Embedded System

Prof. (Dr.)Amiya Kumar Rath

VSSUT, Burla

amiyaamiya@rediffmail.com

Important Features exhibited by Embedded system

1. Single functional system:

Most embedded system perform a single job repetitively.

Example:

A washing machine has an embedded controller that can take user inputs in terms of knob settings and perform the job of washing.

A cell-phone can receive and transmit signals to enable communication between two people.

Single functional system:

- Desktop is capable of doing a lot many operations.
- An embedded system will do the single function efficiently as compared to a general purpose computational system.

All embedded systems are not single functioned

- Example: cell-phones
- Other features: Able to send/receive SMSs, take photograph with add-on camera, tune to a radio station, play music, connect to the Internet and so on.
- But it cannot be utilized to perform complex scientific computation-unlike desktop
- Cannot be programmed for this purpose

2. Interaction with the physical environment

- Most embedded systems interact with the physical environment around them.
- Data collected from environment using sensors while actuators are used to control some of the parameters of the environment.

3. User Interface

- Common user interface in general computing systems: keyboard, mouse, screen
- Dedicated user interface in embedded system: push buttons, LEDs, steering wheels

4. Dependable system

- Ess used in safety-critical applications, like nuclear power plants, medical instrumentation etc.
- This demands a high degree of dependability on such systems.
- Dependable system must ensure easy maintainability, good availability, high degree of safety to the environment and security of information it processes.

5. Tightly constrained system

- ES design is constrained from several angles.
- <u>Example:</u>
- Should be a low-cost solution to the problem so that the overall system is cheap.
- Size of ES, its performance and power budget also put several constraint on the choice of the target implementation.
- For the battery to last long and reduced battery-pack size, the system must be a low power one.

6. Real-time system

- Most Ess are real time in nature.
- They must respond to a request from the environment within a finite and fixed amount of time.
- Failure to do so may lead to a catastrophic situation.

Contd…

- Example:
- Failure to activate fire extinguishers immediately after getting a fire alarm through sensors, may destroy the entire plant.
- Such systems are called hard real time system.

Contd…

- If the effect is not that serious, the system is a soft-real time system.
- Example:
- Failure to process the image just arrived may create some noise in the display of the image for some time.

7. Hybrid systems

- Many of the real-time systems are hybrid in nature, as they include both analog and digital components.

8. Reactive systems

- Reactive systems have continual interaction with the environment.
- Behaviour of the system is very much dependent on the events occurring in the environment.
- This type of systems normally have a set of states. Depending upon the occurrence of events, state transitions in the system take place.

Design metrics

- Design metrics are the optimization goals that an embedded system designer wants to achieve. Commonly used metrics are:

1. System cost

- Consists of two types of cost:
  - Non-recurring engineering (NRE) cost and
  - Recurring cost

NRE cost

- NRE cost is one time-the expenditure incurred in the design stage of the system.
- Once the system has been designed, extra units can be produced at a much lesser cost.
- This type of situation occurs commonly in designing VLSI chips.

Contd…

- NRE cost is very high as it includes the process of generating masks.
- Once the mask preparation is done, it can be replicated over a large silicon die to reduce the cost per unit.

2. Size

- Size of the system is very important.
- For hardware - Size is measured in silicon area.
- For software – Code size for the software portion of the embedded system.

3. Performance

- Refers to speed of the designed system.
- Normally, the specification of the system will have some performance requirements to be met by the design.
- This is one of the vital factors influencing the decision regarding the final implementation.

Contd…

- Example:
- Same functionality implemented in software will have lesser speed than a hardware realization.
- In the hardware realization also, an application specific integrated circuit (ASIC) will have better performance compared to Field Programmable Gate Arrays (FPGAs) or other general purpose processors.

4. Power requirement

- Important design metric
- Ess are expected to have light weight, long battery life.
- This necessitates plastic packaging, absence of cooling fans, etc.
- Thus, power requirement and the associated heat dissipation of the system should be very low.

5. Design flexibility

- It refers to the effort needed to modify a system if the specification changes to some extent later.
- While the software implementation is most flexible, ASIC is the least flexible one, with FPGAs lying at an intermediary stage.
- Main problem in the design change is the repeatation of the NRE cost which is the minimum for software.

6. Design turnaround time

- This is the time needed to complete the design starting from specification up to taking it to the market.
- Due to the very high rate of obsolescence of electronic goods, it is imperative that this time be small.

6. Design turnaround time (contd…)

- The requirement often forces the designers to use off-the-shelf components, rather than doing a costly redesign of system components.
- Design reuse is the key term.

7. System maintainability

- Refers to ease of maintaining and monitoring the health of the system after it has been put into the field.
- A good design is well documented such that even designers excepting those who designed the system originally, can modify the system, if necessary.

8. Testing and verification of functionality

- Refers to the ability to check the system functionality and get confidence regarding the correct operation of it.

Embedded system Design

Consists of several stages

1. System Specification
2. Behavioural specification
3. Register transfer (RT) specification
4. Logic specification

1. System Specification

- Design of any system starts at specification.
- Specification uses a language, may be simple English, some programming language like C or may be some formal technique using PetriNets, State Chart, UML chart and so on.

Contd…

- Automated tool – convert system specification in to a design.
- System synthesis tools – convert abstract system specification in to a set of sequential programs.
- The processes do interact between themselves to realize the overall functionality of the system.

Contd…

- Individual processes can be realized by general purpose processor or through dedicated processor.
- Any task can be implemented by either type of processors, however the speed will vary.

Contd…

- General purpose processor will have a software implementation of the task, while a dedicated processor can be implemented on FPGA or ASIC to have better performance.
- Decision regarding hardware or software implementation determined by the availability of pre-designed modules.
- Such modules form the system level library consisting of complete system solutions to previous problems.

Contd…

- Systems specifications are normally verified by using some formal tools known as model simulators/ checkers.
- These tools prepare a model of the entire system using some mathematical logic.

2. Behavioural specification

- System specifications refines to behavioural specification by the system synthesis tools.
- For each process behavioural specification is obtained.
- Some processes are marked for software implementation on general purpose processor, while some others are on dedicated hardware.

Contd…

- Behavioural specification is verified by hardware-software cosimulation.
- Individual simulation of only hardware or only software cannot bring out the total picture of the system.
- A joint simulation strategy is needed.

3. Register transfer (RT) specification

- Achieved through the refinement of behavioural specification.
- For the processes mapped on to general purpose processor, the software code is translated to the assembly/machine language instructions.
- A processor defines operation at register-transfer level only.

Contd…

- For dedicated hardware realization, synthesis tools (high level synthesis tools) converts the behavioural specification into a netlist of library components.
- This library includes description about RT components that may be used in the design at RT level, EX: registers, counters, ALUs, etc.
- RT specification can be verified by using RTL simulators normally available to simulate descriptions in hardware description languages, such as, VHDL, Verilog etc.

4. Logic specification

- Specifications of the dedicated processors is converted to logic specification.
- Logic specification consists of Boolean equations.
- Equations can now be converted to final implementation in some target technology.

Contd…

- Gate level simulators can be used to simulate the logic specification in terms of gates present in the circuit.

Look at this **sunflower**, a nature's gift – How does the nature embed its software? The flower rotates its face continuously towards the Sun.

Thank You