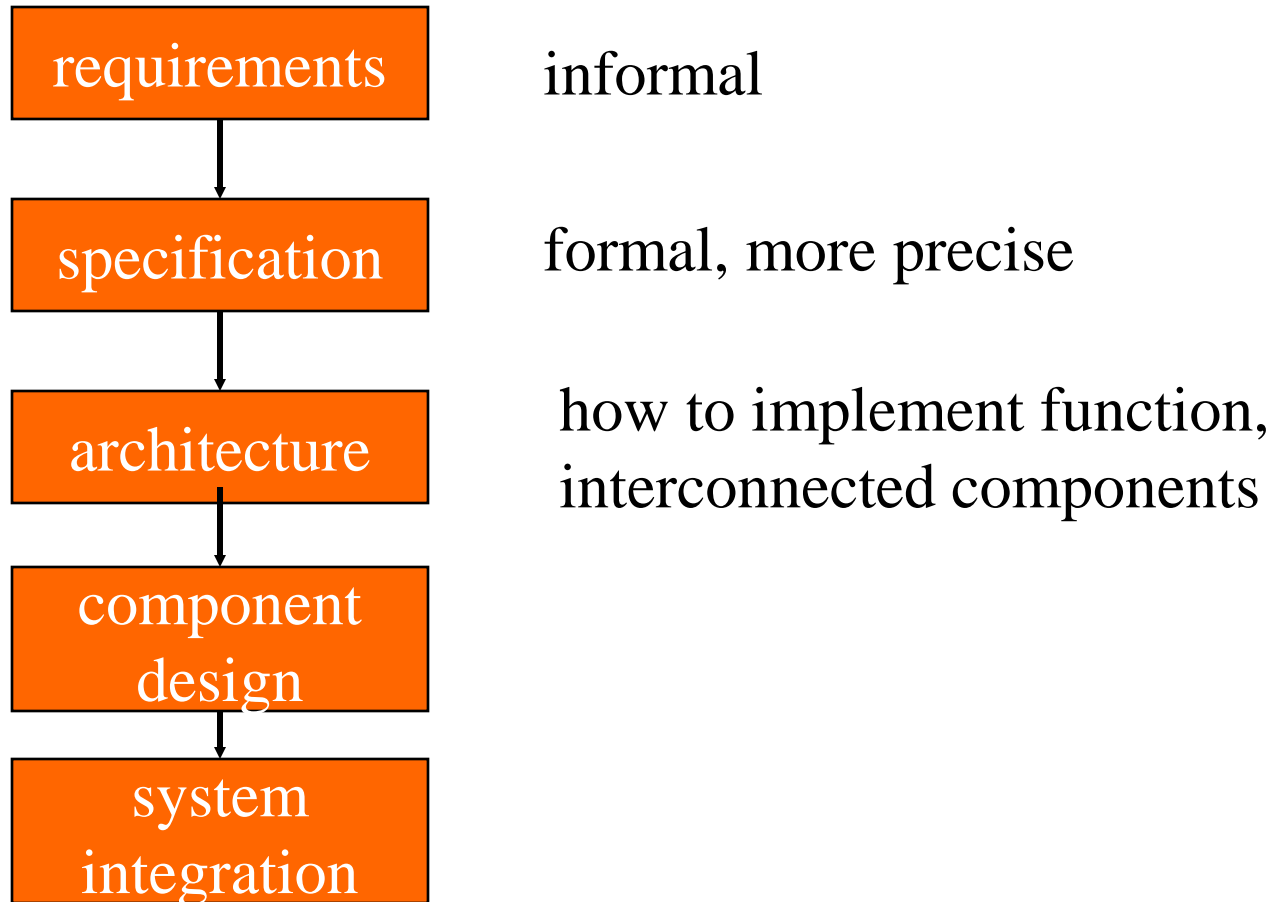
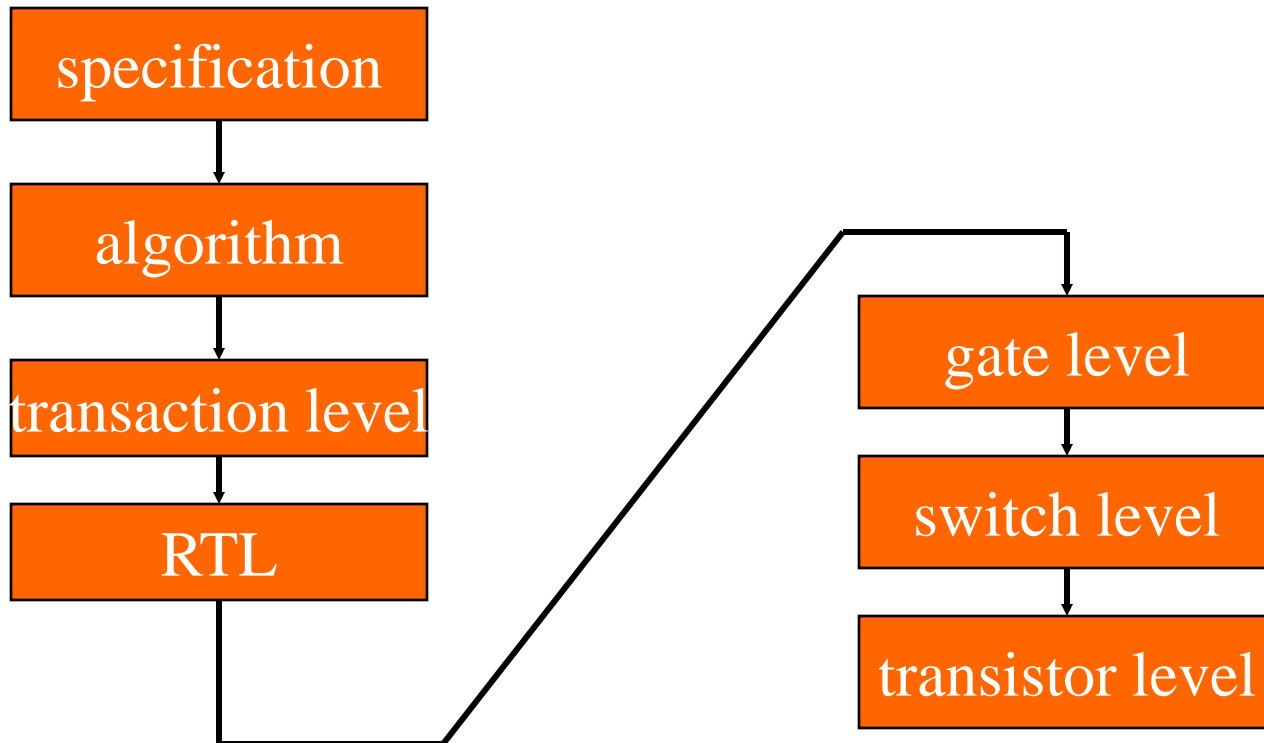


Levels of abstraction



Levels of abstraction (HW)



Top-down vs. bottom-up



⌘ Top-down design:

- ☑ start from most abstract description;
- ☑ work to most detailed.

⌘ Bottom-up design:

- ☑ work from small components to big system.

⌘ Real design uses both techniques.

Design goals



- ⌘ Functionality and user interface.
- ⌘ Performance.
 - ☑ Overall speed, deadlines.
- ⌘ Power consumption.
- ⌘ Manufacturing cost.
- ⌘ Other requirements (physical size, etc.)

Stepwise refinement



⌘ At each level of abstraction, we must:

☑ **analyze** the design to determine characteristics of the current state of the design;

☑ **refine** the design to add detail.

☑ **verify** the design to ensure that it meets all goals

Requirements



- ⌘ Plain language description of what the user wants and expects to get.
- ⌘ May be developed in several ways:
 - ☑ talking directly to customers;
 - ☑ talking to marketing representatives;
 - ☑ Mockup: user interface of a system's requirement
 - ☒ providing prototypes to users for comment.

Functional vs. non-functional

⌘ Functional requirements:

- ☑ output as a function of input.

⌘ Non-functional requirements:

- ☑ time required to compute output;

- ☑ size, weight, etc.;

- ☑ power consumption;

- ☑ reliability;

Sample requirements form



name

purpose

inputs

outputs

functions

performance

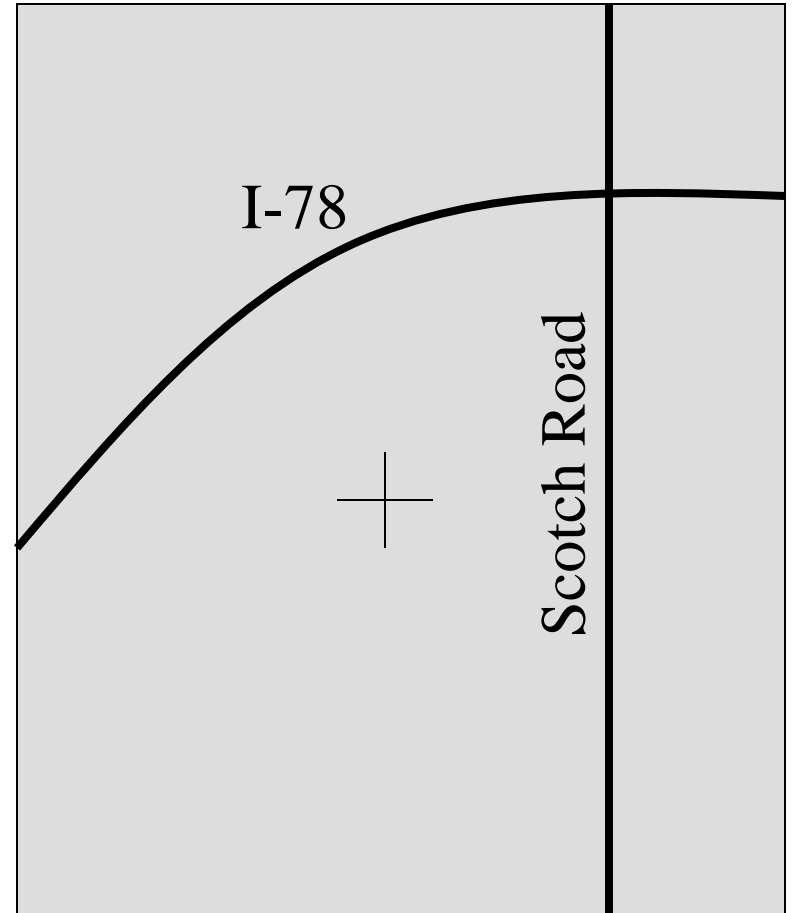
manufacturing cost

power

physical size/weight

Example: GPS moving map requirements

⌘ Moving map obtains position from GPS, paints map from local database.



GPS moving map needs

- ⌘ **Functionality**: For automotive use. Show major roads and landmarks.
- ⌘ **User interface**: At least 400 x 600 pixel screen. Three buttons max. Pop-up menu.
- ⌘ **Performance**: Map should scroll smoothly. No more than 1 sec power-up. Lock onto GPS within 15 seconds.
- ⌘ **Cost**: \$120 street price = approx. \$30 cost of goods sold. = 4~5 x the cost of all components

GPS moving map needs, cont'd.



- ⌘ **Physical size/weight**: Should fit in hand.
- ⌘ **Power consumption**: Should run for 8 hours on four AA batteries.

GPS moving map requirements form

name	GPS moving map
purpose	consumer-grade moving map for driving
inputs	power button, two control buttons
outputs	back-lit LCD 400 X 600
functions	5-receiver GPS; three resolutions; displays current lat/lon
performance	updates screen within 0.25 sec of movement
manufacturing cost	\$ 30 cost-of-goods-sold
power	100 mW
physical size/weight	no more than 2: X 6:, 12 oz.

Specification



- ⌘ A more precise description of the system:
 - ☑ should not imply a particular architecture
 - ☑ provides input to the architecture design process.
- ⌘ May include functional and non-functional elements.
- ⌘ May be **executable** or may be in mathematical form for proofs.

GPS specification



⌘ Should include:

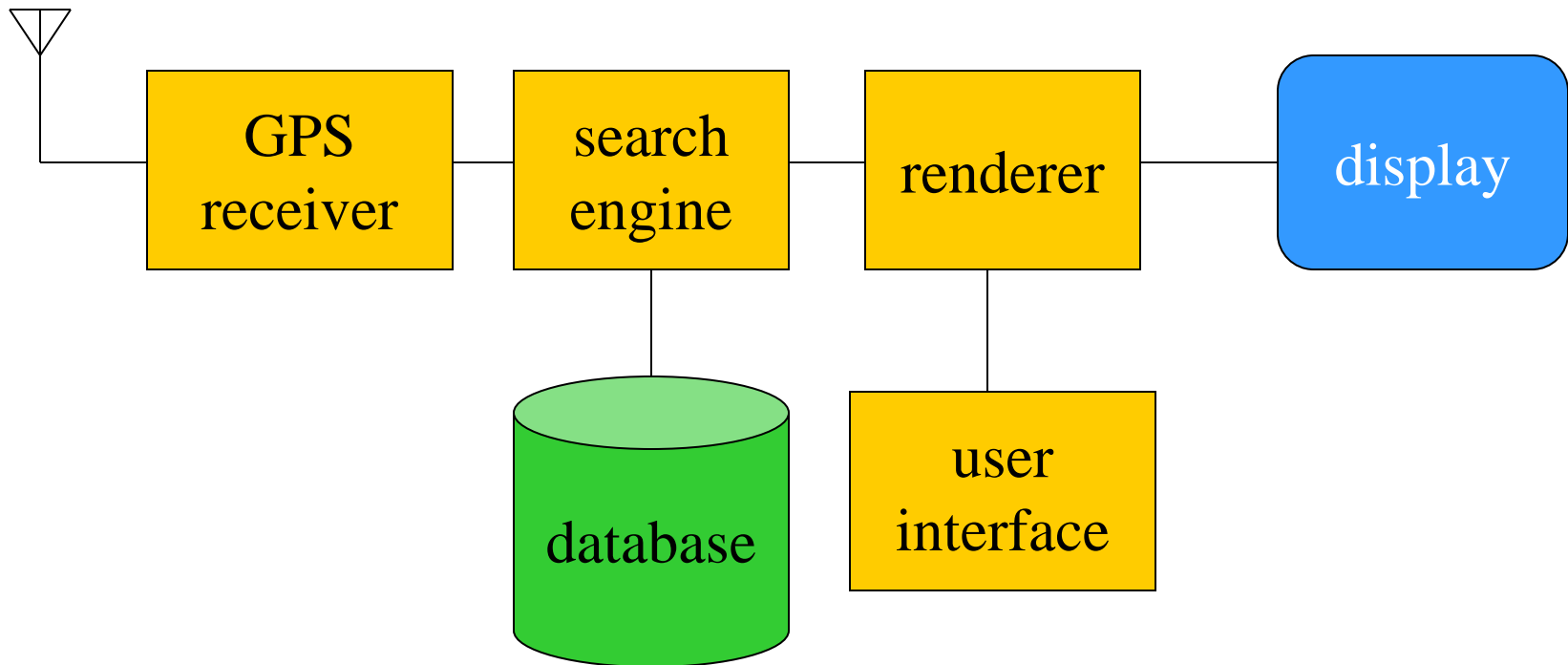
- ☑ data received from GPS;
- ☑ map data;
- ☑ user interface;
- ☑ operations required to satisfy user requests;
- ☑ background operations needed to keep the system running.

Architecture design



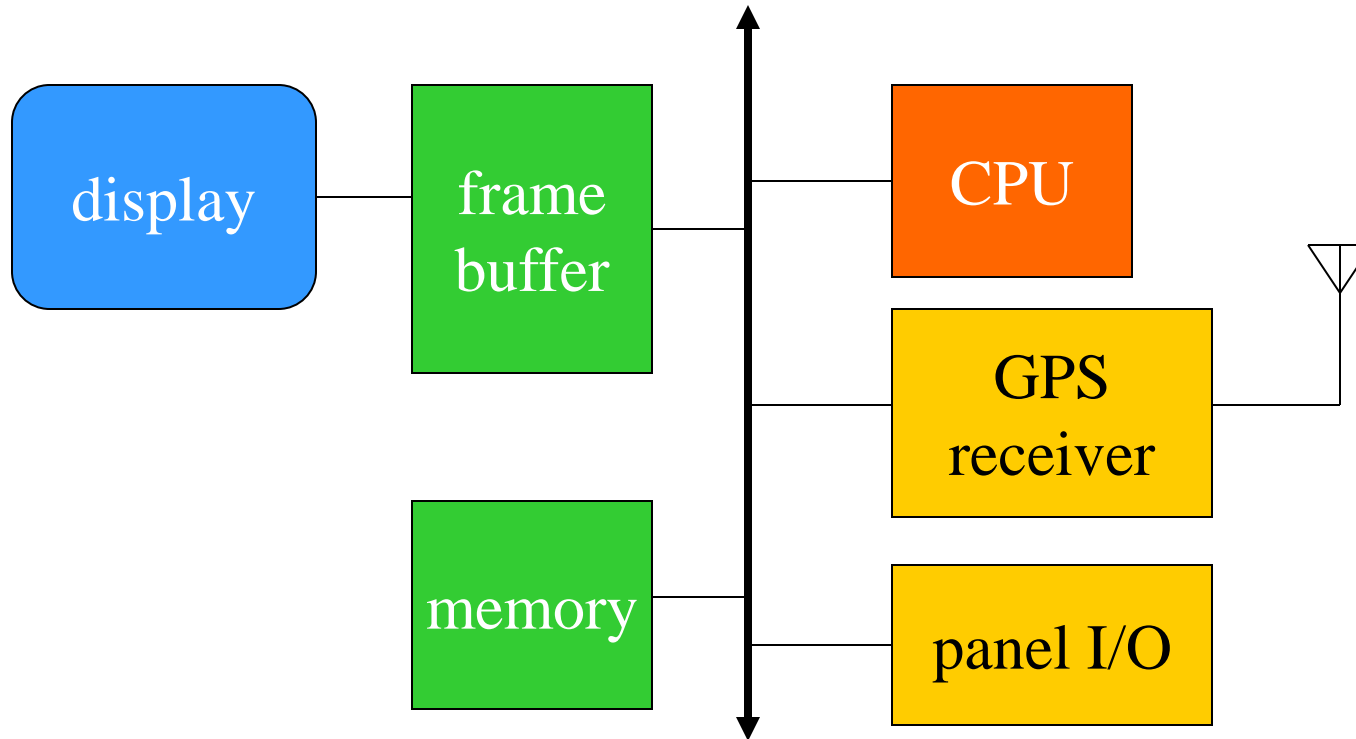
- ⌘ What major components go satisfying the specification?
- ⌘ Hardware components:
 - ☑ CPUs, peripherals, etc.
- ⌘ Software components:
 - ☑ major programs and their operations.
- ⌘ Must take into account functional and non-functional specifications.

Block diagram for GPS moving map

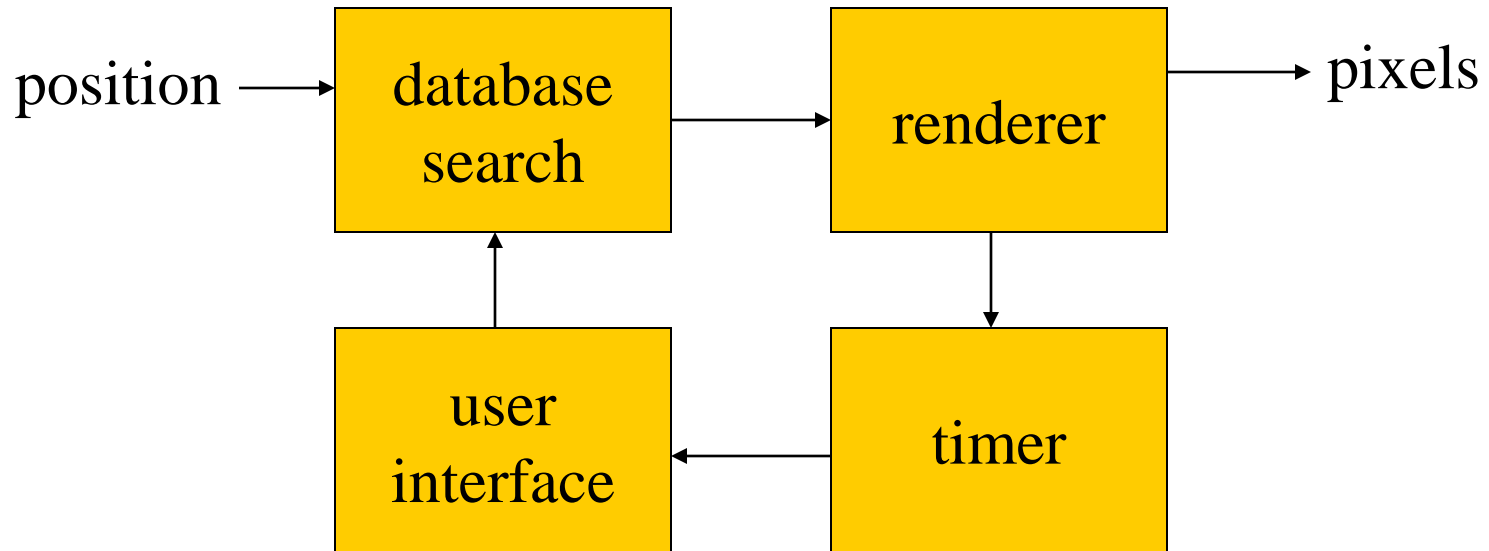


Still quite abstract!

Hardware architecture for GPS moving map



Software architecture for GPS moving map



Designing hardware and software components



- ⌘ Must spend time architecting the system before you start coding.
- ⌘ Some components are ready-made, some can be modified from existing designs, others must be designed from scratch.

System integration



- ⌘ Put together the components.
 - ☑ Many bugs appear only at this stage.
- ⌘ Have a plan for integrating components to uncover bugs quickly, test as much functionality as early as possible.

Summary



- ⌘ Embedded computers are all around us.
 - ☑ Many systems have complex embedded hardware and software.
- ⌘ Embedded systems pose many design challenges: design time, deadlines, power, etc.
- ⌘ Design methodologies help us manage the design process.