A separate off-chip memory management device would have degraded performance or constrained access times and was never seriously considered. Advances in VLSI technology are removing many such constraints, however, and will facilitate both evolution and revolution in the handling of system resources.

The data manipulation model. Sandwiched between the references to source operands and destination fields is the actual manipulation of data. Efficient manipulation of data exploits two notions—parallelism and specialization. Assuming for the moment a very simple, totally uncoupled model with \( c \) processor categories, \( n_i \) processors in each category \( (i = 1, 2, \ldots, c) \), \( \text{eff}_i \) average processor instruction "effectiveness" (e.g., average ratio of source language operations to machine language instructions for a given function), and \( \text{rate}_i \), average instruction execution rate, the total system processing power \( P \) is determined by

\[
P = \sum_{i=1}^{c} n_i \, \text{eff}_i \, \text{rate}_i,
\]

Absolute physical limits (such as the speed of light) as well as physical realities obviously constrain the power that can be provided by any single processor. Parallelism increases the value of \( n_i \), while specialization raises the value of both \( \text{eff}_i \) and \( \text{rate}_i \) by tailoring the design of individual processor categories to particular functional needs. The effect of processor coupling is complex but, simply stated, makes \( \text{rate}_i \), a function of the degree of coupling rather than a constant for a given processor category.

In pure symmetric multiprocessing, \( c \) is set to 1 and \( n_i \) is greater than 1. In simple functional partitioning, \( c \) is greater than 1 and \( n_i \) is equal to 1 for each functional category \( i = 1, 2, \ldots, c \). The general case is a combination of the two, in which multiple categories each contain multiple processors.

The 8086 family implements three processors for the above model: general data processors (8086), numeric data processors (8087), and input/output processors (8089). Operations are partitioned among the three to take advantage of functional specialization and processor concurrency. A simple three-processor example is shown in Figure 9.

General data processors (such as the 8086 or 8088) provide general system control operations, logical operations, short or single-precision integer computation, and string processing. The 8087 numeric data processor provides accurate, anomaly-free, transparent, multiprecision floating-point or fixed-point computation in support of the proposed hardware-independent IEEE floating-point standard. The 8089 I/O processor provides two channels of I/O processing with bit or byte manipulation, on-the-fly computation, and a block transfer rate of two megabytes per second, and supports two-bus (local processor and system I/O) configurations.

The 8086 general data processor and 8087 numeric data processor are tightly coupled (one-on-one), and the pair execute concurrently on a single instruction stream managed by the 8086, with a special synchronization instruction (WAIT) to coordinate their operation.

**Figure 9. A three-processor system.**