SOLUTIONS TO CHAPTER 1 PROBLEMS

1. a. A translator converts programs in one language to another.
b. An interpreter carries out a program instruction by instruction.
c. A virtual machine is a conceptual machine, one that does not exist.

2. It is possible, but there are problems. One difficulty is the large amount of code produced. Since one ISA instruction does the work of many microinstructions, the resulting program will be much bigger. Another problem is that the compiler will have to deal with a more primitive output language, hence it, itself, will become more complex. Also, on many machines, the microprogram is in ROM. Making it user-changeable would require putting it in RAM, which is much slower than ROM. On the positive side, the resulting program might well be much faster, since the overhead of one level of interpretation would be eliminated.

3. During the detailed design of a new computer, the device and digital logic levels of the new machine may well be simulated on an old machine, which puts them around level 5 or 6.

4. Each level of interpretation slows down the machine by a factor of n/m. Thus, the execution times for levels 2, 3, and 4 are kn/m, kn^2/m^2, and kn^3/m^3, respectively.

5. Each additional level of interpretation costs something in time. If it is not needed, it should be avoided.

6. You lose a factor of n at each level, so instruction execution times at levels 2, 3, and 4 are kn, kn^2, and kn^3, respectively.

7. Hardware and software are functionally equivalent. Any function done by one can, in principle, be done by the other. They are not equivalent in the sense that to make the machine really run, the bottom level must be hardware, not software. They also differ in performance.

8. Not at all. If you wanted to change the program the difference engine ran, you had to throw the whole computer out and build a new one. A modern computer does not have to be replaced because you want to change the program. It can read many programs from many CD-ROMs.

9. A typical example is a program that computes the inner product of two arrays, A and B. The first two instructions might fetch $A[0]$ and $B[0]$, respectively. At the end of the iteration, these instructions could be incremented to point to $A[1]$ and $B[1]$, respectively. Before indexing and indirect addressing were invented, this was done.

10. Raw cycle time is not the only factor. The number of bytes fetched per cycle is also a major factor, this increasing with the larger models. Memory speed and wait states play a role, as does the presence of caching. A better I/O architecture causes fewer cycles to be stolen, and so on.

11. The design of Fig. 1-5 does I/O one character at a time by explicit program command. That of of Fig. 1-6 can use DMA to have the controller do all the work, relieving the CPU of the burden, and thus making it potentially better.

12. Each person consumes 730 tags per nonleap year. Multiply by 300 million and you get 219 billion tags a year. At a penny a tag, they cost $2.19 billion dollars a year. With GDP exceeding $10 trillion, the tags add up to 0.02% of GDP, not a huge obstacle.

13. The following appliances are normally controlled these days by embedded systems: alarm-clock radios, microwave ovens, television sets, cordless telephones, washing machines, sewing machines, and burglar alarms.

14. According to Moore's law, next year the same chip will have 1.6 times as many transistors. This means that the area of each transistor will be 1/1.6 or 0.625 times the size of this year's transistors. Since the area goes like the square of the diameter, the diameter of next year's transistors must be 0.079 microns.

15. In three years the power has doubled twice so the execution time is down by a factor of four. Thus, a 4-hour run will take 1 hour then. In 6 years we gain another factor of four in power, so the run time then decreases to 15 min. If we start the simulation now, it will complete in 5 years, but if we start the simulation in 3 years it will run four times faster and thus complete in 1.25 years. If we add the 3-year delay in starting plus the 1.25 year execution time, the run will complete in 4.25 years, which is faster than starting the run now, which would take 5 years from the starting date to complete.

16. The current numbers obviously depend on when the calculation is being made, but for example, assume the instruction execution rate is 1 billion instructions/sec, the memory is 1 GB, and the cost is $500. The speed gain is 2000. The 7090 had 1,179,648 bits of memory, whereas 1 GB is 8,589,934,592 bits giving a memory gain of 7281. Thus, the speed-size product is 14,562,000 times better. And with a price improvement of 6000, the total gain is 87.372 million times better. Now let's scale the aircraft using the same numbers. If it could fly 2000 faster, it would go 1.18 million km/hr, which means it could circle the earth in 2 min. A gain of 7281 in capacity implies about 1.3 million passengers. And our plane would cost about $667.

17. After mainframes, we had personal computers. A current development is moving computation to the cloud, which is basically centralized computing all over again.