e-PG Pathshala

Subject : Computer Science

Paper: Embedded System
Module: Embedded System Design Process
Module No: CS/ES/33

Quadrant 1 – e-text

In this lecture various steps involved in Embedded system design process will be discussed.

1. Design process steps:

There are different steps involved in Embedded system design process. These steps depend on the design methodology. Design methodology is important for optimizing performance, and developing computer aided design tools. It also makes communication between team members easier. Figure1 shows the steps in the design process. They are requirements gathering, specification formulation, architecture design, building of components, and system integration.

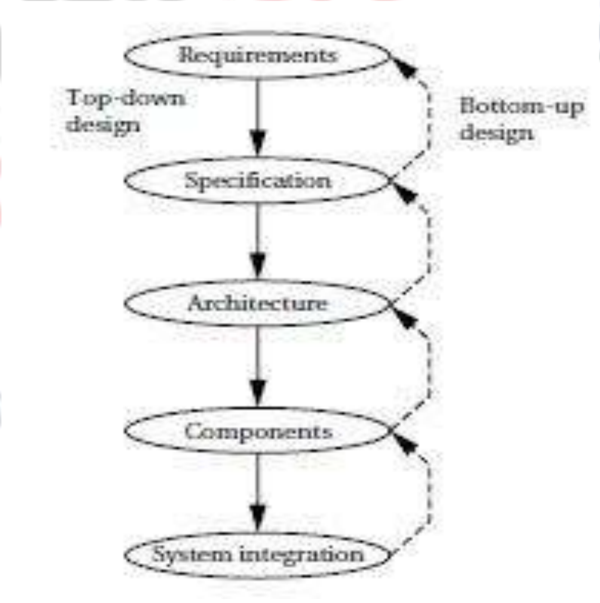


Figure 1. Steps in Design process

The steps in the design process can be viewed as top down view and bottom up view. Top–down view begins with the most abstract description of the system and concludes with concrete details. Bottom–up view starts with components to build a system. Bottom–up design steps are shown in the figure as dashed-line arrows. We need bottom–up design because we do not have perfect insight into how later stages of the design process will turn out.

The major goals of the design to be considered are :

- Manufacturing cost;
- Performance (both overall speed and deadlines) and
- Power consumption.

The tasks which need to be performed at each step are the following.

- We must analyze the design at each step to determine how we can meet the specifications.
- We must then refine the design to add details.
- We must verify the design to ensure that it still meets all system goals, such as cost, speed, and so on.

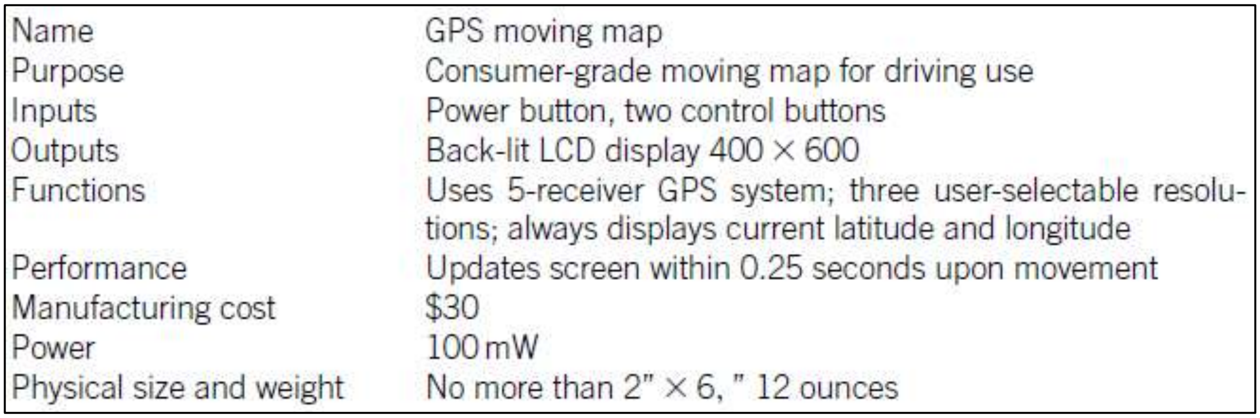
We will discuss each step of the design process in detail.

1.1 Requirements

Informal descriptions gathered from the customer are known as requirements. The requirements are refined into a specification to begin the designing of the system architecture. Requirements can be functional or non-functional requirements. Functional requirements need output as a function of input. Non-functional requirements includes performance, cost, physical size, weight, and power consumption. Performance may be a combination of soft performance metrics such as approximate time to perform a user-level function and hard deadlines by which a particular operation must be completed. Cost includes the manufacturing, nonrecurring engineering(NRE) and other costs of designing the system. Physical size and weight are the physical aspects of the final system. These can vary greatly depending upon the application. Power consumption can be specified in the requirements stage in terms of battery life.

Example:

Let us consider an example of a GPS system. The sample Requirement Form is shown in Figure 2.



Name	GPS moving map
Purpose	Consumer-grade moving map for driving use
Inputs	Power button, two control buttons
Outputs	Back-lit LCD display 400 × 600
Functions	Uses 5-receiver GPS system; three user-selectable resolutions; always displays current latitude and longitude
Performance	Updates screen within 0.25 seconds upon movement
Manufacturing cost	$30
Power	100 mW
Physical size and weight	No more than 2" × 6, " 12 ounces

Figure 2. Requirements chart

1.2 Specification

Requirements gathered is refined into a specification. Specification serves as the contract between the customers and the architects. Specification is essential to create working systems with a minimum of designer effort. It must be specific, understandable and accurately reflect the customer's requirements.

Example:

Considering the example of the GPS system, the specification would include details for several components:

- Data received from the GPS satellite constellation
- Map data
- User interface
- Operations that must be performed to satisfy customer requests
- Background actions

1.3 Architecture Design

The specification describes only the functions of the system. Implementation of the system is described by the Architecture. The architecture is a plan for the overall structure of the system. It will be used later to design the components.The architecture will be illustrated using block diagrams as shown below.

Example:

A basic block diagram of the GPS system shows the major operations and the data flow among the blocks.

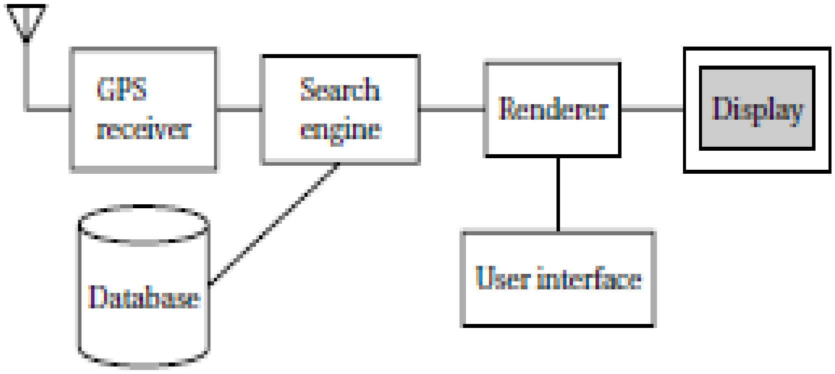


Figure 3. GPS system data flow and operations

This block diagram(figure 3) is an initial architecture that is not based either on hardware or on software but combination of both. This block diagram explains about GPS navigating system where GPS receiver gets current position and the destination is taken from user, digital map for source to destination is found from database and displayed by the renderer. The system block diagram may be refined into two block diagrams - hardware and software.

1.3.1 Hardware block diagram :

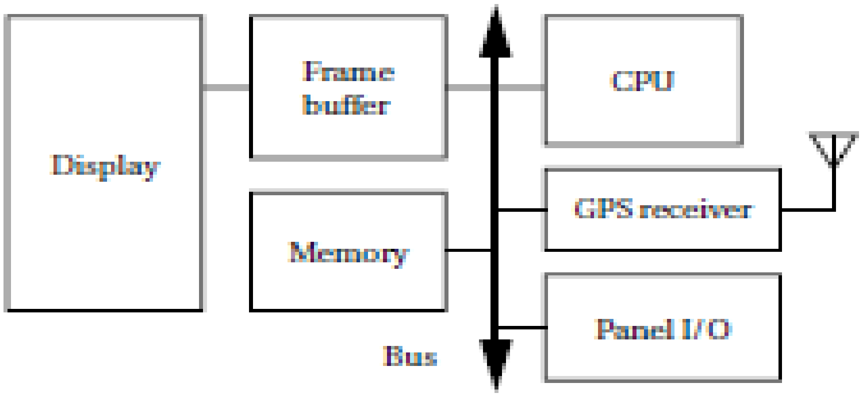


Figure 4. GPS system hardware

Hardware consists of one central CPU surrounded by memory and I/O devices. We have chosen to use two memories that is frame buffer for the pixels to be displayed and separate program/data memory for general use by the CPU. The GPS receiver is used to get the GPS coordinates, and the panel I/O is used to get the destination from the user.

1.3.2 Software block diagram :

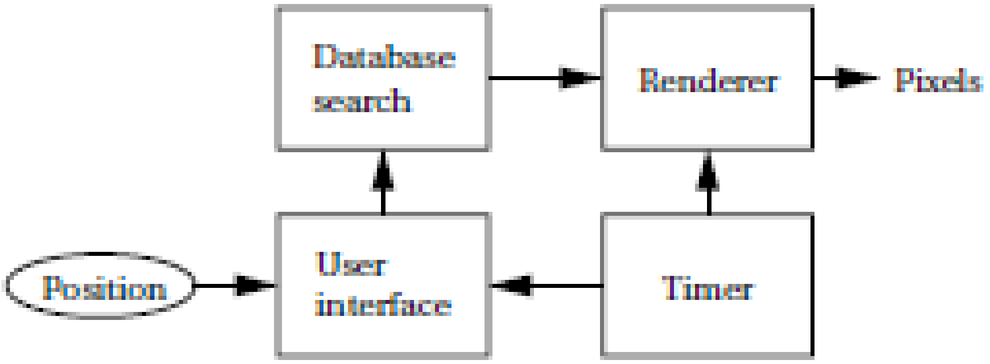


Figure 5. GPS system software

The software block diagram closely follows the system block diagram. We have added a timer to control when we read the buttons on the user interface and render data onto the screen.

To have a truly complete architectural description, we require more details, such as where units in the software block diagram will be executed in the hardware block diagram and when the operations will be performed in time.

Architectural descriptions must be designed to satisfy the functional and non-functional requirements. Not only must all the required functions be present, but we must meet cost, speed, power and other non- functional constraints. Starting out with a system architecture and refining that to hardware and software architectures is one good way to ensure that we meet all specifications. We can concentrate on the functional elements in the system block diagram, and then consider the non- functional constraints when creating the hardware and software architectures.

How do we know that our hardware and software architectures in fact meet constraints on speed, cost, and so on?

- Estimate the properties of the components in the block diagrams (Example: search and rendering functions in the moving map system)
- Accurate estimation derives in part from experience, both general design and particular experience.
- All the non- functional constraints are estimated. If the decisions are based on bad data, those results will show up only during the final phases of design.

1.4 Hardware and Software components

The architectural description tells us what components we need. The component design effort builds those components in conformance to the architecture and specification. The components in general includes both hardware and software modules. Some of the components will be ready-made (example :CPU, memory chips).

Example:

In the moving map, GPS receiver is a predesigned standard hardware component. Topographic software is a standard software module which uses standard routines to access the database. Printed circuit board are the components which needs to be designed. Lots of custom programming is required.

When creating these embedded software modules, ensure the system runs properly in real time and that it does not take up more memory space than allowed. The power consumption of the moving map software example is particularly important. You may need to be very careful about how you read and write memory to minimize power. For example, memory transactions must be carefully planned to avoid reading the same data several times, since memory accesses are a major source of power consumption.

1.5 System integration

After the components are built, they are integrated. Bugs are typically found during the system integration. Good planning can help us to find the bugs quickly. By debugging a few modules at a time, simple bugs can be uncovered. By fixing the simple bugs early, more complex or obscure bugs can be uncovered. System integration is difficult because it usually uncovers problems. The debugging facilities for embedded systems are usually much more limited than the desktop systems. Careful attention is needed to insert appropriate debugging facilities during design which can help to ease system integration problems.

2. Summary

In the Embedded system design process, information is first collected and refined in the Requirement step. Specification uses the refined information to describe the functions of the system which accurately reflects the customer’s requirements and also serves as the contract between the customer and the designer. The functions described by the specification are implemented by the Architecture design. The architectural design describes the components we need which will include both the hardware and software components. After the components are built, they are integrated to get the final system.

3. References

1. Wayne Wolf, “Computers as components: Principles of embedded computing systems design”; Second edition, published by Morgan Kaufmann series(2008).