Design of a Digital/Computer System – which most senior design projects belong to

Charles Kim

Howard University

Objectives

- Identify the major tasks associated with the design of an electronic/digital system
- Familiarize with a range of alternative system implementation methods
- Understand the principal characteristics of the various device technologies and identify the appropriate techniques for a variety of applications
- Recognize the relative merits of programmable and non-programmable systems and to suggest an appropriate strategy for a given task

Process of System Design

- “A good design” : a design that solves a particular problem in the most appropriate and efficient manner.
- How to achieve a good design:
  - Know the problem
  - Know the available techniques and technologies
- Experience is important, so is a systematic and methodical approach
- System Design Methodology (of Top-down Approach) – our focus

Top-Down Design Approach

- **1. Customer Requirements** (“**Functional Requirements**” or “**Problem**”)
  - From Customer or final user
  - Expressed often in non-technical and sometimes imprecise terms
- **2. Top-Level Specification** (← “Design Requirements”← “Problem Definition”)
  - From the Customer Requirements
  - What the system is to do in response to all possible inputs
  - Set out any restrictions on the design of the system – physical size, weight, power consumption, or operating temperature limits
  - Man-machine Interface: how a user interacts with the system
    - This becomes now more important and plays an important part in determining the success or failure of the project.
  - System Block diagram
  - Identification of all inputs and outputs
  - Note: Spec sets out **what** the system is to do, but not to define **how** it is to do it.
  - Defines functions to be performed but does NOT dictate the methods to be used
  - The Specification must be approved by the customer as a correct expression of his/her requirements

Top-Down Design Approach - Continued

- **3. Technology Choice** (→ "Solution Generation" →)
  - What types of solution is suitable for the task?
    - Electronic solution --- analog, digital, or mixture of techniques?
    - Appropriate device technologies – discrete components, standard IC, PLD, FPGA, uP ?
    - Appropriate operating systems
    - Appropriate development environment of S/W and H/W
  - Detailed knowledge on the characteristics of the various components and devices
    - Speed
    - Cost
    - Analog/Digital I/O
    - Power Consumption
    - Operating voltage
    - Noise levels
    - Physical size
    - Temperature range
    - Development cost
  - Correct method of implementation
    - Important factor : success or failure of the project

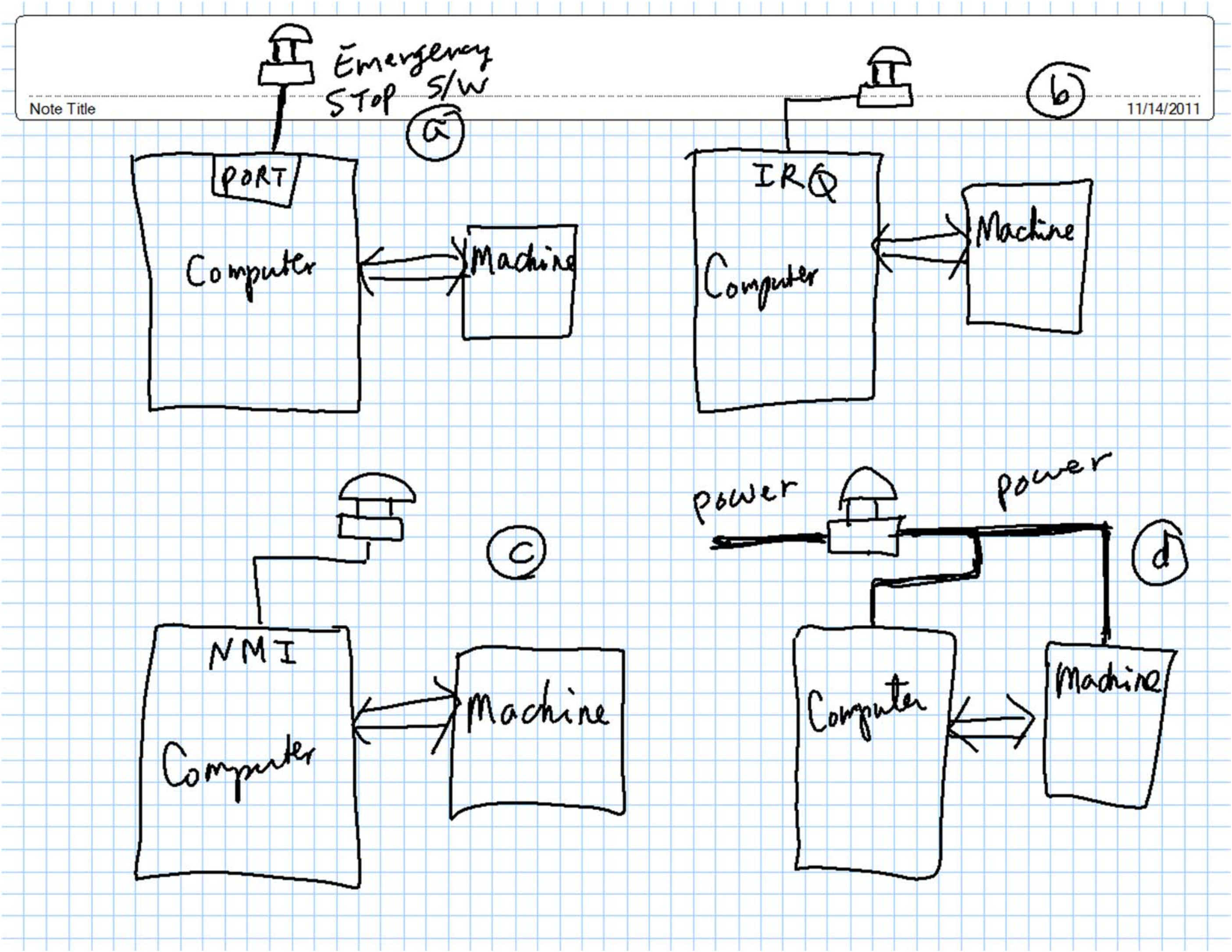
Top-Down Design Approach - Continued

- **4. Top-Level Design**
  - **System Partitioning**: Split the works into a number of modules to produce tasks of manageable size and determine the techniques to be used in each module to achieve its required functions.
  - **Hardware/Software Trade-Off**: Which parts of the system will be implemented in H/W and which in S/W (for uP based system).
  - Block Diagram form of system description
- **5. Detailed Design**
  - Detailed circuitry, H/W/, and S/W design
  - Implementation of various features and characteristics of the specification
- **6. Module Construction and Testing** (→“Implementation”)
  - Construction of modules
  - Testing of individual modules to ensure they conform to the specs
    - Revision of design and construction of the design
- **7. System Testing** (→“Verification and Evaluation”)
  - Assembly of modules and complete a system
  - Verification and assessment to ensure that it meets the top-level specs
  - Modification of the system if necessary

Design Study

- Some aspects of fairly conventional designs may have safety implications which must be treated seriously to ensure that they operate correctly
- Example:
  - Consider how an emergency stop button should be interfaced to a microcomputer based machine control system to ensure its correct operation
  - 4 possible approaches

4 different methods



4 Methods - description

- (a)
  - Parallel input port to S/W
  - Poll periodically: sense and act
  - After system crash?
    - Noise spike → error in execution of program→ missed S/W operation
- (b)
  - IRQ (Interrupt request) line to the S/W
  - After system crash?
    - Maskable interrupt – maybe enables or disabled
- (c)
  - NMI (non maskable interrupt) line to the S/W
  - IRQ always accepted
  - After system crash?
    - Random hardware failure → missed S/W operation
- (d)
  - Main power supply line to S/W operation
  - Safety function is provided by Power S/W **not by computer**
  - **Is this always better?**