LSU EE 7700-2 **Homework 2** **Due: 19 March 2012**

Problem 1: The CUDA assembler below is a simplified version of the `dots` kernel.

```
I0:   IMAD R0, R0, c [0x0] [0x8], R2;
I1:   MOV R3, c [0x2] [0x10];
I2:   IMUL.HI R2, R0, 0x8;
I3:   MOV R6, c [0x2] [0x18];
I4:   IMAD R8.CC, R0, 0x8, R3;
I5:   MOV R5, c [0x2] [0x4];
I6:   IMAD R6.CC, R0, 0x4, R6;
I7:   LD.E R2, [R8];
I8:   FFMA R2, R5, R2, c [0x2] [0x0];
I9:   FFMA R0, R3, c [0x2] [0x8], R2;
I10:  ST.E [R6], R0;
```

(*a*) Show how the code would execute on a multiprocessor on a CC 2.0 device. Show this using the same kind of diagrams used in the scheduling examples in the CC 1.x / 2.x GPU microarchicture notes, `http://www.ece.lsu.edu/gp/notes/set-nv-org.pdf`, such a diagram has been started below. Notice that in the diagram below cycles that end on the right side of the page continue further down on the left side. Notice also that width is 24 cycles, so that an instruction forced to wait for an operand will appear directly below the instruction it depends on.

- Assume that all non-memory instructions have a latency of 24 cycles.
- Be sure to take into account dependencies!
- Assume that scheduler chooses the lowest numbered warp available.
- Show the execution for 4 warps.

```
     0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
W00: [ I0] [I1 ]
W01:
...

     24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
W00: [I2 ]
W01:
...
```

(*b*) Compute the utilization (as described below) of the code above on a CC 2.0 multiprocessor when running a block of 1 warp and when running a block of 2 warps. The utilization is the average percentage of busy CUDA cores. For example, consider the following execution of a kernel with five instructions and one warp on a CC 2.0 device:

```
     0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
W00: [I0 ] [I1 ] [I2 ] [I3 ]         Note: This is unrelated to dots kernel.

     24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
W00:             [I4 ]
```

The utilization is $\frac{5\times32}{30\times32} = .1667$, where the numerator is the number of instructions (each using one CUDA core) and the denominator is the maximum number of instructions that could be executed over the 30 cycles it took the code to execute.

(*c*) Based just on utilization, how many warps are needed to fully utilize the device? *Note: When scheduling is taken into account, more warps might be needed.*

(*d*) Assume that the kernel could be launched with as many threads as was needed. Based on the full utilization computed above, what would be the data bandwidth needed by the kernel running on a GeForce GTX 580?

(*e*) For the following question locate specifications for the GTX 580, in particular the memory bandwidth. Consider the following changes to the GTX 580. For each one indicate whether it would increase the speed of the kernel based on the analysis above. If it would not increase the speed explain why.

- Adding another multiprocessor but not changing anything else.
- Increasing the number of CUDA cores per multiprocessor from 32 to 64; a scheduler could issue an entire warp in one cycle. Other specifications are not changed.
- Doubling the memory bandwidth.