There are several different ways in which the three address-decoding strategies defined in the last section may be implemented. In general, address-decoding techniques may be divided into four groups: address decoding using random logic, address decoding using *m*-line to *n*-line decoders, address decoding using PROMs and address decoding using programmable logic arrays.

5.4.1 Address Decoding Using Random Logic

Random logic is the term which describes a system constructed from small-scale TTL logic, with AND, OR, NAND, NOR gates and invertors. When address decoding with random logic is implemented, the chip-select input of a memory component is derived from the appropriate address lines by means of a number of TTL gates. For example, if a PIA located at 8004 is to be fully decoded, the circuit in Fig. 5.16 is suitable. The PIA has two register select lines, leaving 14 address lines to be decoded. The address of the PIA, $8004_{16} = 1000000000001XX_2$, may be decoded with a 2-input NAND gate, 13 invertors, and a 13-input NAND gate, as in Fig. 5.16.

Full address decoding using random logic is costly in terms of the large number of integrated circuits required, especially in cases where several different memory devices are to be selected. Because a circuit implemented by means of random logic is tailor-made for a specific application, it lacks the flexibility inherent in some of the other forms of address-decoding circuit.

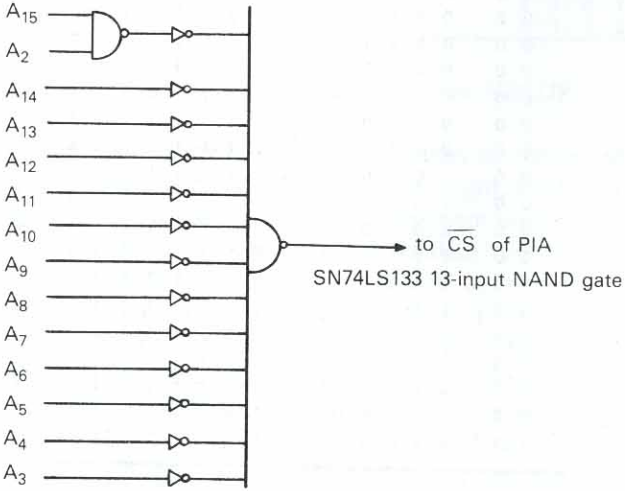

Figure 5.16 Example of address decoding using random logic