California Institute of Technology
Department of Computer Science
Computer Architecture

CS184b, Spring 2005

Assignment P3: Switch Blocks

Monday, April 18

Due: Monday, April 25, 9:00AM

Goals:

- Build switches necessary for interconnect
- Continue building up area and timing model information

Collaboration: This assignment is a group assignment. Each students should have primary responsibility for at least one of the 4 designs and provide the data and writeup for that design.

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

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

Turnin: We have created a directory: `/scratch/ic/cs184b/project/p3` Please put the files requested above in that directory (but keep your a master copy elsewhere, that directory is not backed up). Create a master file `p3.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/p3/p3.html`.
Email `mdel@cs.caltech.edu` when the materials are in the directory and ready for review.

Tasks:

1. Build the following from the switching primitives developed last week:
   - Time-Multiplexed 4×4 Directional Mesh Switch
   - Packet-Switched 4×4 Directional Mesh Switch
   - Time-Multiplexed 4 port ($W = 2$) IO Interface
   - Packet-Switched 4 port ($W = 2$) IO Interface

   All units should be designed to run at 200MHz.
   For each unit:

   (a) Identify who has primary responsibility for the unit and who contributed to the solution.

   (b) Provide a simple diagram like the 2×2 switch diagram showing how the unit is built from switching primitives.

   (c) Write Debug/VHDL (turnin your VHDL code)

   (d) Verify correct operation (turnin your testing script) Use queue depths of 16, 4 word packets, and 16 or 32 cycle time-multiplexed memories for testing.

   (e) Synthesize/place/route

   (f) Provide a function for the area of each switch. Variables:
   - Q=Q-depth
   - T=TM cycles
   - W=network width (tracks/channel)
   - P=wide word width for packets

   Note the active datapath width (serial width) is 15b. Variables relevant to each switch:
   - Time-Multiplexed 4×4 Directional Mesh Switch – $\{T\}$
   - Packet-Switched 4×4 Directional Mesh Switch – $\{Q\}$
   - Time-Multiplexed 2W port IO Interface – $\{W, P, T\}$
   - Packet-Switched 2W port IO Interface – $\{W, P, Q\}$

   Expressions in T and Q can be given in multiples of 16 if appropraite.

   (g) Identify the minimum (unloaded) latency, in clock cycles, for each path through the switch or IO Interface.

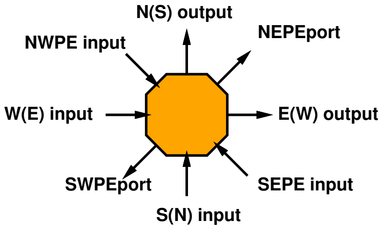
2. Optimize each of the switching elements above attempting to reduce area and latency without sacrificing throughput.

   (a) Can you provide a smaller/lower-latency design that meets the throughput? (This is not necessarily possible, so “no” is a valid answer; if so, identify the key properties of the design preventing further optimization.)

   (b) Provide a diagram showing the design of the switch and any new switching primitives, along with a brief description of the new design.

   (c) Provide the information requested above.

4×4 Directional Mesh Switch



Input: Four 15b data stream
Outputs: Four 15b data streams
Operation:

- Based on the Time-Multiplexed Instruction Control or the Packet Routing Information, send the packet to one of the four output ports.

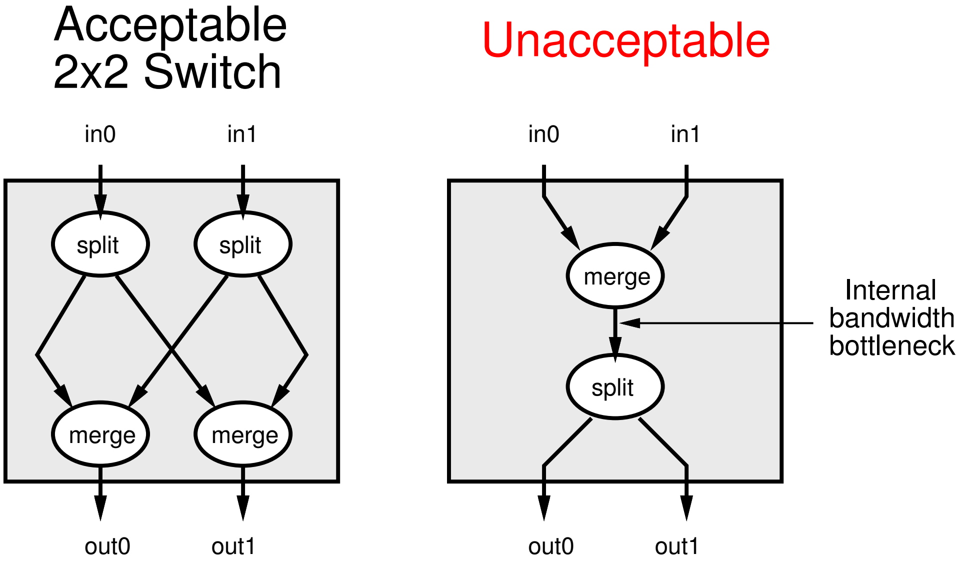
This will be used as a mesh routing primitive.

The inputs are all functionally identical. As shown two serve as inputs from processing elements and two as network inputs.

The outputs are all distinct. Two of the outputs go to adjacent nodes. One of the outputs goes East or West (depending on the switch location in the mesh – see mesh picture two pages forward) and the other goes North or South.

If the deadlock-free routing algorithm permits, it may be possible route either out the E/W port or the N/S port.

The switch should be able to support the full bandwidth of its inputs and outputs.



Since the switch ports have different assignments, you should try to minimize the latency between network ports, perhaps at the expense of the latency associated with network ingress/egress.

The packet-switched header word specifies the destination and is fielded:

14:12	11:9	8:6	5:3	2:0
ChipZ	ChipY	ChipX	Y	X

With 3b for each field, we have have an 8×8×8 board Dishoom, with each node holding up to 8×8 PEs.

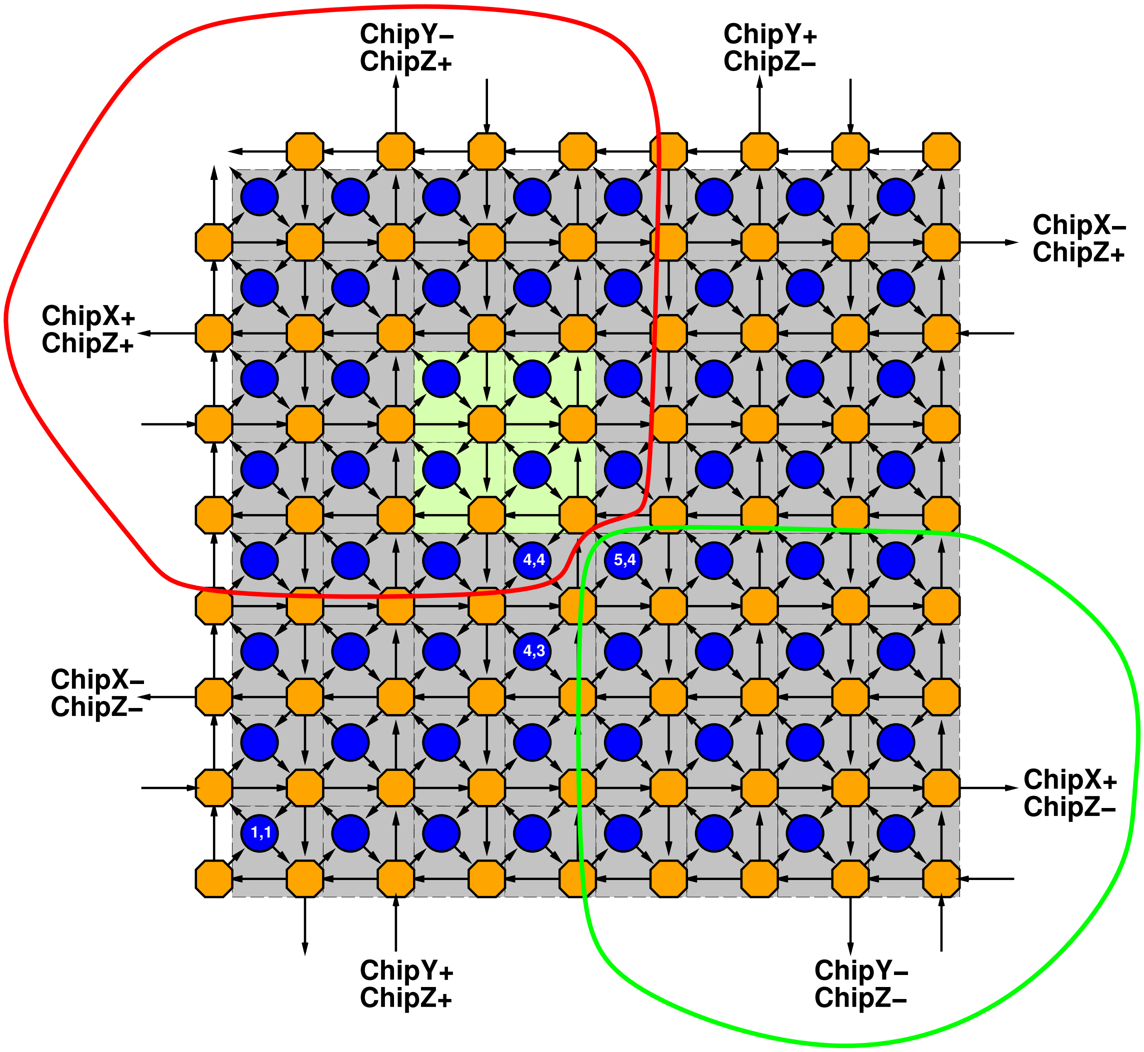
For the sake of evaluation in the packet-switched case, consider that this switch is located at position (X=4, Y=4, ChipX=4, ChipY=4, ChipZ=4) in the network. Consequently, the route function is:

```
if ((ChipX==4)&&(ChipY==4)&&(ChipZ==4))
   {
     if ((X==4)&&(Y==3))
        route out SWPEport
     else if ((X==5)&&(Y==4))
        route out NEPEport
   }
else if (   ((ChipX==4)&&(ChipY==4)&&(ChipZ==4)&&(Y>3)&&(X<6))
         || ((ChipZ>4)&&(ChipX>4))
         || ((ChipZ>4)&&(ChipY<4)) )
        route out N
else if (   ((ChipX==4)&&(ChipY==4)&&(ChipZ==4)&&(Y<5)&&(X>4))
         || ((ChipZ<4)&&(ChipX>4))
         || ((ChipZ<4)&&(ChipY<4)) )
        route out E
else if (PreferNtoE)
        route out N
     else
        route out E
```

PreferNtoE is the signal you calculate to decide which way to route a packet when there is choice. This should look at output fullness to avoid blocking; it may also take a state or random bit to break the tie.

Note that this is not relevant to the time-multiplexed case.

We derive the previous switching function from the following routing scenario:

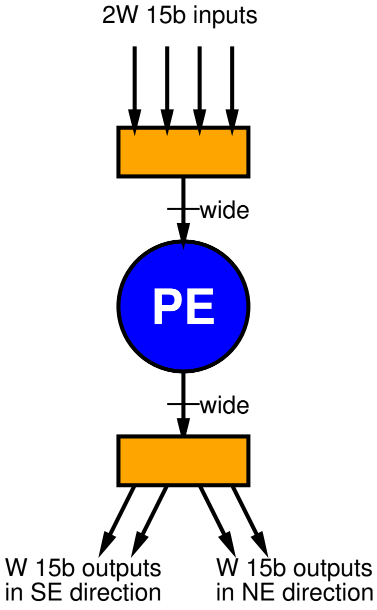


To avoid having to write separate VHDL for every switch, you will eventually want to parameterize your switch by its location in the grid (*e.g.* MyChipZ, MyChipY, MyChipZ, MyX, MyY). This may lead to more generic code for a NE switch which looks like:

```
if ((ChipX==MyChipX)&&(ChipY==MyChipY)&&(ChipZ==MyChipZ))
   {
     if ((X==MyX)&&(Y==(MyY-1)))
        route out SWPEport
     else if (((X==(MyX+1))&&(Y==MyY))
        route out NEPEport
   }
else if (   ((ChipX==MyChipX)&&(ChipY==MyChipY)&&(ChipZ==MyChipZ)
             &&(Y>(MyY-1))&&(X<(MyX+2)))
         || ((ChipZ>MyChipZ)&&(ChipX>MyChipX))
         || ((ChipZ>MyChipZ)&&(ChipY<MyChipY)) )
        route out N
else if (   ((ChipX==MyChipX)&&(ChipY==MyChipY)&&(ChipZ==MyChipZ)
             &&(Y<(MyY+1))&&(X>MyX))
         || ((ChipZ<MyChipZ)&&(ChipX>MyChipX))
         || ((ChipZ<MyCHipZ)&&(ChipY<MyChipY)) )
        route out E
else if (PreferNtoE)
        route out N
     else
        route out E
```

However, there would also need to be code for NW, SE, and SW switches and there are probably still some exceptions around the edge of the chip. Alternately, the code might look at MyX and MyY, determine if it is a NE, NW, SE, or SW switch and then select the appropriate switching function. In any case, this does not impact the functionality you need to provide this week or, we believe, the complexity of each individual switch; that is, we believe this switching function is indicative of the complexity of the route decision being made at each switch point so the area and timing of this switch will be comparable to what you will see for other locations in the mesh. We will need to revisit the more generic route function in later labs.

IO Interface



Input: $2W$ 15b data stream, 1 wide-word data stream
Outputs: $2W$ 15b data streams, 1 wide-word data stream
Operation:

- Based on the Time-Multiplexed Instruction Control or the Packet Routing Information, serialize the the wide word input and send it to one of the output streams; deserialize the input data streams and deliver to the wide-word output.

This provides the input and output to the processing element node. Recall the PE can produce/consume one wide-word input per cycle. In the packet-switched case, flow control continues through the wide-word input/output.

This design will be parameterized by the width of the wide-word, in addition to the number of time-muliplexed cycles and the network channel width (W).

The packet-switched route decision here is similar to the switch. Based on the quadrant of the destination (or chip exit), egress should pick the appropriate port through which to enter the network.

For sake of packet-switched evaluation, consider the PE located at position (X=4,Y=4, ChipX=4,ChipY=4,ChipZ=4).

The previous mesh (three pages back) had $W = 1$. Shown below is a snippet for a $W = 2$ mesh. The 4×4 switching element remains the same. The IO Node gets larger as parameterized above.