Energy Efficient Computer Design Guest Lecture

Marius Grannæs

Energy Micro

Who are we?

- Founded in 2007
- Worldwide Offices
  - HQ Oslo (Norway), Krakow, Chicago, Raleigh, Munich, Paris, Hong Kong, Kuala Lumpur, Seoul
- Team with extensive experience
  - Chipcon, Texas Instruments, Atmel, Nordic Semiconductor, Freescale, Infineon and Silicon Laboratories
- Experts in ultra low power mixed-signal technologies

Minimum 4x battery lifetime…

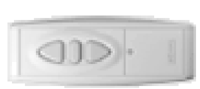
… versus any other 8, 16 or 32-bit microcontroller

5-Feb-13

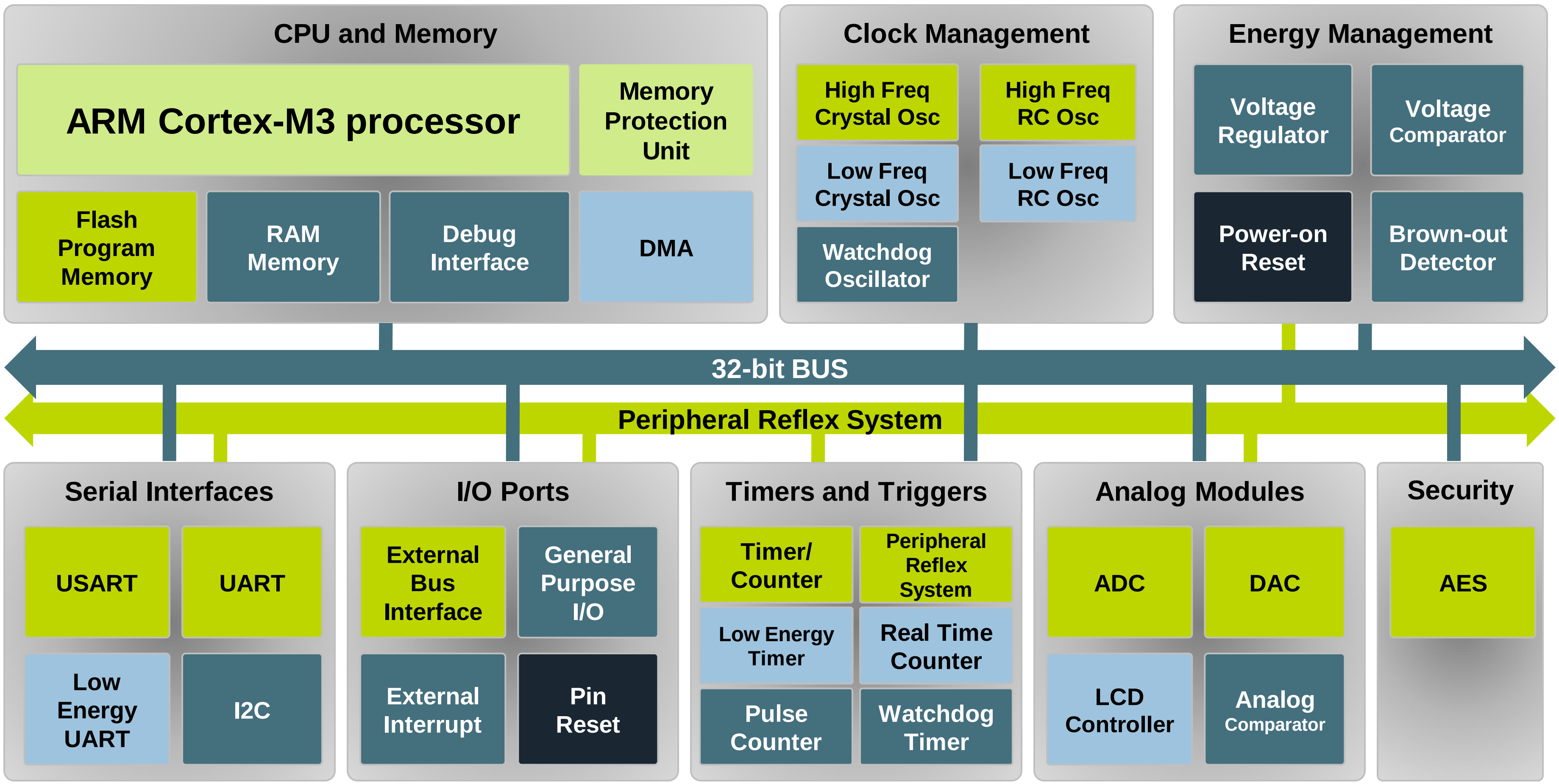
Why does battery life matter?

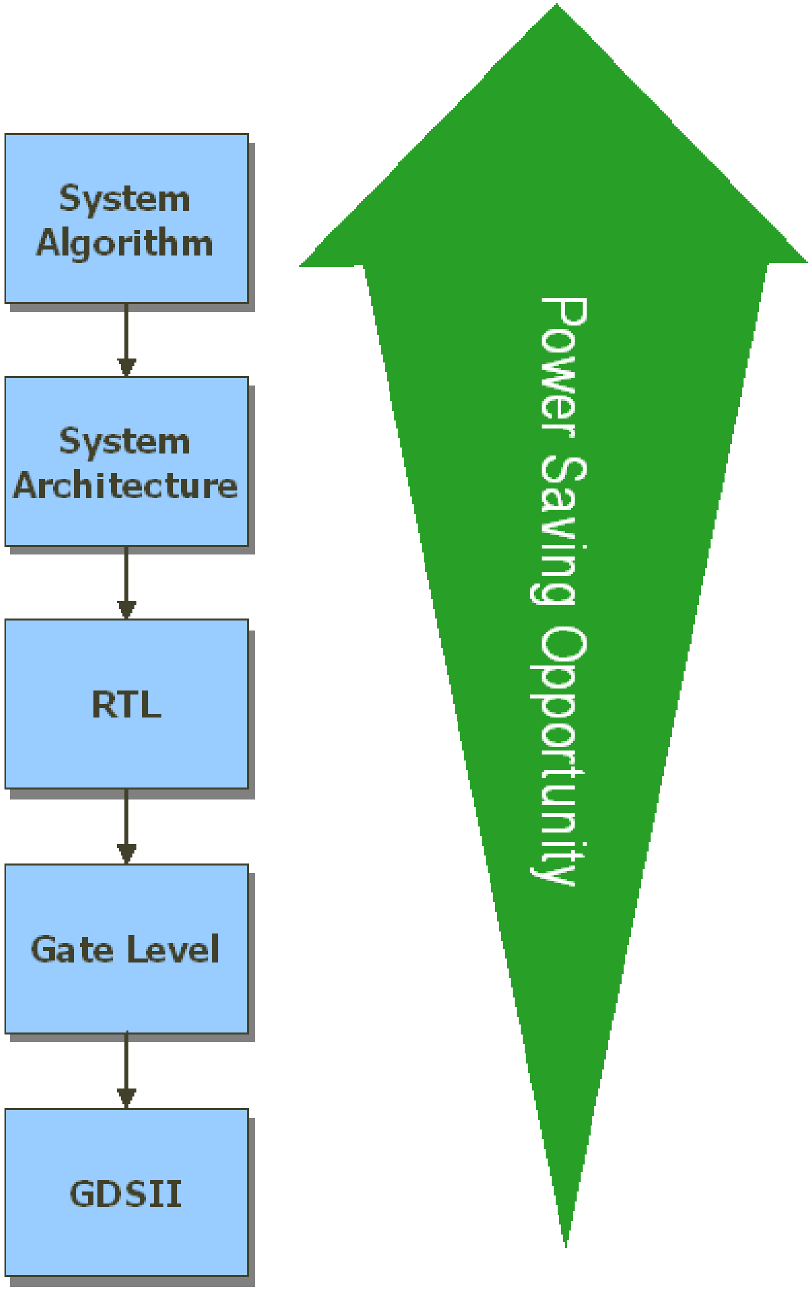
1. The Environment
2. Long lasting gadgets
3. Cost of battery / Replacing battery
4. Impossible to replace battery

Energy Sensitive Applications

Energy, Gas, Water and Smart Meters	Home and Building Automation	Alarm and Security Systems	Health and Fitness	Smart Accessories
 Power Meters	 Access Control	 Fire and Safety	 Portable Medical	
 Water Meters	 Light Control	 Burglar Alarms	 Remote Diagnostics	 Ultra low power add-ons and dongles
 Gas Meters	 Climate Control	 Motion Sensors		
 Smart Meters	 Remote Control			

EFM32 Architecture





EE Times Design

5 factors that made it possible to make the world's most energy friendly microcontroller

How to achieve low energy consumption?

1. Power efficient MCU core (ARM Cortex-M3)
2. Low leakage process technology (TSMC)
3. The ”right” IP
   - Std. Cell library
   - Memory (FLASH and RAM)

How to achieve low energy consumption?

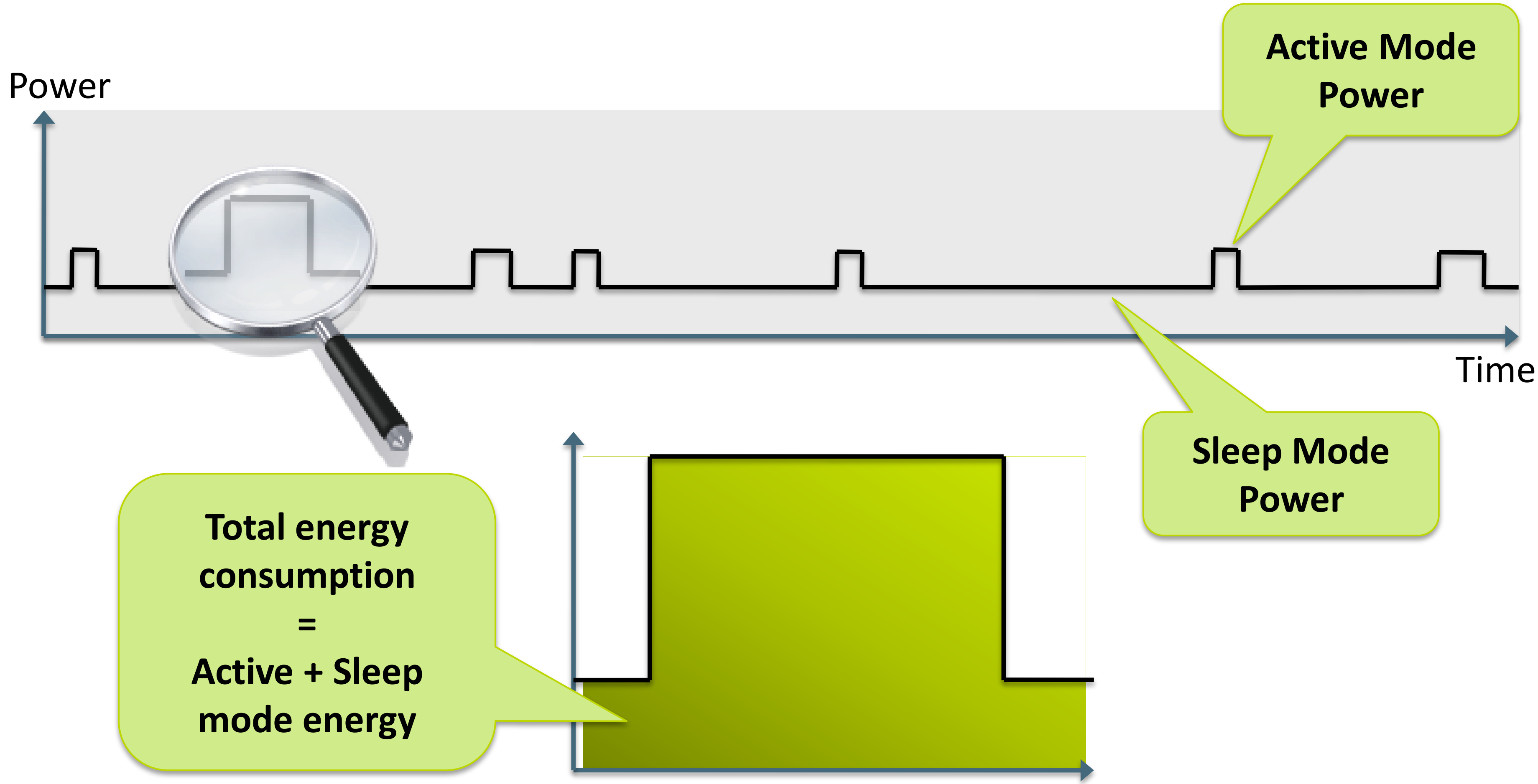
4. System architecture
   - Short wake-up time
   - Autonomous system
   - The right peripherals (focus on applications)
5. Innovative low power design techniques
   - Enable design for clock gating
   - Minimize combinational depth
   - Smart methodology to monitor power consumption
   - Efficient methodology (auto-generation!!!)
   - Sub-threshold design
   - Low power peripherals
   - Joulemeter
   - FOCUS!

10 factors that make the EFM32 the world's most energy friendly microcontroller

5-Feb-13

Battery life vs energy consumption

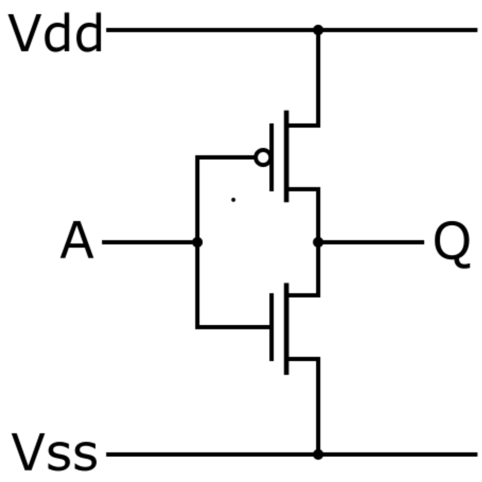
Energy = Power · Time



What Consumes Power?

Dynamic Power

- Caused by switching transistors
- Decreases with smaller feature size
- Caused when switching from a ‘1’ to a ‘0’
- Dynamic Voltage and Frequency Scaling



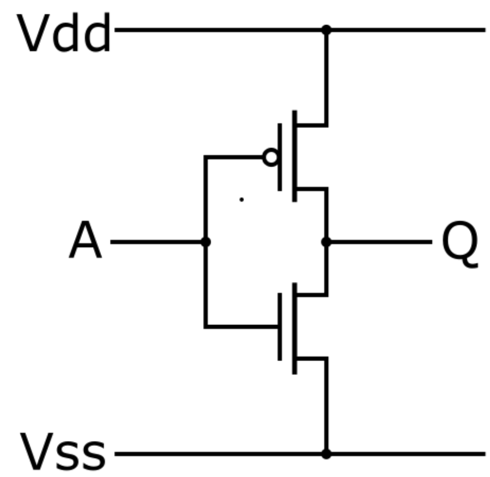
$$P_{Dynamic} \propto \text{Capacitance} \bullet \text{Frequency} \bullet \text{Voltage}^2$$

What Consumes Power?

Static Power

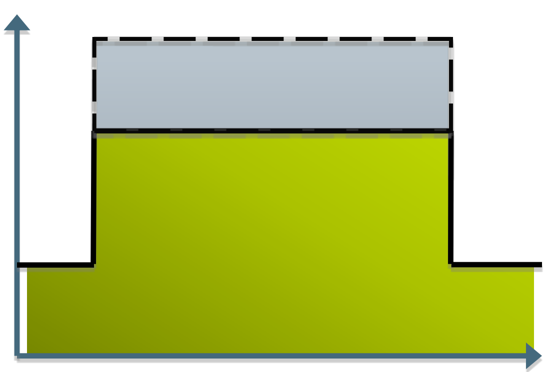
- Caused by leaking transistors
- Increases with smaller feature size
- Turning the power off

$$P_{Static} = C$$



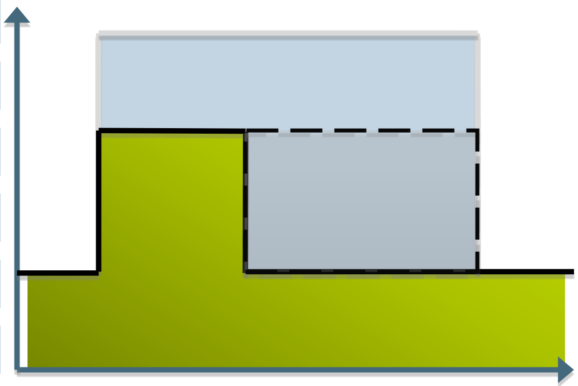
Very low active power consumption 1

µA/MHz @3V @1 MHz	µA/MHz @3V @25 MHz	µA/MHz @3V @32 MHz
220	*180*	*180*



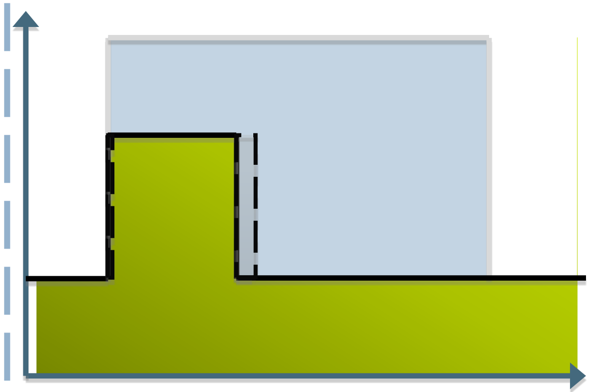
Reduced processing time 2

Cortex-M3 DMIPS/MHZ	µA/MHz @3V @25 MHz
1.25	*180*



Very fast wake-up time 3

Wake-up time from sleep modes
2 µs

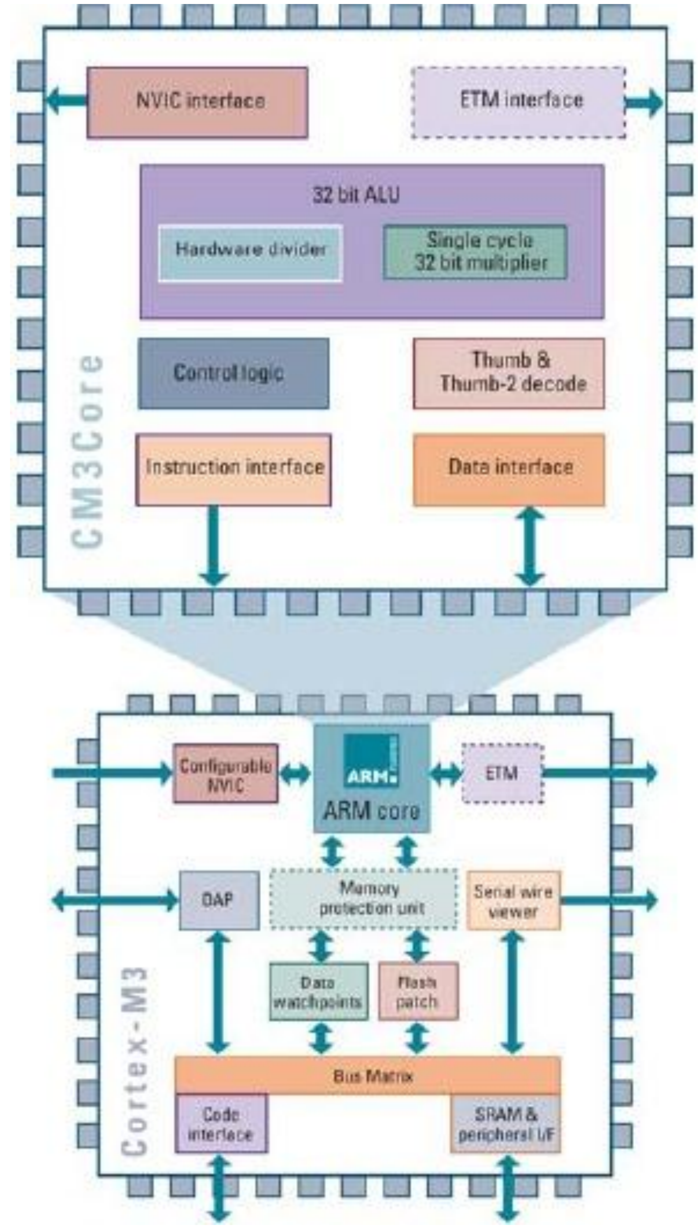


ARM Cortex-M3

2

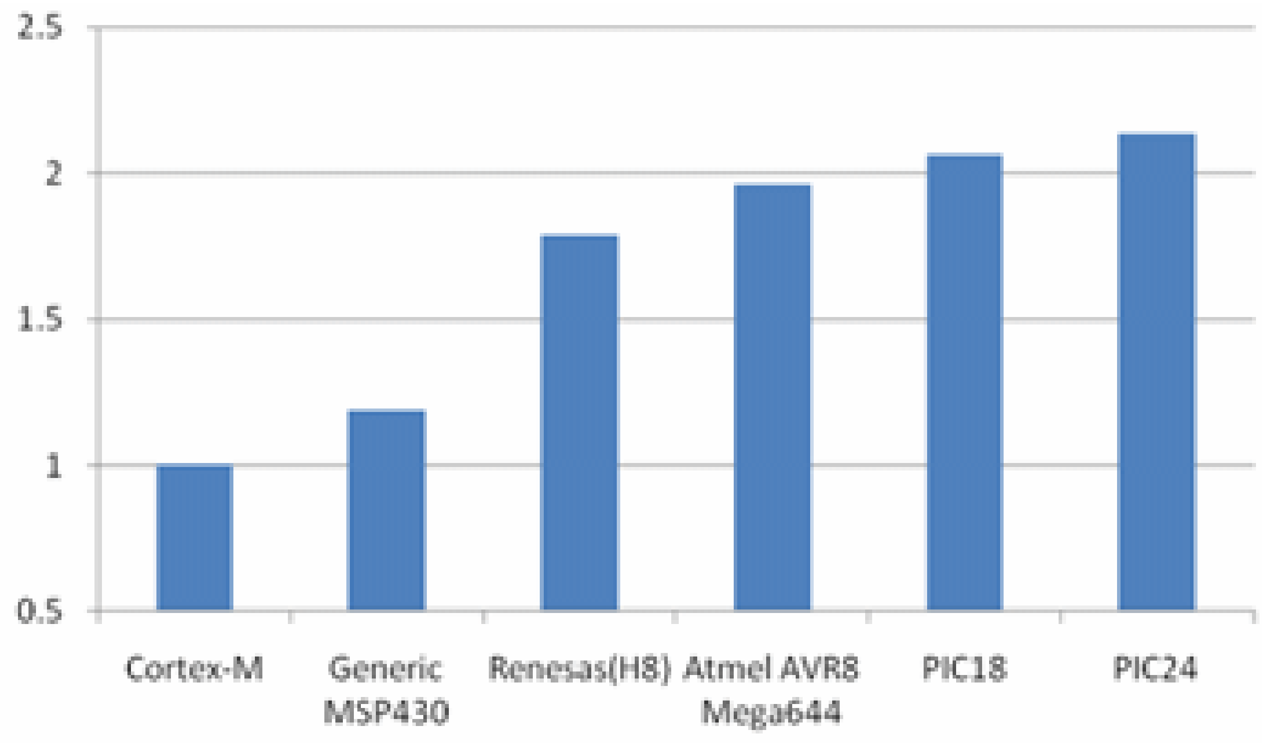
Cortex-M3 highlights

- 1.25 DMIPS/MHz
- Harvard architecture
- Thumb-2 ISA
- Small core, high performance
- Integrated power mode support
- Optimized for low latency, nested interrupts
- Hardware debug support
  - Total 8 breakpoints / watchpoints
  - Flash patching
  - PC sampling
  - Profiling support



ARM Cortex-M3 benchmarks

2



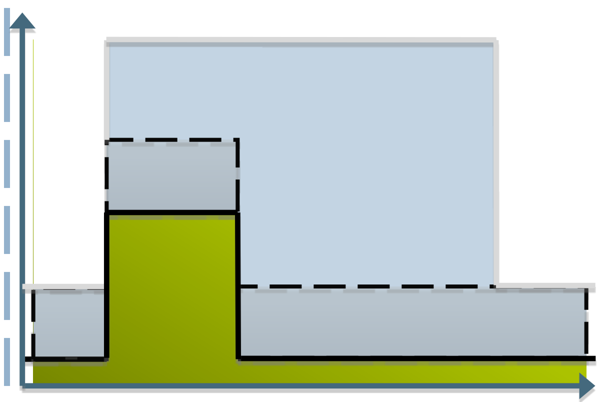
Code size comparison using relative EEMBC CoreMark test size.

EFM32

ENERGY micro

Ultra-low standby current 4

Shutoff current @ 3V	Deep Sleep @ 3V *incl. POR, BOD, RTC, RAM and CPU retained*
20 nA	*900 nA*

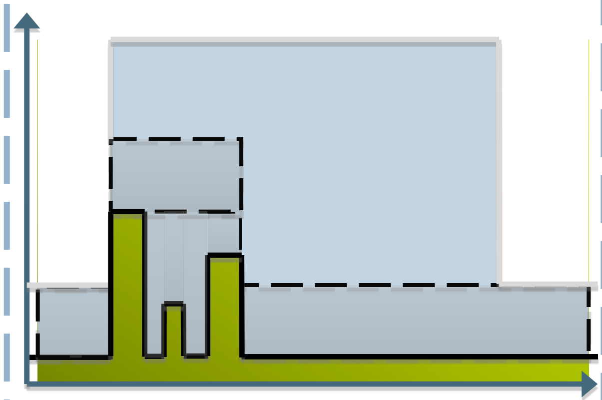


Autonomous peripherals 5

Operation while CPU sleeps

Extensive DMA Support

All peripherals can operate autonomously

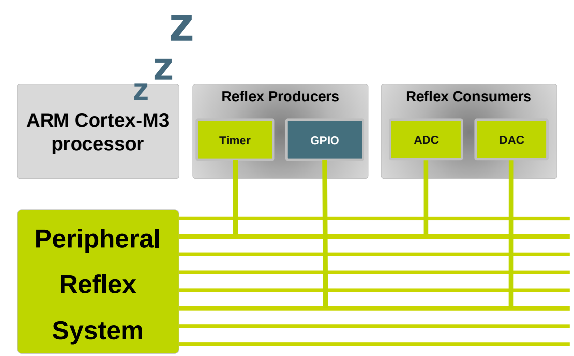


Peripheral Reflex System 6

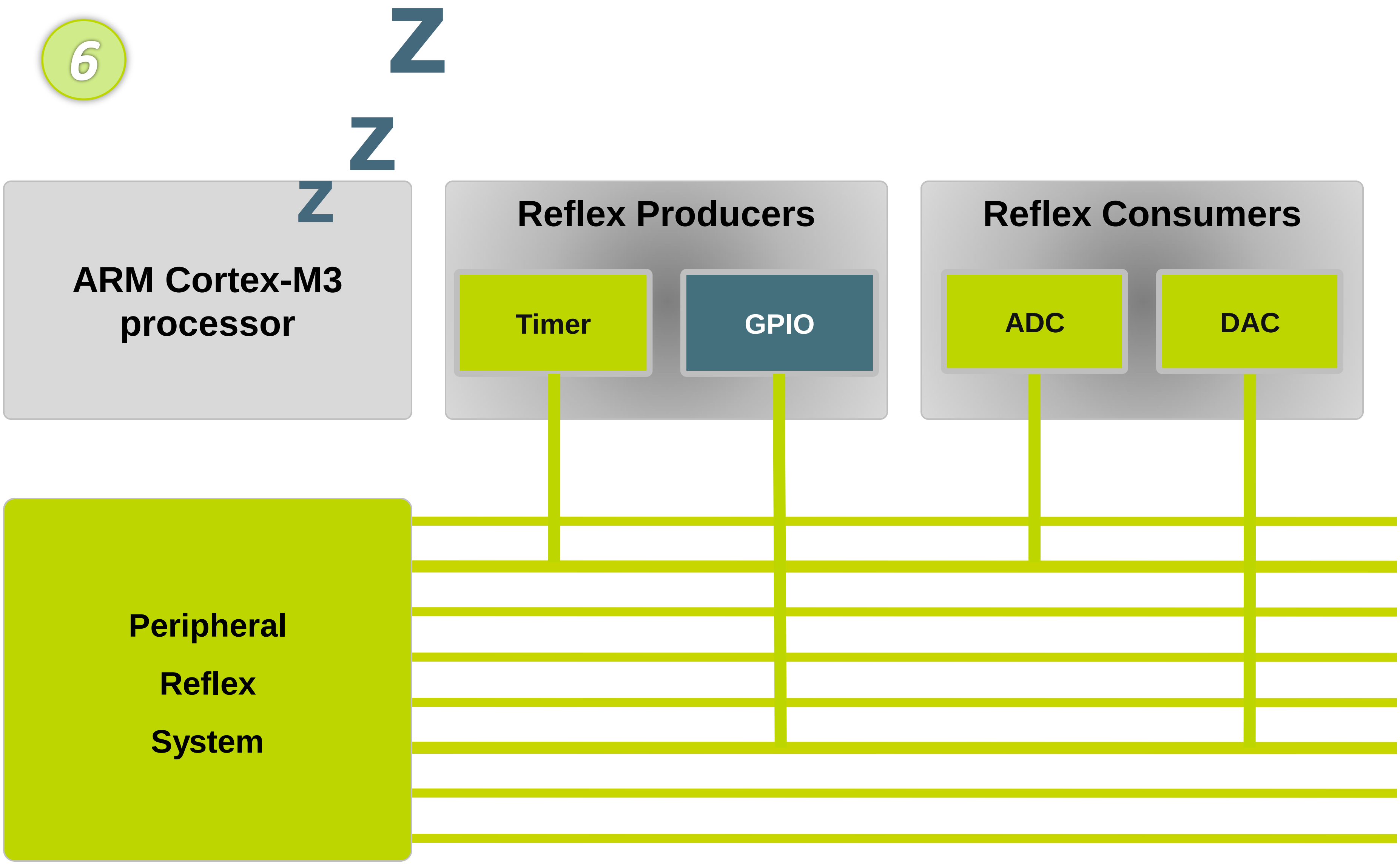
Direct peripheral interconnection system

Boosting the value of autonomous operation

Highly configurable

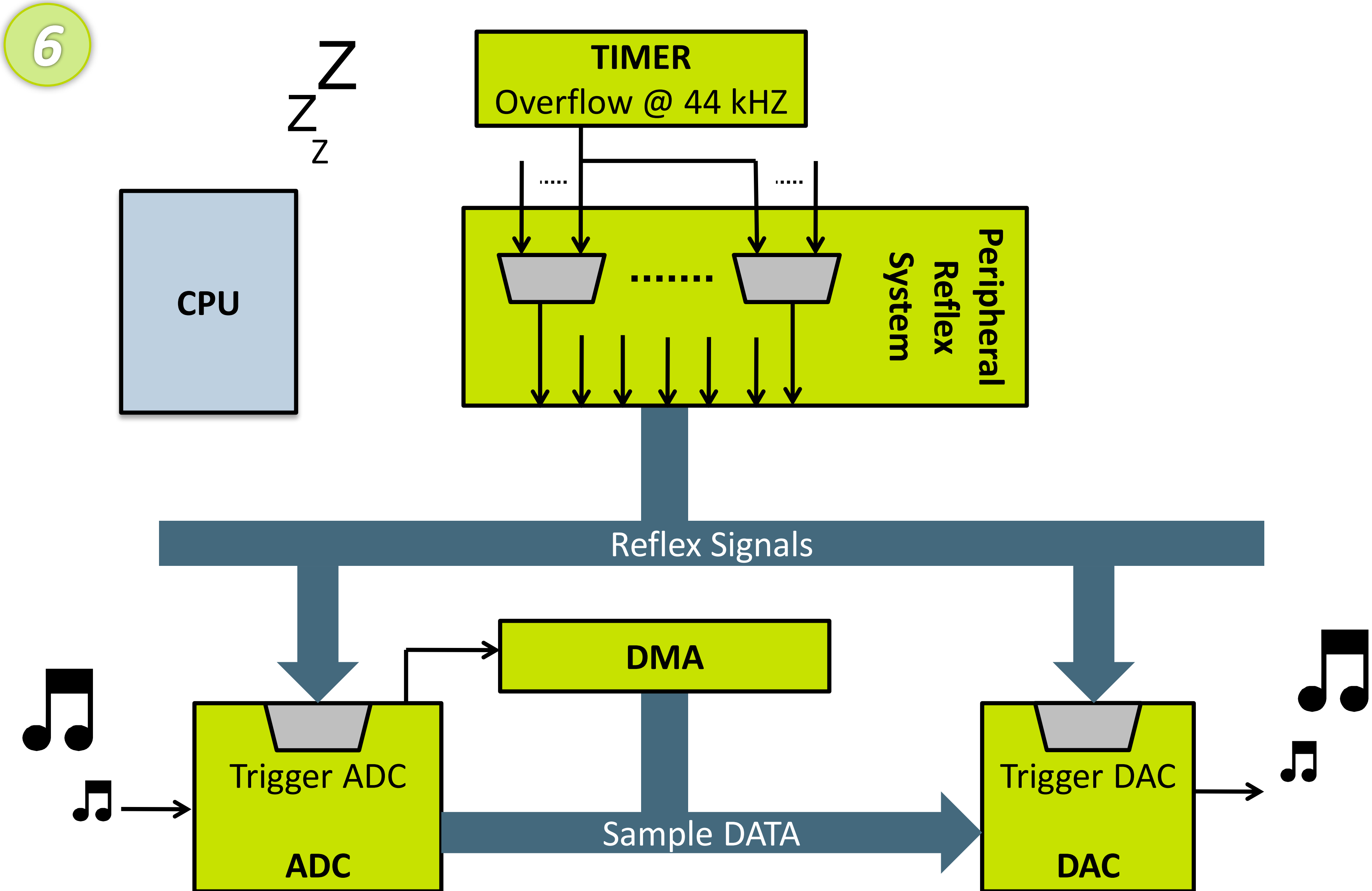


Peripheral Reflex System



5-Feb-13

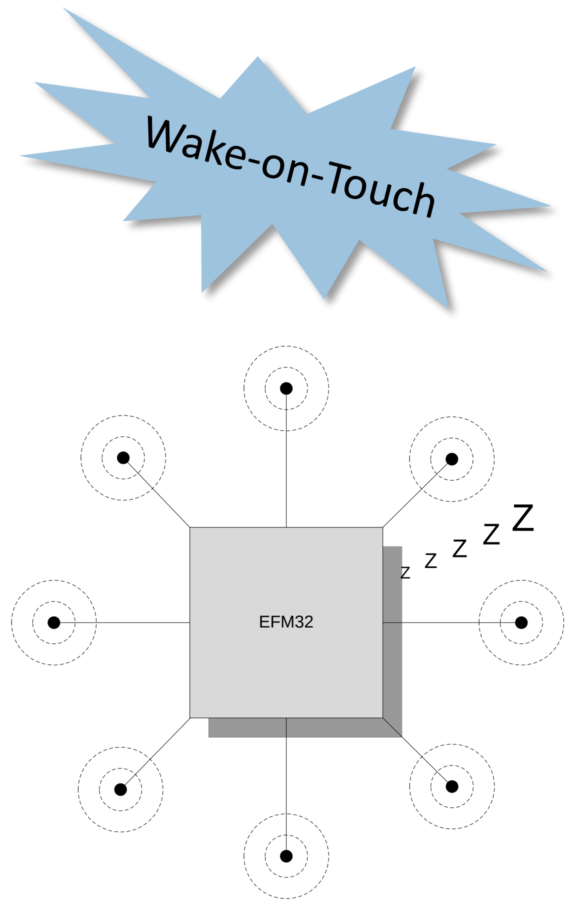
Autonomous sound loop



Low Energy Sensor Interface (LESENSE)

LESENSE Highlights

- **Autonomous sensing in Deep Sleep**
  - **LESENSE with central control logic**
  - **ACMP for sensor input**
  - **DAC for reference generation**
- **Measure up to 16 sensors**
  - **Inductive (LC)**
  - **Capacitive (eg. capacitive buttons)**
  - **General analog sensors (e.g. resistive)**
- **Programmable state machine**
  - **16 states, 4 input channels**
  - **Can do quadrature decoding**
- **Interrupt/PRS on sensor events**



Well architectured Energy Modes 7

EM0 “Run Mode”: 180 µA/MHz

EM1 “Sleep Mode”: 45 µA/MHz

EM2 “Deep Sleep Mode”: 900 nA
RTC, Power-on Reset, Brown-Out Detection
RAM and CPU retained

EM3 “Stop Mode”: 600 nA
Power-on Reset
RAM & CPU retained

EM4“Shutoff Mode”: 20 nA
Power-on Reset

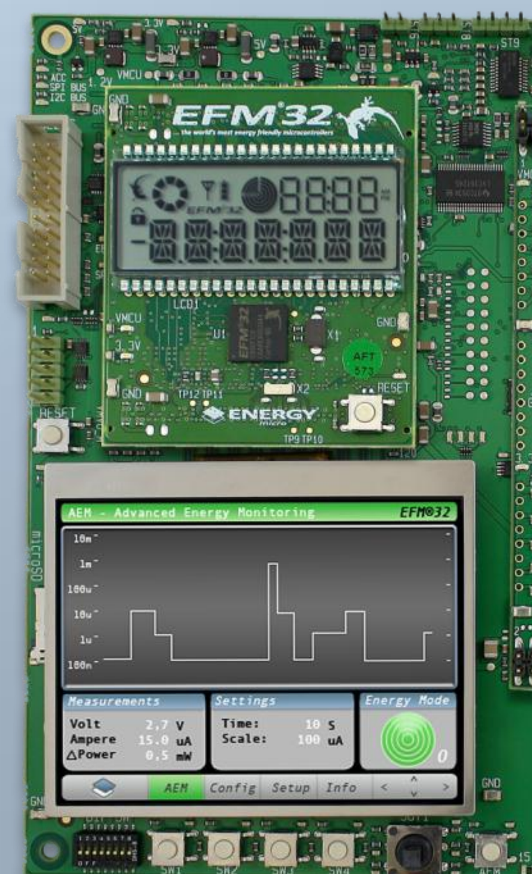
Ultra energy efficient peripherals 8

Analog to Digital Converter
12-bit @ 1 MSamples/s: 350 µA
6-bit @ 1 kSamples/s: 500 nA

Low Energy UART
Full UART with 32 kHz clock
150 nA at 9600 baud/s

LCD Controller
Directly driving 4x40 segment LCD
Boost/Contrast/Animation/Blink
550 nA

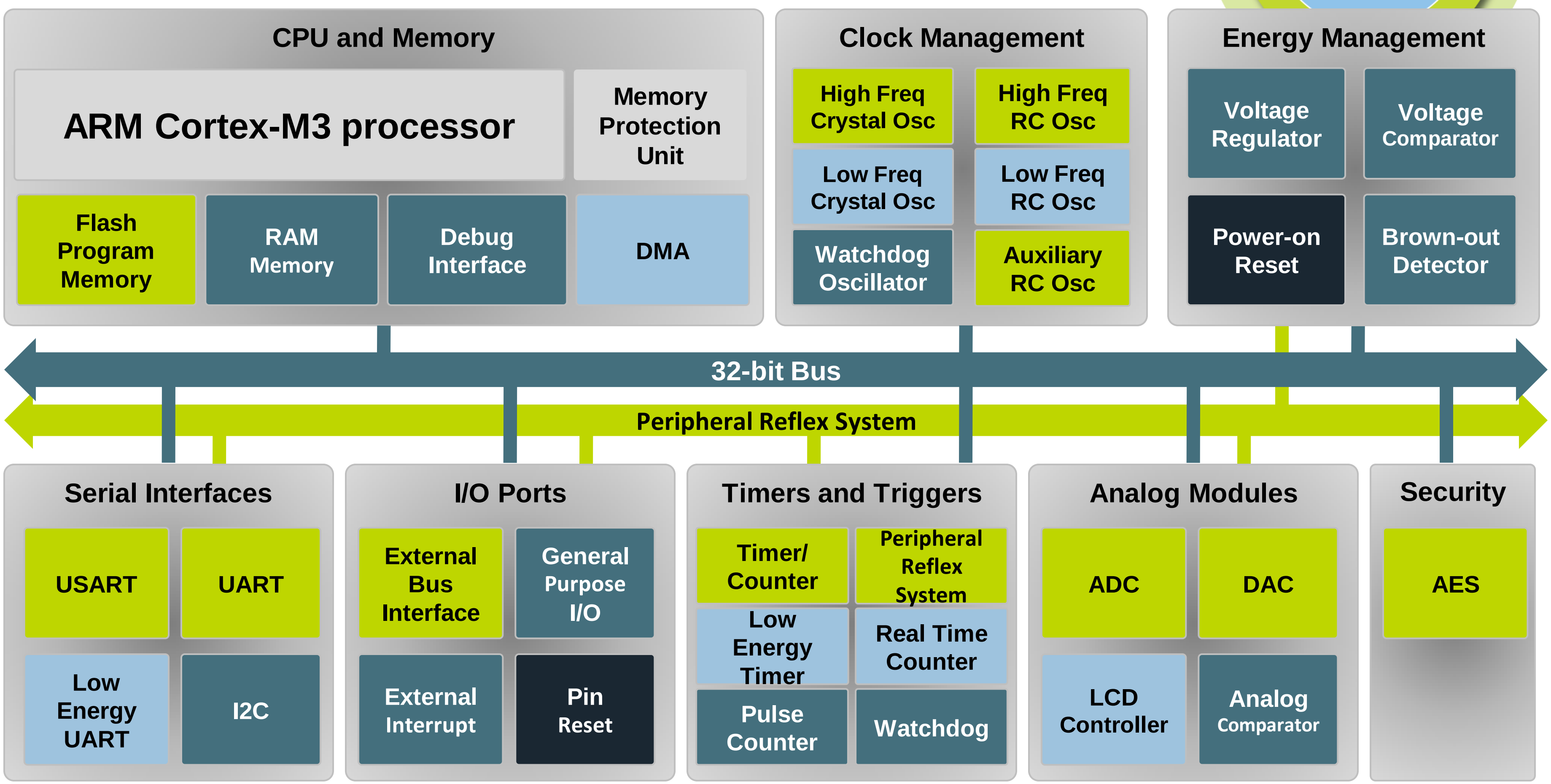
Advanced Energy Monitoring 9



Well architectured Energy Modes

7

EM1 “Sleep Mode”: 45 µA/MHz



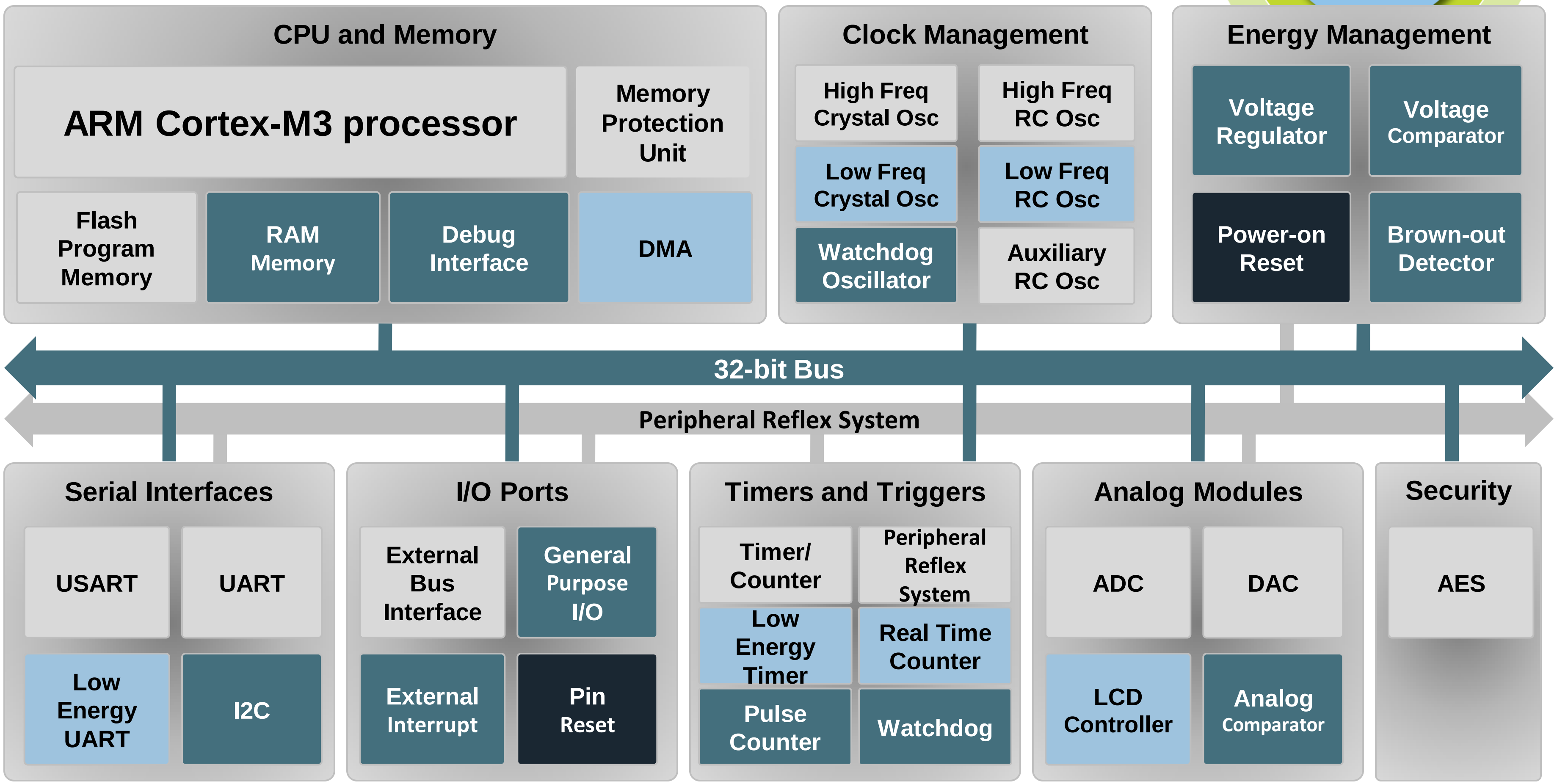
5-Feb-13

Well architected Energy Modes

7

EM2 “Deep Sleep Mode”: 900 nA

(RTC, POR, BOD & RAM & CPU retention active)



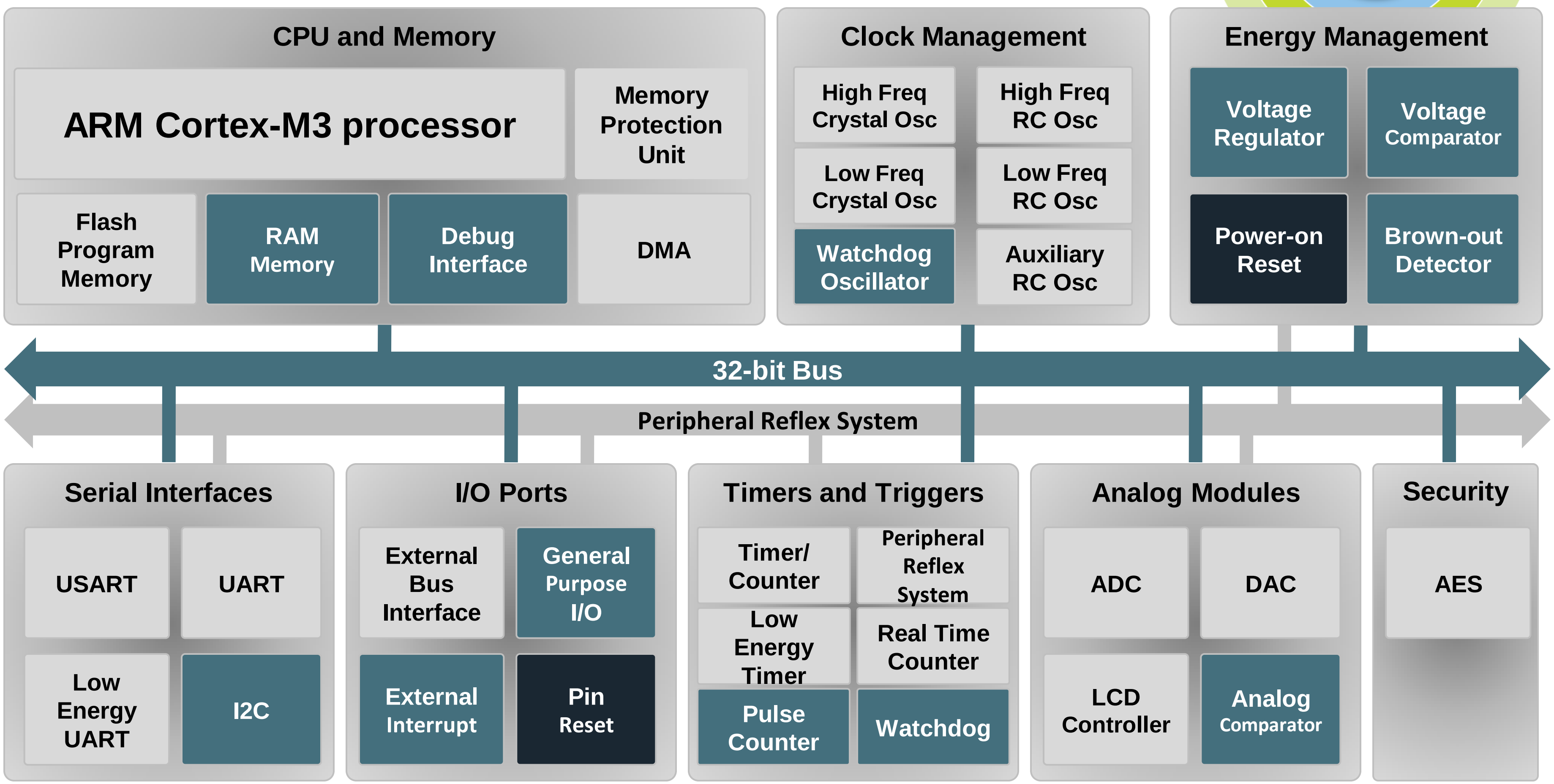
5-Feb-13

Well architectured Energy Modes

7

EM3 “Stop Mode”: 600 nA

(POR, BOD and RAM & CPU retention)



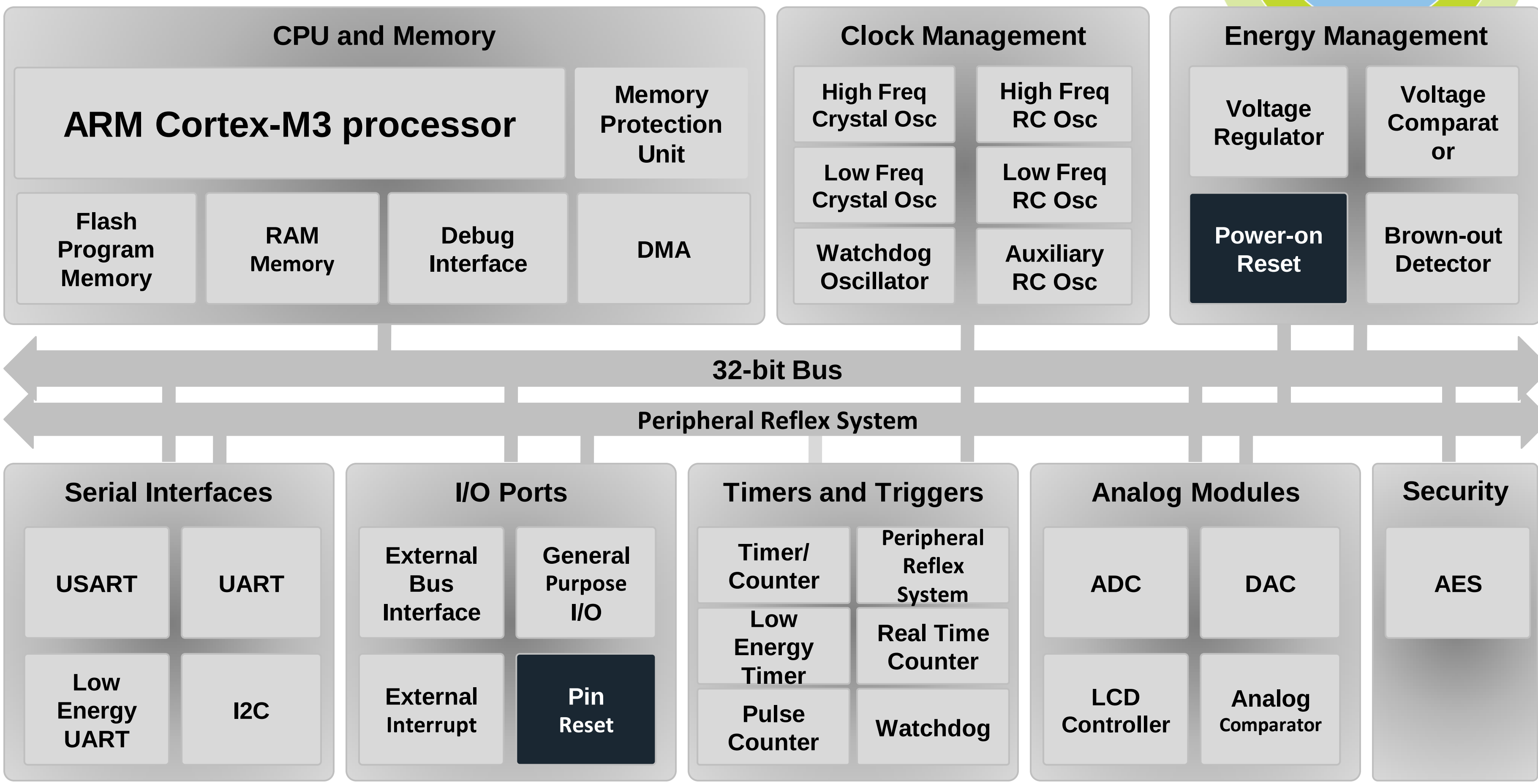
5-Feb-13

Well architectured Energy Modes

7

EM4“Shutoff Mode”: 20 nA

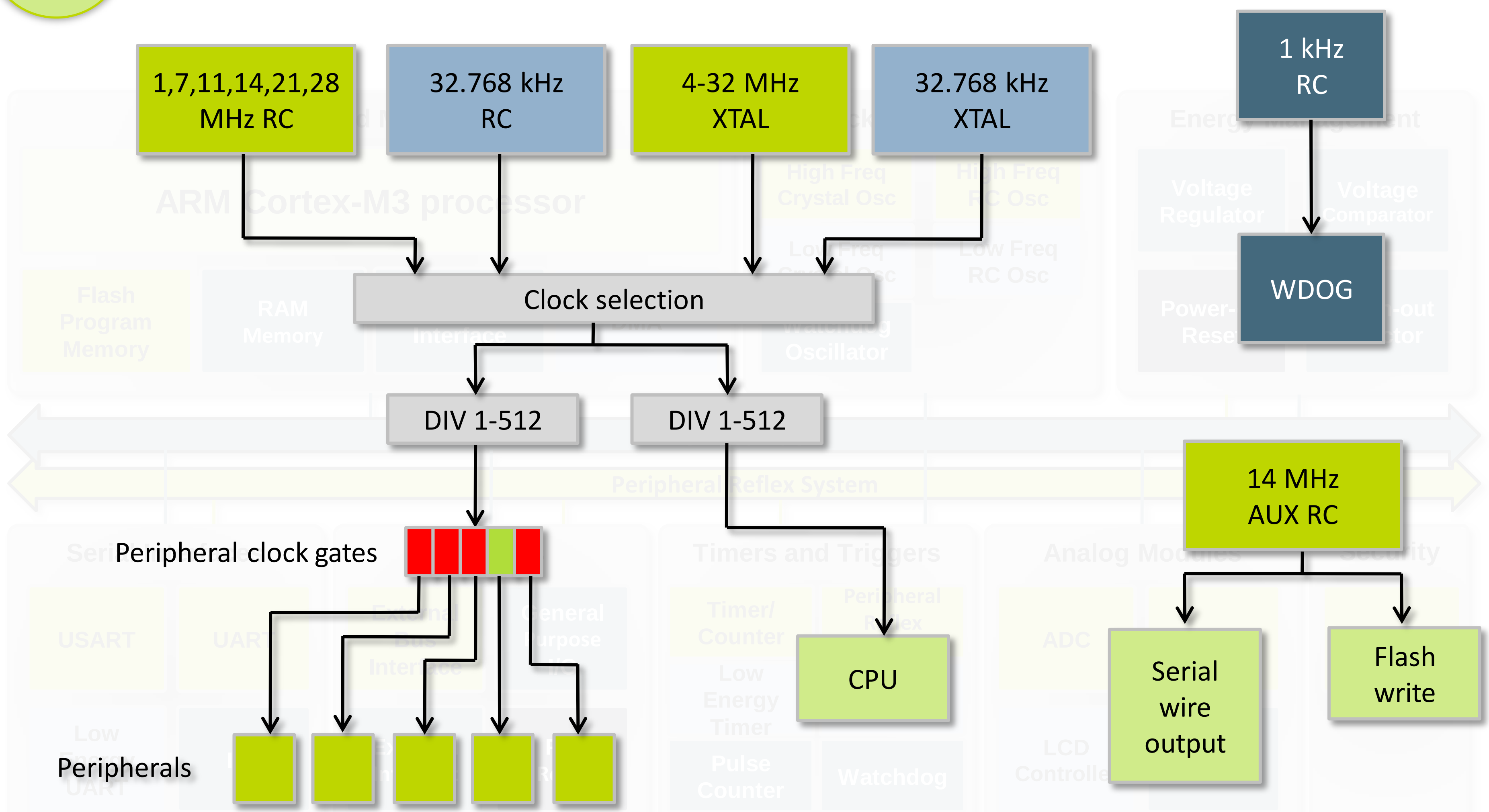
(Power-on Reset)



5-Feb-13

Clocks and Oscillators

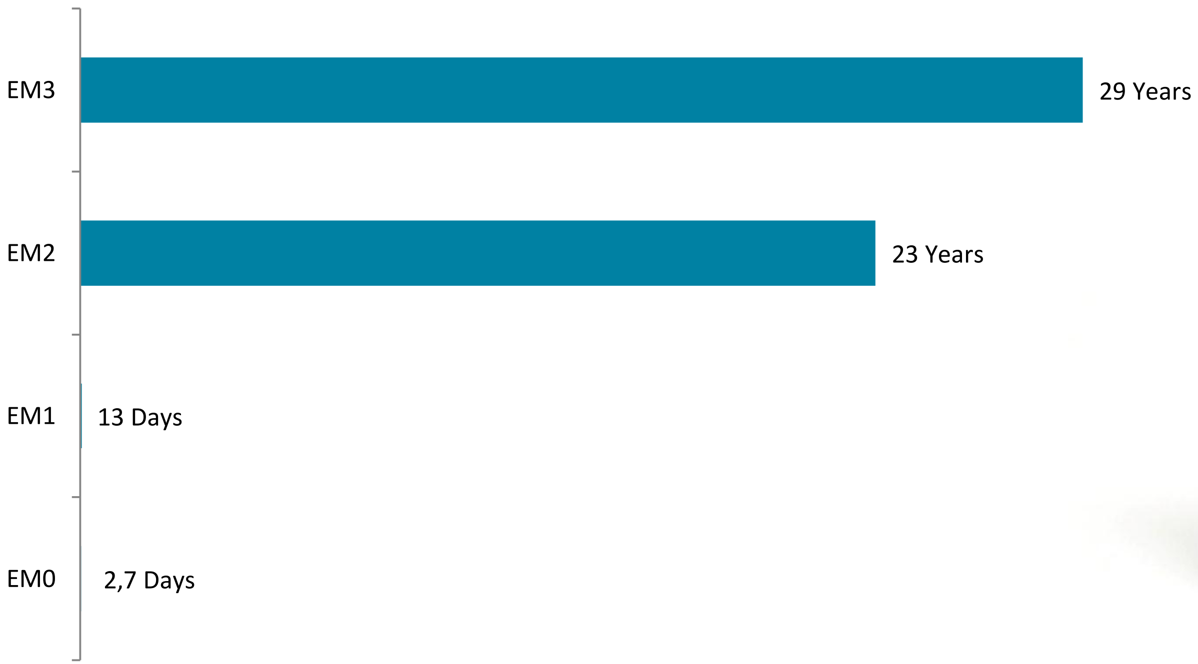
8

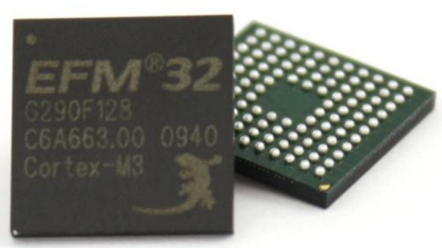


Creating Great Software

Why?

Save power by using the hardware properly!
What do you really need?





5-Feb-13

Creating Great Software

How?

1. Energy Friendlyness is an attitude!
2. Explore Early. Once the code is written, it won't change.
3. Hardware resources are often, but not always, cheaper.
4. "What is the minimum resources I need at this point?"
5. Polling vs Interrupts vs DMA
6. Profile your code to find bugs and opportunities.

Design Space Exploration

Blood Glucose Meter

Typical use case:

•Most of the time not in use (Standby)

•Want quick measurements when in use

What is the impact of changing things?

Standby → Wakup, Greeting → Measurement → Data Logging → Display Data

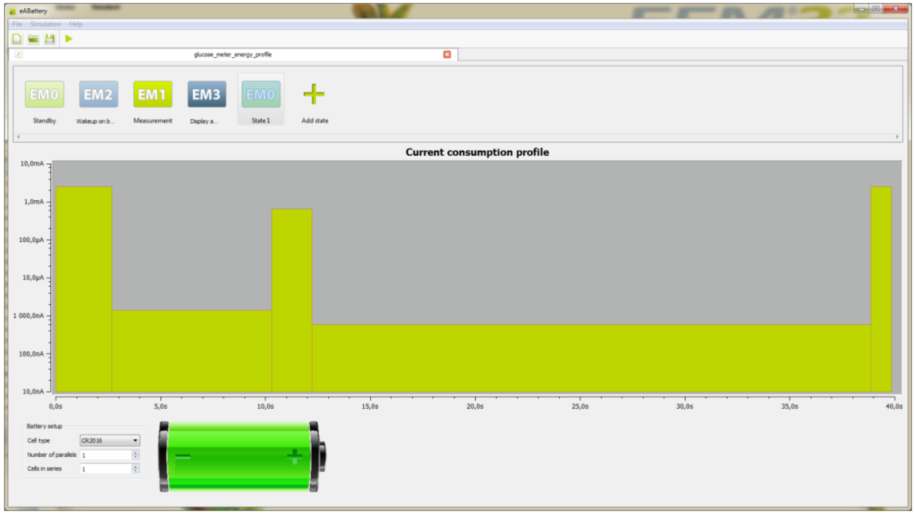
EM2 | EM2 | EM0 | EM0 | EM2

5-Feb-13

energyAware Battery Estimator

Tools will let you explore ”what if” scenarios

- What if I want to display something in Standby?
- What if I use more time to do the measurements?
- What if I have an external component drawing power?



Sampling the ADC

Using CPU + TIMER + DMA

CPU + TIMER [EM0]

CPU + ADC [EM0]

```
While () {
   While  (TIMER->CNT < 100) {}
   Sample = ADC->DATA;
   TIMER->CNT = 0;
}
```

Using ADC + TIMER + PRS + DMA

TIMER [EM1]

PRS [EM1]

ADC [EM1]

DMA [EM1]

Memory [EM1]

Using RTC + ADC + CPU

RTC [EM2]

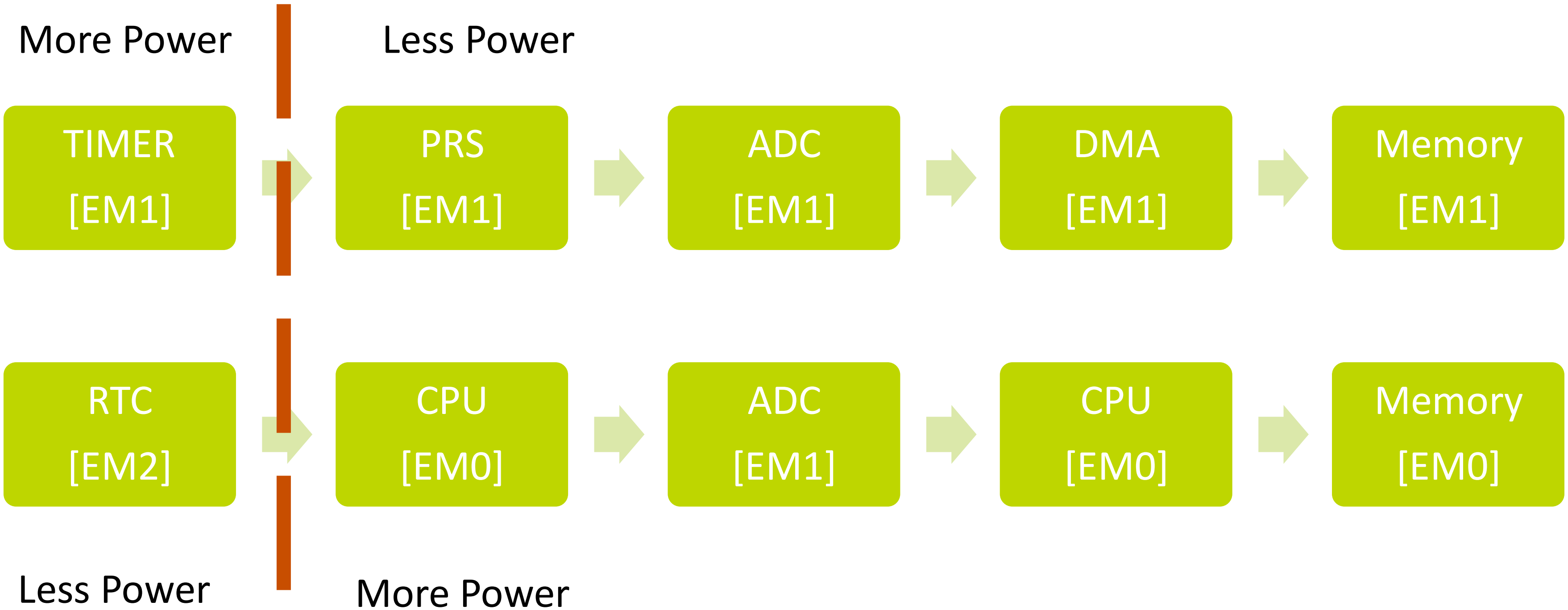
CPU [EM0]

ADC [EM1]

CPU [EM0]

Memory [EM0]

Sampling the ADC II



Left side is sampling frequency dependant. Right side is not.

Use top for high frequencies, bottom for low frequencies.

But.... What frequency?

EFM®32 Energy Aware Profiling

Code Development

C source code

↓

Compiler IDE

↓

ELF/BIN output

→

EFM32 Tools

EFM32 Starter + Development Kit

AEM* Advanced Energy Monitor

Program Counter Sample

Current Voltage Time

→

energyAware Profiler

libelf/ libdwarf

↓

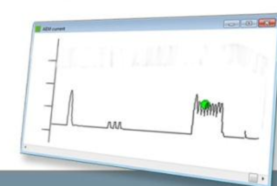
File Function Library

↓

Data Correlation

↓

GUI

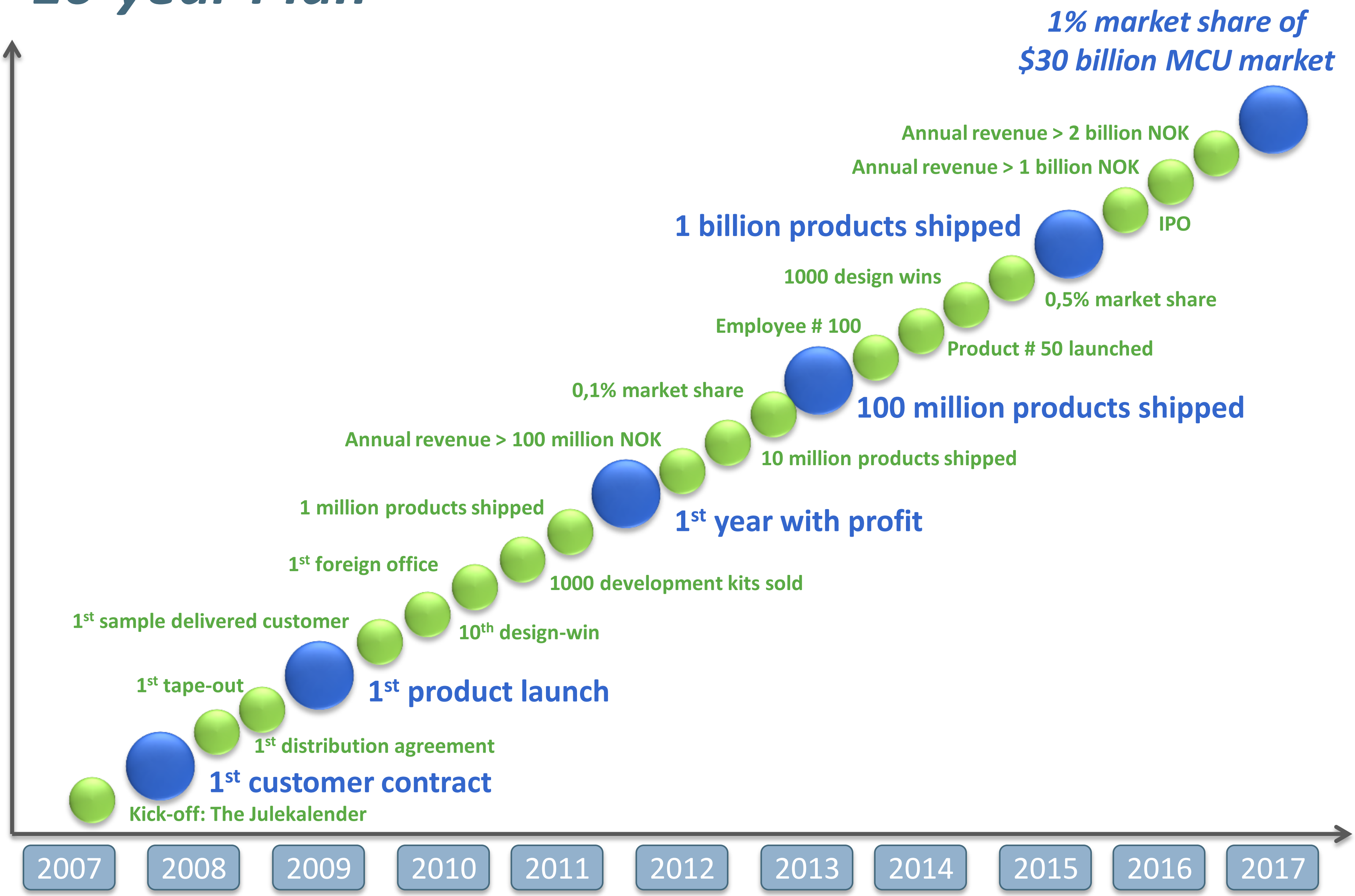


5-Feb-13

ENERGY micro

www.energymicro.com

10-year Plan



5-Feb-13

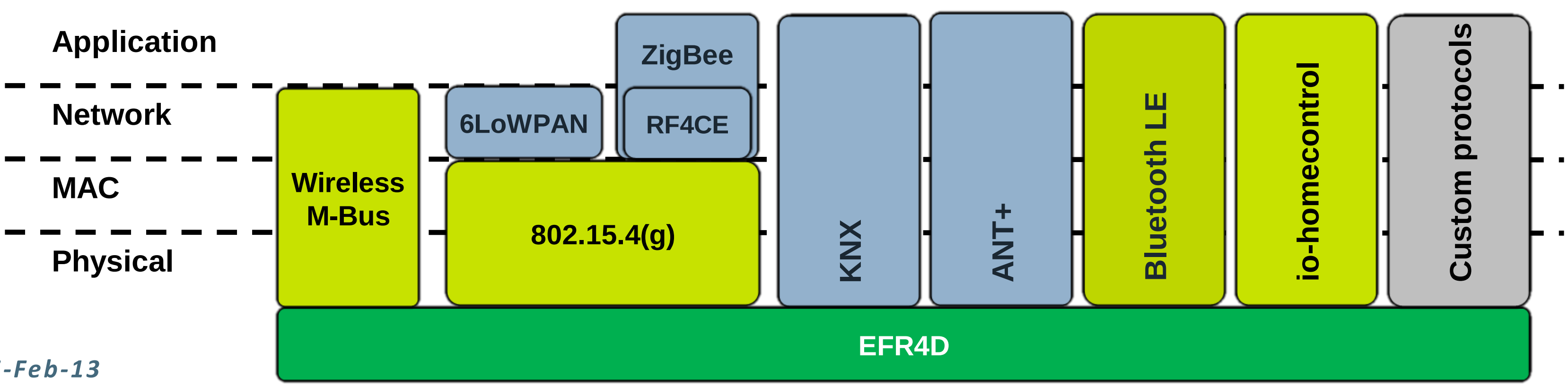
EFR4D – High Performance and Low energy consumption

World class energy consumption

- Continuous RX mode: 4 mA
- Continuous TX mode:
  - 5 mA, 0 dBm @ 868 MHz
  - 6 mA, 0 dBm @ 2.4 GHz
  - 14 mA, +10 dBm @ 868 MHz
- ARM Cortex-M3
- Up to 256 KB Flash and 16 KB RAM

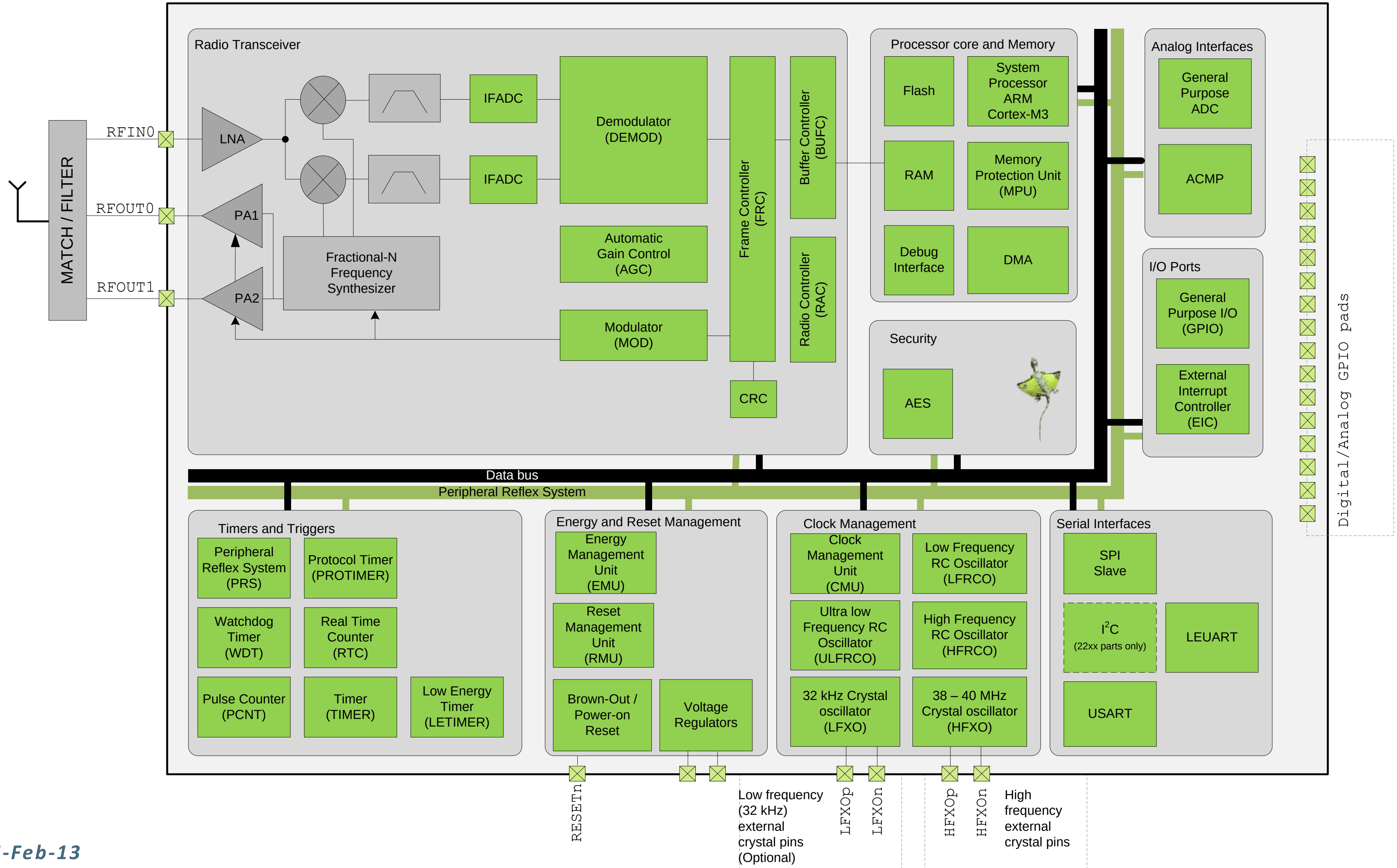
World class radio performance

- Sensitivity:
  - -121 dBm, FSK 1.2 kbit/s
  - -110 dBm, FSK 38.4 kbit/s
  - -101 dBm, O-QPSK DSSS 250 kbit/s
  - -95 dBm, GFSK 1 Mbit/s
- Max output power:
  - +17 dBm @ 868 MHz
  - +13 dBm @ 2.4 GHz
- (4)(G)FSK/(G)MSK/O-QPSK/(D)BPSK/OOK/ASK
- Selectable baudrates: 300 bit/s - 4 Mbit/s



EFR®

System-on-Chip RF devices with ARM Cortex-M3



5-Feb-13

Gecko and Draco?

- Cold-blooded amphibians and reptiles are the most successful creatures ever to walk on land
- Only 10% energy consumption compared to mammals of similar size

"Reptiles and amphibians are sometimes thought of as slow, dim-witted and primitive. In fact they can be lethally fast, spectacularly beautiful, surprisingly affectionate and extremely sophisticated"

- Sir David Attenborough

5-Feb-13

240+ Scalable Low Energy EFM32s

Software Compatible Pin Compatible within each package

Family	Optional Features	QFN24	QFN32	QFN64	QFP48	QFP64	QFP100	BGA48	BGA112	BGA120
Wonder	USB LCD TFT DSP with FPU			EFM32WG940 EFM32WG330 EFM32WG840 EFM32WG230		EFM32WG942 EFM32WG332 EFM32WG842 EFM32WG232	EFM32WG980 EFM32WG380 EFM32WG880 EFM32WG280		EFM32WG990 EFM32WG390 EFM32WG890 EFM3WG290	EFM32WG995 EFM32WG395 EFM32WG895 EFM32WG295
Giant	USB LCD TFT			EFM32GG940 EFM32GG330 EFM32GG840 EFM32GG230		EFM32GG942 EFM32GG332 EFM32GG842 EFM32GG232	EFM32GG980 EFM32GG380 EFM32GG880 EFM32GG280		EFM32GG990 EFM32GG390 EFM32GG890 EFM32GG290	EFM32GG995 EFM32GG395 EFM32GG895 EFM32GG295
Leopard	USB LCD TFT			EFM32LG940 EFM32LG330 EFM32LG840 EFM32LG230		EFM32LG942 EFM32LG332 EFM32LG842 EFM32LG232	EFM32LG980 EFM32LG380 EFM32LG880 EFM32LG280		EFM32LG990 EFM32LG390 EFM32LG890 EFM32LG290	EFM32LG995 EFM32LG395 EFM32LG895 EFM32LG295
Gecko	LCD		EFM32G210 EFM32G200	EFM32G840 EFM32G230	EFM32G222	EFM32G842 EFM32G232	EFM32G880 EFM32G280		EFM32G890 EFM32G290	
Tiny	LCD	EFM32TG110 EFM32TG108	EFM32TG210	EFM32TG840 EFM32TG230	EFM32TG822 EFM32TG222	EFM32TG842 EFM32TG232		EFM32TG825 EFM32TG225		
Zero		EFM32ZG110 EFM32ZG108	EFM32ZG210		EFM32ZG222					

ZERO
Total 16 MCUs
Flash: 4 - 32
RAM: 2 - 4

TINY
Total 35 MCUs
Flash: 4 - 32
RAM: 2 - 4

GECKO
Total 31 MCUs
Flash: 16 – 128
RAM: 8 - 16

LEOPARD
Total 60 MCUs
Flash: 64 - 256
RAM: 32

GIANT
Total 40 MCUs
Flash: 512 - 1024
RAM: 128

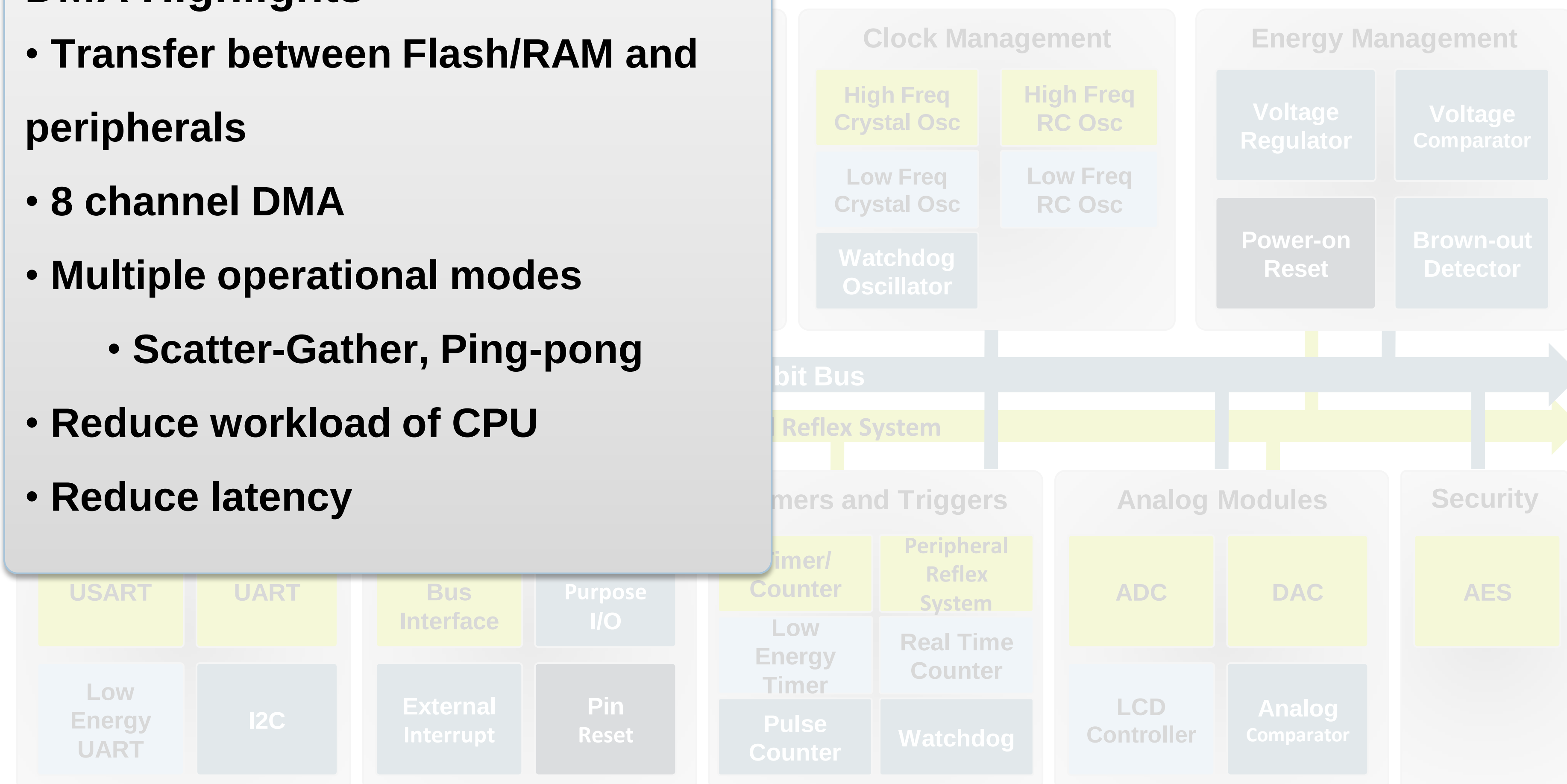
WONDER
Total 60 MCUs
Flash: 64 - 256
RAM: 32

Direct Memory Access Controller

5

DMA Highlights

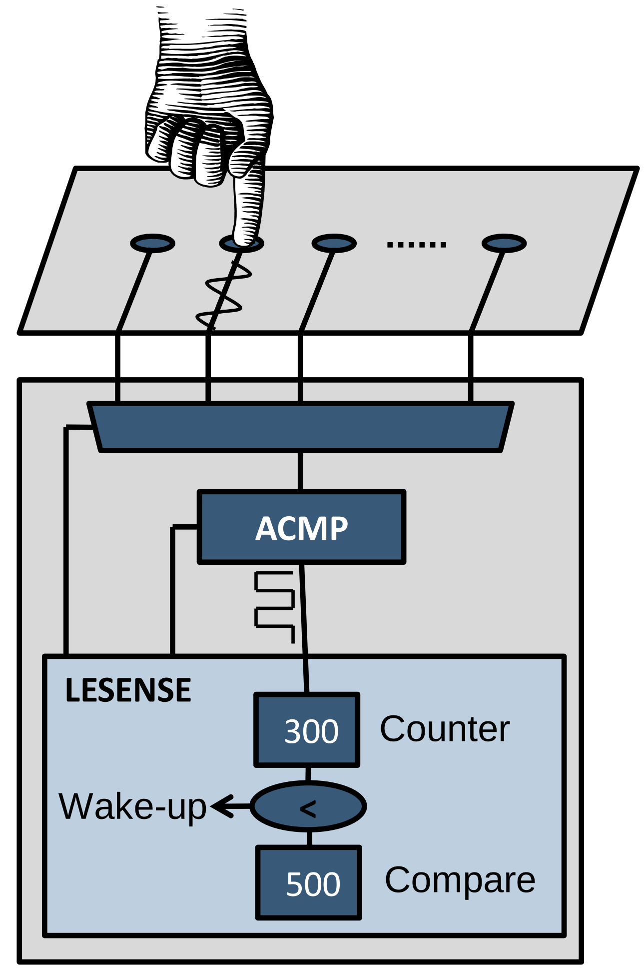
- Transfer between Flash/RAM and peripherals
- 8 channel DMA
- Multiple operational modes
  - Scatter-Gather, Ping-pong
- Reduce workload of CPU
- Reduce latency



5-Feb-13

LESENSE – Capacitive Example

- Analog Comparators measure one input at a time
- Counts oscillations for a given time period
- Touched sensor gives lower frequency
- Performs action if threshold is breached
  - Wake-up
  - State-machine input
  - Buffer results
- 1.2 µA @ 20 Hz



LESENSE – Resistive Example

- Capacitor charged to VDD during excitation
- Sample ACMP output after a programmable time
  - Wake-up
  - Buffer results
  - State Machine input
- Adjustable time period before sampling