Chapter 4 Solutions

Case Study: Implementing a Vector Kernel on a Vector Processor and GPU

4.13 a. 1.5 GHz × 0.80 × 0.85 × 0.70 × 10 cores × 32/4 ¼ 57.12 GFLOPs/s

b. Option 1:
1.5 GHz × 0.80 × 0.85 × 0.70 × 10 cores × 32/2 ¼ 114.24 GFLOPs/s
(speedup ¼ 114.24/57.12 ¼ 2)
Option 2:
1.5 GHz × 0.80 × 0.85 × 0.70 × 15 cores × 32/4 ¼ 85.68 GFLOPs/s
(speedup ¼ 85.68/57.12 ¼ 1.5)
Option 3:
1.5 GHz × 0.80 × 0.95 × 0.70 × 10 cores × 32/4 ¼ 63.84 GFLOPs/s
(speedup ¼ 63.84/57.12 ¼ 1.11)
Option 3 is best.

4.14 a. Using the GCD test, a dependency exists if GCD (2,4) must divide 5 — 4. In this case, a loop-carried dependency does exist.

b. *Output dependencies*
S1 and S3 cause through A[i]
Anti-dependencies
S4 and S3 cause an anti-dependency through C[i]
Re-written code

```
for (i=0;i<100;i++) {
  T[i] = A[i] * B[i]; /* S1 */
  B[i] = T[i] + c; /* S2 */
  A1[i] = C[i] * c; /* S3 */
  C1[i] = D[i] * A1[i]; /* S4 */}
```

True dependencies
S4 and S3 through A[i]
S2 and S1 through T[i]

c. There is an anti-dependence between iteration i and i+ 1 for array B. This can be avoided by renaming the B array in S2.

4.15 a. Branch divergence: causes SIMD lanes to be masked when threads follow different control paths.

b. Covering memory latency: a sufficient number of active threads can hide memory latency and increase instruction issue rate.

c. Coalesced off-chip memory references: memory accesses should be organized consecutively within SIMD thread groups.

d. Use of on-chip memory: memory references with locality should take advantage of on-chip memory, references to on-chip memory within a SIMD thread group should be organized to avoid bank conflicts.

4.16 This GPU has a peak throughput of $1.5 \times 16 \times 16 ¼ 384$ GFLOPS/s of single-precision throughput. However, assuming each single precision operation requires four-byte two operands and outputs one four-byte result, sustaining this throughput (assuming no temporal locality) would require 12 bytes/FLOP $\times$ 384 GFLOPs/s ¼ 4.6 TB/s of memory bandwidth. As such, this throughput is not sustainable, but can still be achieved in short bursts when using on-chip memory.