**California Institute of Technology**
**Department of Computer Science**
**Computer Architecture**

CS184b, Spring 2005 Assignment P4: Switch Blocks Monday, April 25

---

**Due:** Monday, May 2, 9:00AM

**Goals:**

- Build complete chip network
- Complete Area and Timing Model

**Collaboration:** This assignment is a group assignment. Each students should have primary responsibility for one of the 4 network cases and provide the data and writeup for that design. Nonetheless, the answers to all 4 cases should make similar assumptions and should be informed by lessons learned across all four designs.

**Team:** The class makes up 1 team = {hbarnor, nmehta, nachiket, mwilson}

**Target**: All designs should target a Virtex2-6000-4.

**Q-depth**: Assume $Q = 16$.

**Turnin:** We have created a directory: `/scratch/ic/cs184b/project/p4`. Please put the files requested below in that directory (but keep your a master copy elsewhere, that directory is not backed up). Create a master file `p4.html` in that directory which points to all the constituent files and provides any necessary explanations. An html template is at:

- `/cs/courses/cs184/spring2005/assign/p4/p4.html`

Email `mdel@cs.caltech.edu` when the materials are in the directory and ready for review.

**Tasks:**

1. Write an equation for number of LUTs required per chip to support the following networks as a function of the channel width ($W$) and the number of PEs per chip ($N$):
   - Packet-Switched ConceptNet (assume PE requires 512 LUTs)
   - Packet-Switched SMVM (PE provided)
   - Time-Multiplexed ConceptNet (assume PE requires 512 LUTs) [also a function of TM-cycles $T$]
   - Time-Multiplexed SMVM (PE provided) [also a function of TM-cycles $T$]

2. Guided by your answer above, assemble the largest network ($N$) you can fit on the XC2V-6000 with $W = 2$ for each of the SMVM cases. Add pipelining (EQRs) for a 140MHz clock frequency for your interconnect. If the PE does not perform at 140MHz the frequency of the interconnect can be tested by replacing the PE VHDL operator with the identity operator as a dummy. Our goal here is that the network should not be the performance bottleneck.
   - Packet-Switched SMVM
   - Time-Multiplexed SMVM ($T = 256$)

   Note that we provide you with IO locations for the FPGA pins.

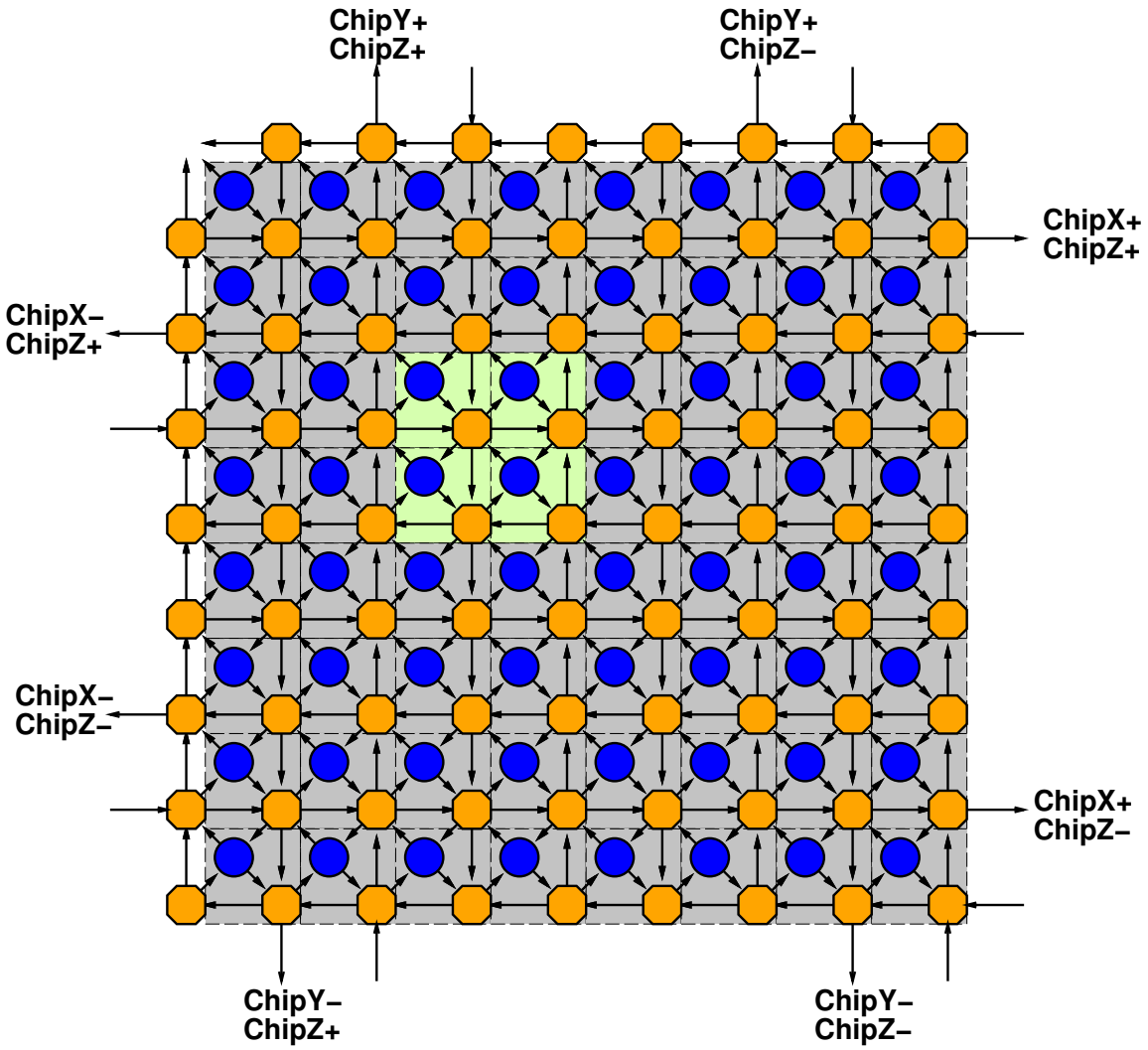
3. Based on your experience in the previous problem, describe the feasibility contour for the parameter set which will fit onto a single XC2V-6000 for each of the 4 cases listed in problem one.
   - Packet-Switched ConceptNet (contour is in $(N, W)$ space)
   - Packet-Switched SMVM (contour is in $(N, W)$ space)
   - Time-Multiplexed ConceptNet (contour is in $(N, W, T)$ space)
   - Time-Multiplexed SMVM (contour is in $(N, W, T)$ space)

   Your solution to problem 2 should have given you insight into how full you can actually pack the device.

4. For each case, write an equation to estimate the number of clock cycles it takes to route between two points ($X_1$,$Y_1$) and ($X_2$,$Y_2$) on a single chip.

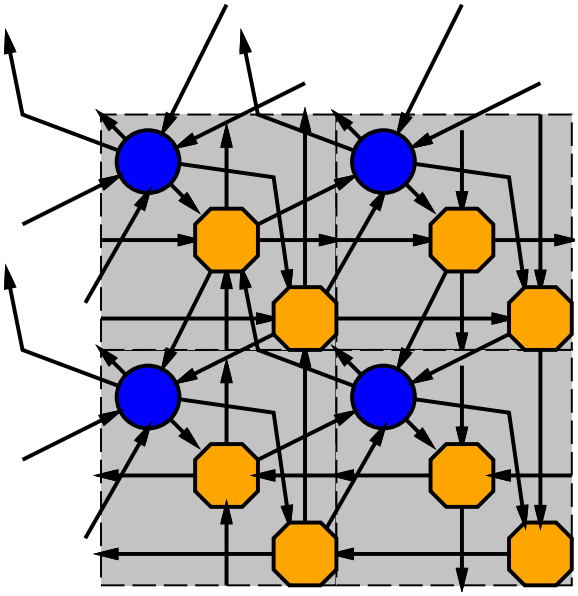
# Network Topology

The basic network topology is shown below. The off-chip IO designations have changed from the version shown on P3. Otherwise, the network remains the same.



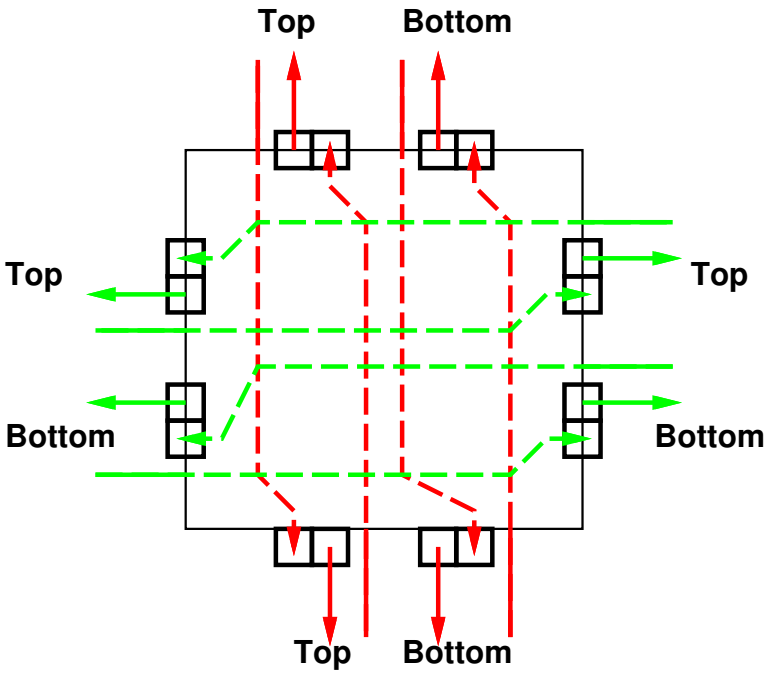

You may choose any rectangle ($j \times k = N$) for your designs as long as $j \leq 8$ and $k \leq 8$. Always have switched routers on all 4 sides of the design (so there are always $W \times (j+1) \times (k+1)$ switch blocks).

Wider channels ($W > 2$) are accommodated with multiple tracks as shown on P3 and repeated below.



Chip I/O's remain on the chip edges and sides as shown. The channel directions alternate between route tracks. Consequently, the direction of the edge routing track will differ depending on the number of PEs in a dimension. When there are an even number of PEs in a dimension, there will be an odd number of channels and the two channels on opposite sides will be in the same direction; when there are an odd number of PEs in a dimension, there will be an even number of channels and the two channels on opposite sides will be in opposite directions.

# Chip IO




The IO interface in VHDL is demonstrated in:

- `/cs/courses/cs184/spring2005/assign/p4/DishoomApp.vhd`

There is an input port and an output port for each of eight combinations of (ChipX+ or ChipX-), (ChipY+ or ChipY-), and (ChipZ+ or ChipZ-).

Keep the VHDL entity name and the port names the same for your top level VHDL file.You may need to place ports using the floorplanner in order to achieve the desired frequency.

# SMVM PE

One SMVM PE exists for each of the static and dynamically switched cases. Files for the dynamically switched PE are in:

- `/cs/courses/cs184/spring2005/assign/p4/PEdynamic`

Files for the statically switched PE are in:

- `/cs/courses/cs184/spring2005/assign/p4/PEstatic`

`PEdynamic/PEdynamicWrap.vhd` is the VHDL wrapper for the dynamic PE. `PEdynamic/*.edf` files must be placed in the same directory as the `.npl` project to be included in compilation. `PEstatic` files are the same as `PEdynamic`.

`PEstaticWrap` inputs recv and outputs send. There are five 15-bit wide words in a single message. Your TM switches may be parameterized by the message depth to save memory, so in this case the input ratio is five.

`PEdynamicWrap` inputs recv with recvV and recvB and outputs send with sendV and sendB. There are six 15-bit words in a message to be sent and five 15-bit words in a received message. The most significant word in a sending message is the header. The five least significant should be received by the PE addressed by the header.