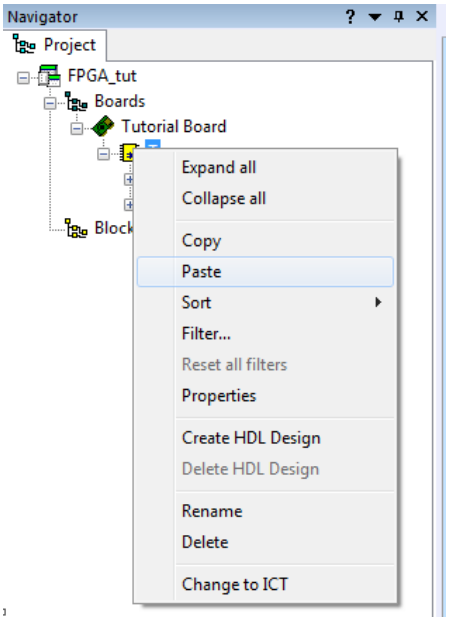
4) Make a copy of your FPGA, signal names and all. To do this make sure that "Unique names on copy" is disabled. In Setup -> Settings -> Advanced -> Unique names on copy. This setting will allow us to copy all of the names and symbols 1 to 1. The downside of this is there will be conflicts with the symbol reference designators that we will need to fix ourselves.



Make a copy of the sheet



Paste the sheet



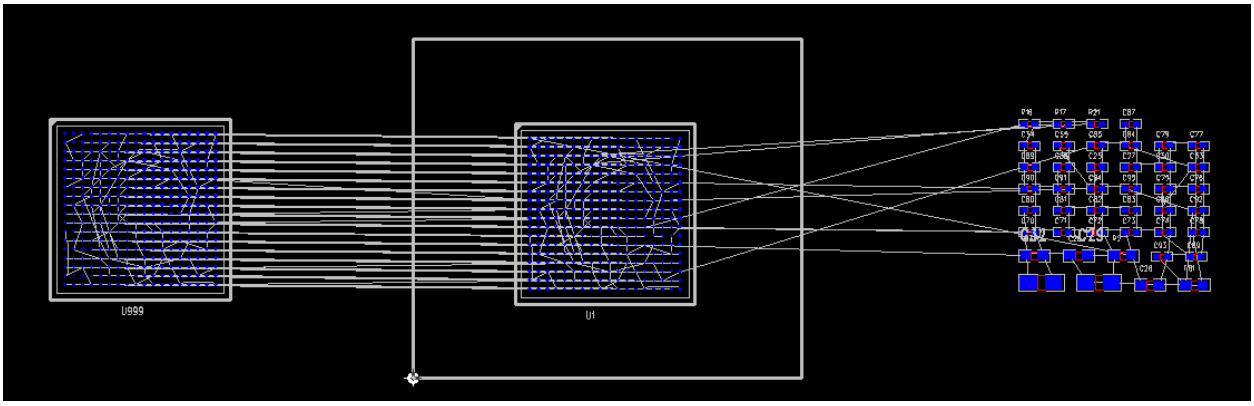
And it should give you an exact copy of that sheet

5) Change your copy FPGA reference designator to something abstract so you know it's your copy device. For this example we will use U999. Make sure you change *all* the reference designators on your copy FPGA, otherwise it will not work properly.

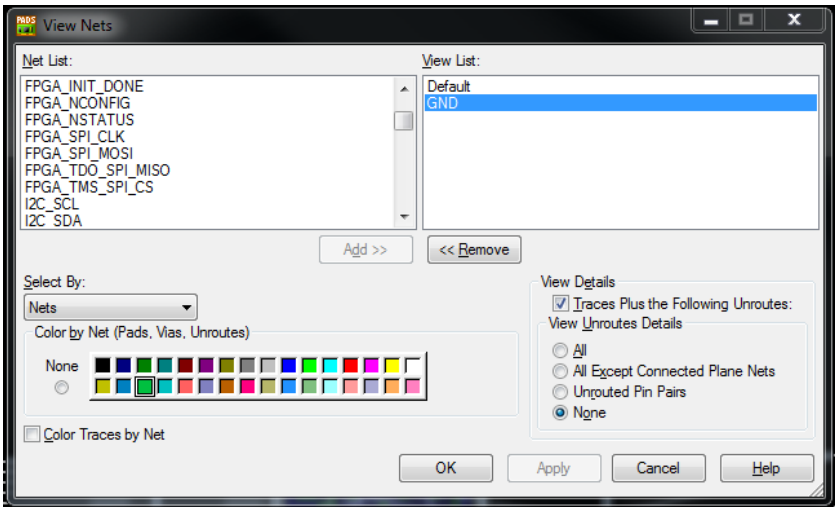
6) Delete the capacitors and resistors connected to U999 but leave the attached signals. We do not want extra unwanted components.

This copy FPGA will allow us to partially hook up our actual FPGA. PADs will not let you route from unconnected pins and it will unconnected signals that only go to a single pin. This is why we are creating a mirror part.

7) Forward your design to your PCB, place your FPGA, move U999 off screen and set your decoupling capacitors off to the side. They will be placed when we have broken out the FPGA.

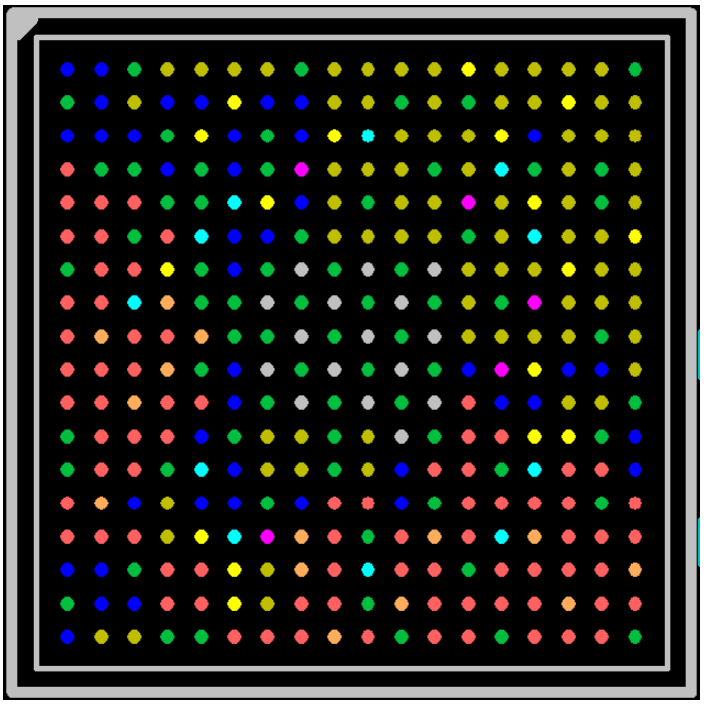


8) PADS allows for nets to be colored and the visibility of the un-routes to be modified. Press ctrl+alt+n and it should open up the *View Nets* dialog menu. Browse to GND on the left, select it and press add. Set *View Unroutes Details* to *None*. Select the color to be a shade of green and leave *Color Traces by Net* unselected.



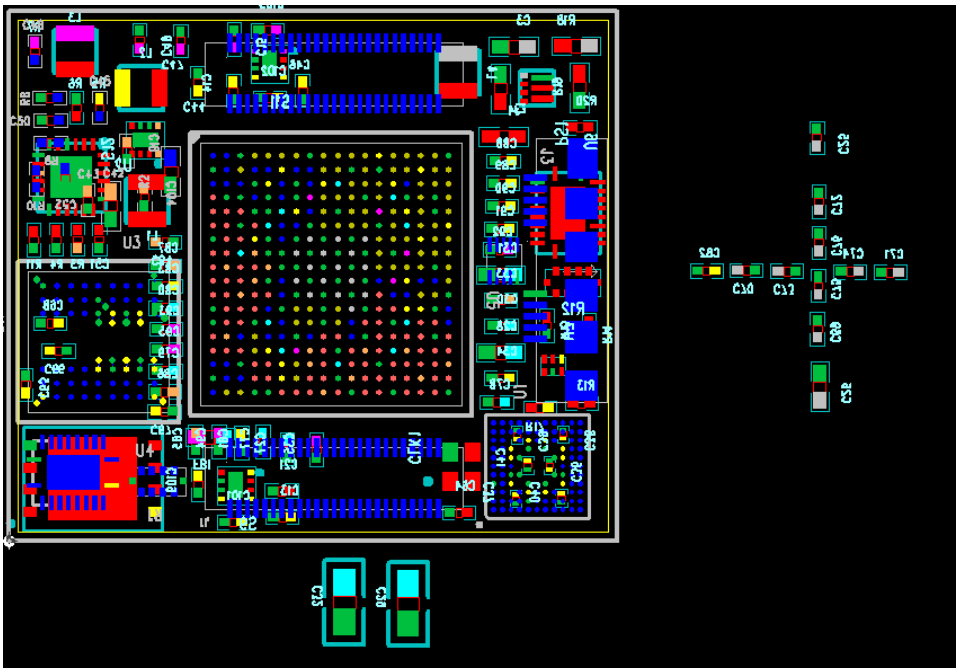
What this is going to do is color all pins associated with the net *GND* green and make their unroutes invisible until you try and route the individual pin. This is useful for nets that have a large number of connections (such as power or ground) so you don't have white lines all over your screen.

9) Repeat step 8 for the different voltages and the I/O pins on the FPGA. Color the I/O pins by the voltage of the bank.



Yellow: 1.8V
Green: GND
Orange: 3.3V
Grey: 1.1V
Maroon: 2.5V
Teal: Filtered 2.5V
Gold: 1.8V I/O
Salmon: 3.3V I/O
Blue: Default/not colored

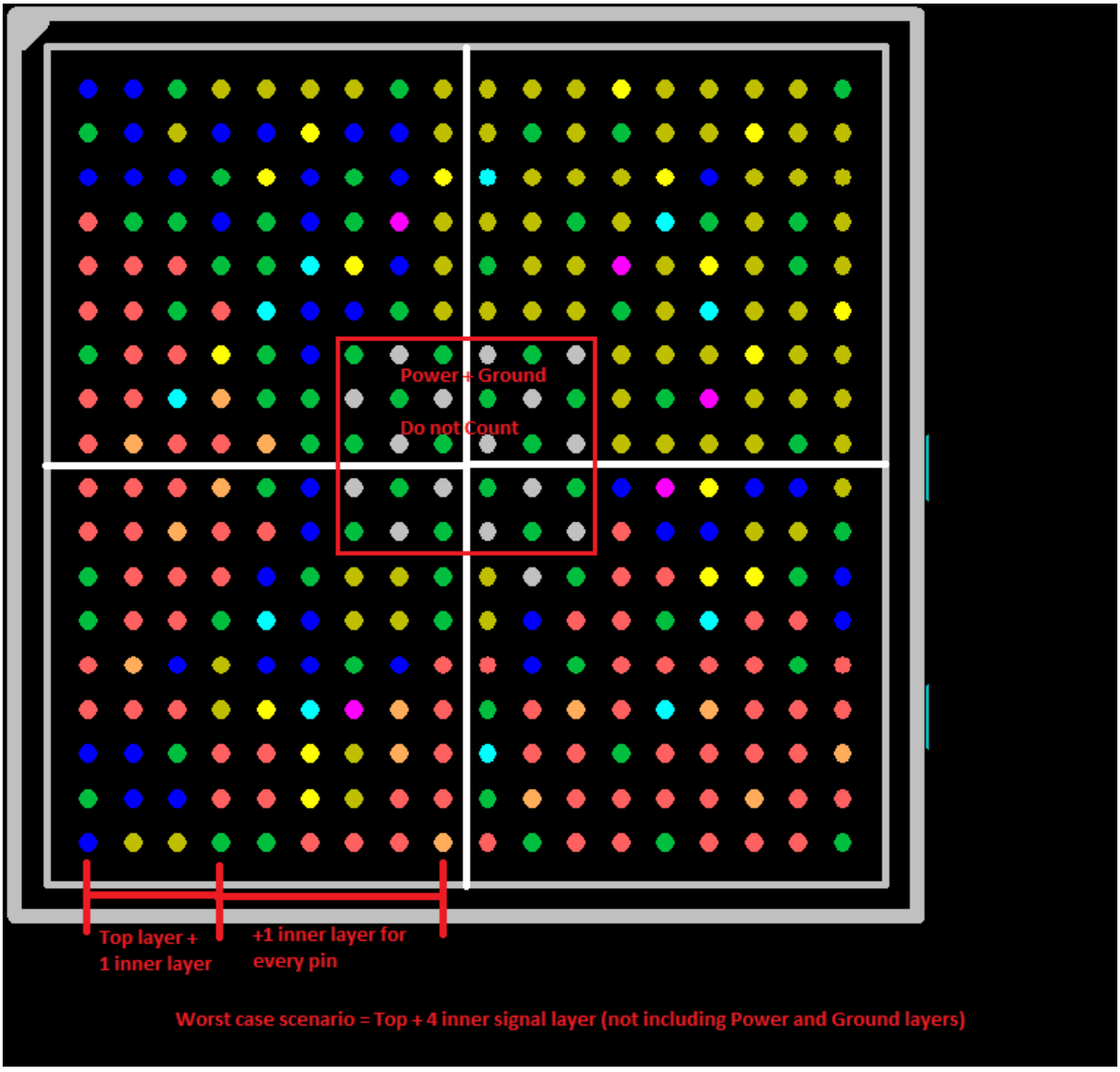
10) Finish your schematic in DxDesigner and place all your parts except for your decoupling capacitors. Don't place any components underneath the FPGA until it is broken out.



Note: It is important to have all your components placed before you route, but if you are following along you may skip step 10

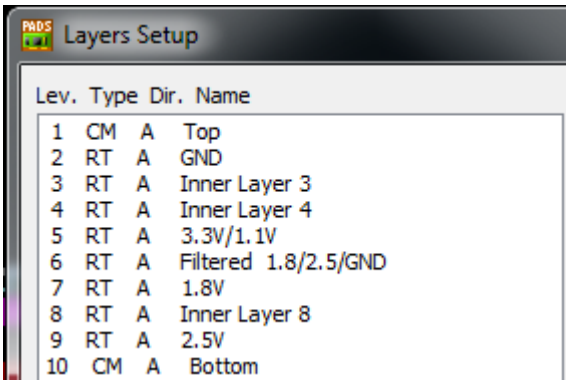
11) Next you need to determine how many layers you are going to need. Fabrication houses make boards in 2 layer sets. You can get an odd number of layers but it typically doesn't cost anything more to have that extra layer to make your number of layers even. When determining number of *signal layers* needed a good rule of thumb is top + 1 signal layer for the first 4 pins and an additional signal layer after that.

Example:



So for this example we would need a minimum of 8 layers to accommodate this FPGA. Top + 4 signal layers + Power + Ground + Bottom. This is not accounting for multiple needed voltages for power either, so more realistically it would be 10 layers to use this package. You don't count the bottom layer as a signal layer as we will be placing decoupling capacitors down there.

Define your layers in PADS. For this example we are using a 10 layer board. We have top and bottom, 3 inner signal layers 1 dedicated GND plane, and 4 split power and ground planes. We don't need all of the I/O pins so we used 3 inner signal layers instead of 4.



When assigning layers try and make is so that a signal layer is next to either a power or ground layer. Do not have a signal layer in-between two other signal layers.

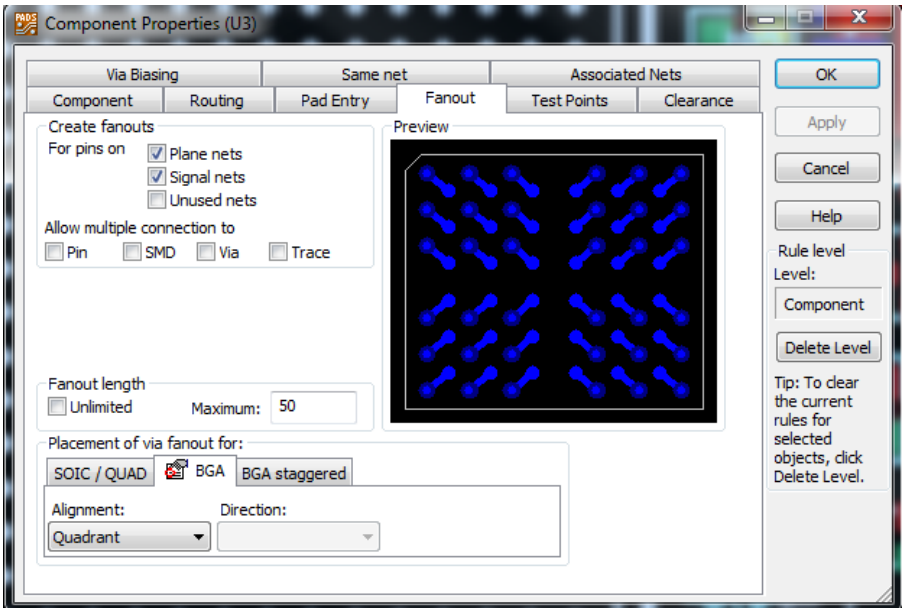
12) Now that you have your layers defined, open your design in PADS Router. We are going to break out the FPGA.

12a) Click on your component and open up the Properties window. Navigate to the *Fanout* tab.

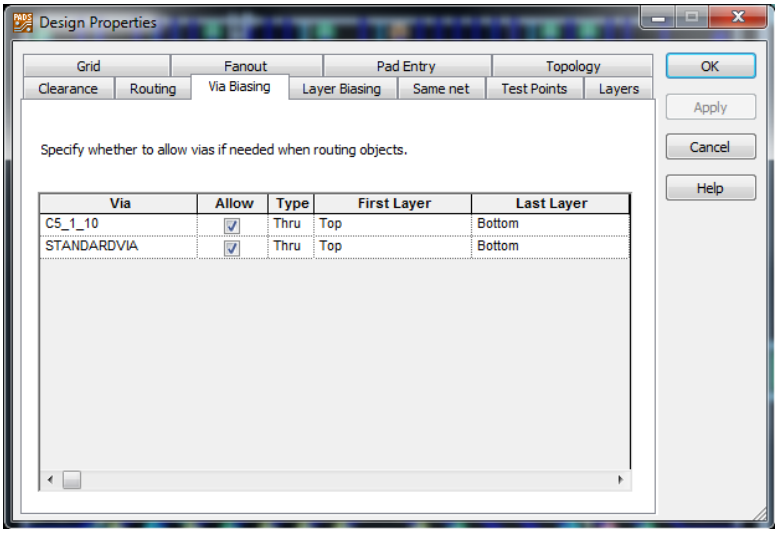
12b) Under the *Create fanouts* section select *Plane nets* and *signal nets.*

12c) Under *Fanout length* set the maximum to something small. If you have the length too long it will automatically attempt to connect the pins to their respective location. We do not want that at the moment.

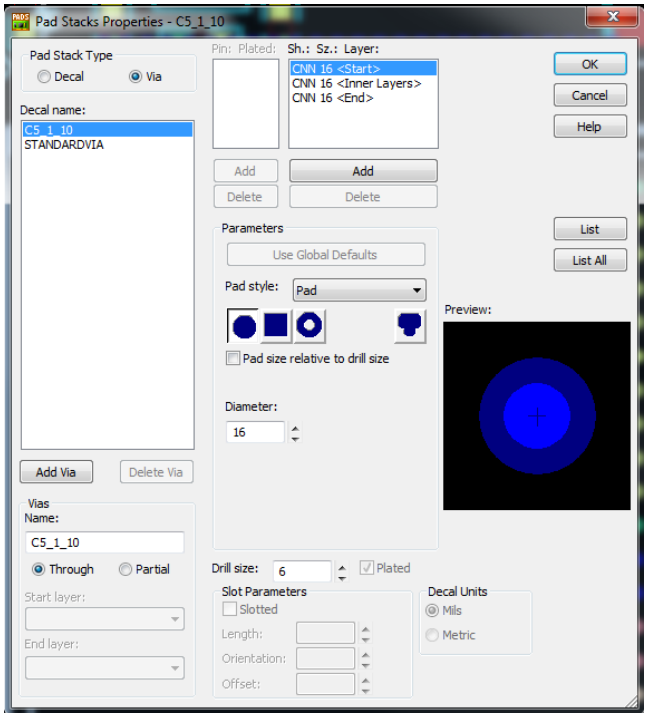
12d) In the *Placement of via fanout for:* select BGA. There are several options here but I have had the best results with *Quadrant*

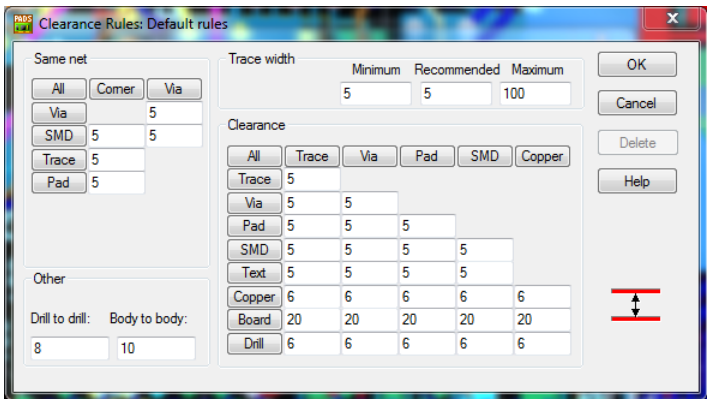


12c) Navigate to the *Via Biasing* tab and ensure that the via you wish to be placed underneath your BGA package is selected and at the top of the list.



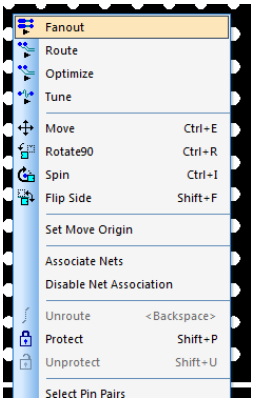
Side note about via sizes. Check with your board fabrication house what their requirements are and what will work underneath your package. For this particular example with 0.8mm spaced pins we will be using 6mil drill with a 5mil annular ring for a total diameter of 16 mils.

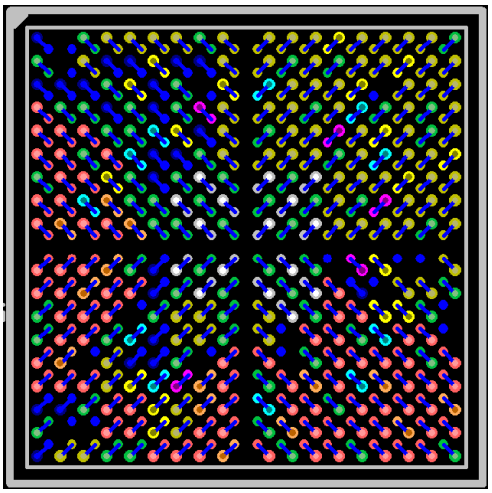




In addition to you your via sized you need to make sure that your clearances are set before you start routing. For this example we will be using 5 mil space/trace.

12d) Right click on your component and click on *fanout*. This should fan out your component

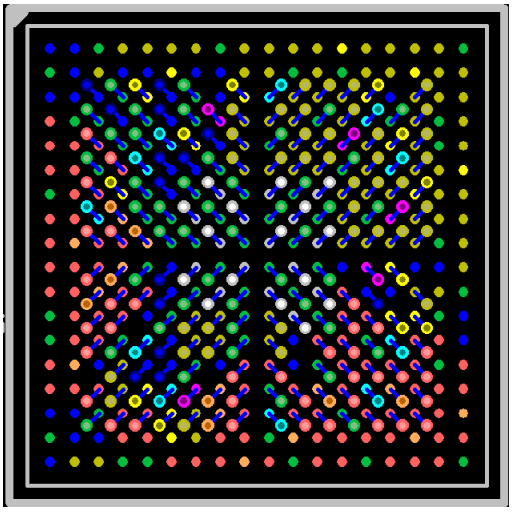




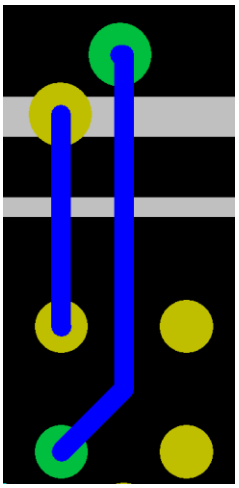
Notice how some pins are not connected. This is because we did not connect them in our schematic as they do not have an associated net.

13) At this point it is ok to place your decoupling capacitors as this is worst case scenario for your FPGA.

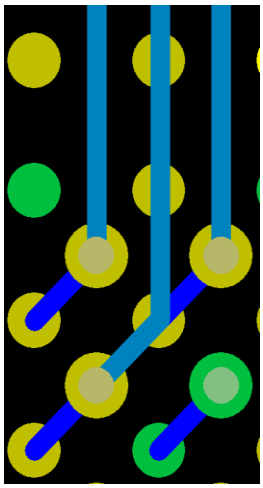
14) Delete all of the routes on the outer 2 pins. We will manually place these ourselves as their current positioning is sub optimal.



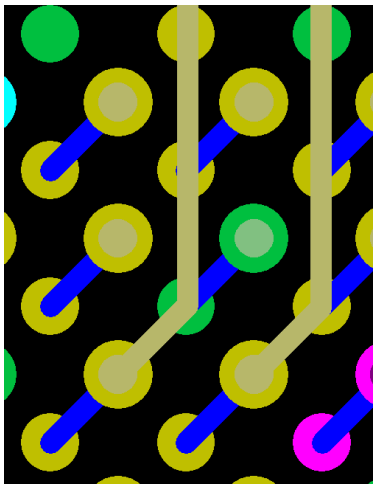
15) Route your traces to the edge of your package. This will allow us to just grab a trace that we need without worry how to break it out first. A basic break out pattern is shown below.



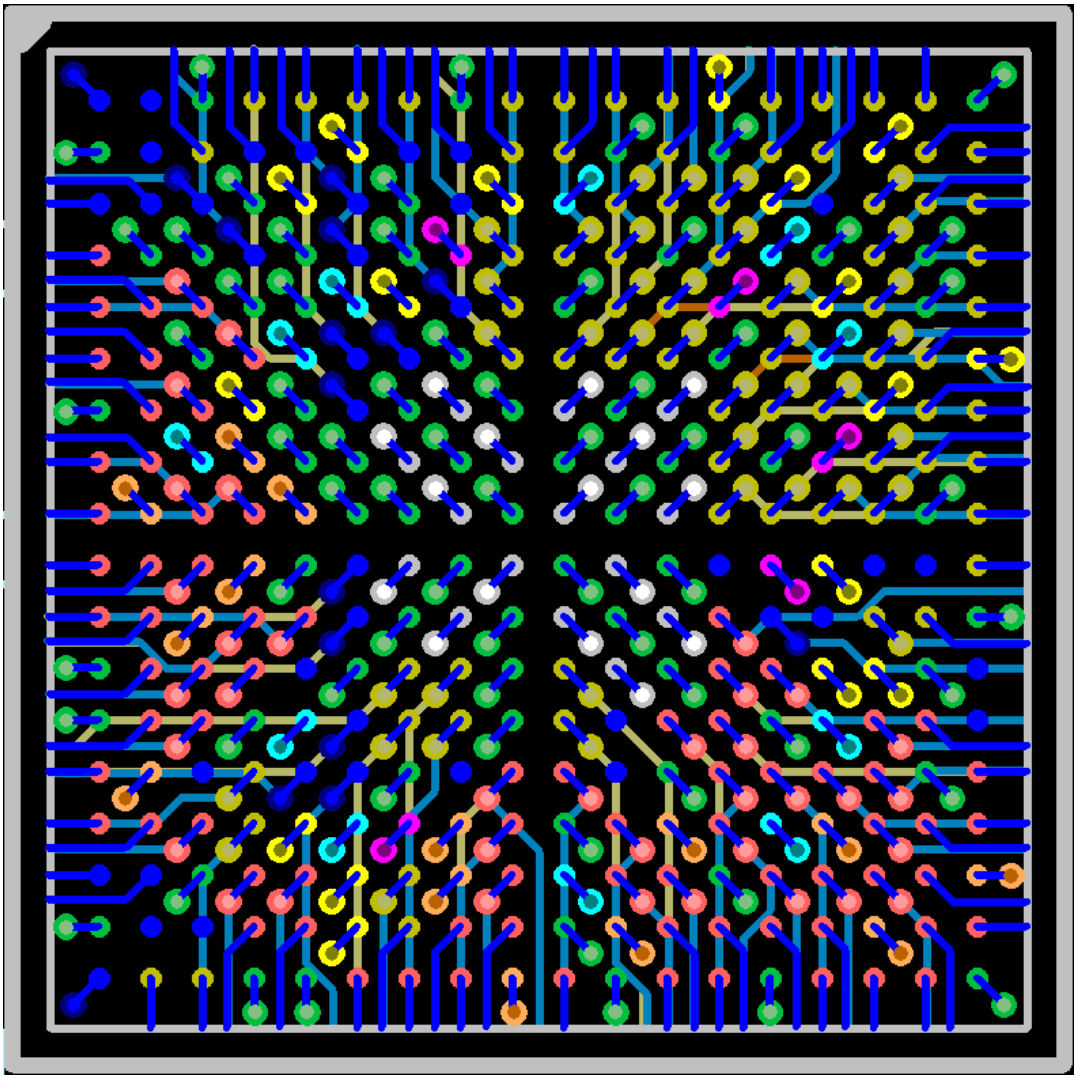
First 2 rows on top layer



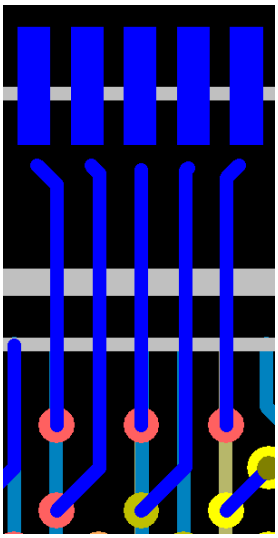
Next 2 rows on an inner layer



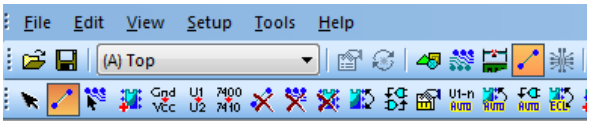
Subsequent rows on inner layers



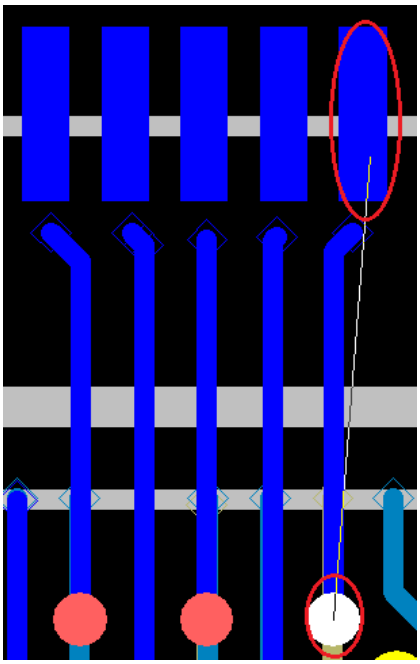
16) We can now begin routing. Since most of our FPGA is hooked up to nothing we need to hook up the pins properly. Go to one of your components that you need to hook up to the FPGA and route the *voltage appropriate* pins from the FPGA to the pins on the device. You won't be able to connect to the pins yet but this will give you an idea of the physical layout so you can hook up the pins after.



17) Open your design back up in Layout and open up the ECO toolbar and click on *add connection*

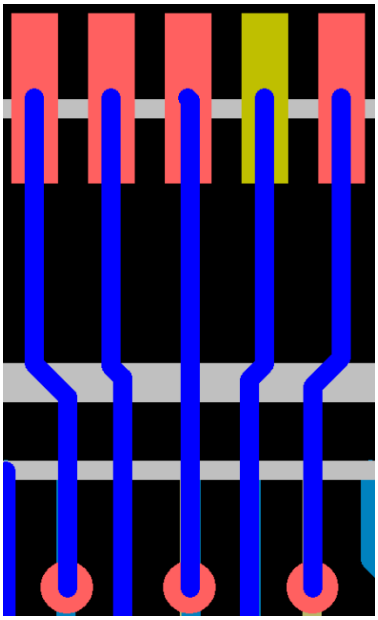


Click on the FPGA pin you want and then follow its trace to the pin you want to connect.



Ideally the non FPGA pin will not be connected anywhere else. If it is not just choose which name you think fits the pin best.

18) Open your design back up in Router and finish hooking up the pins.



19) The final thing to do is to update the pins *in the schematic* with the changes that you made in the PCB. To do this, navigate to the pins that you added connections to and make sure that they are hooked together in the schematic. You may change names if necessary, just make sure that the connections reflect the changes made in the layout. Push your updated schematic back to PADs to check that you made the proper changes.

20) Delete your copy FPGA (U999) an unused signals when you are all done routing.

It is recommended to use *Electroless Nickel Immersion Gold (ENIG)* for your PCB finish when working with BGA and fine-pitch pads. However please research the finishing process to ensure that it is the right finish for your design.

A Brief introduction to Decoupling Capacitors

Decoupling capacitors are important to a device for 2 reasons: They reduce unwanted frequencies in your power and can provide additional power for a brief amount of time if your device has a sudden increase in power consumption. Now ideally your device manufacturer has recommended capacitors for their device or a development board you can reference. For our example above we used a combination of both. Altera had a power calculator that would recommend capacitors based on power needs. We also had a reference design to work off of for our device that helped fill in the gaps from the power calculator.

If your device manufacturer does not have either of these things a good rule of thumb to use is a single 0.1uF X5R decoupling capacitor placed as close to every power pin as physically possible.

Further Reading

Effective BGA Fanout Patterns: Presentation by Mentor Graphics. Requires Flash. Highly Recommended
Via In Pad Guidelines: App Note by Screaming Circuits. Recommended
Introducing Via-in-Pad Blind Via Technology to Any PCB Multilayer Fabricator: Paper. Recommended
Designing With High-Density BGA Packages for Altera Devices: App Note by Altera
PCB Design Guidelines for 0.5mm Package-on-Package Applications Processor, Part I: App Note by TI
PCB Design Guidelines for 0.4mm Package-on-Package (PoP) Packages, Part I: App Note by TI
Choosing and Using Bypass Capacitors: App Note by Intersil
PCB Surface Finishes: Presentation by Viasystems