Embedded systems have always been very cost-sensitive. Up till recently, the main focus has been on area-efficient designs meeting constraints due to performance and design time. During the last couple of years, power dissipation has become an additional major design measure for many systems next to these traditional measures. More and more applications become portable because this is felt as an added value for the product (e.g., wireless phones, notebook computers, multimedia terminals). The less power they consume, the longer their batteries last and the lighter their batteries can be made. Higher power consumption also means more costly packaging and cooling equipment, and lower reliability. The latter has become a major problem for many high-performance (non-portable) applications. As a consequence, power efficient design has become a crucial issue for a broad class of applications.

Many of these embedded applications turn out to be data-dominated, both in the multimedia domain and in the telecommunication domain. Experiments have shown that for these applications, a very large part of the power consumption is due to data storage and data transfer. Also the area cost is for a large part dominated by the memories. Hence, we believe that the memory architecture should be optimized as a first step in the design methodology for this type of applications. This has been formalized in our Data Transfer and Storage Exploration (DTSE) methodology.

This book provides a summary of the research activities in the context of the system-level DTSE at IMEC. In this introductory chapter, only a high-level view is provided which will be refined in the subsequent chapters.

1.1 CONTEXT AND PROBLEM FORMULATION

For most real-time signal processing applications many ways exist to realize them in terms of a specific algorithm. As reported by system designers, in practice this choice is mainly based on "cost" measures such as the number of components, performance, pin count, power consumption, and the area of the custom components. Currently, due to design time restrictions, the system designer has to select — on an ad-hoc basis — a single promising path in the huge decision tree from abstract specification to more refined specification (figure 1.1). To alleviate this situation, there is a need for fast and early feedback at the algorithm level without going all the way to assembly code or hardware layout. Only when the design space has been sufficiently explored at a high level and when a limited number of
promising candidates have been identified, a more thorough and accurate evaluation is required for the final choice (figure 1.1).

Experience shows that the initial specification heavily influences the outcome of the underlying architectural estimation or mapping tools (e.g., for data-path allocation, memory allocation, address generation). Therefore, transforming this specification is one of the most prominent tasks during the early system-level exploration of cost measures. This task is both very difficult to explore globally using ad hoc strategies, and very tedious and error-prone if done fully manually.

To remove this time bottle-neck, we are developing systematic methodologies, partly supported with automatatable steering methods, for the set of system-level transformations that have the most crucial effect on the system exploration decisions. Such transformations change the control/data-flow graph or other specification models on which the subsequent estimation/synthesis tasks work. For the most important categories of transformations in our target domain, an appropriate model has been identified which allows to (as much as possible globally) explore the effect of these transformations on the cost measures relevant to the system designers. Based on this model, efficient steering techniques are established. These guide the transformations such that the system designer becomes less dependent on the original specification description while still arriving at a reasonable cost estimate for a particular figure of merit, without having to explore all the possible transformations manually. This approach will make the system designer much less dependent on the way the original specification is described.

Most research and development efforts in digital electronics have focused on increasing the speed and integration of digital systems on a chip while keeping the silicon area as small as possible. This has resulted in a powerful, but power hungry, design technology, which enabled a whole series of new applications such as real-time 3D rendering, multi-media terminals, video compression, speech recognition, and so on. While focusing on speed and area, power consumption has long been ignored. This situation is, however, rapidly changing [335]. The main reason is the increasing demand for portable systems in the areas of communication (e.g., cellular phones and pagers), computation (e.g., notebook computers and personal digital assistants), and consumer electronics (e.g., multi-media terminals and digital video cameras). These systems require sophisticated and power-hungry algorithms for high-bandwidth wireless communication, video compression and decompression, handwriting recognition, speech processing, and so on. As relatively little improvement in battery technology is anticipated (expected battery life time will increase with no more than 30 to 40% over the next 5 years), portable devices will suffer from either very short battery life or unreasonably heavy battery packs unless a low power design approach is adopted. Hence, for portable applications, average power consumption has become a critical design concern which is starting to replace speed and area as the most important implementation constraints.

Portability is, however, not the only driving force behind the move towards low-power design. Heat dissipation is another one. The cost associated with packaging and cooling high-performance devices such as modern microprocessors is becoming prohibitive. Since core power consumption