Internet of Things

Lecture 14

Energy 1

Power and energy modelling and analysis and energy sources

with material by Peter Marwedel

Michael Engel

Power and energy in the IoT

Challenge:

Enable IoT edge devices to conserve energy as far as possible and to run with intermittent unreliable energy sources such as energy harvesting devices

Questions:

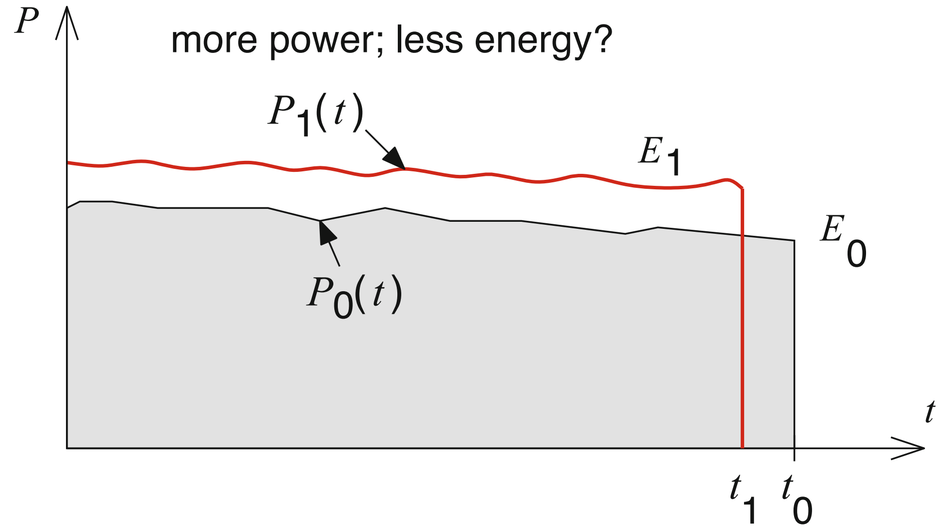
- How can we assess power and energy consumption?
- Which components influence power and energy consumption?
- How can energy be harvested from the environment?

Power and energy in CMOS semiconductors

Power dissipation in semiconductors is classified into

- **static power dissipation**
  - "*leakage power*" due to subatomic effects, e.g. tunneling
  - independent of frequency and switching of the system
  - dependent on supply and threshold voltage and the relation of transistor width W and length L: $P_{static} = \mathrm{f}(V_{dd}, V_{th}, W/L)$
- **dynamic power dissipation**
  - "*switching power*" due to switching between logic levels 1 and 0 (charge moving across the chip) and temporary short circuit paths from supply to ground
  - proportional to frequency f and square of supply voltage V_{dd}: $P_{dynamic} \approx f \cdot C_{eff} \cdot V_{dd}^2$
- Total power dissipation $P_{total} = P_{static} + P_{dynamic}$

Power and energy

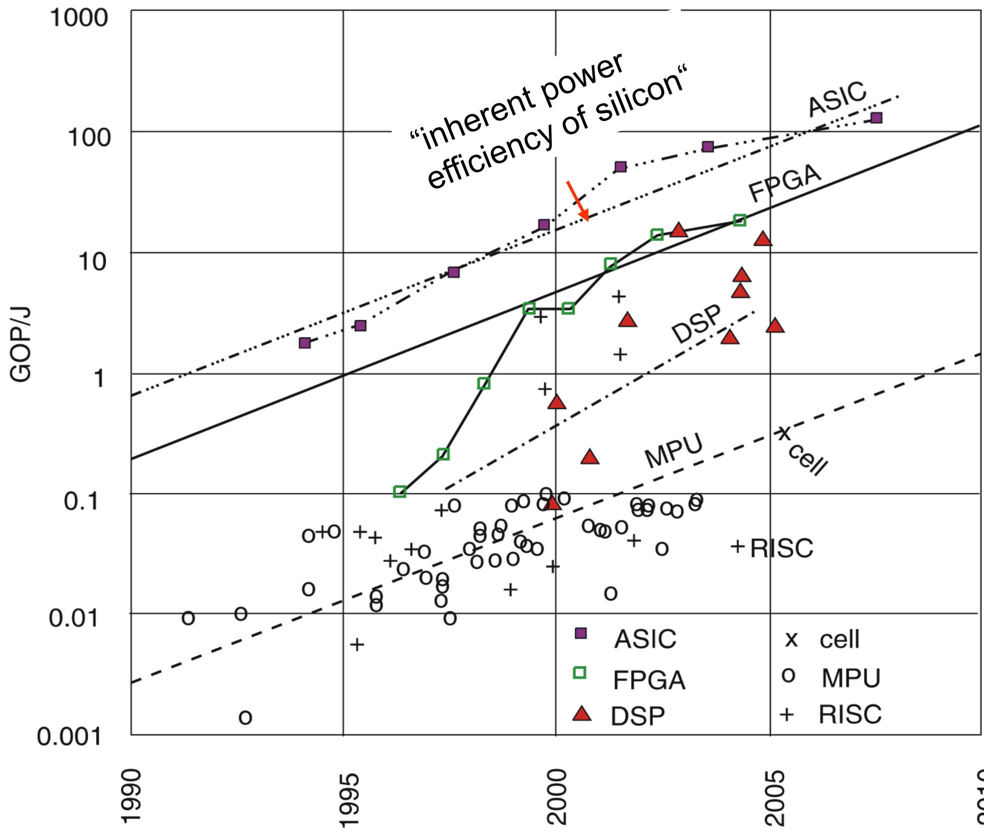


$$E = \int_t P\,dt$$

- The **energy** E is the **integral over the power dissipation** P over time t
- Power/energy models become increasingly important
  - due to mobile computing,
  - since energy availability becomes more relevant due to increased performance, and
  - due to environmental issues
- A higher power consumption can result in a lower energy consumption for a task if the task is finishes faster (see figure)
  - However, higher power consumption results in higher heat dissipation

Energy consumption trends

- Amount of computing per energy unit (Joule) steadily increasing
  - one order of magnitude difference between regular CPUs, digital signal processors (DSPs) and custom hardware (ASICs)



Unplugged embedded computing (or high-performance plugged)

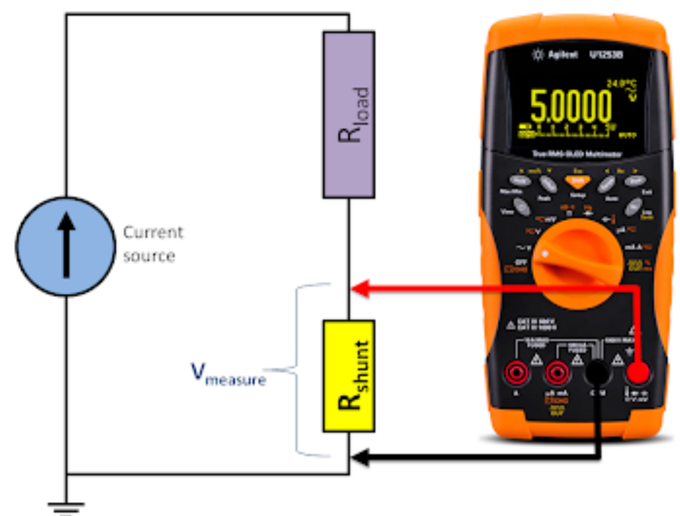
Computing, plugged

© Hugo De Man: From the Heaven of Software to the Hell of Nanoscale Physics: An Industry in Transition, *Keynote Slides*, ACACES, 2007

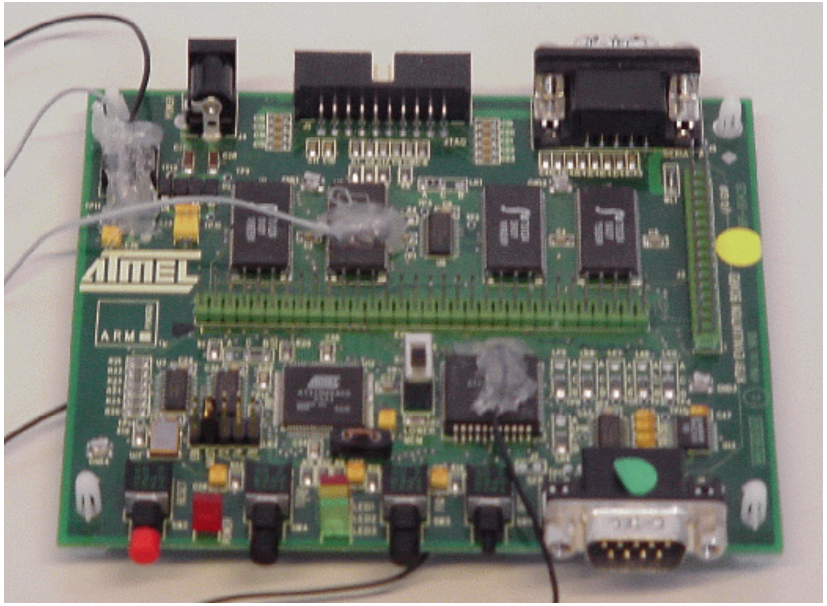
Measuring power

- Electrical power $P = U \cdot I$
  - product of supply voltage and current

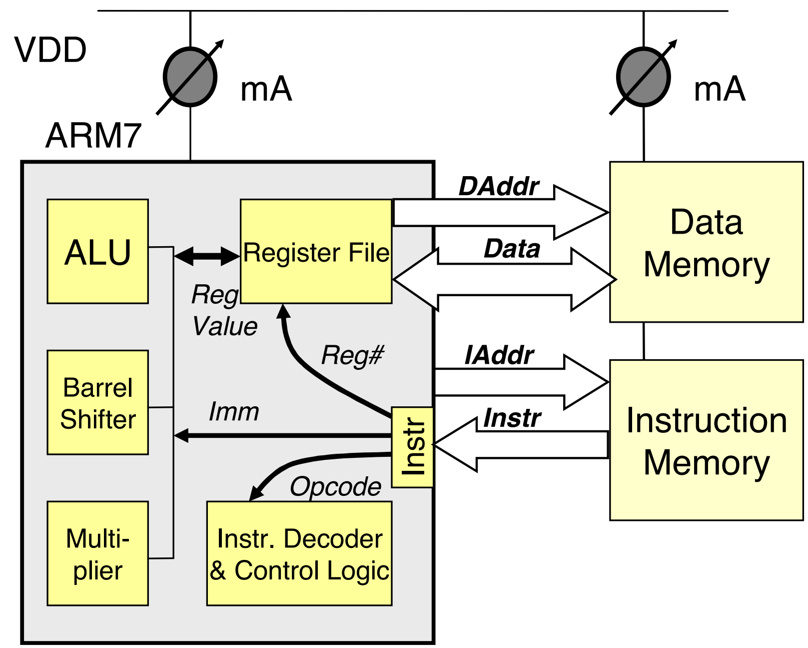
- Measurement using a *shunt resistor* R_{shunt}
  - highly precise resistor with very low resistance in the IC's (modelled as load resistance R_{load}) supply voltage line
  - low resistance (e.g. 0.1 V) important so the voltage drop across the resistor is low
  - *Assumption:*
    supply voltage is constant (e.g., 3.3V)



Steinke and Knauer model



Used measurements on an ATMEL board with ARM7TDMI CPU and external SRAM



The overall energy consumption is derived from instruction and data accesses in the CPU and memory:

$$E_{total} = E_{cpu\_instr} + E_{cpu\_data} + E_{mem\_instr} + E_{mem\_data}$$

Example: Instruction-dependent costs in CPU

- Cost for a sequence of *m* machine instructions:

$$
\begin{aligned}
E_{cpu\_instr} = \sum \mathrm{MinCostCPU}(\boldsymbol{Opcode_i}) &+ \mathrm{FUCost}(\boldsymbol{Instr_{i-1}}, \boldsymbol{Instr_i}) \\
\alpha_1 * \sum \mathrm{w}(\boldsymbol{Imm_{i,j}}) &+ \beta_1 * \sum \mathrm{h}(\boldsymbol{Imm_{i-1,j}}, \boldsymbol{Imm_{i,j}}) + \\
\alpha_2 * \sum \mathrm{w}(\boldsymbol{Reg_{i,k}}) &+ \beta_2 * \sum \mathrm{h}(\boldsymbol{Reg_{i-1,k}}, \boldsymbol{Reg_{i,k}}) + \\
\alpha_3 * \sum \mathrm{w}(\boldsymbol{RegVal_{i,k}}) &+ \beta_3 * \sum \mathrm{h}(\boldsymbol{RegVal_{i-1,k}}, \boldsymbol{RegVal_{i,k}}) + \\
\alpha_4 * \sum \mathrm{w}(\boldsymbol{IAddr_i}) &+ \beta_4 * \sum \mathrm{h}(\boldsymbol{IAddr_{i-1}}, \boldsymbol{IAddr_i})
\end{aligned}
$$

w: number of ones;
h: Hamming distance;
FUCost: cost of switching functional units
α, ß: determined through experiments

Other costs

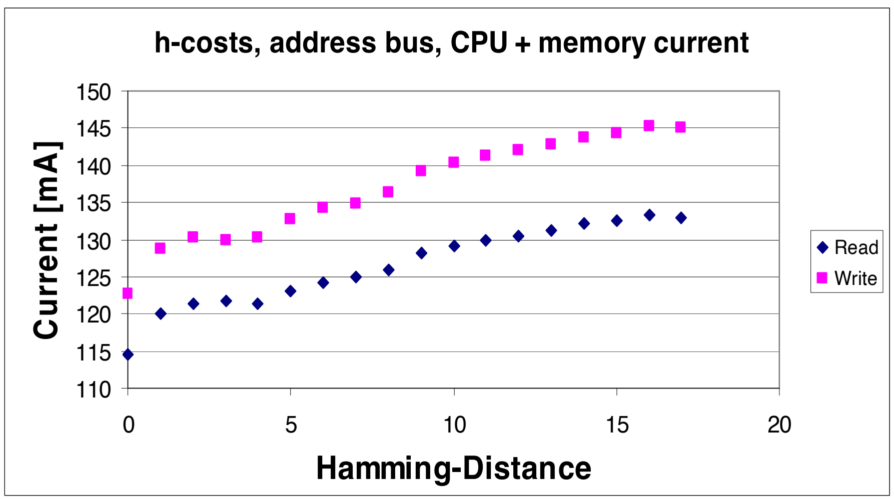
$$E_{cpu\_data} = \sum \alpha_5 * w(DAddr_i) + \beta_5 * h(DAddr_{i-1}, DAddr_i) + \alpha_6 * w(Data_i) + \beta_6 * h(Data_{i-1}, Data_i)$$

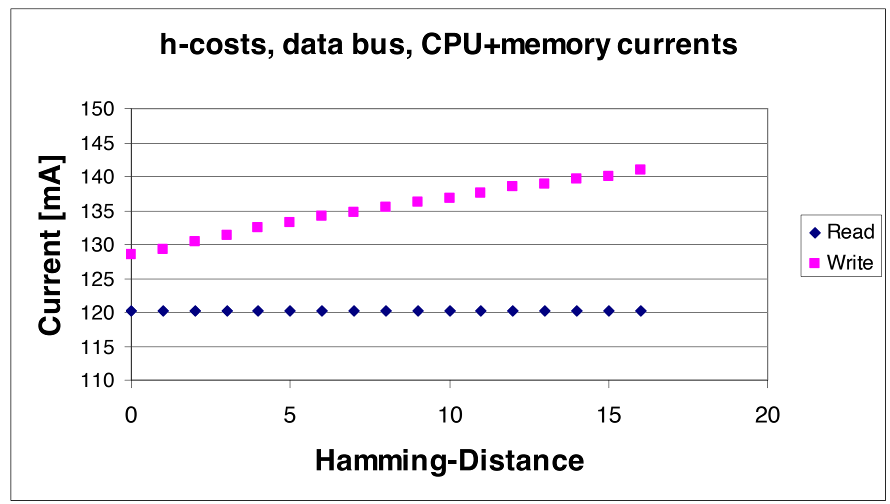
$$E_{mem\_instr} = \sum \text{MinCostMem}(InstrMem, Word\_width_i) + \alpha_7 * w(IAddr_i) + \beta_7 * h(IAddr_{i-1}, IAddr_i) + \alpha_8 * w(IData_i) + \beta_8 * h(IData_{i-1}, IData_i)$$

$$E_{mem\_data} = \sum \text{MinCostMem}(DataMem, Direction, Word\_width_i) + \alpha_9 * w(DAddr_i) + \beta_9 * h(DAddr_{i-1}, DAddr_i) + \alpha_{10} * w(Data_i) + \beta_{10} * h(Data_{i-1}, Data_i)$$

Which software effects influence power?

- Software behavior influences the switching behavior of a circuit
- Relevant effects:
  - *Hamming distance* between adjacent memory **addresses**
    - Instructions are fetched linearly (if no branch occurs)
  - *Hamming distance* between adjacent **values** on the data bus
    - Data bits switching between 0 and 1





Steinke's results

- It is not important which address bit is set to '1'
- The number of '1's in the address bus is irrelevant
- The cost of flipping a bit on the address bus is independent of the bit position.
- It is not important, which data bit is set to '1'
- The number of '1's on the data bus has a minor effect (3%)
- The cost of flipping a bit on the data bus is independent of the bit position

More on energy modelling

Energy analysis and optimization is still an active research topic

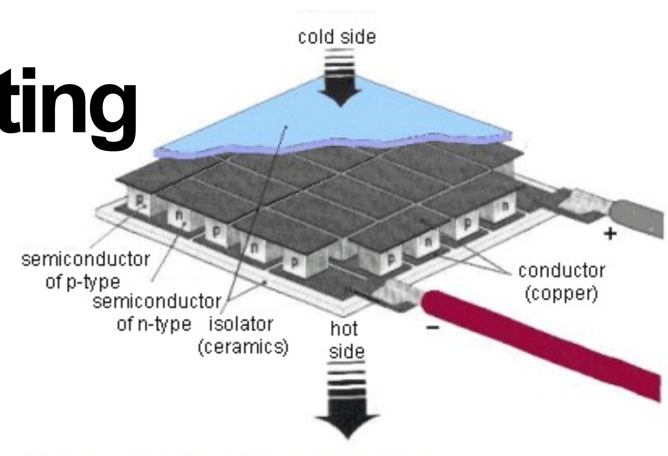
See e.g.

- Tiwari (1994): Energy consumption within processors
- Simunic (1999): Using values from data sheets. Allows modeling of all components, but not very precise.
- Russell, Jacome (1998): Measurements for 2 fixed configurations
- Steinke et al., University of Dortmund (2001): mixed model using measurements and prediction
- CACTI [2]: Predicted energy consumption of caches
- Wattch [3]: Power estimation at the architectural level, without circuit or layout

Energy sources

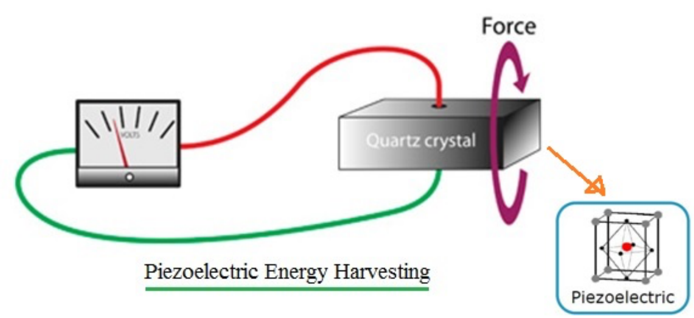
- Many IoT devices will not be connected to a permanent power source
- Instead, they have to rely on other energy sources
  - Batteries – low capacity, expensive, environmental problems
- Alternative: ***energy harvesting***
  - use of physical effects to convert mechanical or other forms of energy into electric energy
    - **Photoelectric** effect: light
    - **Piezoelectric** effect: mechanical strain
    - **Thermoelectric** generators: temperature difference
    - **Kinetic** energy: movement
    - Ambient **electromagnetic radiation**

Examples for energy harvesting



Thermoelectric Energy Transfer

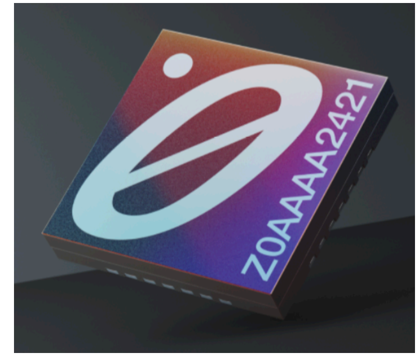
- **Thermoelectric generators** use ambient solar energy to convert temperature differences across a material into equivalent electric voltage or electric current
- The temperature difference on two sides of a crystal (i.e. one warmer and the other cooler) causes voltage across crystal device
  - Steady voltage is available when difference in temperature remains unchanged
- **Piezoelectric energy harvesting** converts mechanical stress into an electrical signal
  - The charge gets accumulated in solid materials due to application of mechanical strain
  - Common sources which can be exploited are low frequency vibrations, acoustic noise, human motion etc.



Piezoelectric Energy Harvesting

Examples for energy harvesting

- **Ambient electromagnetic radiation** harvesting: **ONiO.zero** chip [4]
  - 32-bit RISC-V based microcontroller
  - 1KB of mask ROM (stdlib, math etc)
  - 2KB RAM
  - 8/16/32KB ultra low power Flash



- Internal RF Rectifier:
  - Multi frequency 800/900/1800/1900/2400MHz bands (ISM and GSM) supported
- Internal Power Generation:
  - Voltaic cells down to 400mV (DC) Solar, piezoelectric and thermal (1V8 to 3.6V)
- Operating conditions
  - Operating voltage: 450 mV–1.8 V
  - Operating frequency: Asynchronous to 24 MHz

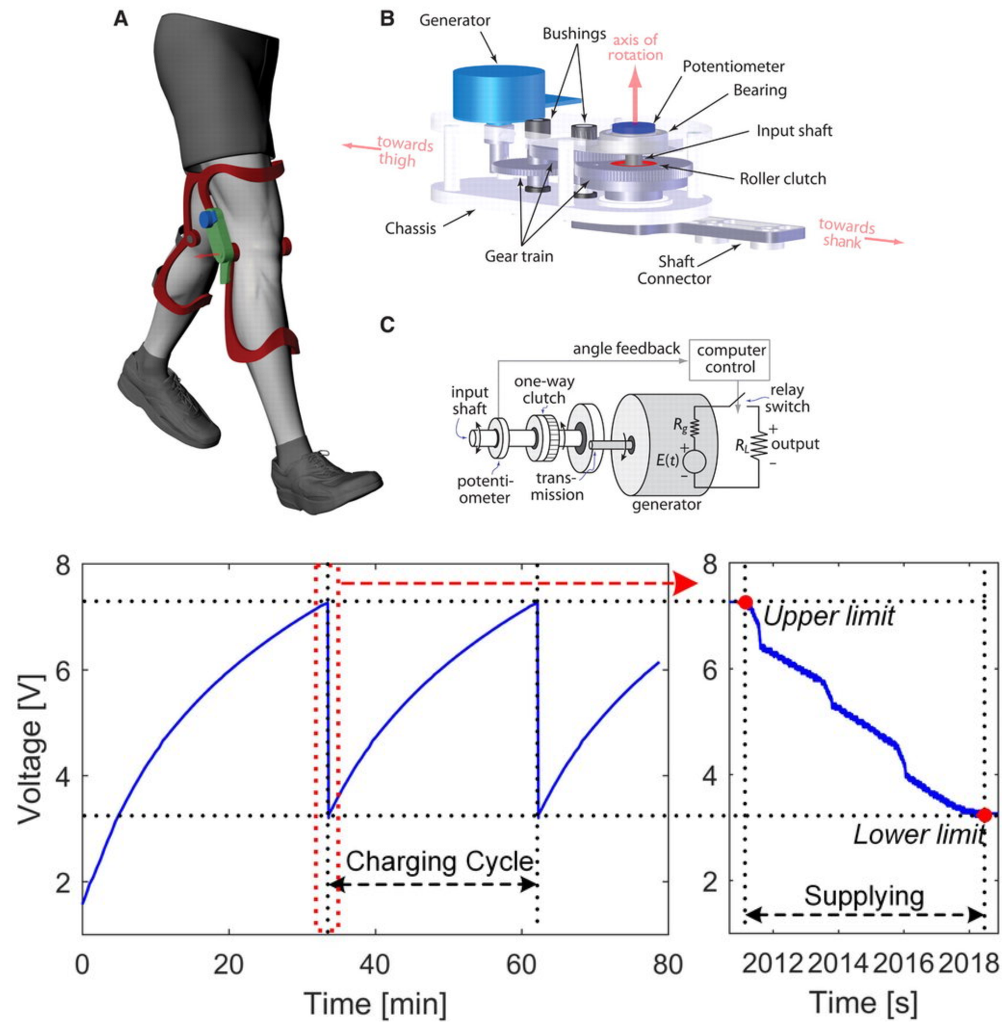
Energy storage

- Harvested electric energy is not always available
  - Store currently not needed energy for later use

- Two major storage devices:
  - **Capacitors**
    - Advantages: potentially fast charging process, very high output currents, close to 100% efficiency, low leakage currents (for high-quality capacitors), large number of charge/discharge cycles [5]
    - Disadvantage: limited amount of energy that can be stored
  - **Rechargable batteries**
    - based on chemical processes
    - limited amount of charge/discharge cycles

Energy harvesting example

- Idea: Use rotational energy as source [6]
  - low-frequency broad-band rotational energy harvesting solutions for self-powered sensing
- Approaches include electromagnetic, piezoelectric resonant, and piezoelectric non-resonant harvesters
  - **discharge duration << charge time**



Conclusion

- Energy consumption and generation is one of the central problems of IoT nodes
- Power and energy consumption of software can be modelled with high accuracy

- Energy to operate IoT nodes is often harvested from the environment
    - exploit physical effects to convert other forms of energy (e.g. mechanical) into electric energy
    - storage of energy in rechargeable components (capacitors or batteries)
- Often, recharging takes *significantly longer* than discharging
    - operation only possible during short amounts of time

References

[1] Stefan Steinke, Markus Knauer, Lars Wehmeyer and Peter Marwedel, *An Accurate and Fine Grain Instruction-Level Energy Model Supporting Software Optimizations,* Proc. of the International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2001

[2] Naveen Muralimanohar, Rajeev Balasubramonian and Norman P. Jouppi, *CACTI 6.0: A tool to model large caches*, HP laboratories 27 (2009):28

[3] David Brooks, Vivek Tiwari and Margaret Martonosi, *Wattch: A framework for architectural-level power analysis and optimizations,* ACM SIGARCH Computer Architecture News 28.2 (2000): 83-94

[4] ONiO: https://www.onio.com

[5] Sehwan Kim and Pai Chou, *Energy harvesting: Energy harvesting with supercapacitor-based energy storage*. Smart Sensors and Systems. Springer, 10.1007/978-3-319-14711-6_10

[6] Hailing Fu, *Rotational Energy Harvesting for Low-Power Electronics,* PhD thesis, Imperial College, London/UK, 2017