Consider integer matrix multiplication $C = A \times B$, where $A$, $B$, and $C$ each are $n \times n$ 32-bit integers. Assuming all instructions and data are 32-bit based, estimate the total numbers of clocks to compute $C$ on the following platforms.

(a) A standard single-issue pipelined computer consisting of five stages with 1 adder and 1 multiplier.
(b) A 20-stage Pentium 4 with 1 integer adder and 1 integer multiplier.
(c) A quad-core processor, where each core has 1 integer adder and 1 integer multiplier. The cores share the DIMMs.
(d) A 16-way symmetric multiprocessor (16 processors SMP) sharing the DIMMs but separate L3 caches, where each processor has 1 adder and 1 multiplier.
(e) A GPU with 16 SIMD processors each of which has 16 lanes, where each lane can execute the same instruction on 16 32-bit elements.
(f) A cluster of 128 PCs connected through a Gbps switch. Each PC has Gbps NIC. Assume the switch and NICs are 100 times slower than the processors, in other words, sending an integer to another machine takes 100 clocks. The matrices are originally stored in PC0 and the final matrix will be stored in PC0. Assume collective communication constructs such as MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Barrier, etc.

Assume the following:
- An integer addition takes 1 clock.
- An integer multiplication 2 clocks.
- Finding $C[i][j]$ requires $n$ multiplications and $n-1$ additions.
- A memory load/store takes 4 clocks.
- Ignore cache effect.
- For MMX/XMM/AVX, mult/add instructions can operate on 4 32-bit integers for XMM and , 8-bit byte, 16-bit short, 32-bit word, or 64-bit long.

State your own assumptions that are reasonable and short.