Industrial Embedded Systems
- Design for Harsh Environment -

Dr. Alexander Walsch
alexander.walsch@ge.com

IN2244

Part I

WS 2013/14

Technische Universität München

Motivation

- What is critical infrastructure?

Motivation

- What is critical infrastructure?
  (from wikipedia) a term used by governments to describe assets that are essential for the functioning of a society and economy. Most commonly associated with the term are facilities for:
  - electricity generation, transmission and distribution
  - oil and gas production, transport and distribution
  - telecommunication
  - water and food supply
  - heating (e.g. natural gas, fuel oil, district heating)
  - public health (hospitals, ambulances)
  - transportation systems (fuel supply, railway network, airports, harbors)
  - financial services (banking, clearing)
  - security services (police, military).

Source: GE O&G

Motivation II

- How does this relate to this lecture?
  Critical infrastructure relies on computer systems, sometimes deeply embedded (depending on the hierarchical control level and the control strategy) – a hidden technology:
- No-one cares if they do their job. A disaster if they fail.
- It is not only important what we do but also how and how well we do it
  - process (especially in a regulated environment)
  - integrity (there are a few but one counts more than others): *reliability*
- Why will it be even more important in the future?
  - More and more electric systems (electrification)
  - Autonomous systems (no human in the loop or teleoperation)

Learning Objective

- Requirements analysis (strategy, tools)
- Reliability in embedded systems design (HW + SW)
- Functional safety
- Design patterns (HW+SW)
- Documentation guidlines
- V+V basics

We will focus on small footprint systems. A detailed understanding of hardware is essential. An organized approach to software development is key to address quality.

Why you Should attend

- You get credit
- Computer system design is about understanding and answering high-level requirements. Those requirements are usually very similar at a high level:
  - Reduced life-cycle cost
  - Increased availability
  - (sometimes) Safety
  - Added functionality
  - User experience – this is a rather new topic
  - (sometimes) Security – this is a rather new topic

Some Books I Recommend

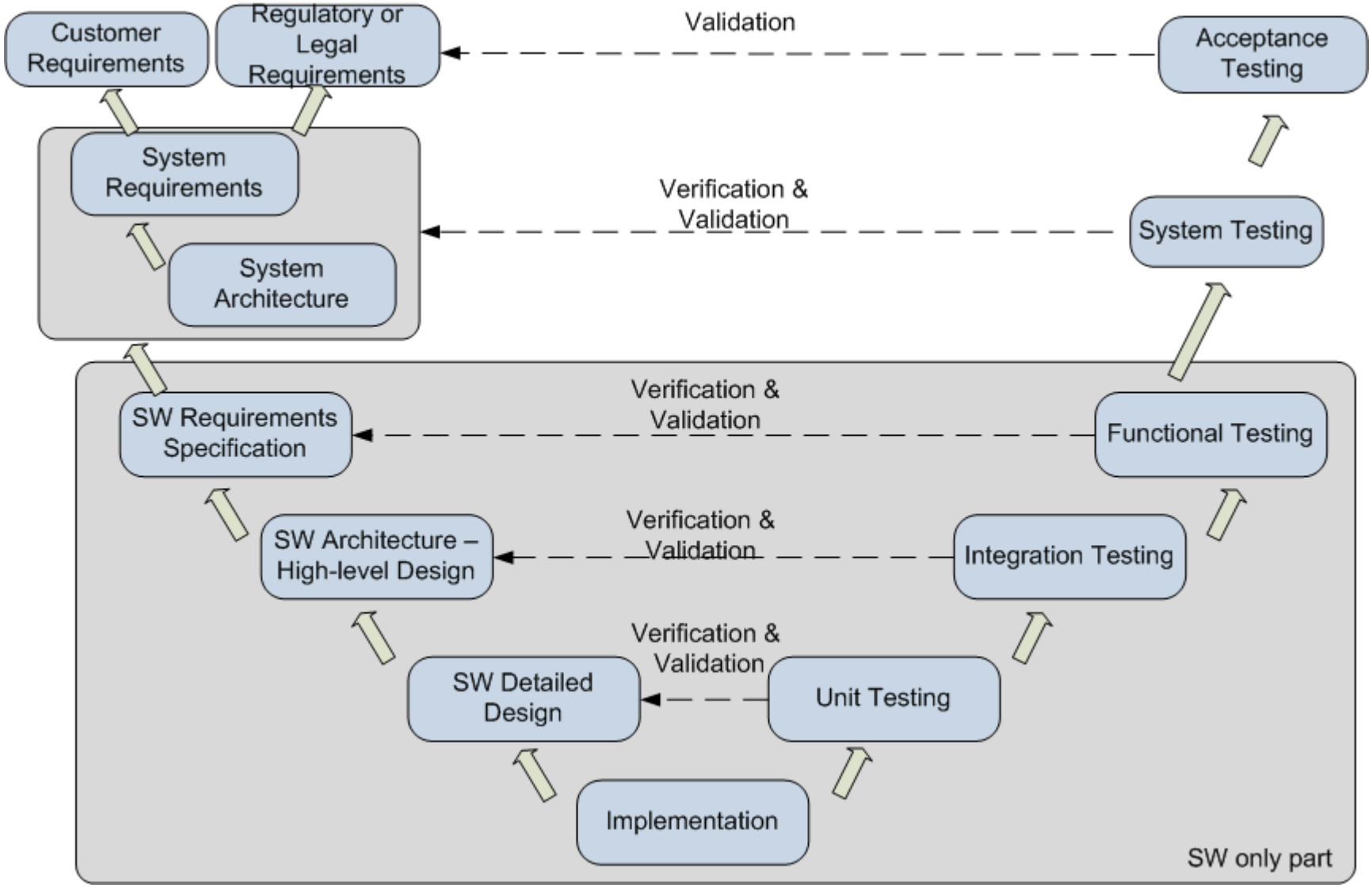
D. Patterson, J. Hennessy	Computer Organization and Design: The Hardware/Software Interface	Computer architecture textbook
A. Tanenbaum	Operating Systems Design and Implementation	Operating systems textbook
J. Labrosse	MicroC/OS-II	Good introduction to RTOS
D. Smith	Reliability, Maintainability and Risk	Reliability textbook
B. Kernighan, D. Ritchie	The C Programming Language	Introduction to C
L. Hatton	Safer C	Introduction to common pitfalls when using C
D. Smith	Safety Critical Systems Handbook	Introduction to safety critical systems, especially taking IEC61508 into account
C. Ericsson III	Hazard Analysis Techniques for System Safety	Various techniques (FTA, ETA, FMEA, Markov, ...)
A. Spillner, T. Linz	Basiswissen Softwaretest	General textbook on testing

Lecture Organization

21.10.	Organization, Prerequisites
04.11.2013	Requirements, Reliability
18.11.2013	Safety, Risk, Hazards, Architectures
02.12.2013	Example system: PMU
16.12.2013	Hardware + Software Patterns
13.01.2014	Example PMU
27.01.2014	V&V
03.02.2014	Tbd/Exam (oral or written)

Homepage:

The V-Model


Customer Requirements
Regulatory or Legal Requirements
Validation
Acceptance Testing
System Requirements
System Architecture
Verification & Validation
System Testing
SW Requirements Specification
Verification & Validation
Functional Testing
SW Architecture – High-level Design
Verification & Validation
Integration Testing
SW Detailed Design
Verification & Validation
Unit Testing
Implementation
SW only part

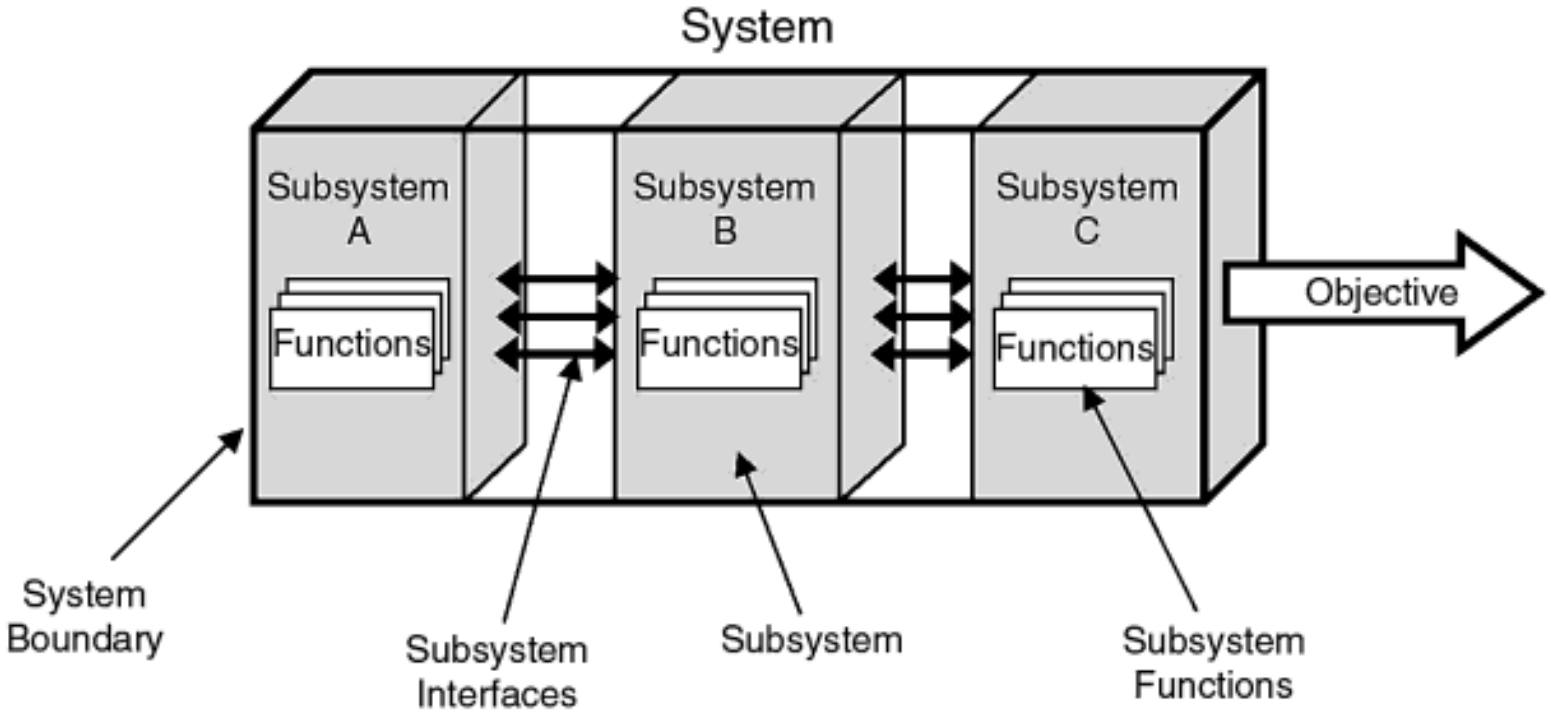
The V-Model II

- V-model on previous slide (covers system and software, y (vertical) = refinenement, x (horizontal) = time)
- A requirements driven approach
- Design gets more detailed as it proceeds in time
- Tests are defined at the time of design (testability - left wing of V)
- Tests are executed at the V+V stage (e.g. pass/fail - right wing of V)
- Iterative process that defines phases for design and testing
- Used for system, HW, SW
- Other processes: waterfall, spiral, agile
- Development process in general: defines steps, deliverables, reminds us so we do not forget anything

Systems
- What is a System? -

System border, subsystems, interfaces, functions, objectives, environment, external interfaces

High-level functions can be spread across subsystems. Components are basic building blocks (e.g. CPU).



Source: Ericson, Hazard Analysis Techniques for System Safety

Systems
- Environmental Constraints -

- Important environmental constraints:
  - Temperature (e.g. -40 to +85 °C ambiant)
  - Electromagnetic compatibility – EMC (conductive, inductive, capacitive, radiative coupling)
  - Shock (e.g. 2000 g)
  - Vibration (displacement, velocity, acceleration)
  - Radiation (increases with altitude)
- Impact: accelerated aging, more sensors, soft and hard errors

Probability Theory

- Uncertain (statistical) vs. deterministic outcome
- Think about a percentage (e.g. 1 out of a million) as in terms of frequency of occurrence
- Probabilistic models express scenarios with uncertain outcome and help us to assign a probability to certain sets of outcome (e.g. the probability that a certain component (or a set of components) will fail after a certain time).

We need this for reliability and safety calculations.

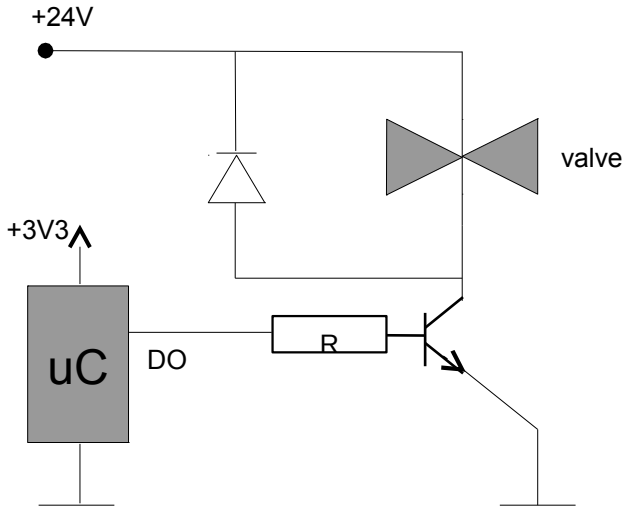
System Classification

- Control systems – open loop/closed loop
- Monitoring systems – monitoring, diagnostics, visualization
- Protection systems – emergency shutdown (ESD)
- Combination of the above (not always possible because of mixed criticality)

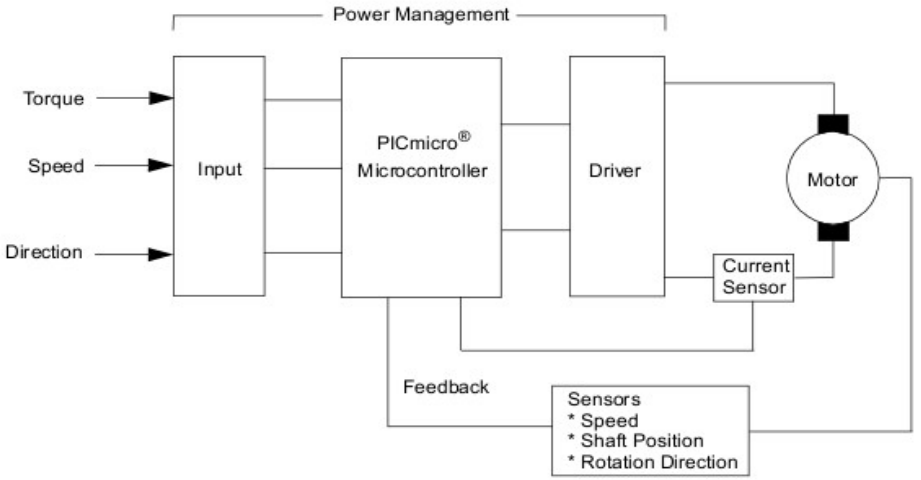
Control Systems

- Control systems – open loop (on/off) and closed loop (feedback)

open loop: solenoid valve



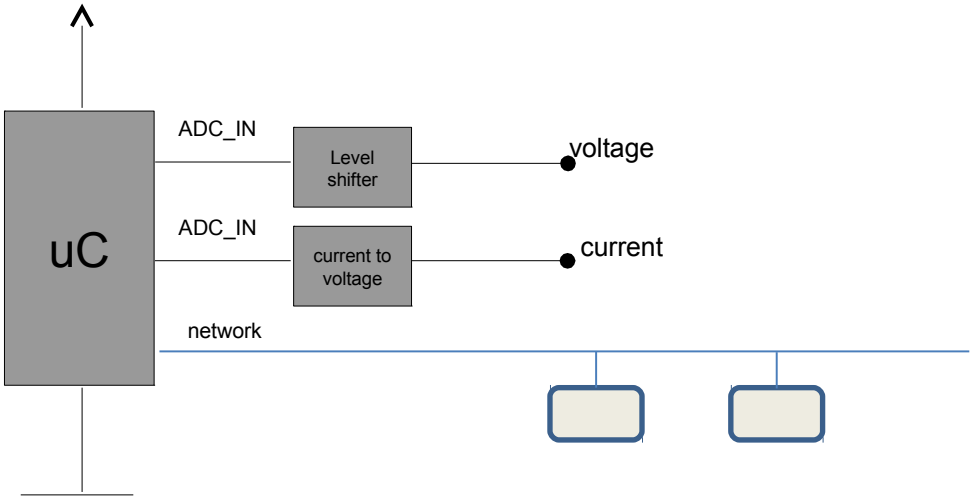
closed loop: BLDC motor



Source: Microchip AN894

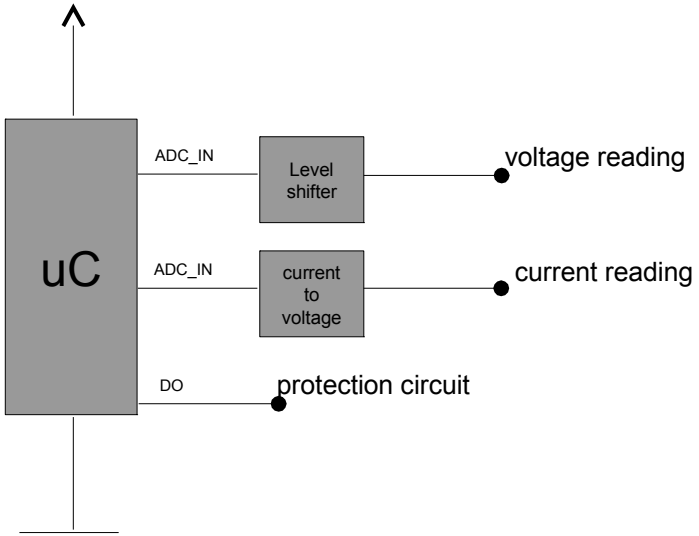
Monitoring Systems

- Monitoring systems – process/environment monitoring and internal diagnostics (e.g. sensors)
- Collection of data (use: on-board diagnostics, RM&D, CBM; SCADA).
- Recording systems (”flight recorder”).



Protection Systems

- Protection systems – ESD (fail safe/off)



Source: VW

- Protection systems have the sole purpose of putting the system into a safe state upon fault detection (fail safe). The safe state needs to be maintained until a clearance operation has been carried out (e.g. safety chains).

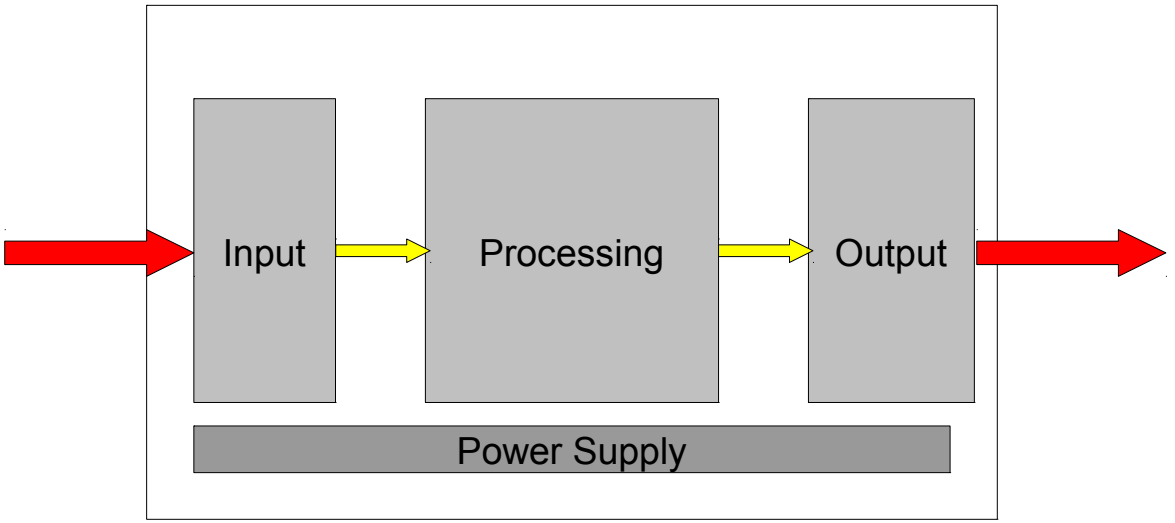
Electronics

- Software is a configuration of the hardware
- Hardware is used to run
  - Analog functions (continuous in time and signal)
  - Digital functions (discrete in time and signal)
- Software functions can fail because hardware
  - is used outside its specification
  - de-rates (gets old)
  - exhibits faults that are linked to the manufacturing process

We will use electronic circuit diagrams in this lecture.

Computer Engineering

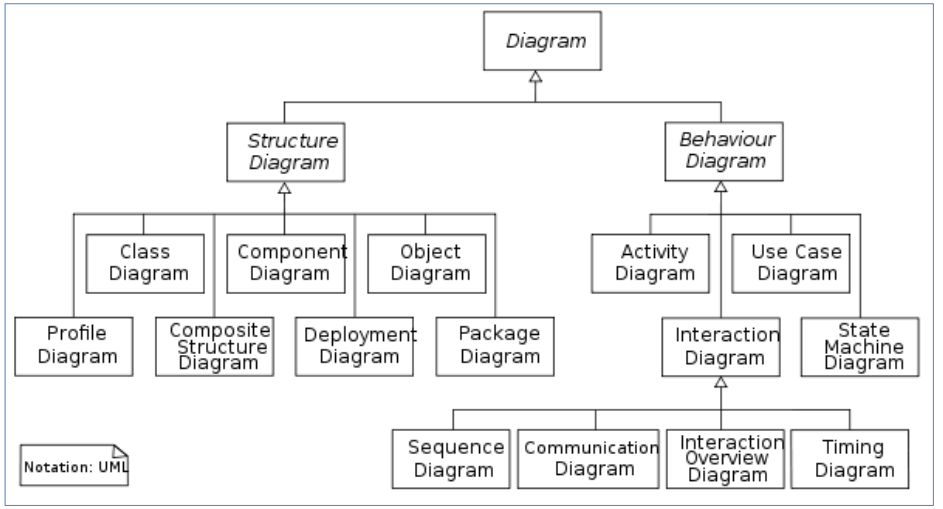
- A function is mapped to an embedded computer system.
- The embedded computer system has input, processing logic, and output.



We need a general understanding on computer system components.

Software Engineering

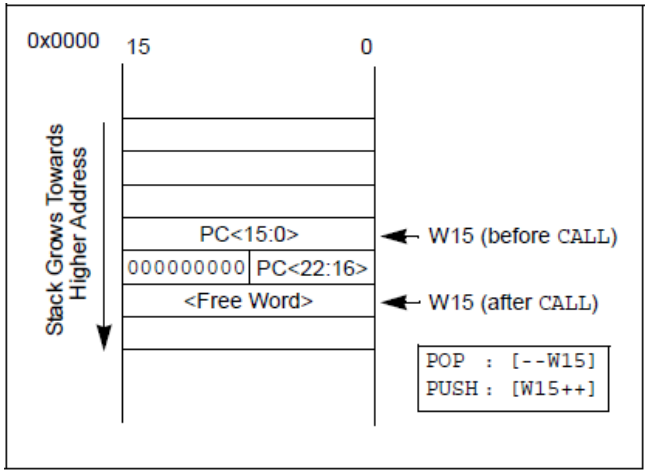
- A function is (partially) mapped to software.
- Software can be specified by models (OO - UML, functional – Simulink, block diagram,...) or programming languages.

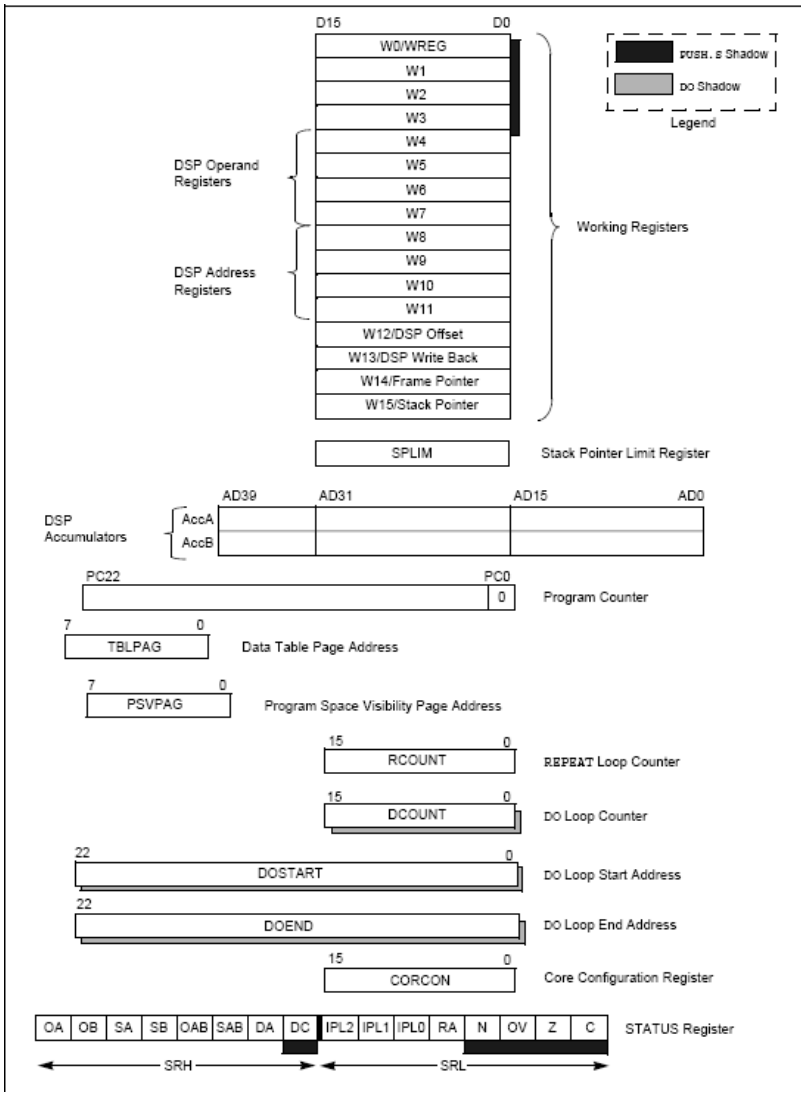


We will use some graphical model and C notation.

Programmer's Model

- A programmer's model is an abstract or conceptual view of the structure and operation of a computing system. A microprocessor's programming model shows its register file (e.g. data, address and special registers like stack pointer (here W15)).





Source: microchip.com

IEC 61508

- General safety standard as a guideline
- Risk based approach to safety
- Importance of architecture
- Contains a lot of guidelines on hardware and software design
- We will try to not use terminology of the standard but rather concentrate on its concepts
- Link: http://en.wikipedia.org/wiki/IEC_61508

The safety standard will serve as a design cookbook.

Example System - PMU

- PMU: Pressure Measurement Unit
- Will be used as an example virtual technology development
- Measures pressure, temperature compensation, inexpensive, CAN interface
- We will work on the design throughout the lecture
- Will use as a demonstrator for our safety and reliability methodology

Backup

Probability
- Example -

A patient is admitted to the hospital and a potentially life-saving drug is administered. The following dialog takes place between the nurse and a concerned relative.

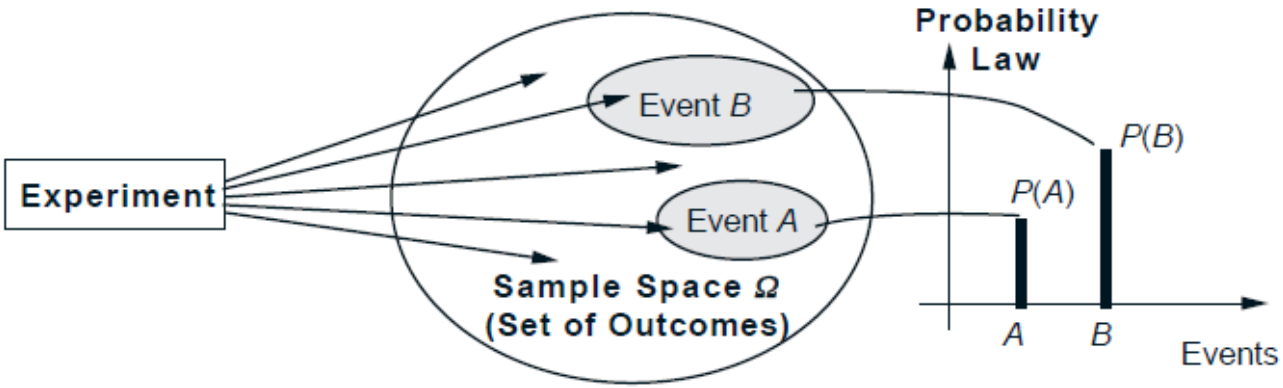
RELATIVE: Nurse, what is the probability that the drug will work?
NURSE: I hope it works, we'll know tomorrow.
RELATIVE: Yes, but what is the probability that it will?
NURSE: Each case is different, we have to wait.
RELATIVE: But let's see, out of a hundred patients that are treated under similar conditions, how many times would you expect it to work?
NURSE (somewhat annoyed): I told you, every person is different, for some it works, for some it doesn't.
RELATIVE (insisting): Then tell me, if you had to bet whether it will work or not, which side of the bet would you take?
NURSE (cheering up for a moment): I'd bet it will work.
RELATIVE (somewhat relieved): OK, now, would you be willing to lose two dollars if it doesn't work, and gain one dollar if it does?
NURSE (exasperated): What a sick thought! You are wasting my time!

Source: Bertsekas, Tsitsiklis Introduction to Probability

Probability
- Probabilistic Model -

Elements of a Probabilistic Model

- The **sample space** Ω, which is the set of all possible outcomes of an experiment.
- The **probability law**, which assigns to a set A of possible outcomes (also called an **event**) a nonnegative number $\mathbf{P}(A)$ (called the **probability** of A) that encodes our knowledge or belief about the collective "likelihood" of the elements of A. The probability law must satisfy certain properties to be introduced shortly.



Source: Bertsekas, Tsitsiklis Introduction to Probability

Probability
- Probability Axioms and Law Properties -

Probability Axioms

1. **(Nonnegativity)** $\mathbf{P}(A) \geq 0$, for every event A.
2. **(Additivity)** If A and B are two disjoint events, then the probability of their union satisfies

$$\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B).$$

Furthermore, if the sample space has an infinite number of elements and $A_1, A_2, \ldots$ is a sequence of disjoint events, then the probability of their union satisfies

$$\mathbf{P}(A_1 \cup A_2 \cup \cdots) = \mathbf{P}(A_1) + \mathbf{P}(A_2) + \cdots$$

3. **(Normalization)** The probability of the entire sample space Ω is equal to 1, that is, $\mathbf{P}(\Omega) = 1$.

Some Properties of Probability Laws

Consider a probability law, and let A, B, and C be events.

(a) If $A \subset B$, then $\mathbf{P}(A) \leq \mathbf{P}(B)$.

(b) $\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B) - \mathbf{P}(A \cap B)$.

(c) $\mathbf{P}(A \cup B) \leq \mathbf{P}(A) + \mathbf{P}(B)$.

(d) $\mathbf{P}(A \cup B \cup C) = \mathbf{P}(A) + \mathbf{P}(A^c \cap B) + \mathbf{P}(A^c \cap B^c \cap C)$.

Source: Bertsekas, Tsitsiklis Introduction to Probability

Probability
- Conditional Probability -

Properties of Conditional Probability

- The conditional probability of an event A, given an event B with $\mathbf{P}(B) > 0$, is defined by

$$\mathbf{P}(A \mid B) = \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(B)},$$

  and specifies a new (conditional) probability law on the same sample space Ω. In particular, all known properties of probability laws remain valid for conditional probability laws.

- Conditional probabilities can also be viewed as a probability law on a new universe B, because all of the conditional probability is concentrated on B.

- In the case where the possible outcomes are finitely many and equally likely, we have

$$\mathbf{P}(A \mid B) = \frac{\text{number of elements of } A \cap B}{\text{number of elements of } B}.$$

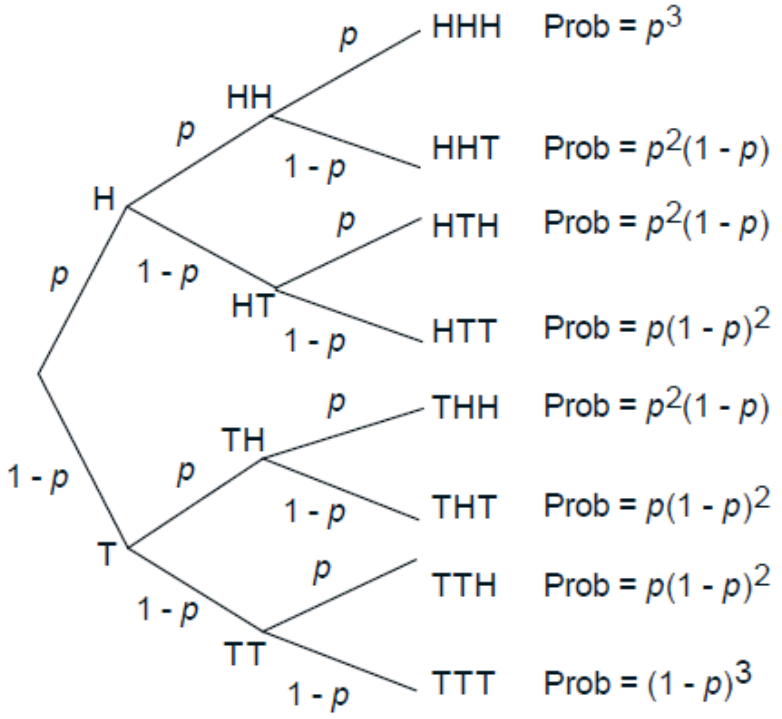
$$\mathbf{P}(A \cap B) = \mathbf{P}(A)\mathbf{P}(B)$$

If A and B are independent

Source: Bertsekas, Tsitsiklis Introduction to Probability

Probability
- Bernoulli Experiment -

- An experiment involves a sequence of independent but identical stages -> independent trials
- In the case of two possible outcomes (heads or tails)



Source: Bertsekas, Tsitsiklis
Introduction to Probability

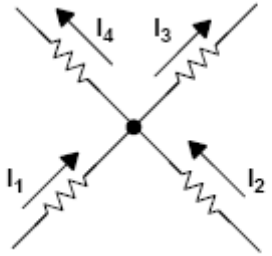
$$p(k) = \binom{n}{k} p^k (1-p)^{n-k},$$

where

$\binom{n}{k}$ = number of distinct n-toss sequences that contain k heads.

Electrical Engineering
- Analog Electronics -

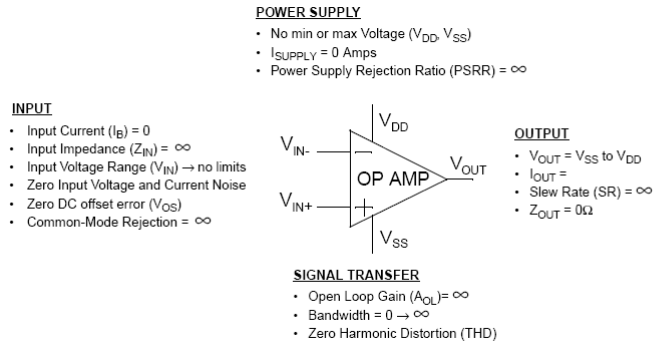
- An analog frontend to a digital computer must be able to provide elements for signal conditioning or math.
- A single component, the operational amplifier, is able of doing that depending on its configuration.
- An (ideal) op amp is an amplifier circuit with differential input and (infinite) large gain.
- Op amp is always used with a component network (R, C) and negative feedback
- Voltage calculations based on Kirchhoff's law, superposition, and „virtual mass" (Ud=0)



$$\sum V_{SOURCES} = \sum V_{DROPS}$$

$$\sum I_{IN} = \sum I_{OUT}$$

$$I_1 + I_2 = I_3 + I_4$$

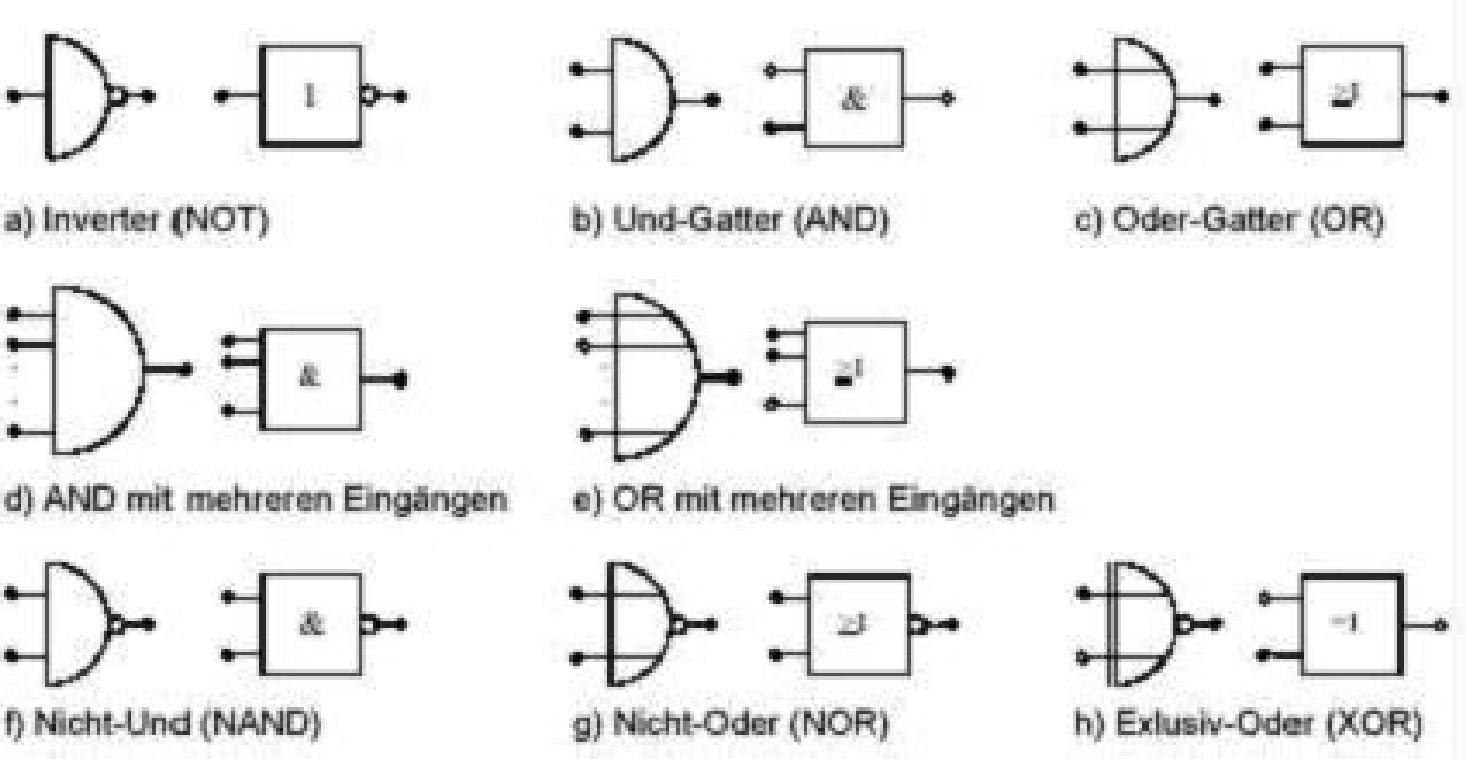


Source:
TI, Op Amps for Everyone

Source:
Microchip, AN722

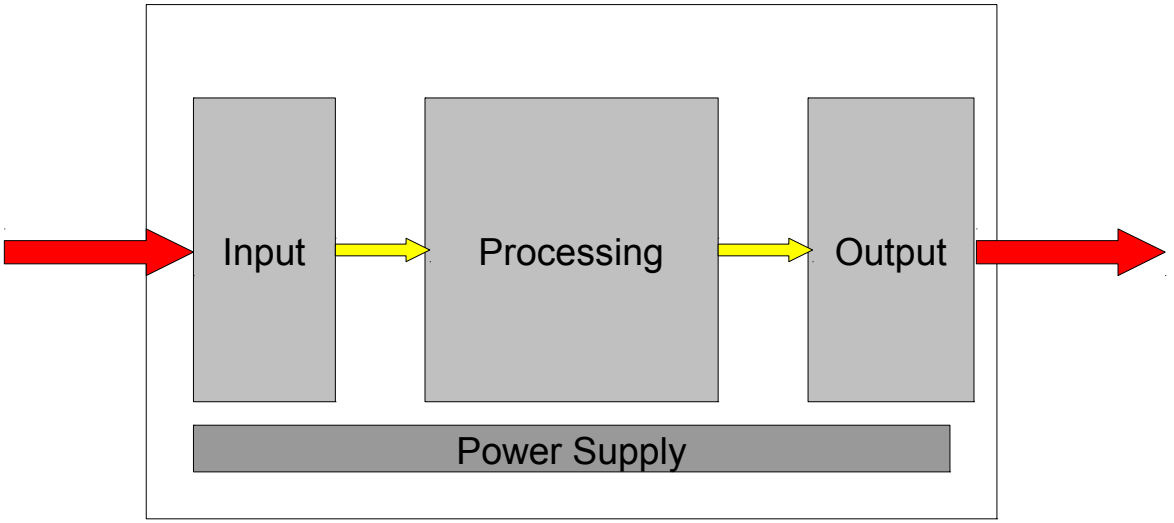
Electrical Engineering
- Digital Electronics -

- Combinatorial circuits – the output is a Boolean function of the input



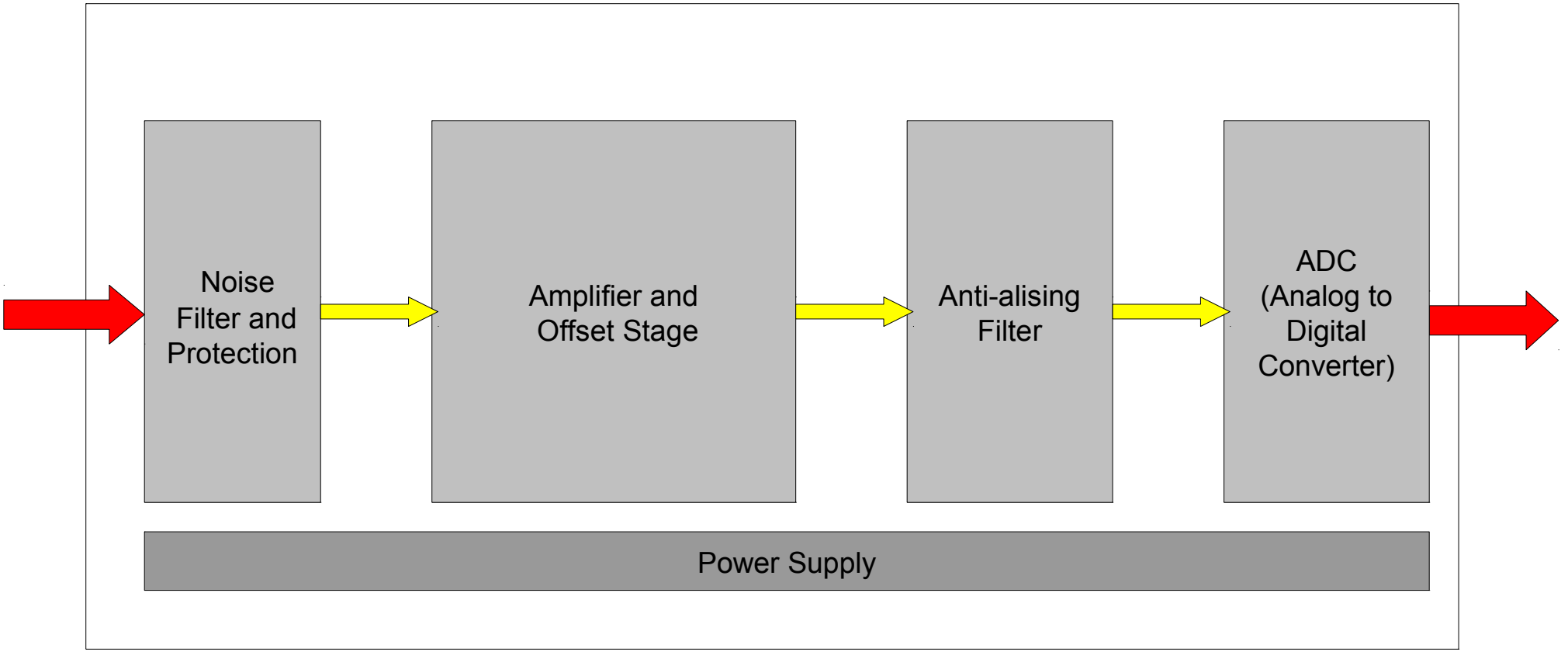
Source: http://www.virtualuniversity.ch/

Computer Science
- Digital Computer -



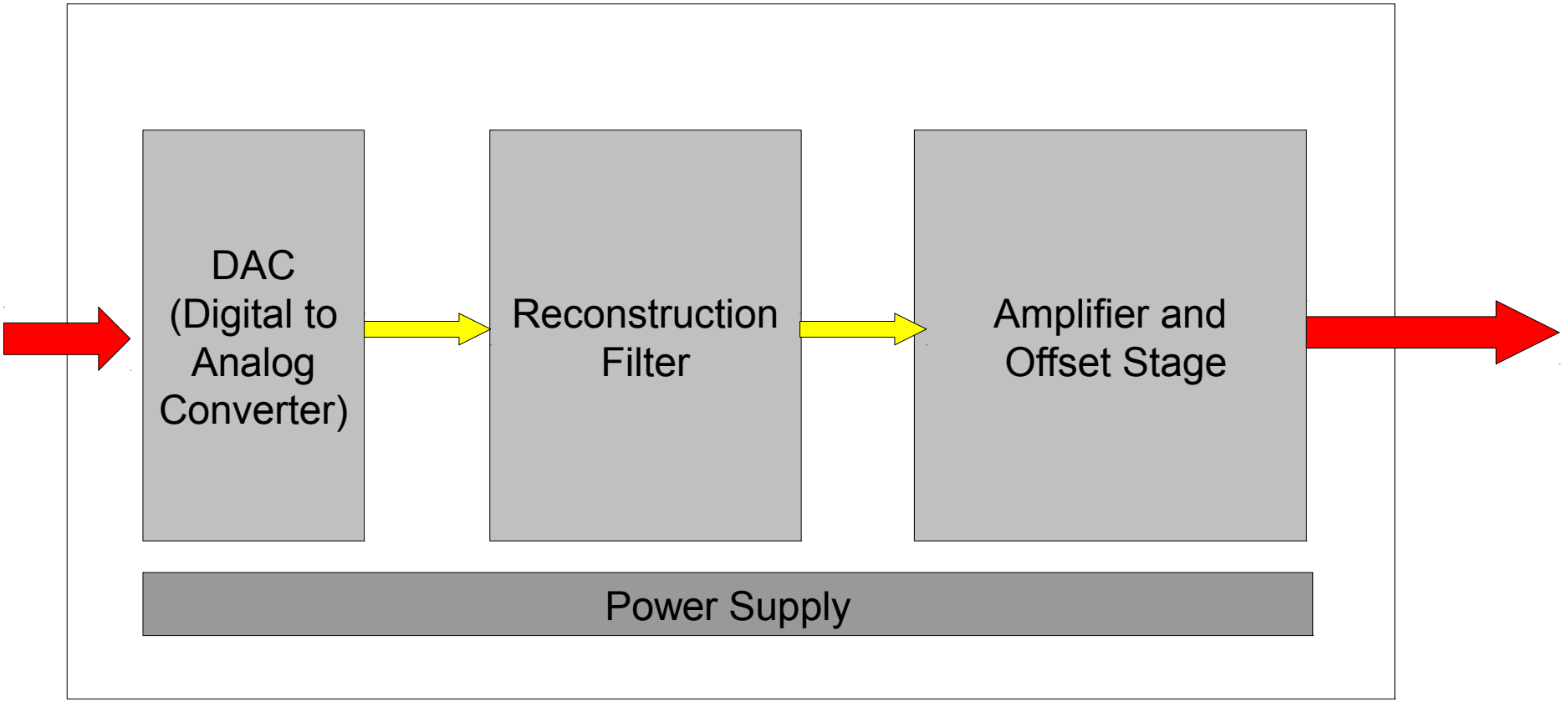
- The digital computer computes a result
  - Computer architecture (e.g. von Neumann) independent of calculation
  - The functionality is translated into an algorithm (= step by step recipe)
  - Real-time performance depends on I/O, instruction set, clock speed
- The digital computer is easily reconfigurable (software, stored program computer).
- The digital computer has (in theory) unlimited precision.

Computer Science
– Analog Input -

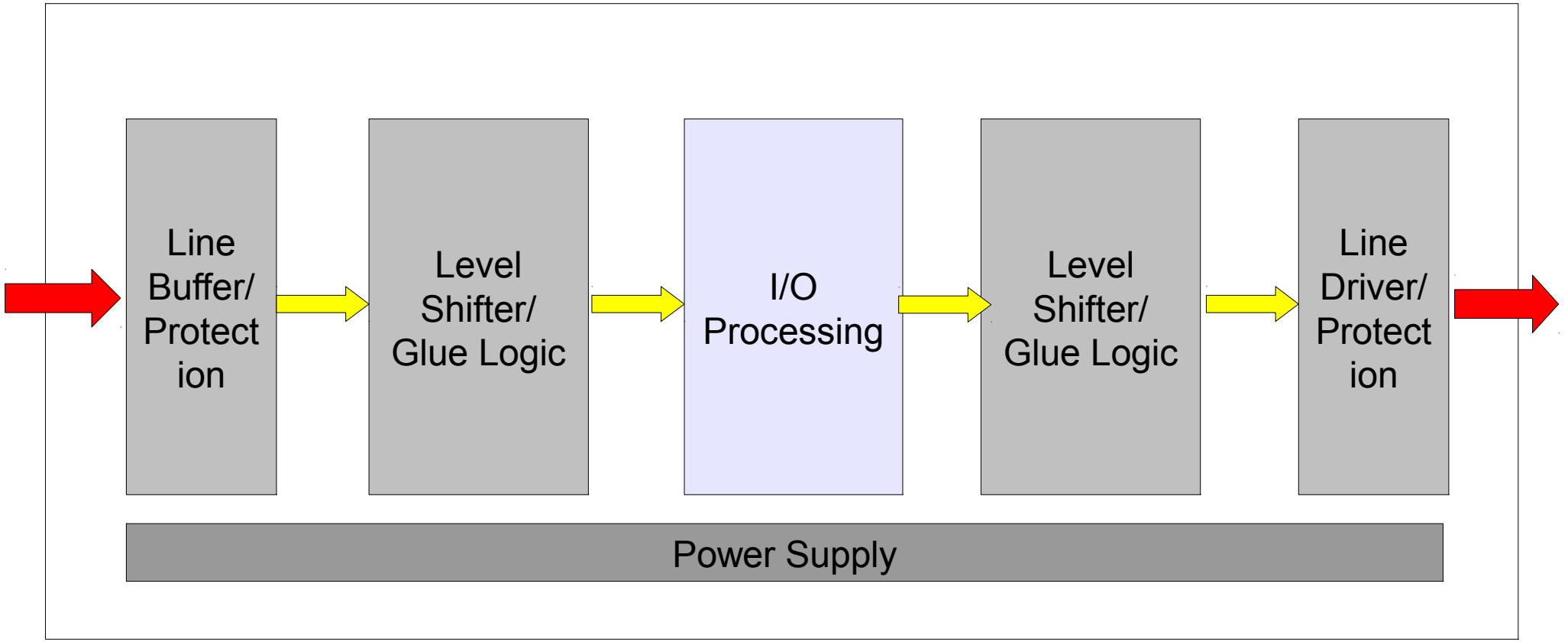


- Nyquist frequency fN = 0.5 x fS; fN is the highest frequency component that must be present at the ADC input → proper reconstruction

Computer Science
– Analog Output -



Computer Science
- Digital IO -



- Protective circuits can contain both transient, over/under-voltage protection and galvanic isolation.

Computer Science
- Definitions -

- Embedded computer systems are inside another device used for running one predetermined application or collection of software (according to this a computer without software is not an embedded computer system).
- Real-time performance requirement means that a function of the embedded computer system has an absolute maximum execution time.
- Hard real-time means that severe damage to assets or loss of life results on violation of the performance requirement
- Soft real-time means that the average time for a function is constrained and loss of performance (or quality) results on violation.

Computer Science
- Some More Definitions -

- Cyber-physical systems are networked embedded systems with a connections amongst each other or to the outside world.
- RTOS-based embedded systems run a real-time operating system. A real-time operating system provides support to meet real-time performance requirements.
- Bare-metal embedded systems are not running an RTOS. All requirements are realized without an abstraction layer running on the bare metal.
- Small footprint systems refer to limitations with respect to size, power, memory, etc.