Energy Autonomous Wireless Systems (EAWS)

Prof. Catherine Dehollain
EPFL-IEL RFIC
Prof. Anja Skrivernik
EPFL-TCL LEMA
Prof. Franco Maloberti
Guest Lecturer, Univ. of Pavia
Prof. Andreas Burg
EPFL-IEL TCL

Course contents

Objectives :

PART-1

- To provide a basic introduction to power consumption in digital VLSI circuits

- To give an overview of the key techniques for designing low power digital VLSI circuits

- To raise awareness and introduce potential solutions for the difficulties associated with ultra-low-power design in advanced technology nodes

PART-2

- Provide an example of a low-power system design (ECG monitoring)

- Get acquianted with the anatomy of a basic low-power embedded system based on commercial off the shelf (COTS) components

- Lear about the various system components and modes to be able to make design decisions when using or selecting an embedded platform

- Collect some initial experience in using an embedded low-power system and its low-power modes

LESSON 2b – Low Power System Design

Prof. Andreas Burg

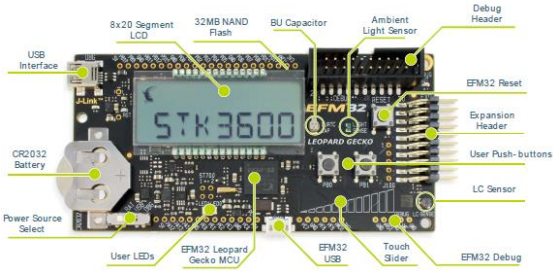
Case study: Energy Consumption of Wireless Transmission

Embedded Low-Power Microcontrollers

- Overview and Low-Power Modes
- Key Components

Hands-on Example: EFM32 ULP MCU

- Power Consumption and Low-Power Modes

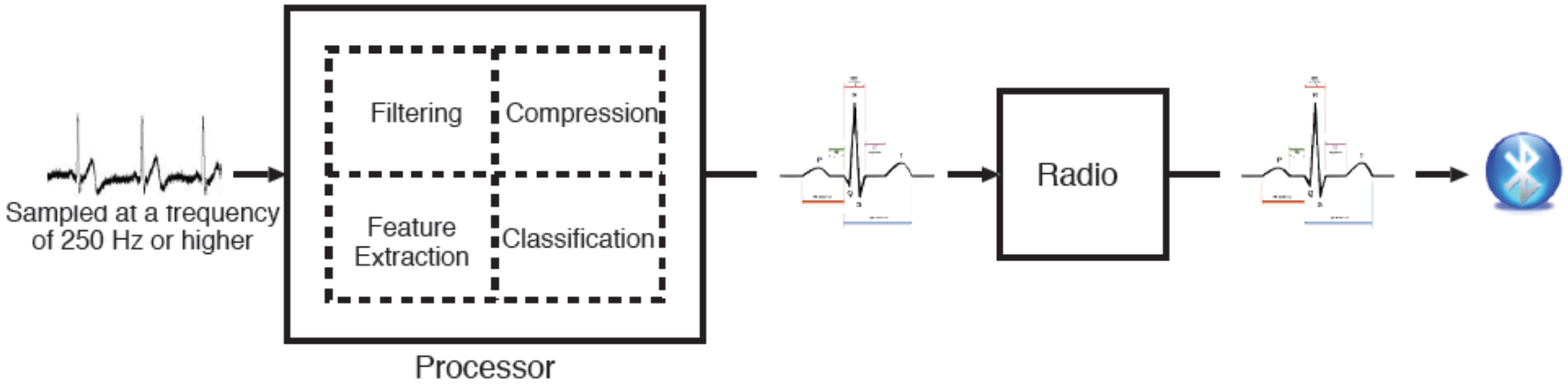


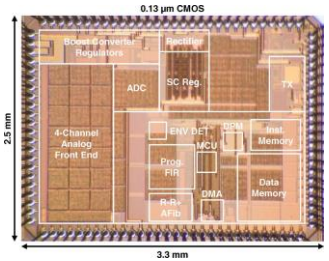
Case study: energy consumption analysis and optimization of wireless transmission

7 Reducing Radio Interface Power win On-Node Processing

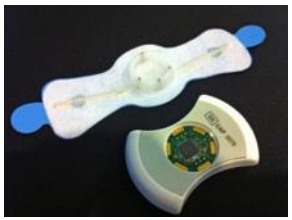
Reducing the amount of data to be transmitted generally reduces radio interface power consumption, but requires

- **reasonably efficient radio protocol (power-down between bursts)**
- **processing resources on the node to reduce data to the essence**





Zhang (2012)



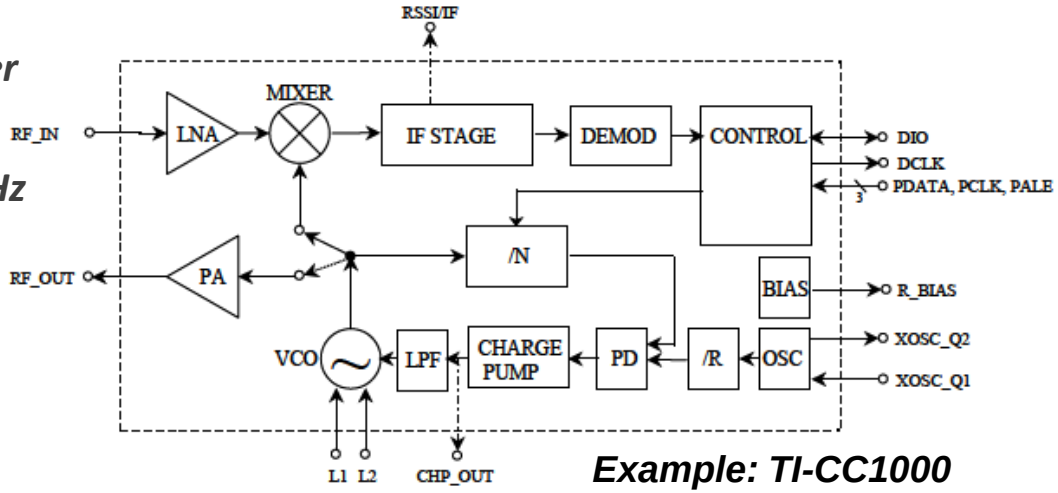
IMEC cardiac patch (Yazicioglu,2009)

Holst Centre (Masse, 2010-13)

Shimmer (shimmerresearch, 2010-13)

- Consider a system that comprises only the physical layer
- Simple RF radios are available as discrete components
  - TI, Analog Devices, Freescale, Microchip (often combined in SoCs with low power uControllers)

- A simple single-chip (UHF) transceiver
  - RF frequency synthesis from a low-frequency reference quartz (few MHz)
  - Modulation: alters the RF signal according to the data
  - Demodulation: analyzes the received signal to recover the data
  - Received signal strength indicator (RSSI): detects presence of an RF signal and its strength
- Example: TI-CC1000
  - *Combined transmitter and receiver*
  - *Software-programmable freq. band: 315 / 433 / 868 and 915 MHz*
  - FSK modulation up to 76.8 kbit/s
  - Programmable output power

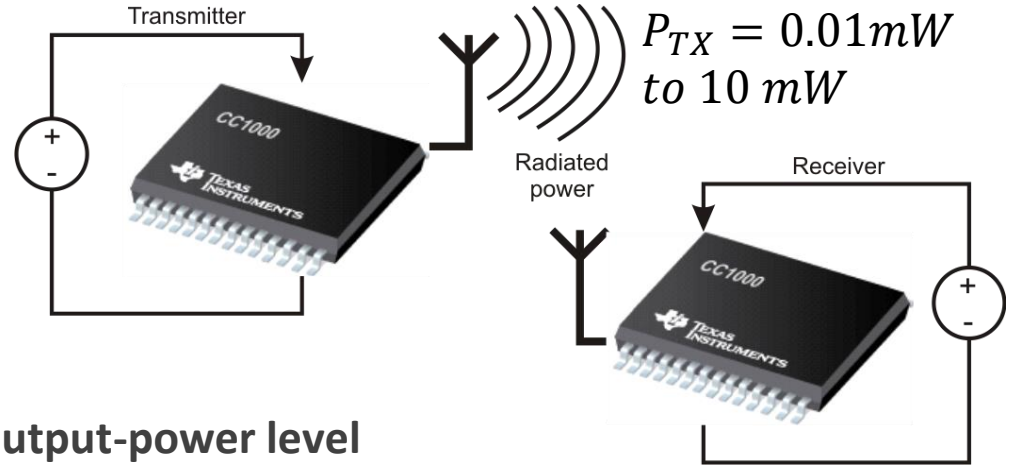


Example: TI-CC1000

7 Radio Interface: Power Consumption

Parameter	Typ.	Max.	Unit
Current Consumption, receive mode 433/868 MHz	7.4/9.6		mA
P=0.01mW (-20 dBm)	5.3/8.6		mA
P=0.3 mW (-5 dBm)	8.9/13.8		mA
P=1 mW (0 dBm)	10.4/16.5		mA
P=3 mW (5 dBm)	14.8/25.4		mA
P=10 mW (10 dBm)	26.7/NA		mA
Current Consumption, crystal osc.	30 80 105		µA µA µA
Current Consumption, crystal osc. And bias	860		µA
Current Consumption, crystal osc., bias and synthesiser, RX/TX	4/5 5/6		mA mA

- Transmitter W_{TX} :
  - 433 MHz: 16 – 80.1 mW
  - 868 MHz: 25.8 – 76.2 mW
- Receiver W_{RX} :
  - 433 MHz: 22.2 mW
  - 868 MHz: 28.6 mW



Transmitter

- Power consumption depends on the output-power level
- Radio output-power level P_{TX} is far below transmitter power consumption W_{TX}

Receiver

- Power consumption is comparable to the transmitter (at low power levels)

7 Radio Interface: Energy Efficiency

- For sensor node applications, power consumption is not the most relevant metric
  - Energy efficiency relates battery lifetime to the amount of data that can be transmitted
  - Energy efficiency is measured in Joules per (information) bit [J/bit] and depends on the active power consumption and the data rate

$$E\left[\frac{nJ}{bit}\right] = W_{Tx/Rx} \cdot T_{bit} = \frac{W_{Tx/Rx}}{f}$$

- Increasing the transmission rate *f*
  - improves energy efficiency, but also reduces radiated energy per information bit
  - Receiver: sensitivity drops for higher rates

toward higher data rates ↓

Data rate [kBaud]	Separation [kHz]	433 MHz			868 MHz		
		NRZ mode	Manchester mode	UART mode	NRZ mode	Manchester mode	UART mode
0.6	64	-113	-114	-113	-110	-111	-110
1.2	64	-111	-112	-111	-108	-109	-108
2.4	64	-109	-110	-109	-106	-107	-106
4.8	64	-107	-108	-107	-104	-105	-104
9.6	64	-105	-106	-105	-102	-103	-102
19.2	64	-103	-104	-103	-100	-101	-100
38.4	64	-102	-103	-102	-98	-99	-98
76.8	64	-100	-101	-100	-97	-98	-97
Average current consumption		9.3 mA			11.8 mA		

↑ toward better sensitivity

7 Radio Interface: Energy Efficiency

- **Maintain a given signal to noise ratio as data rate f increases**
  - **Need to increase transmit power to maintain constant received energy per bit**
  - **From an information theoretic perspective we would expect**

$$\frac{P_{RX}}{f} \equiv const. \Rightarrow P_{TX} \sim f \qquad W_{TX} \sim P_{TX} \sim f$$

- **"Luckily" reality is different:** $P_{TX} < O(f)$ **and** $W_{TX} < O(P_{TX}) < O(f)$

Rate *f*	Duty cycle	Tx power	Sensitivity	Margin	Tx	Rx
4.8 kbit/s	1	-5 dBm	-107 dBm	102 dB	26.7 mW	22.2 mW
38.4 kbit/s	1/8	0 dBm	-102 dBm	102 dB	31.2 mW	22.2 mW

3x ↓ (Tx power)

1.2x ↓ (Tx)

Overall improvement in energy efficiency with higher data rates

Rate *f* [kbps]	Tx energy efficiency	Rx energy efficiency
4.8 kbit/s	5.6 uJ/bit	4.6 uJ/bit
38.4 kbit/s	0.8 uJ/bit	0.6 uJ/bit

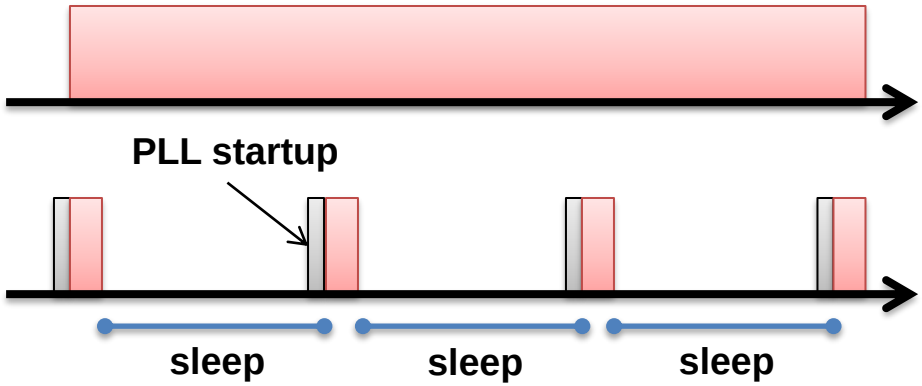
7 Radio Interface: Energy Efficiency

- **Constant throughput requirement**
  - **Higher rate allows to duty-cycle both transmitter and receiver**

Duty cycle	Rate *f*	Tx	Rx
1	4.8 kbit/s	26.7 mW	22.2 mW
1/8	38.4 kbit/s	3.9 mW	2.8 mW

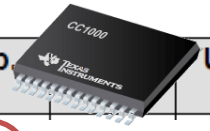
  - **Duty cycling is limited by the startup/ locking time of the frequency synthesizer (typ: 0.1-1 ms)**

$$L_{Burst}[bits] \gg T_{Startup} \cdot f$$



- **Example: TI-CC1000**
  - $T_{Startup}$=250 us
  - f=38.4 kbit/s

$$L_{Burst} \gg 10\ bit$$

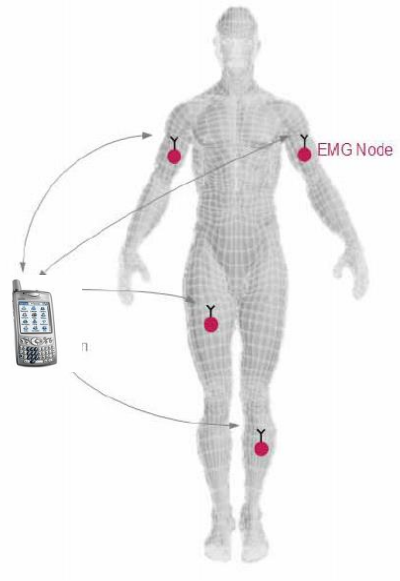


Parameter	Min.	Typ.		Unit	Condition / Note
PLL turn-on time, crystal oscillator on in power down mode		250		µs	Crystal oscillator running
Current Consumption, crystal osc. And bias		860		µA	
Current Consumption, crystal osc., bias and synthesiser, RX/TX		4/5 5/6		mA mA	< 500 MHz > 500 MHz

7 Processing Energy vs. Transmit Energy

Sensor data is usually highly redundant (i.e., compressible)

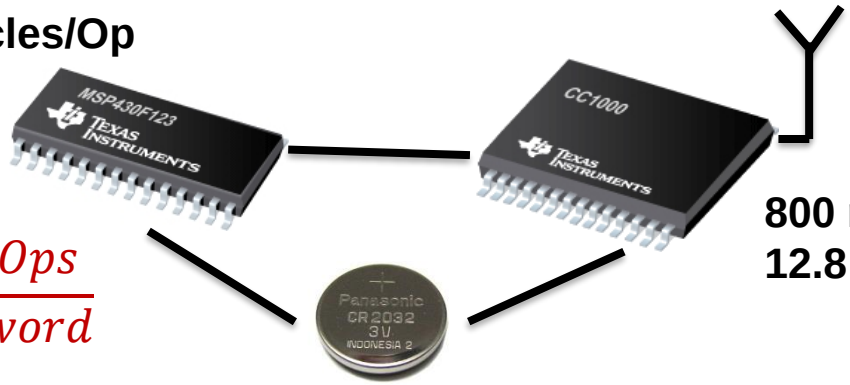
- **Data reduction through analysis and/or compression**
- **Requires computations (i.e., consumes power)**
- **Computations can often be done either after transmission on the central node *or* before transmission on the sensor node**



What is the cost of computation vs. the cost of transmission??

Example: MSP430 16-bit low-power uController with TI-CC1000 transceiver

600 uW/MHz, 1-3 cycles/Op
200 pJ/Operation



800 nJ/bit or
12.8 uJ/word (16-bit)

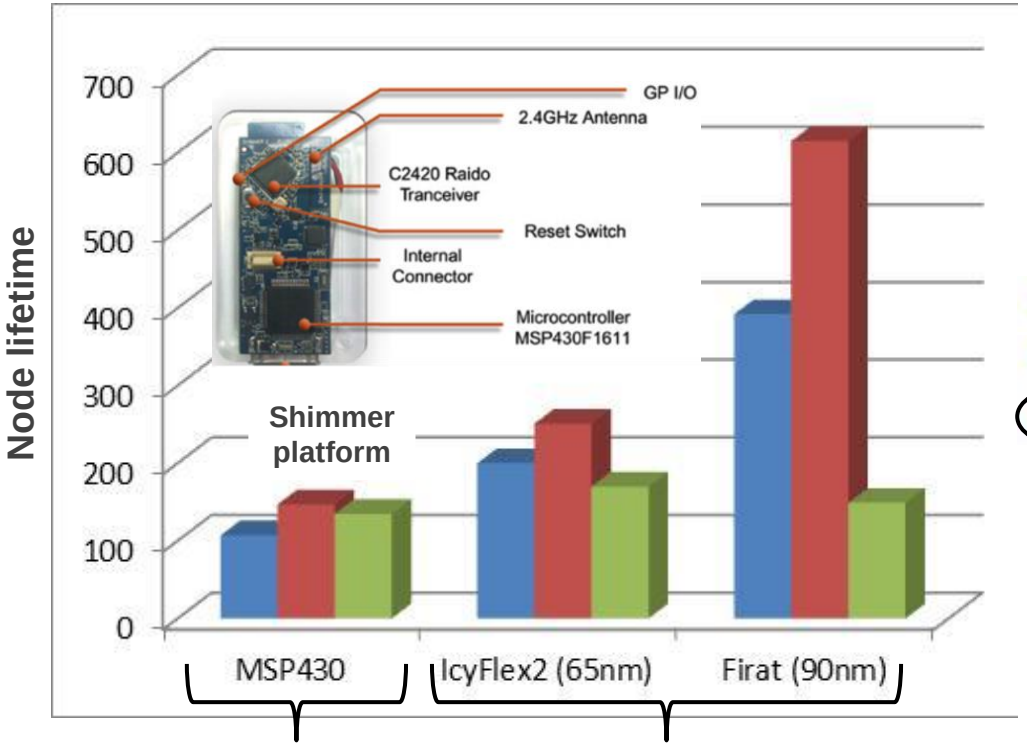
$$\frac{12.8\ \frac{uJ}{word}}{200\ \frac{pJ}{Op}} > 64'000\ \frac{Ops}{word}$$

7 Lifetime extension from compression

Compression requires processing resources and consumes power

- **Energy savings from reduced radio activity can get eaten up by**
  - **Poor energy efficiency of the embedded processor**
  - **Overly complex compression/analysis algorithms**

- **Burden shifts to more energy-efficient data processing**



No comp.
DWT
CS

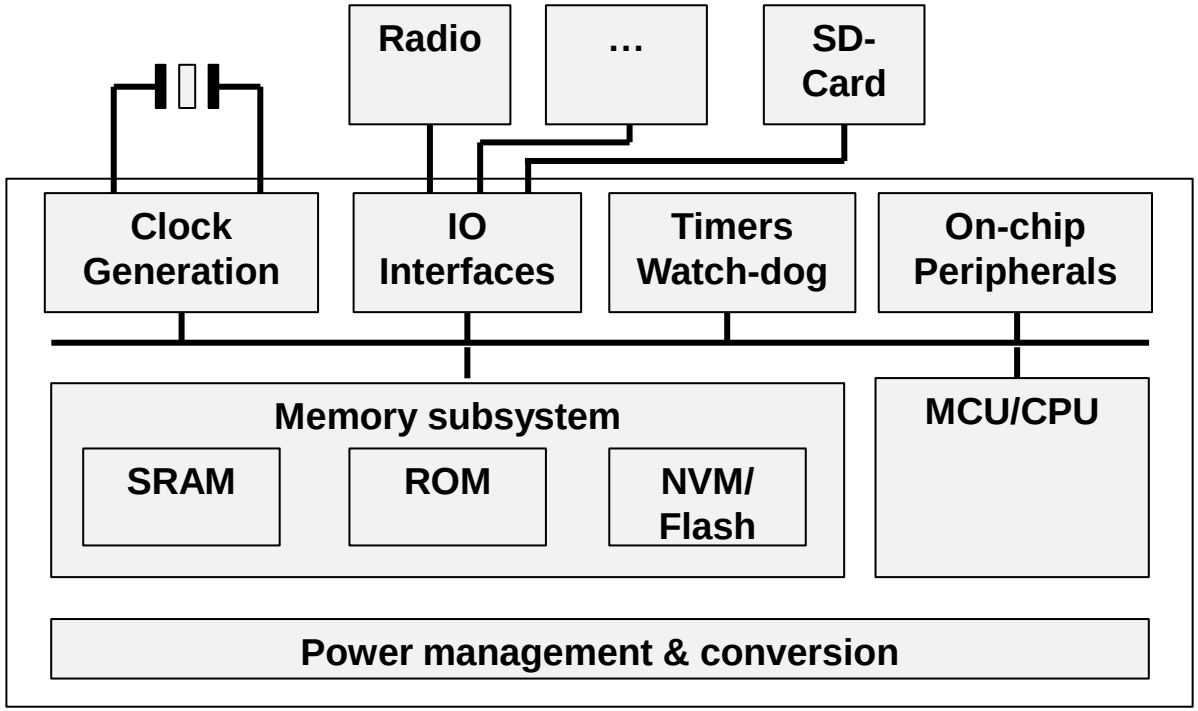
Low complexity compression algorithm

[Rincon et al., ITAB, 2011]

Embedded Low-Power Microcontrollers:
Overview and Low-Power Modes

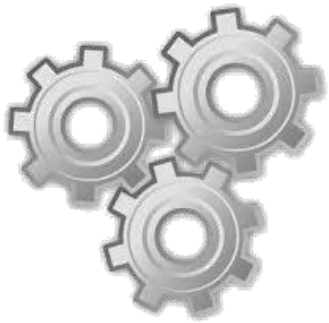
Low Power Sensor Nodes: System Components

- Microcontroller
- Memory subsystem
  - SRAM
  - ROM
  - FLASH
- IO interfaces
- Clock generation and clock distribution
- Timers
- Interface peripherals
- Power management



Wireless Sensor Nodes: Operation

- **Typical activities in a wireless sensor node**
  - **Sampling/sensing**
  - **Processing data on the node**
  - **Storing data**
  - **Transmitting data**
  - **Waiting for an event or a fixed period**



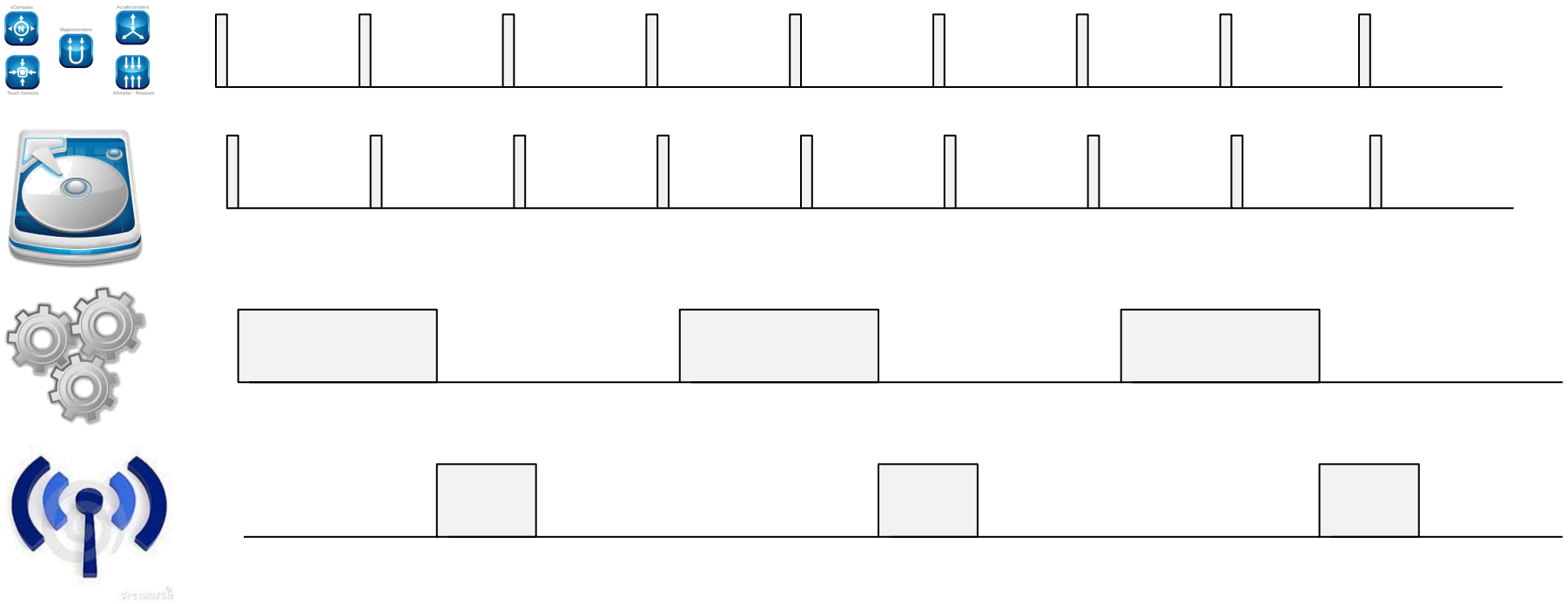
Handled by various system components in a more or less independent fashion from the processor core

Processing schedules

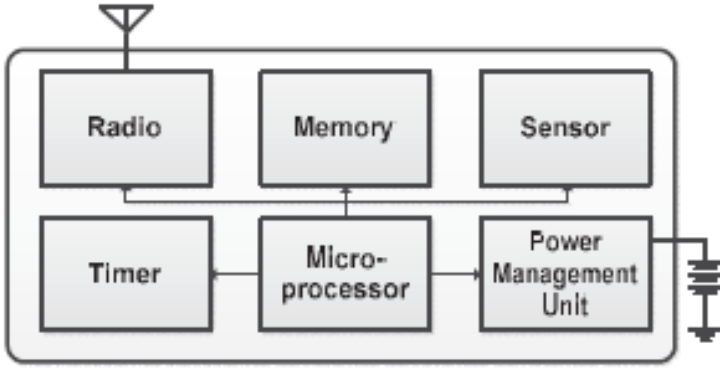
- A typical application schedule
  - Data is sampled in regular intervals (typically relatively slowly)
    - Data acquisition handled by the core itself
    - Smart data converter with direct memory access (DMA)
  - Data can be processed each time a new sample arrives OR in larger junks (typically more efficient)
  - Processing speed is governed by deadlines
    - Processing speed (clock frequency) adjusted precisely to meet the deadline
    - Processing in short bursts with high clock frequency while sleeping in between

Wireless Sensor Nodes: Operation Scheduling

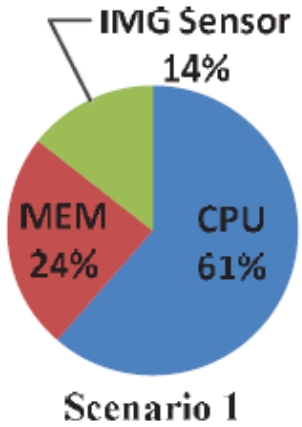
- Multiple parallel processes (partially overlapping) with different characteristics
  - Sampling and storage of data: on small chunks of data with hard real-time constraints
  - Processing and transmission: more energy efficient on larger data blocks
- Idle periods almost unavoidable
- Operation is driven by interrupts



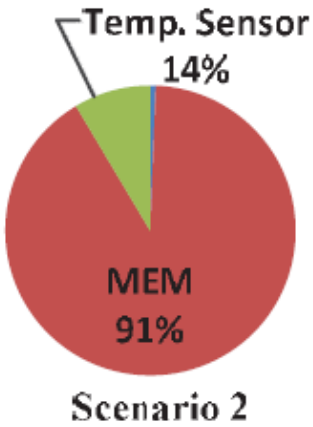
Power Consumption Depends on Usage Scenario



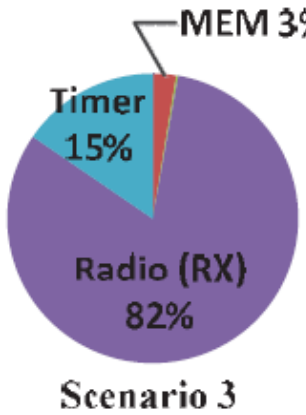
Unit	Scenario 1 (Surveillance)	Scenario 2 (Temp. monitor)	Scenario 3 (Temp. monitor /w symmetric communication)
(wake up period)	5 sec	10 min	10 min
Microprocessor	100k inst. / wake up	2k inst. / wake up	2k inst. / wake up
Memory	16kB, 229nW	1kB, 14.3nW	1kB, 14.3 nW
Timer	Not used	Not used	8.6 nW
Radio – TX	1 kb / wake up	100 b / 1 hr	100 b / 1 hr
Radio – RX	Not used	Not used	1.64mW for mismatch
Image Sensor	1 frame / wake up	Not used	Not used
Temperature Sensor	Not used	1 meas. / wake up	1 meas. / wake up



Scenario 1



Scenario 2



Scenario 3

Optimization and Design Strategy for Low Power

- Start from a “reasonable” system configuration
- Decide on a processing schedule that maximizes the exploitation of low power modes available from the system
- Identify the distribution of power consumption among the system components and processing phases
- Address the part or phase with the largest contribution first by

Reminder: Sources of Power Consumption

- Dynamic power consumption
  - Active power of functional blocks
    - Power is consumed while circuit/component is active (operating)
      - Without DVFS: Power depends (linearly) on the operating frequency
      - With DVFS: Power depends super-linearly on frequency/voltage (often not supported in ULP embedded systems due to complexity of generating a variable voltage in an efficient way)
  - Power for clock generation and clock distribution

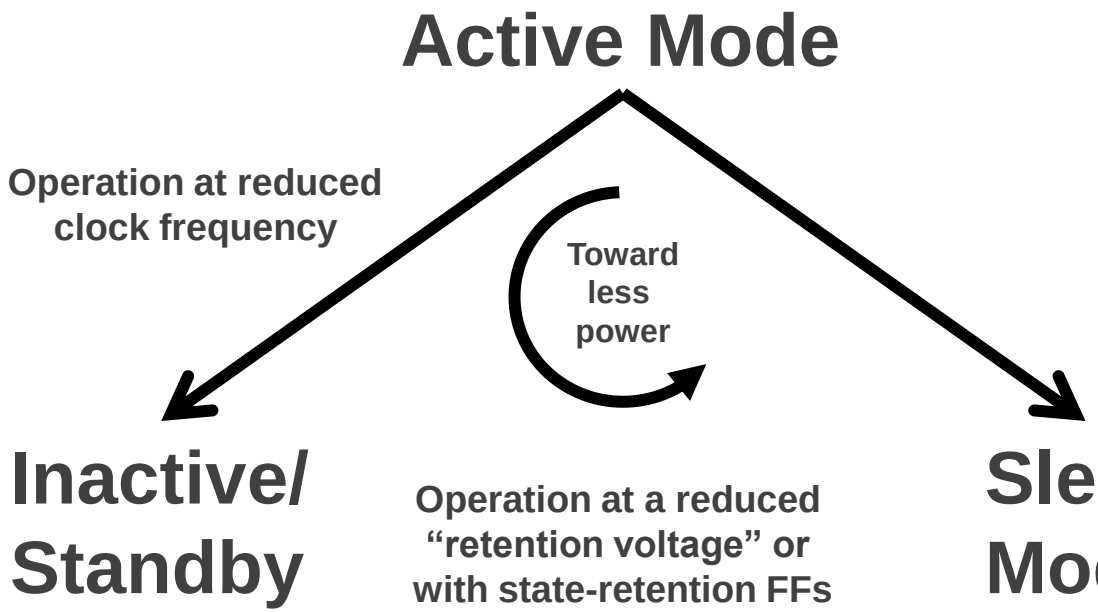
- Static power consumption
  - Static power of digital functional blocks
    - Power is consumed during both active and inactive periods
      - Often negligible while circuit is active (except at low frequencies)
      - Without power-gating: dominant (only) power during inactive periods
      - With power gating: can be very low
  - Static power consumed in analog circuitry (e.g., amplifiers)

Low-Power and Sleep Modes

Low power modes typically differ in

- Energy consumption
- CPU activity
- Reaction time
- Wake-up triggers
- Active peripherals
- Available clock sources

Low-Power and Sleep Modes: Two main tuning knobs

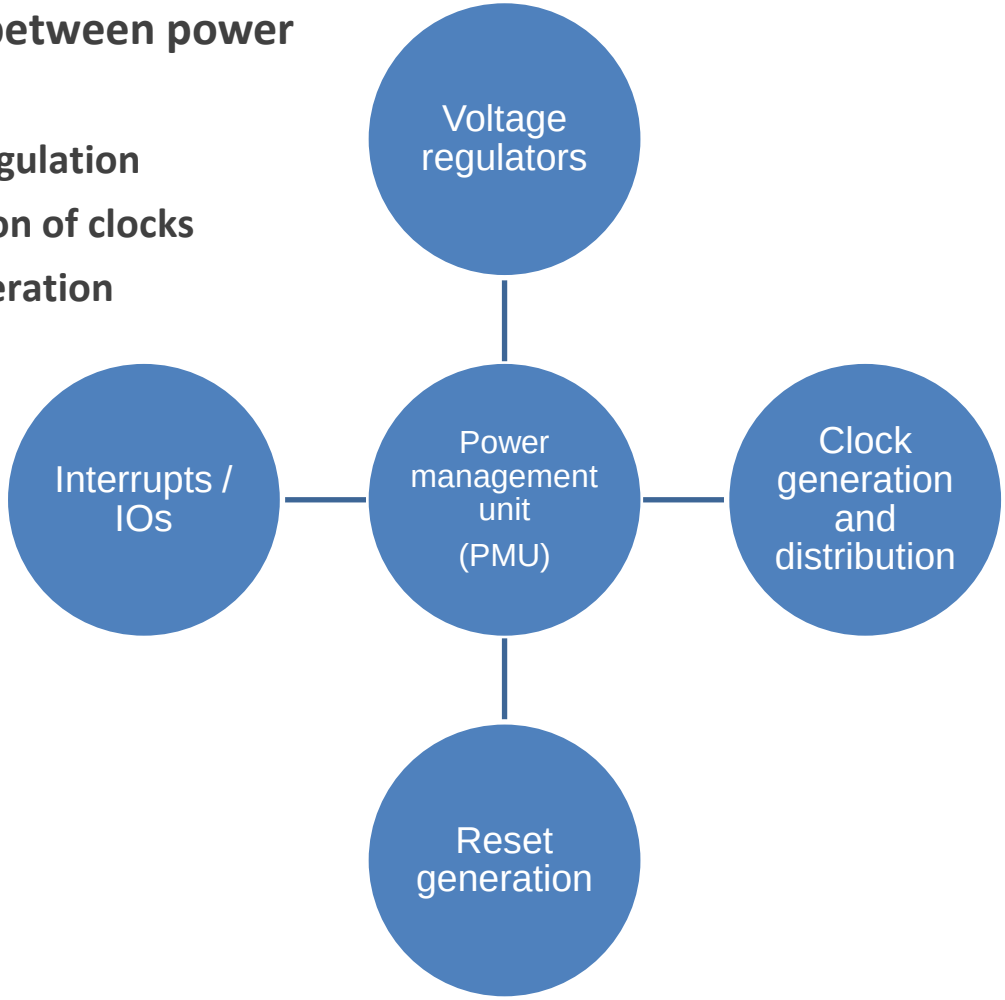


- Clock is deactivated, but supply is still on
  - No activity
  - State is retained
  - Rapid wakeup

- Clock is deactivated and supply is off (power-gating)
  - No activity
  - No state retention
  - Slow wakeup
    - Power-up delay
    - Need to restore state

Components involved in energy management

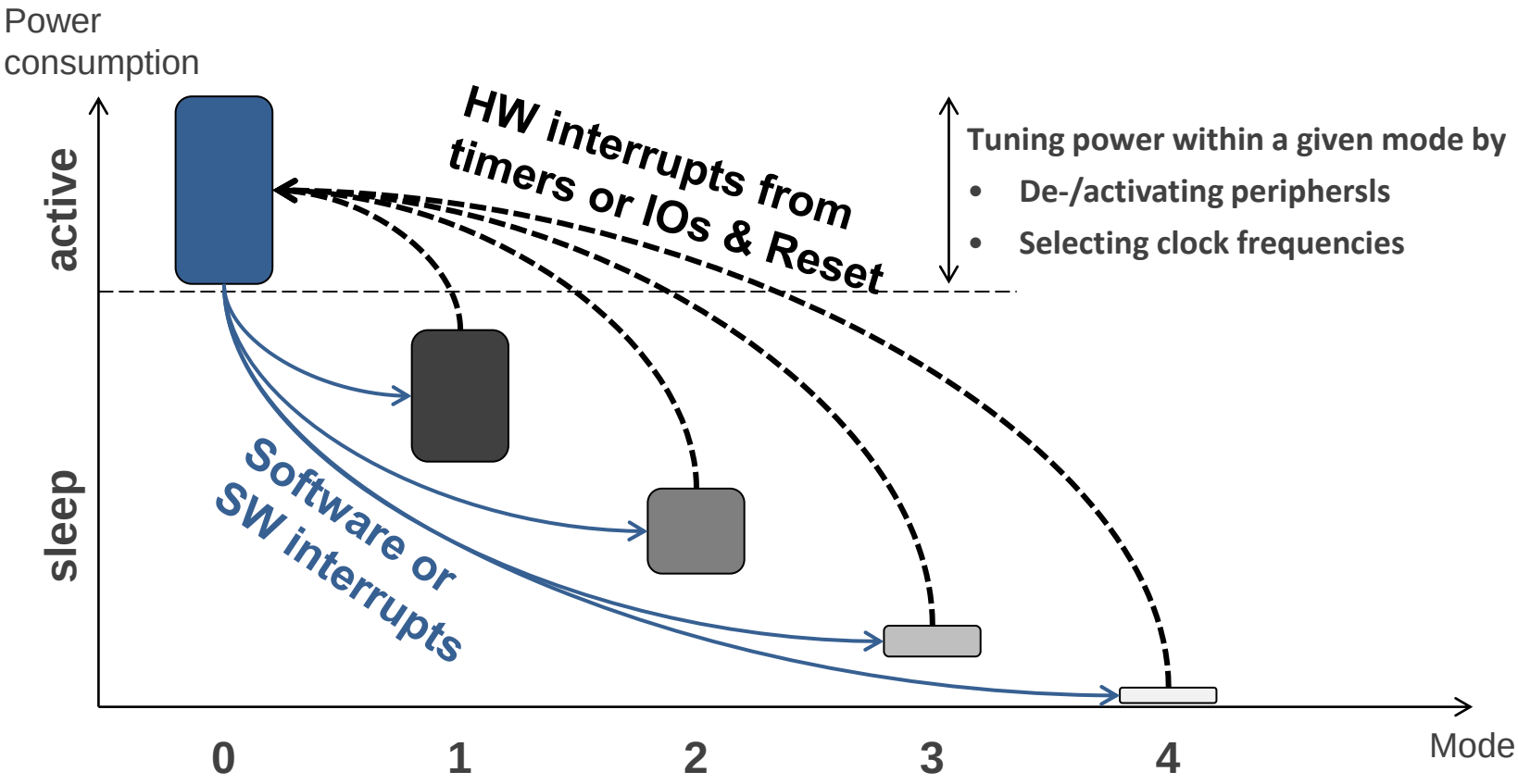
- **PMU: controls and coordinates the transition between power modes**
  - **Voltage regulation**
  - **Stabilization of clocks**
  - **Reset generation**



Transition between power modes

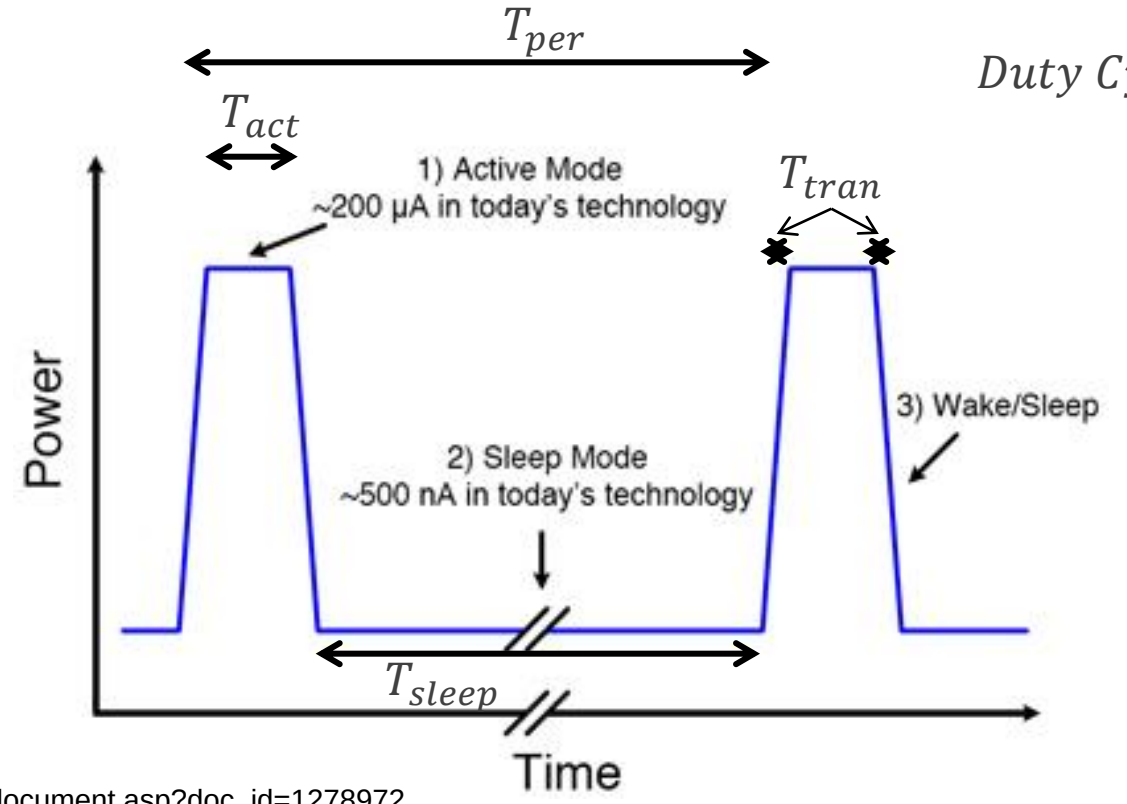
Different sleep modes and one configurable “active” mode:

- **Transition from “active” into one of the “sleep” modes: triggered by software**
- **Waking up “sleep” to “active”: triggered by hardware**



Transition between power modes

- **Break energy consumption into three parts**
  - **1) active energy**
  - **2) sleep energy**
  - **3) wake/sleep transition energy**
    (energy for moving between sleep and active modes)

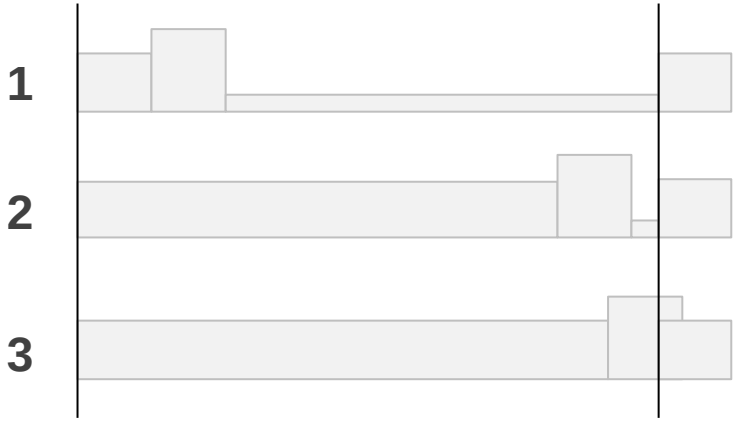


$$Duty\ Cycle = \frac{T_{act}}{T_{per}}$$

http://www.eetimes.com/document.asp?doc_id=1278972

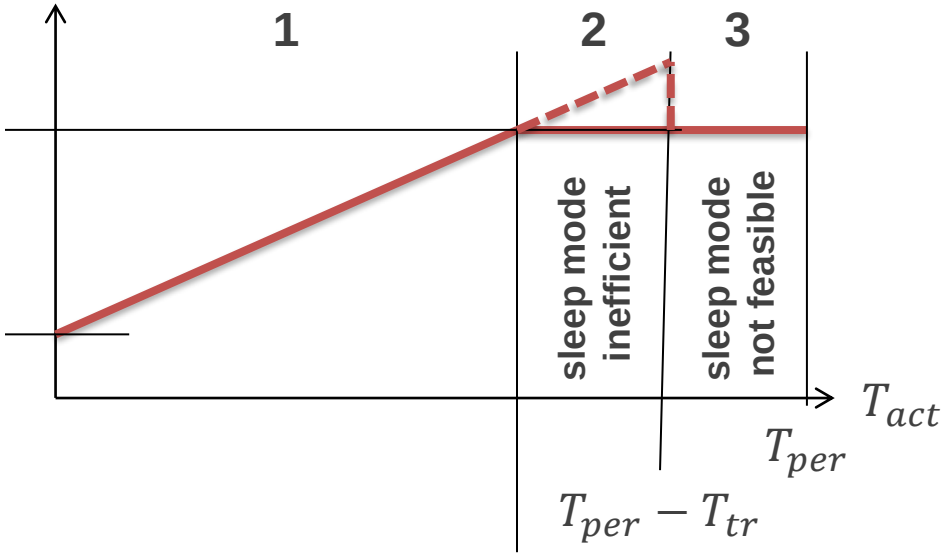
Power budget

- **Power figures for active/transition/sleep:** $P_{act}, P_{tr}, P_{sleep}$
  - **Assume:** $P_{tran} = P_{act}$
- **Time for period/active/transition:** $T_{act}, T_{tr}, T_{per} = T_{act} + T_{sleep}$



$$P = \frac{T_{act}P_{act} + T_{tr}\left(P_{tr} - P_{sleep}\right) + T_{sleep}P_{sleep}}{T_{per}}$$

$$P = \frac{T_{tr}\left(P_{tr} - P_{sleep}\right) + T_{sleep}P_{sleep}}{T_{per}}$$



$$T_{per} - \frac{T_{tr}P_{tr} + P_{sleep}\left(T_{sleep} - T_{tr}\right)}{P_{act}}$$

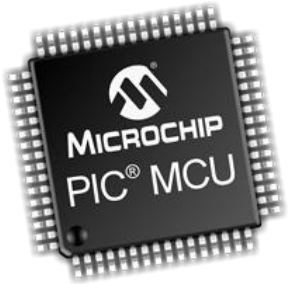
Embedded Low-Power Microcontrollers:

Key Components

Active Power Consumption (COTS products)

- **Prominent examples:**

More and more vendors offer ARM core based solutions



PIC Microcontroller
- 8 bit
- 16 bit
- 32 bit

MSP 430 Microcontroller
- 16 bit

ARM Microcontroller Series
- ARM Cortex M0(+)
- ARM Cortex M3

Active power consumption
(CPU/Clock Generation/Regulators)
ARM based: 80-250 uA/MHz @ 1.8 – 3.3V
Older models: 250-500 uA/MHz @ 1.8 – 3.3V

Comparison of COTS MCUs to State-of-the-Art Processor Cores

State-of-the-art cores generally outperform available microcontrollers

- **COTS microcontrollers often use mature technology nodes (>130nm)**
- **Core-only specifications and research cores often exclude peripherals/SRAM/clocking**
- **Commercial MCUs typically do not yet exploit voltage/frequency scaling**

research

	Energy per Ops @ 1.0 V	Technology
TamaRISC	12.1 pJ	90 nm
16-bit [Kwong,2011]	> 47 pJ	130 nm
32-bit [Ickes,2011]	19.7 pJ-27 pJ	65 nm

commercial

ARM Cortex-M0 Implementation Data***			
	180ULL 7-track, typical 1.8v, 25C)	90LP 7-track, typical 1.2v, 25C)	40LP 9-track, typical 1.1v, 25C)
Dynamic Power	66µW/MHz	12.5µW/MHz	5.3µW/MHz
Floorplanned Area	0.11 mm^2	0.03 mm^2	0.008 mm^2

Best ARM M0+ based COTS MCU from ST Micro with approximately 85 uW/MHz including memory & clocking

Clock Generation and Clock Distribution

- **Motivation:**
  - **Clock frequency determines power of active components**
  - **Clock generation and distribution can be a significant part of the overall power consumption (e.g., when most of the chip is inactive)**

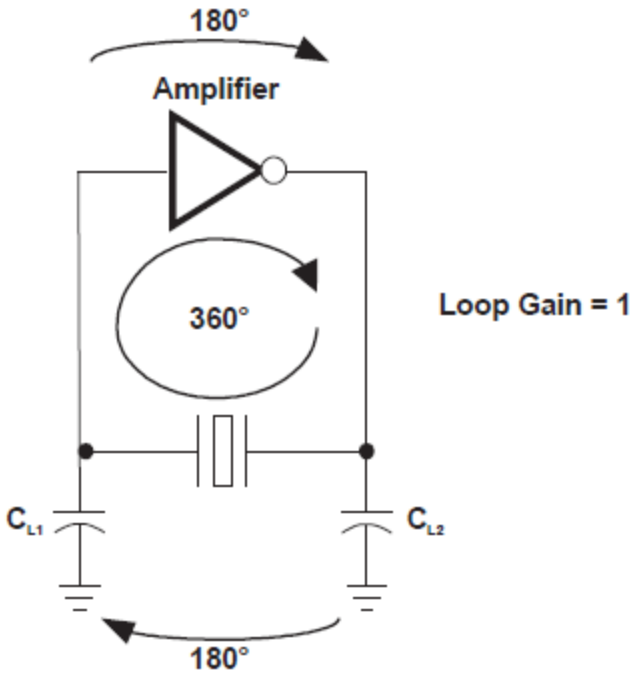
- **Performance criteria of clocking subsystem**
  - **Power consumption**
  - **Flexibility to generate adjust frequency to the needs**
  - **Agility to change frequency during operation**
  - **Accuracy compared to a golden reference:** measured in parts-per-million (PPM)
  - **Startup time:** time to restore a stable clock after shutdown

- **Tradeoffs associated with clock generation**
  - **Fast clock vs. slow clock: jitter, frequency, basis for derived clocks**
  - **On-chip clock generation vs. off-chip components: cost, space on PCB, power consumption of external components, accuracy**
  - **Fixed vs. programmable: power consumption, area, analog components, ability to precisely adjust to the needs**

Clock Generation and Clock Distribution

- **Quartz based frequency reference**
  - Available in some standard frequencies from kHz to >100 MHz
  - Accuracy 5-20 ppm
  - High cost and power consumption

- **External quartz & capacitors with on-chip oscillator**
  - On-chip feedback amplifier (often two for high and low target frequencies)
  - Accuracy 1-20 ppm (low cost)
  - Startup time in the milliseconds (ms) range and increases for lower frequencies
  - Power consumption
    - Linear in frequency
    - Typical values in the ~50uW / MHz range
      - STM32L15xx6: 1.1uA@3V for 32 KHz



Example: MSP430 oscillator startup times

OSCILLATOR FREQUENCY	OSCILLATOR TYPE	V_{CC} (V)	START-UP TIME (ms)
32 kHz C = none	Quartz crystal	3	281
		5	96.4
455 kHz C = 82 pF	Ceramic resonator	3	1.77
		5	—
1 MHz C = 47 pF	Quartz crystal	3	5.82
		5	3.81
4 MHz C = 56 pF	Quartz crystal	3	4.54
		5	3.07
8 MHz C = 39 pF	Quartz crystal	3	1.97
		5	1.27

http://www.st.com/web/en/resource/technical/document/datasheet/CD00277537.pdf

http://www.gaw.ru/pdf/TI/app/msp430/slaa081.pdf

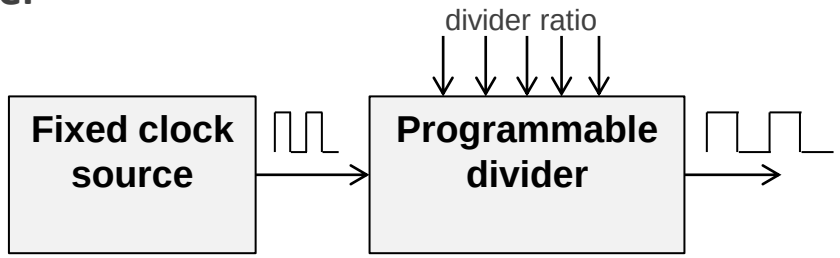
Clock Generation and Clock Distribution

- **On-chip digitally controlled oscillator (DCO)**
  - Programmable ring oscillator
    - Based on RC delays (low-power)
    - Few preselected frequency settings
    - DC reference controlled by an external or internal reference resistor (for calibration)
  - Mostly used for low power consumption at very low frequencies (e.g., during sleep periods as clock source for wakeup timers)

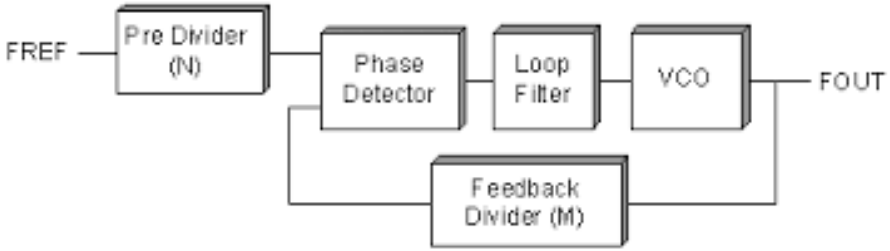
  - Startup times similar to clock period (1MHz: 1us, 100kHz: 10us)
  - Power consumption: ~50uW / MHz range
  - Accuracy is very poor: few % (@1MHz: 10’000 ppm)

Clock Generation and Clock Distribution

- **Reference clock with synchronous clock divider**
  - Can only reduce clock frequency compared to reference input
  - Only division by integers (coarse grained) for clocks not too far below the reference.

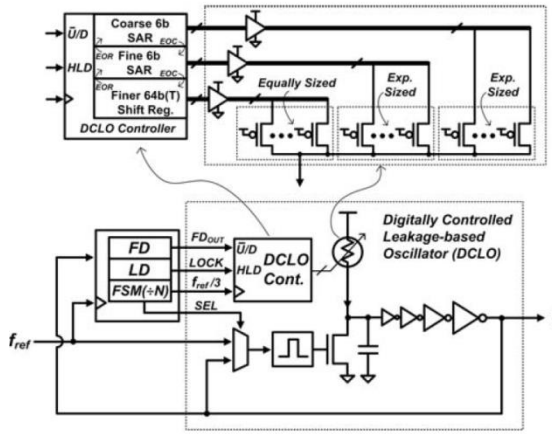


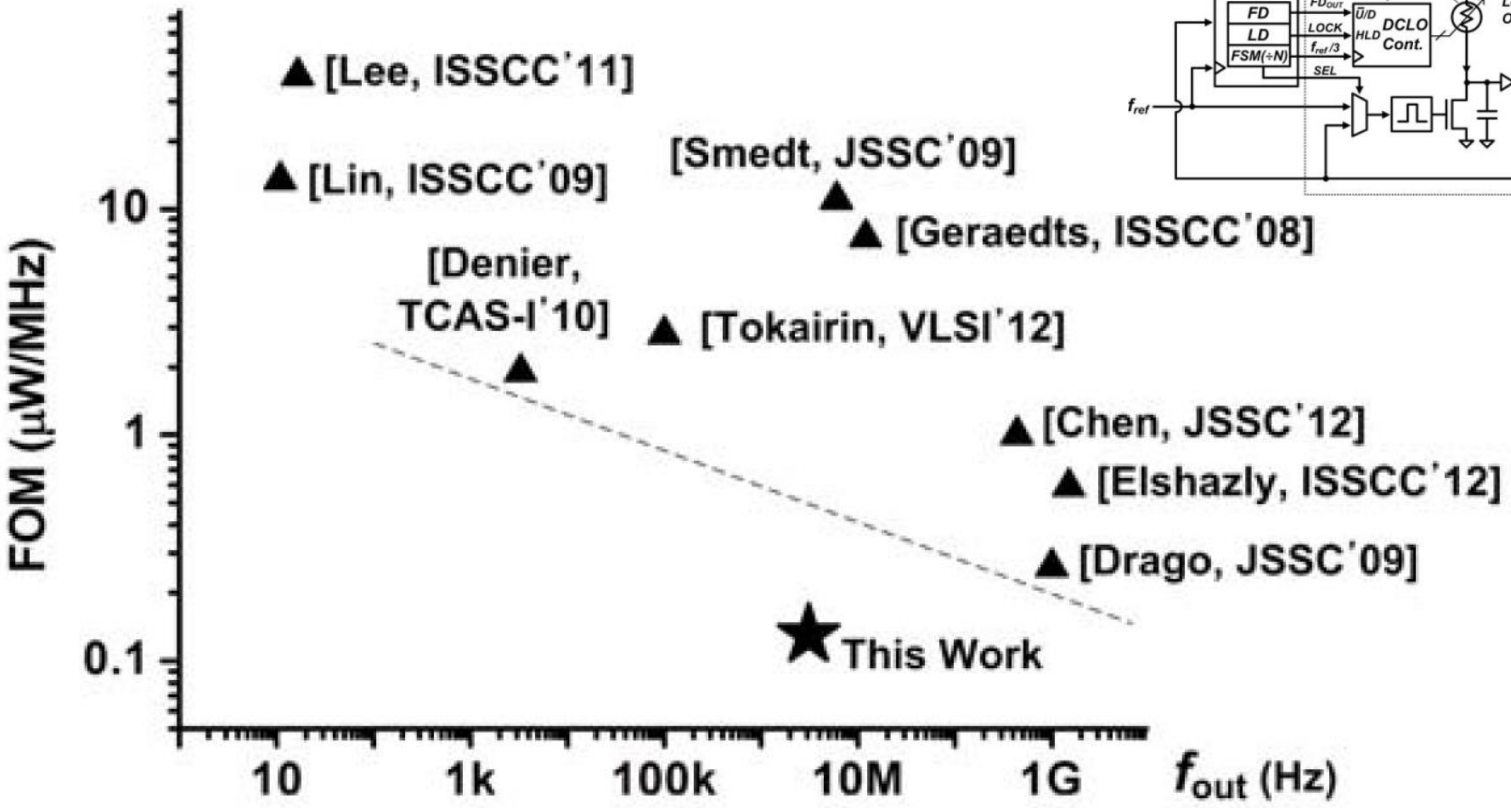
- **Reference clock with Phase Locked Loop (PLL)**
  - PLL is comprised of
    - Voltage Controlled Oscillator
    - Phase/frequency detector
    - Loop filter
  - Alternatives:
    - AD-PLL: all digital implementation
    - Frequency locked loop (FLL): phase not locked to the reference oscillator



Circuits for Clock Generation (Research State-of-the-Art)

- **Figure of merit (FOM): uW/MHz**
- **Main contributor to power consumption is the DCO**
  - **Implementations based on leakage currents**

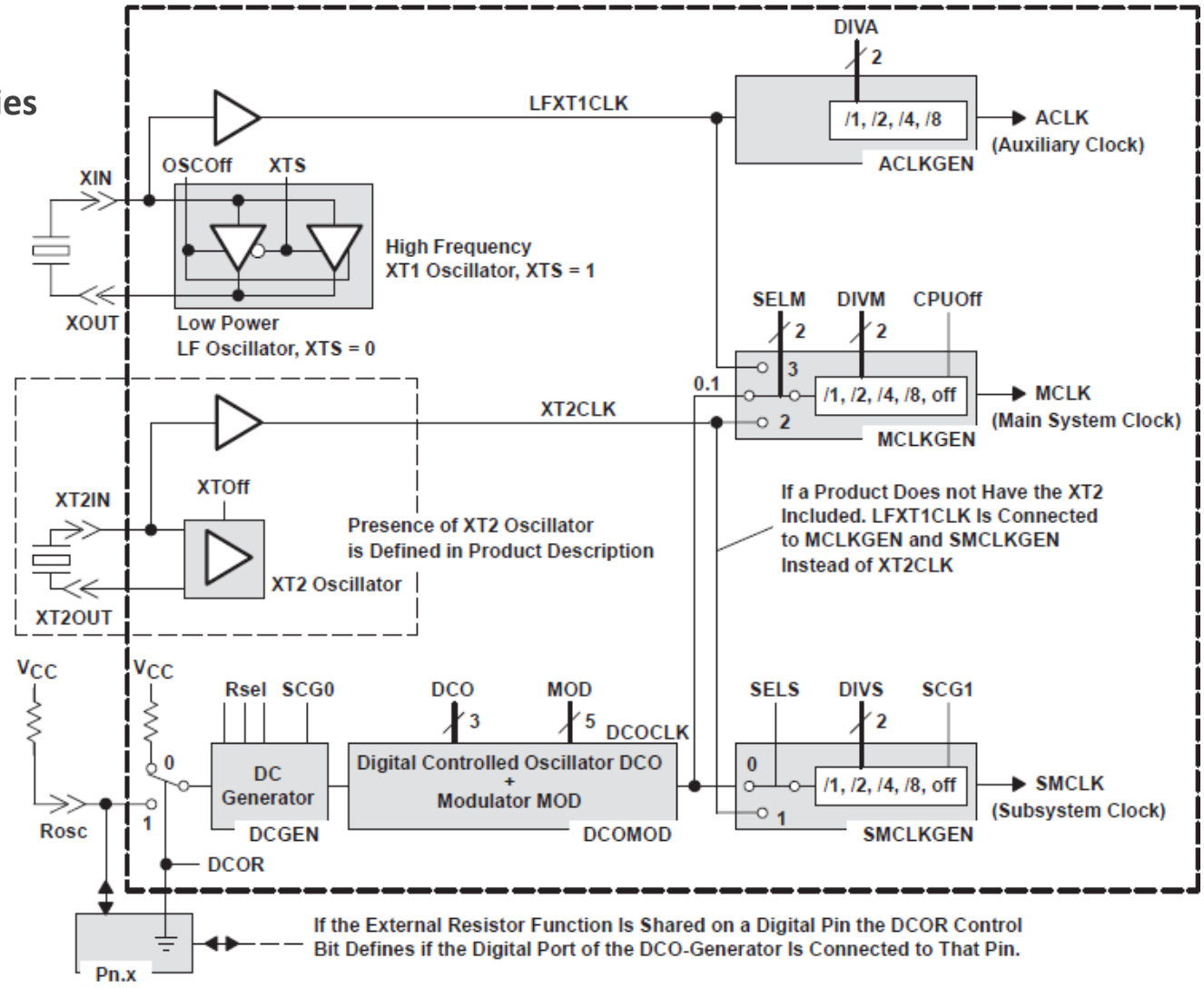




Dong-Woo Jee; Sylvester, D.; Blaauw, D.; Jae-Yoon Sim, "A 0.45V 423nW 3.2MHz multiplying DLL with leakage-based oscillator for ultra-low-power sensor platforms," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*

Clock Generation and Clock Distribution

- **Example: TI-MSP430 clock generation and distribution subsystem**
  - **Different clock sources generate different frequencies**
  - **Rapid selection of the best clock for different system components**



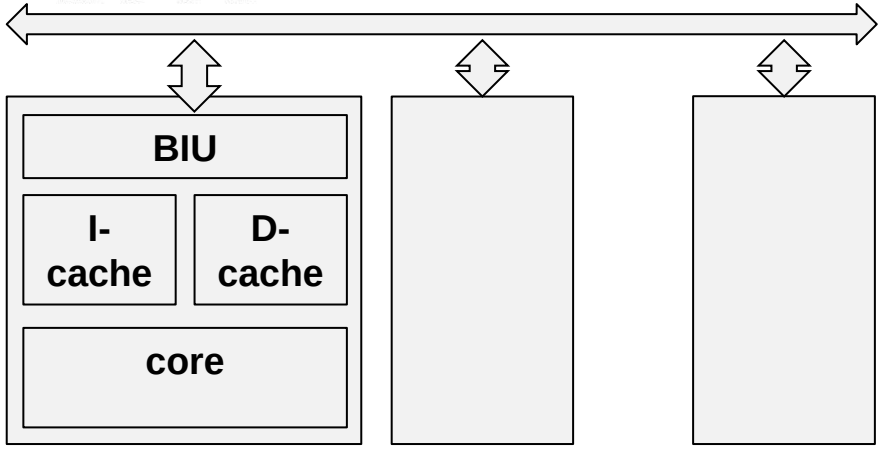
Memory/Storage Subsystem and Components

Architectures of the storage subsystem of two types of embedded systems

- **Program and data: shared address space (von Neumann - like)**

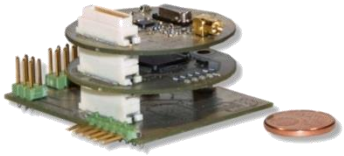
High performance

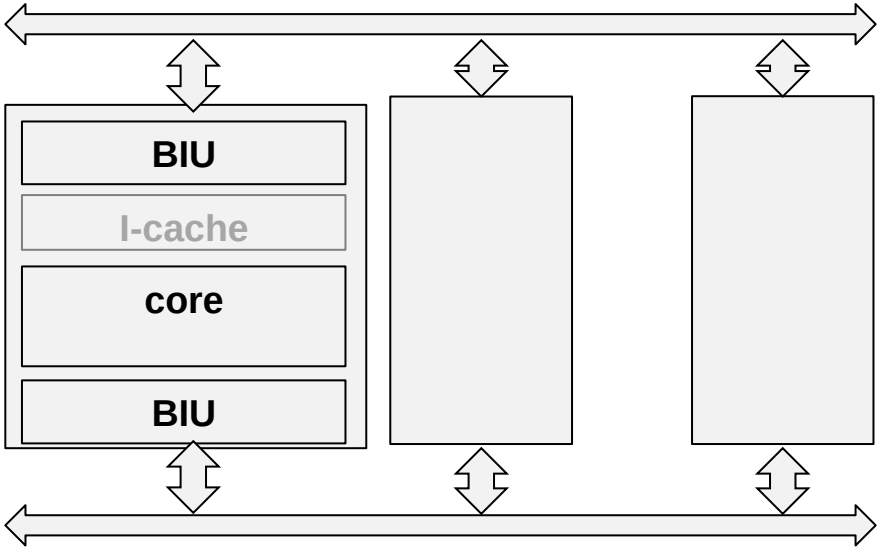
- **Often single bus interface, but large caches for data and instruction**



Ultra low power

- **Harvard architecture core with separate bus interfaces for data and instruction**
- **Multiple or multi-layer busses**
- **No or only small cache, often only for instructions**

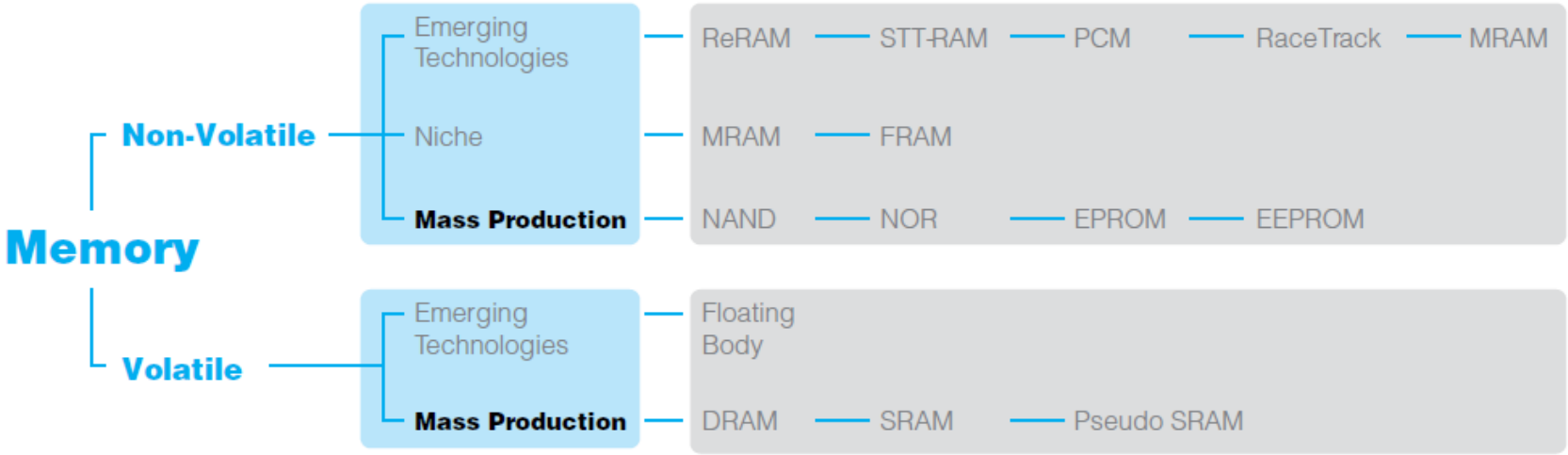




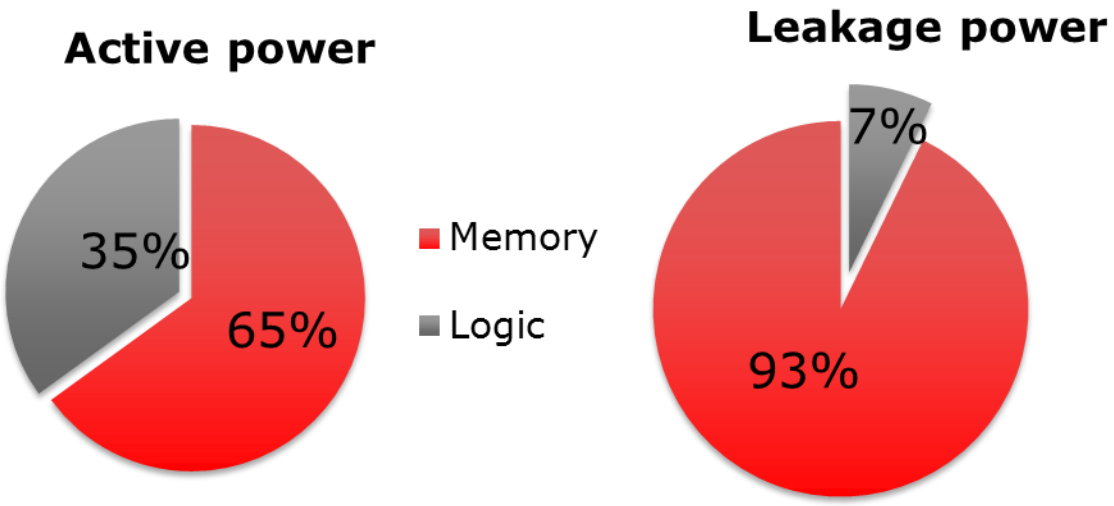
Memory/Storage Subsystem and Components

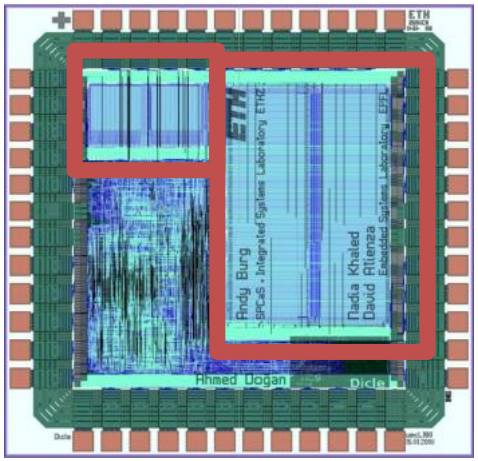
- Type of information to be stored
  - Program code (instructions)
  - Runtime intermediate variables (data)
  - Acquired data
  - System settings and state (registers in peripherals)

- Different types of storage
  - (Registers)
  - Volatile memory: retains its content only with power on, typically SRAM
  - Non-volatile memory: retains content after power-off



SRAMs Consume a Considerable Amount of Power





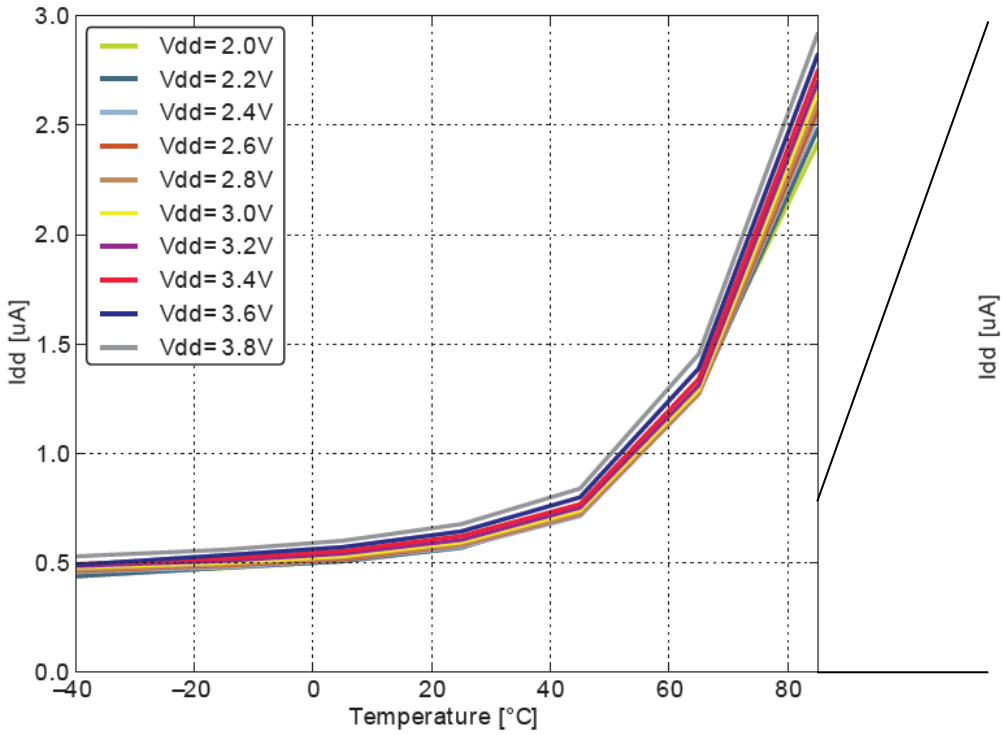
typical embedded processor

- For embedded processors, memories occupy a large percentage of the silicon area
- Active mode:
  - Data and program memory can consume 2/3 of total power
  - Low-frequency: SRAM leakage becomes visible
- Sleep modes:
  - Generally, no power gating to retain SRAM content
  - SRAM leakage becomes dominates system power consumption (3-4 pJ/bit in 180nm): 32kByte -> 400nW @ 1.8V

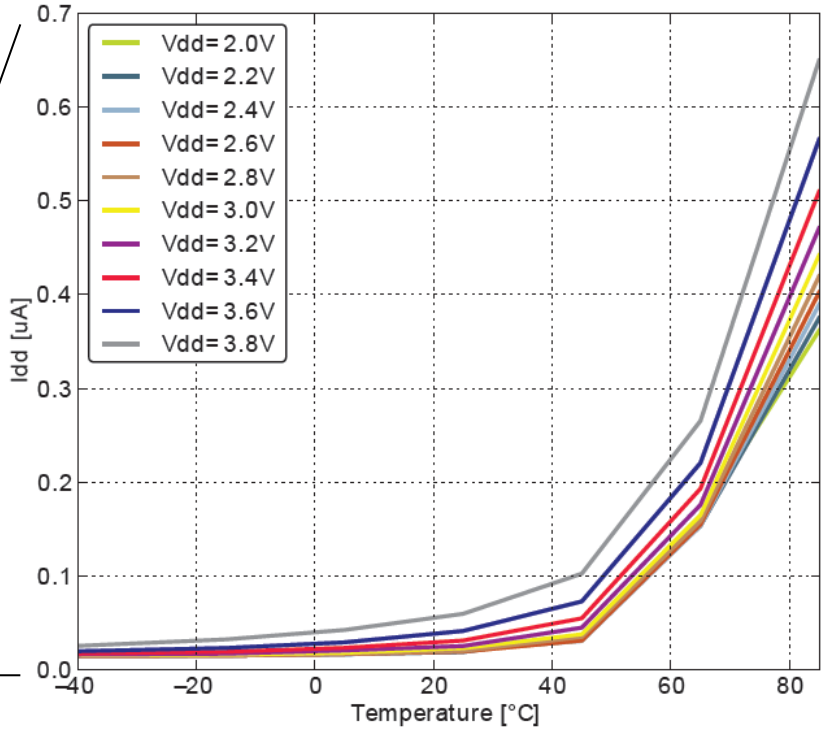
SRAM Leakage over Temperature

Example: EFM32 MCU (ARM-M3) with 32kByte SRAM

- **Significant difference leakage reduction when SRAM is disabled**
- **Leakage increases exponentially with temperature**



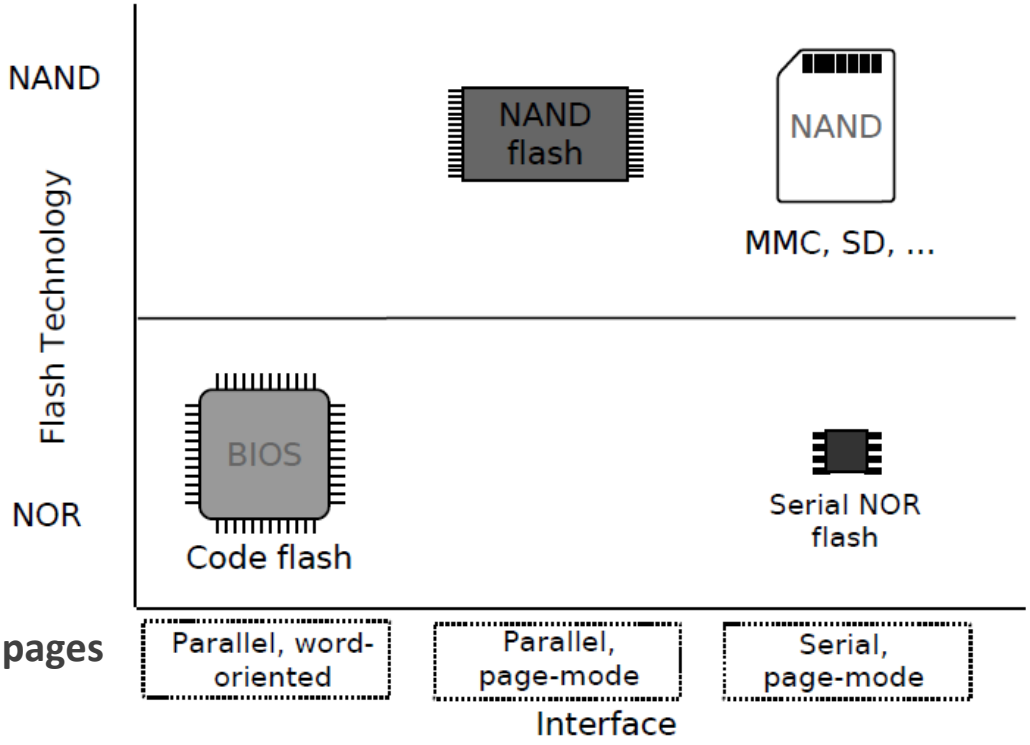
SRAM on



SRAM off

Non Volatile Memory

- Typically implemented as Flash
  - NOR or NAND flash
- Flash vs. SRAM: Flash has
  - Slower read access
  - Much slower write access
  - Much higher write power
- NAND Flash
  - High density, organized in large pages
  - Used for storing large amounts of data
  - Usually off chip
- NOR Flash
  - Medium density with word access possible
  - Can store program code
  - Program code can be executed directly from Flash

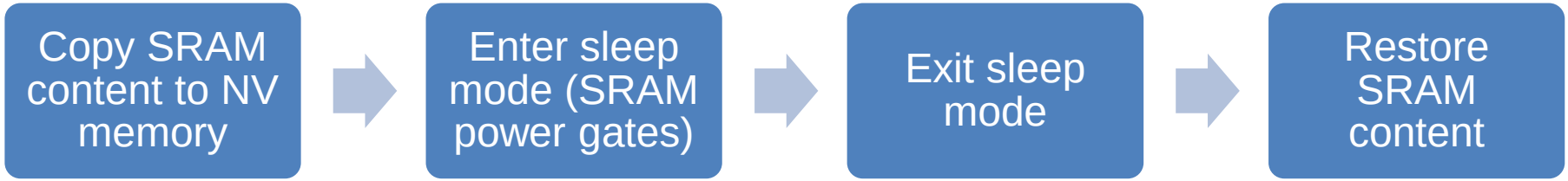


	NOR	NAND
Memory size	<= 512 Mbit	1–8 Gbit
Sector size	~1 Mbit	~1 Mbit
Output parallelism	Byte/Word/Dword	Byte/Word
Read parallelism	8–16 Word	2 Kbyte
Write parallelism	8–16 Word	2 Kbyte
Read access time	<80 ns	20 µs
Program Time	9 µs/Word	400 µs/page
Erase time	1 s/sector	1 ms/sector

Non Volatile Memory as Deep Sleep Backup

- Almost all components are OFF
- Leakage becomes THE dominant factor in power consumption
- Memory (SRAM) is often the largest contributor to leakage:
  - E.g., 3.5 pW/bit with 32kByte => 900 uW
- Memory power gating leads to a loss of data

Flash SRAM based backup:

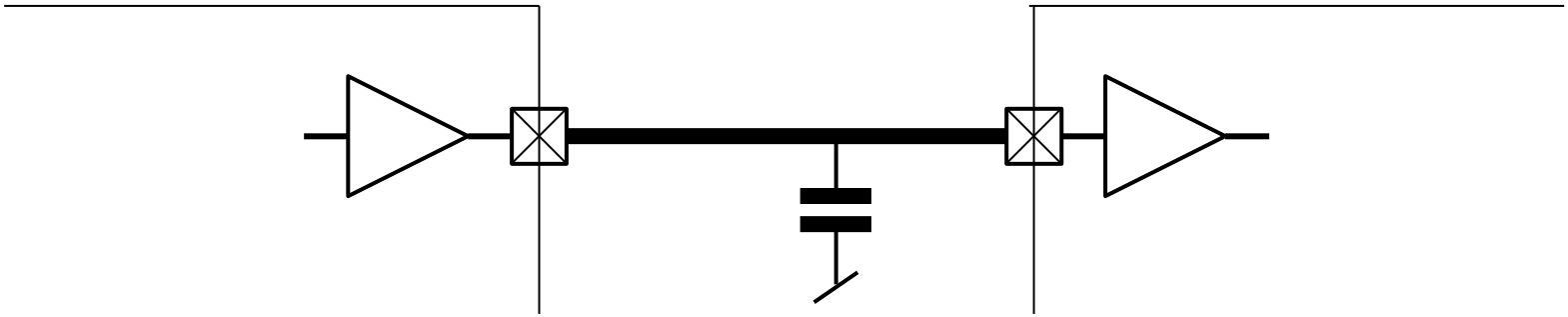


- Caveat: writing to NV memory requires a lot of energy
  - Transition to deep-sleep is expensive
  - Backup only the required data
- Deep-sleep with SRAM backup to NV memory only efficient for long sleep periods
  - Alternative: very small backup SRAM with state retention for essential data (few Bytes)

The cost of off-chip communication

Off-chip communication consumes dynamic power

- **IO-pad capacitance is around 2.5-10pF**
- **Typical voltages for IO-standards: 1.8V, 2.5V, 3.3V**



- **Energy per transition and pin:**

$$E = \frac{1}{2} V_{IO}^2 C$$

- **Power consumed by 8 IO pins α =50% activity @ f =1 MHz ($C = 5pJ, V_{IO} = 1.8V$)**

$$P_{IO} = \alpha f \frac{1}{2} V_{IO}^2 C = 32uW$$

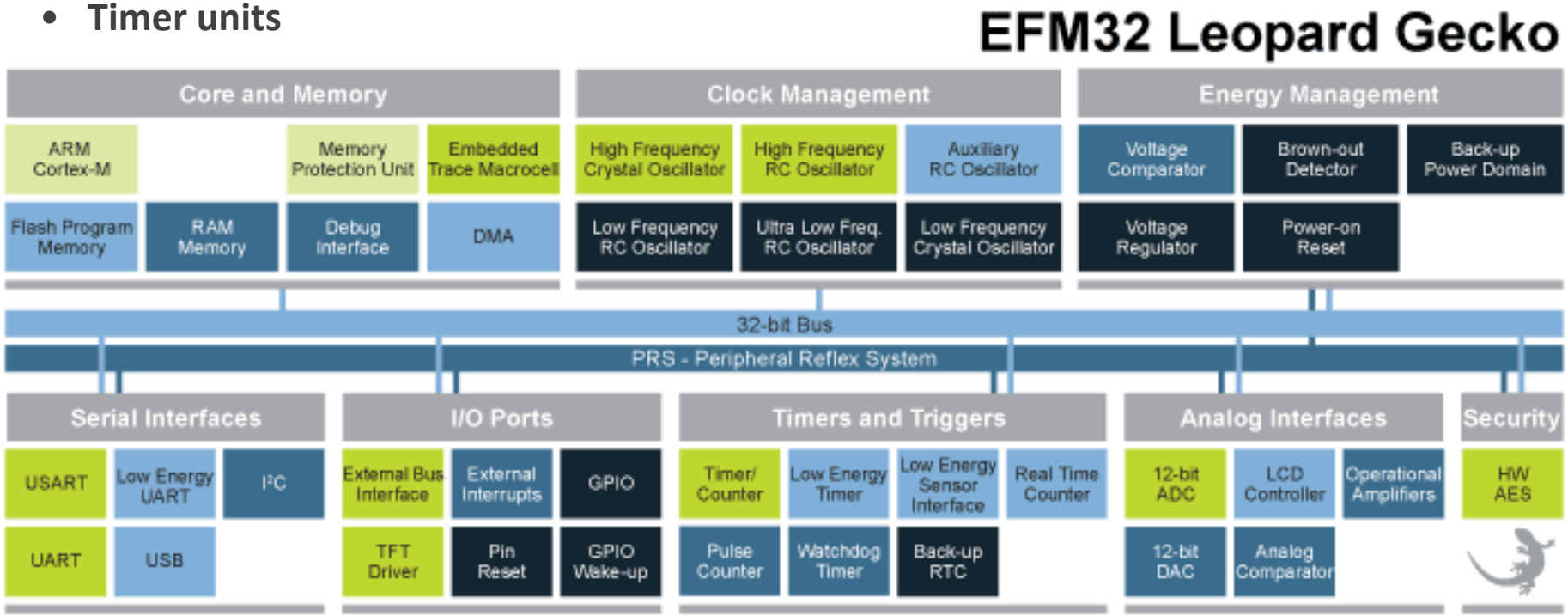
Hands-on Example: EFM32 ULP MCU:
Power Consumption and Low-Power Modes

Energy Micro EFM32LG (Leopard Gecko) MCU Overview

EFM32 is a series of microcontrollers with different capabilities and peripherals

EFM32LG: medium range ULP device

- **ARM-M3 low-power 32 bit CPU up to 48MHz**
- **SRAM and Flash memory (read and write access)**
- **Clock generation and energy management unit**
- **Support for many peripherals**
- **Timer units**



5 basic energy modes with different default settings for peripherals and clocks

EM0 – Energy Mode 0

(Run mode)

In EM0, the CPU is running and consuming as little as 211 µA/MHz, when running code from flash. All peripherals can be active.

EM1 – Energy Mode 1 (Sleep Mode)

In EM1, the CPU is sleeping and the power consumption is only 63 µA/MHz. All peripherals, including DMA, PRS and memory system, are still available.

EM2 – Energy Mode 2 (Deep Sleep Mode)

In EM2 the high frequency oscillator is turned off, but with the 32.768 kHz oscillator running, selected low energy peripherals (LCD, RTC, LETIMER, PCNT, LEUART, I^2C, LESENSE, OPAMP, USB, WDOG and ACMP) are still available. This gives a high degree of autonomous operation with a current consumption as low as 0.95 µA with RTC enabled. Power-on Reset, Brown-out Detection and full RAM and CPU retention is also included.

EM3 - Energy Mode 3 (Stop Mode)

In EM3, the low-frequency oscillator is disabled, but there is still full CPU and RAM retention, as well as Power-on Reset, Pin reset, EM4 wake-up and Brown-out Detection, with a consumption of only 0.65 µA. The low-power ACMP, asynchronous external interrupt, PCNT, and I^2C can wake-up the device. Even in this mode, the wake-up time is a few microseconds.

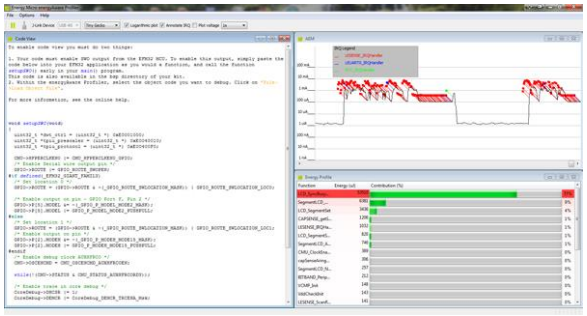
EM4 – Energy Mode 4

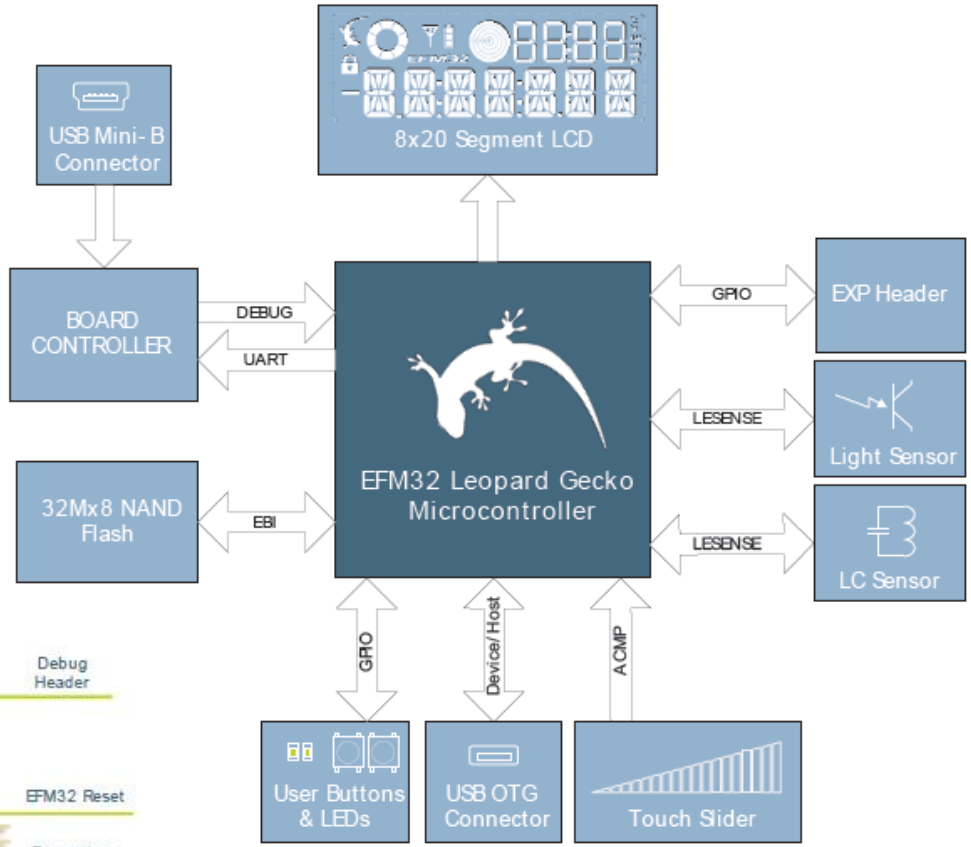
(Shutoff Mode)

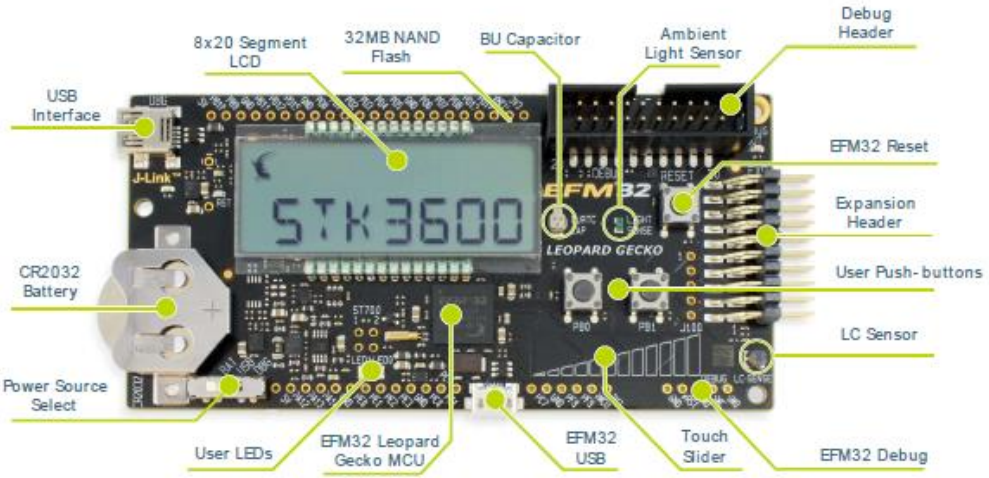
In EM4, the current is down to 20 nA and all chip functionality is turned off except the pin reset, GPIO pin wake-up, GPIO pin retention, Backup RTC (including retention RAM) and the Power-On Reset. All pins are put into their reset state.

EFM32LG – STK 3600 Evaluation Board & Power Monitor

- EFM32LG990F256 MCU
- **Energy Monitoring system for current tracking**







- Various peripherals and IOs
- Power sources:
  - Battery
  - USB
  - 0.03 F supercap for backup