• Finish HW1 peer reviews by Friday, please.

• HW2 out, due Nov. 2.

• Check if Cython works for you in HW2, now-ish.

• Log into Odyssey before lecture on Friday.
Review: Threads
Partial Iterations in AVX

• How do we iterate some parts of an AVX register, but not all of them?

• Recall the GPU branching picture from last lecture...
What about Branches?

Time (clocks)

ALU 1 ALU 2 ... ... ALU 8

if x > 0:
    tmp = x ** 5.0
else:
    tmp = 2 * tmp

result = tmp + 1

Not all ALUs do useful work!

Worst case: 1/8 peak performance
AVX branching

- Comparisons yield either:
  - all 1s (0xFFFFFFFF) == -NaN
  - all 0s (0xFFFFFFFF) == 0.0

- We can use these with the AVX.and(a, b) function to mask certain values.
if x > 0:
   tmp = x ** 5.0
else:
   tmp = 2 * tmp
result = tmp + 1

mask = x > 0
notmask = not(mask)
tmp = (and(mask, x) ** 0.5 +
       and(notmask, 2 * tmp))
result = tmp + 1

We didn’t give you AVX.not, but you can add it if needed. Post on Piazza…
parallel() vs. prange()

cdef:
    int val
    int[:] val

with nogil, parallel(num_threads=N):
    # This statement will be executed N times,
    # once for each thread.
    # val will be thread-private
    val = threadid()

for i in prange(L, nogil=true, num_threads=N):
    # This statement will be executed L times,
    # somehow split between N threads
    val[i] = i
parallel() vs. prange()

with nogil, parallel():
    buf = <int *> malloc(sizeof(int) * size)

    # populate thread-local buffer in a sequential loop
    # (one per thread)
    for i in range(size):
        buf[i] = i * 2

    # share the work using the thread-local buffers
    for i in prange(n):
        func(buf)

    free(buf)
Debugging

- Run with one thread.
- with gil: print ....
- gdb or lldb - C level debuggers
- cygdb - part of cython, but needs gdb (pain on OSX).
Deadlock

- Deadlock:
  - A locks L1, L2
  - B locks L2, L1
- Avoid by always locking in the same order
  - it doesn’t really matter which order, just be consistent.
Double locking

• You can’t do this in one thread:

    acquire(lock_A)
    acquire(lock_A)

• There are Reentrant locks in Python (nested locks in OpenMP) where this is allowed.

• We didn’t provide code for these in HW2, P2, but you could add it.
Getting integers out of AVX

- Can’t read integers directly out of AVX float register.
- Write out to memory using AVX.to_mem()
- Loop over 8 values and convert to int
  - use <int> if needed.