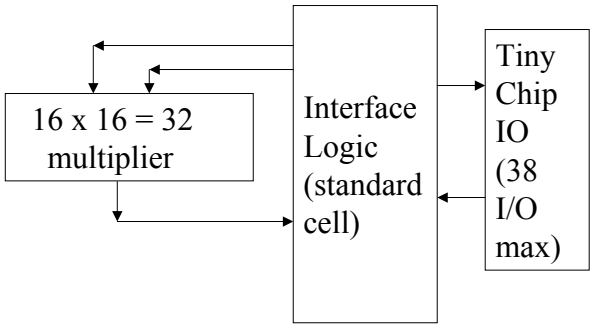
Interface Questions

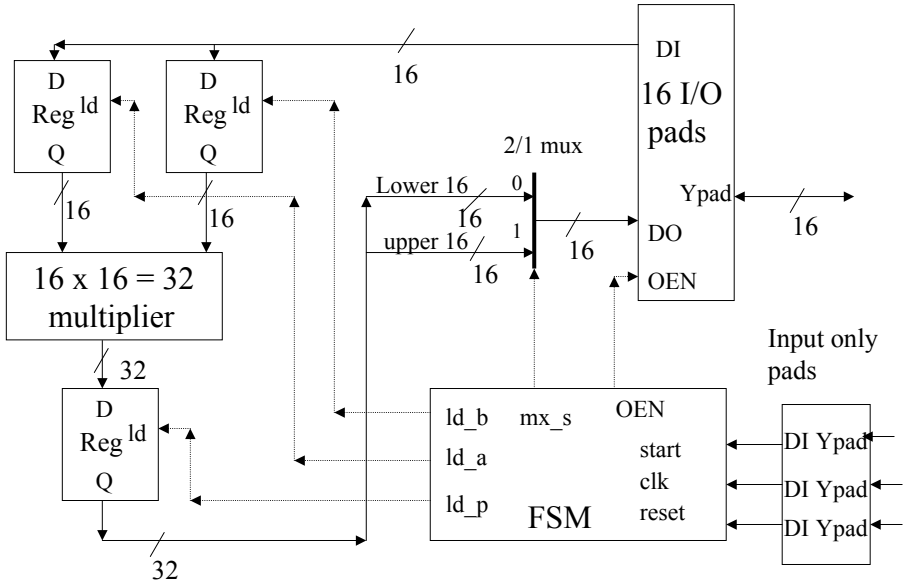
- I have received lots of questions about the details of the interface logic between your design and the limited IO of the tinychip padframe.
- Assume a 16x16 = 32 multiplier



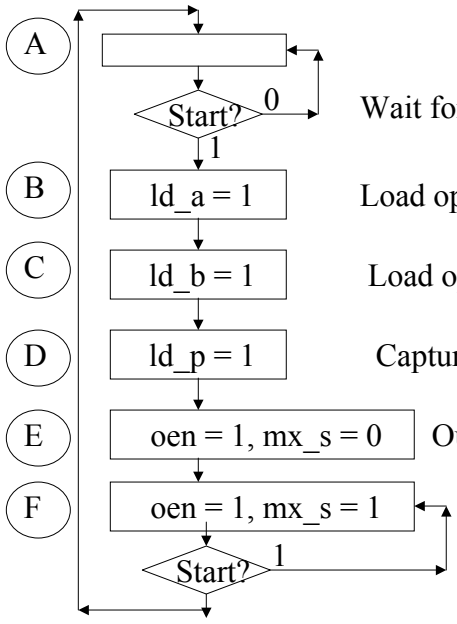
Decisions

- Need to decide how many external data pins you actually want to deal with in testing
- For a 16x16 = 32 multiplier, could have an external 32-bit interface that would be used input the two 16-bit operands, output the 32-bit product
  - Testing: Do you want to connect 32+ bits to a logic analyzer? Another common way to test is to use a small microcontroller to do the testing – not many microcontrollers have have 32+ ports
- Could use a 16-bit external data interface
  - Less pins to connect, but more clocks required to get data in/out – number of clocks to get data in/out is not really an issue, will simply require more states in FSM
- Assume a 16 bit data interface, clock, reset at minimum
  - Do you want handshaking for data exchange?

Block diagram



FSM



Wait for start

Load operand A

Load operand B

Capture Product

Output lower half of product

Output upper half of product, wait for start to go low if still high.

Use a logic analyzer or microcontroller to drive inputs, capture outputs

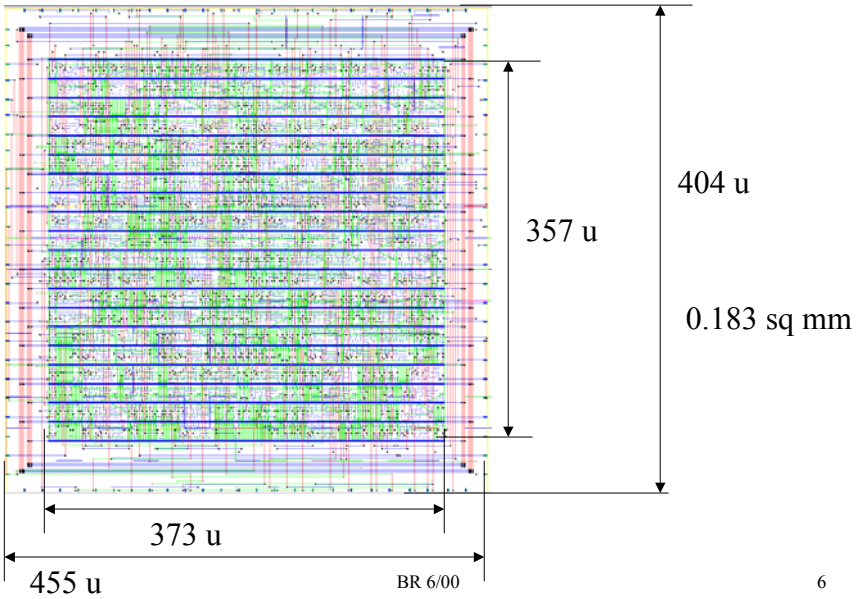
VHDL

The *interface.vhd* file contains the RTL for this interface logic. Interface logic synthesizes to 353 gates (67 DFFs) using jennings.db with synopsys.

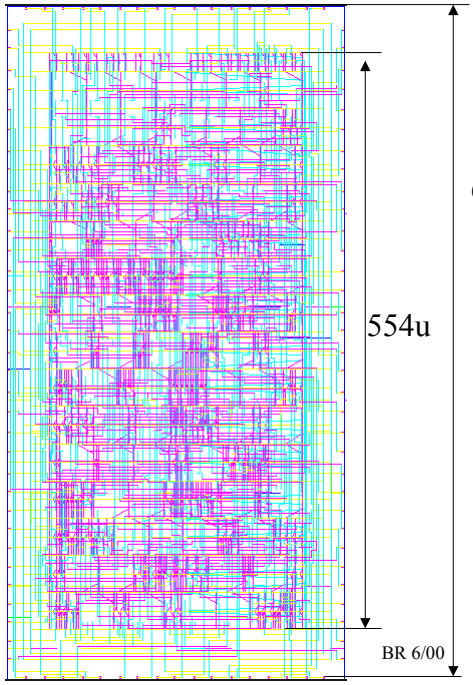
The tutorial library is very simple – only contains a nor2, nand2, DFF, inverter, pullup, pulldown cells so synthesized designs will have high cell counts.

If you want to lower cell counts and improve area, add more cells to the library.

Layout (1.0 x 1.0 Aspect Ratio)



Layout 0.5 Aspect Ratio



X dim (inner) 250u

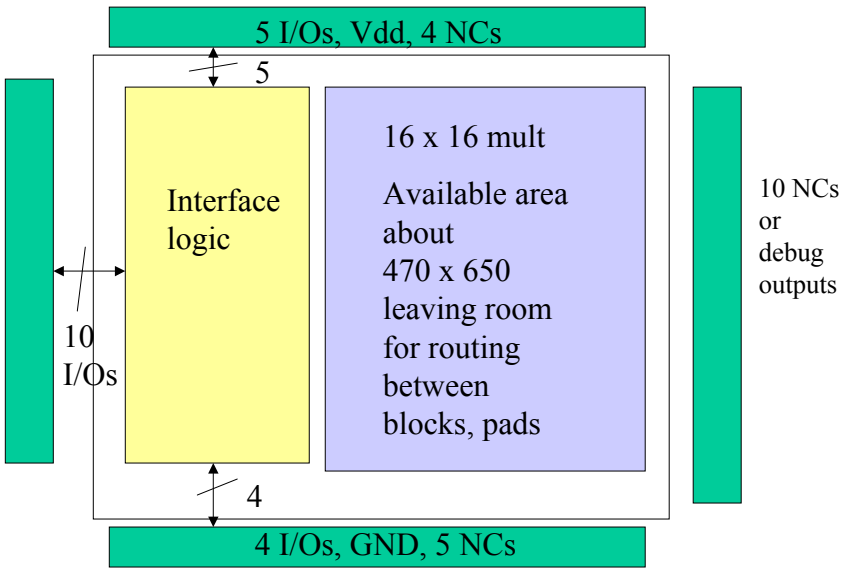
X dim (outer) 330u

0.215 sq mm

Need to tailor your aspect ratio to your floorplan

Floor Plan

900u x 900u core, 10 IO pins per side



This design needs 19 I/O

Standard Cell blocks - pins

- The tutorial has you use the 'random' pin assignment when creating a standard cell block (Place→ IOs step)
  - This will bring the pins (I/Os of the block) to random sides
- You can use a constrained IO file to bring pins out to a particular side
  - Eg., I would bring the interface pins for the multiplier block to the right side of the standard cell layout, and the pins needed for the pads to the top, bottom, right side of the standard cell layout.
  - Look at the help documentation for the required format for the IO file.

Needed Area for Multiplier?

The basic multiplier cell is a FA + AND gate. The complexity is about the same as a DFF.

Assume the cell area is about the same the DFF (33u x 36u).

For a CSA multiplier, need a 16 x 15 array (528 u x 540).

Also need room for the Carry-LookAhead adder at the end of the array.

Looks to be a tight fit. The numbers given in the tutorial for the standard cell floorplan concerning the 'block halo' around the layout as well as the row utilization can be adjusted to produce a tighter layout, or other cells added to the library to reduce the total number of cells in the layout.

The message from this is that you will need be concerned with area.