suburbs or in large cities with sizable appropriations for computer-based instruction. Small towns and rural districts, as well as low-income cities, lag behind, except where a state has promoted and supported a computer initiative. More boys than girls are encouraged to take up computer use, except in the lower grades. Minorities are inadequately represented in advanced computer science courses.

Sixth, software is still driven by what most educational publishers have reluctantly had to buy or develop to supplement their textbooks to meet adoption requirements. These multimillion-dollar businesses control the market and determine what children learn. Being businesses, they cannot afford to take risks, so few of them publish innovative educational software.

So what have schools been using on the reported figure of 1.6 million computers in the schools? More precisely, what and how have they been teaching with them?

Help Wanted

In the initial onslaught of the software shortage, a few aficionados wrote their own instructional software. A teacher who could see how to make a computer do something more easily might design a simple program for local use or might collaborate with a programmer or engineer friend. If a colleague wanted to try out the product, so much the better. A few small publishers polished such programs up and offered them inexpensively. Sometimes the programs were full of bugs. Generally, they looked primitive. A great many single-concept science programs began this way.

Other programmers took the educational software and ideas that had already been developed for timesharing systems and rewrote them for micros. Very few of these early programs offered more than a textbook or a workbook, but they fit right into the curriculum and gave the computer something to do that didn't strain its powers.

After a while, complaints began to emerge about the rough quality of the available educational software. As computers gained more memory capacity, authors began to write more polished programs. Many of these were so formulaic that authoring systems were published that allowed teachers to develop their own drills or branching multiple-choice programs. Even when such programs are professionally prepared and are supplied with ornamental graphics and sound, they are what teachers and students have come to describe as dull, at best.

Only when educational software began to look like a commercially worthwhile field did graphics designers, software engineers, and teachers come out of hiding. Conventions evolved about what an ideal program should look like and contain.

Hands-off or Hands-on

When "computer literacy" entered the curriculum, it focused on hardware, not software. It meant courses on "How the Computer Works," "Inside the Computer," "The Impact of Computers on Our Society," or "Our Friend, the Computer." Children were given paper-and-pencil tests on what they learned about computers. We all know youngsters who had to draw the CPU, memory banks, input, ouput, and keyboard. Step-by-step lesson plans were designed by curriculum developers. It is little wonder that many high schoolers now hate computer courses.

Apparently, however, educators thought that was the way to incorporate computer consciousness into the curriculum. But awareness is only a first step in literacy. People don't learn to drive a car by reading a manual.

Practice is the best teacher. We learn

continued