Embedded Systems

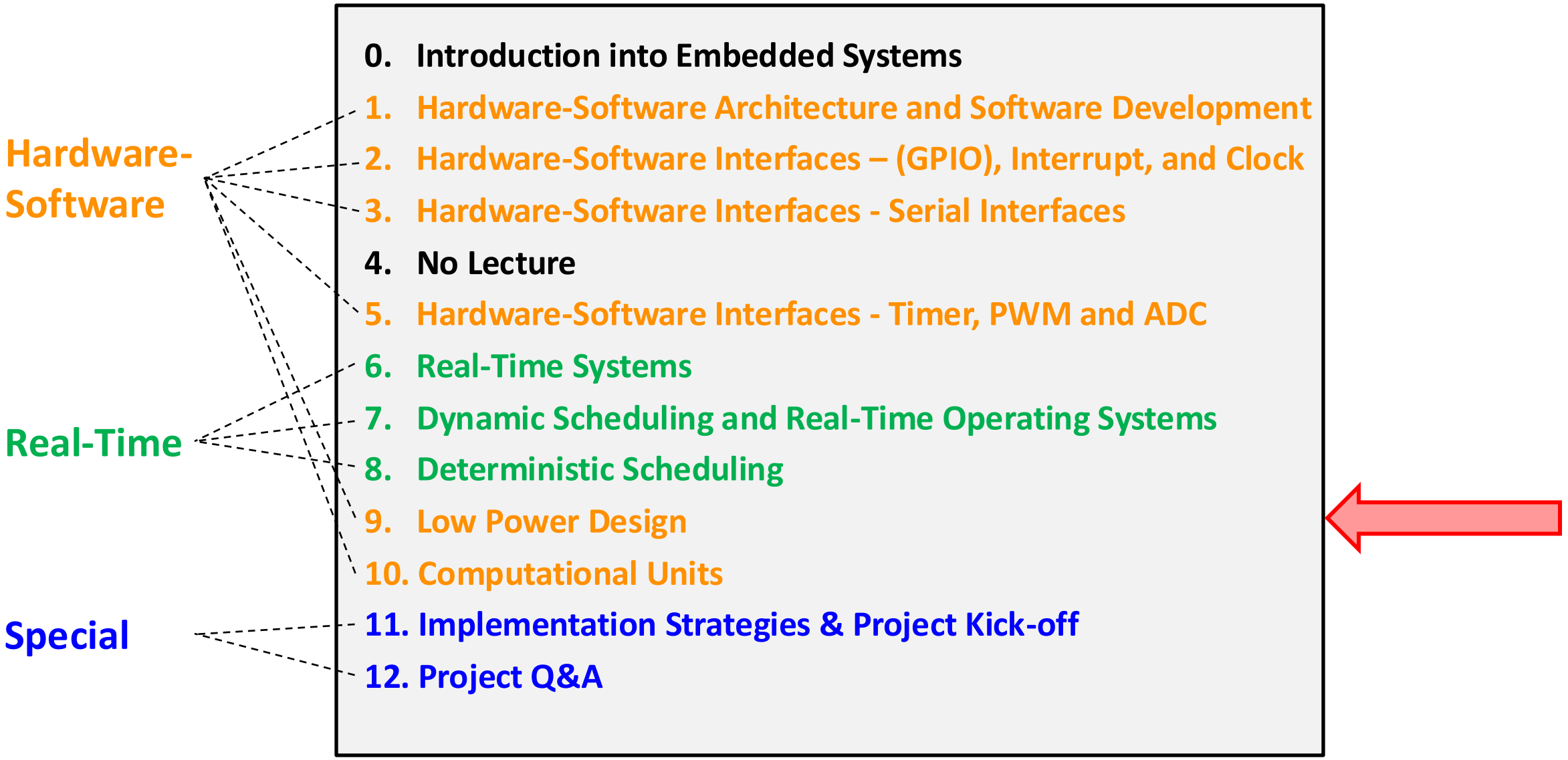
Lecture 9
Low Power Design

Michele Magno

D-ITET Center for Project-Based Learning

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Where We are


Hardware-Software
Real-Time
Special
0. Introduction into Embedded Systems
1. Hardware-Software Architecture and Software Development
2. Hardware-Software Interfaces – (GPIO), Interrupt, and Clock
3. Hardware-Software Interfaces - Serial Interfaces
4. No Lecture
5. Hardware-Software Interfaces - Timer, PWM and ADC
6. Real-Time Systems
7. Dynamic Scheduling and Real-Time Operating Systems
8. Deterministic Scheduling
9. Low Power Design
10. Computational Units
11. Implementation Strategies & Project Kick-off
12. Project Q&A

General Remarks

Power and Energy Consumption

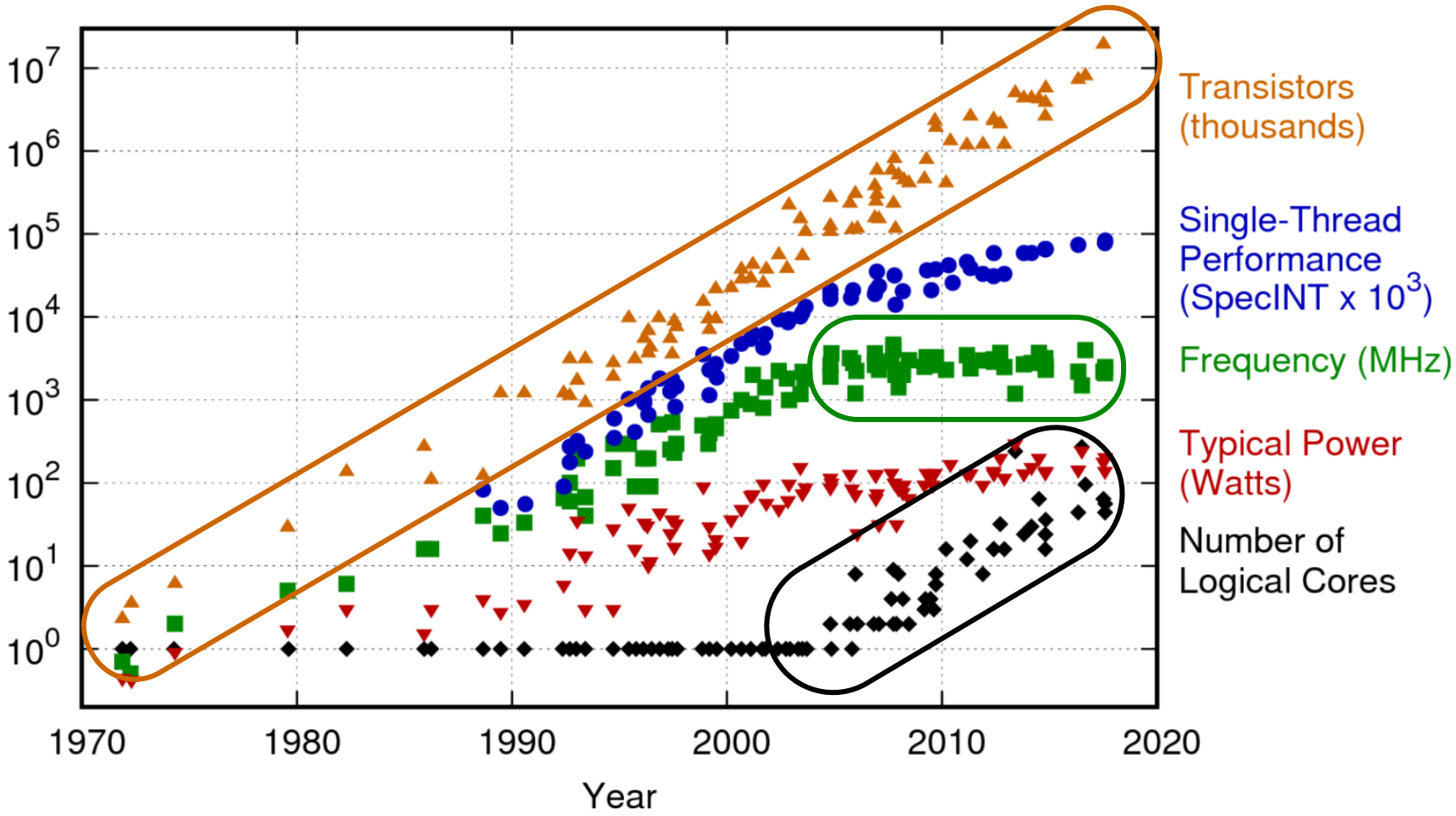
- Statements that are true since a decade or longer:

  "Power is considered as the most important constraint in embedded systems."
  [in: L. Eggermont (ed): Embedded Systems Roadmap 2002, STW]

  "Power demands are increasing rapidly, yet battery capacity cannot keep up."
  [in Diztel et al.: Power-Aware Architecting for data-dominated applications, 2007, Springer]

- Main *reasons* are:
  - power provisioning is expensive
  - battery capacity is growing only slowly
  - devices may overheat
  - energy harvesting (e.g. from solar cells) is limited due to the relatively low energy density available

Trends in Electronics

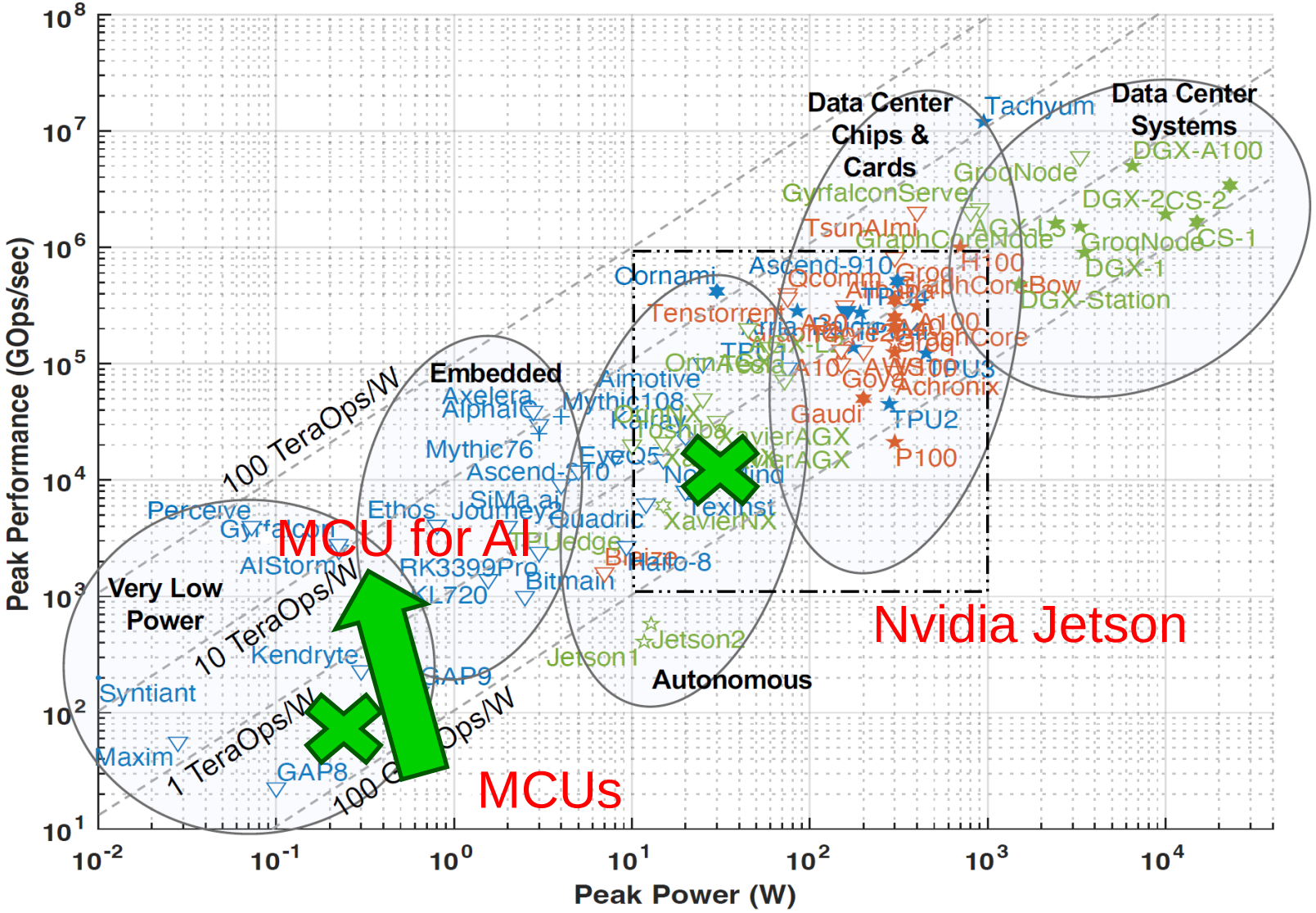


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

- Number of Transistors experience exponential growth: More's Law
- Clock frequency saturates
- Multi-Core systems

Setting Things into Perspective

Reuther 22: arXiv:2210.04055



Legend

Computation Precision

- analog
- int1
- int2
- int4.8
- int8
- Int8.32
- int16
- int12.16
- int32
- fp16
- fp16.32
- fp32
- fp64

Form Factor

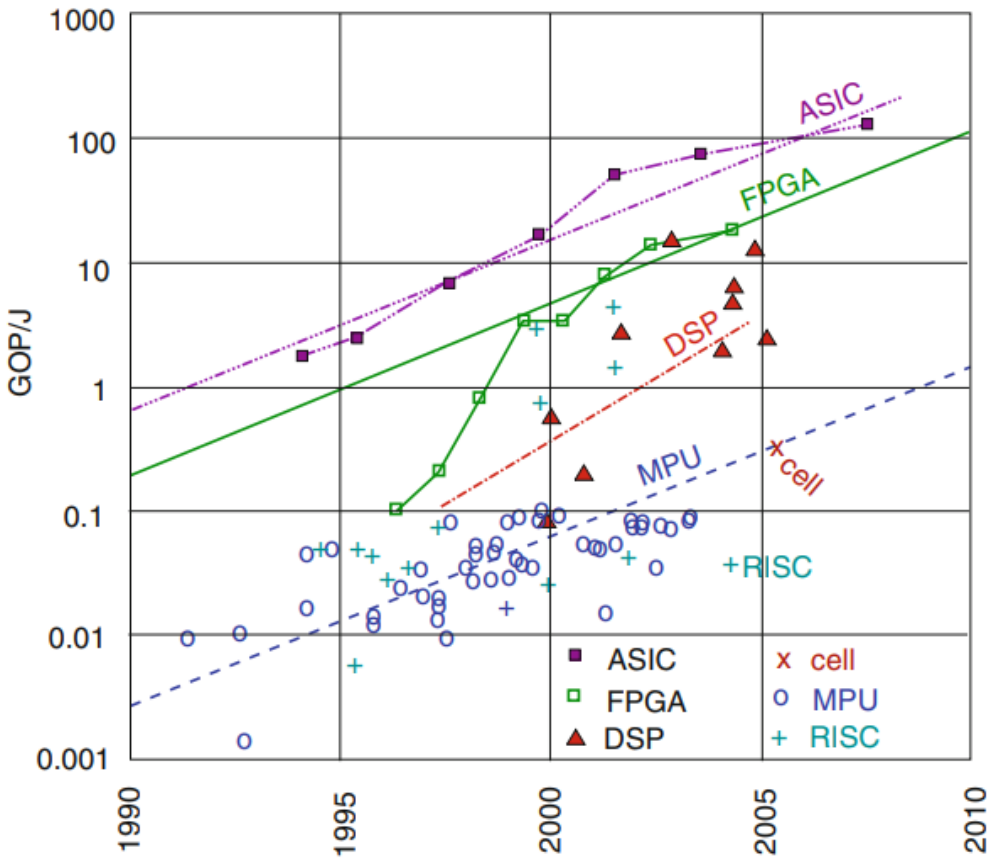
- Chip
- Card
- System

Computation Type

- Inference
- Training

Energy Efficiency

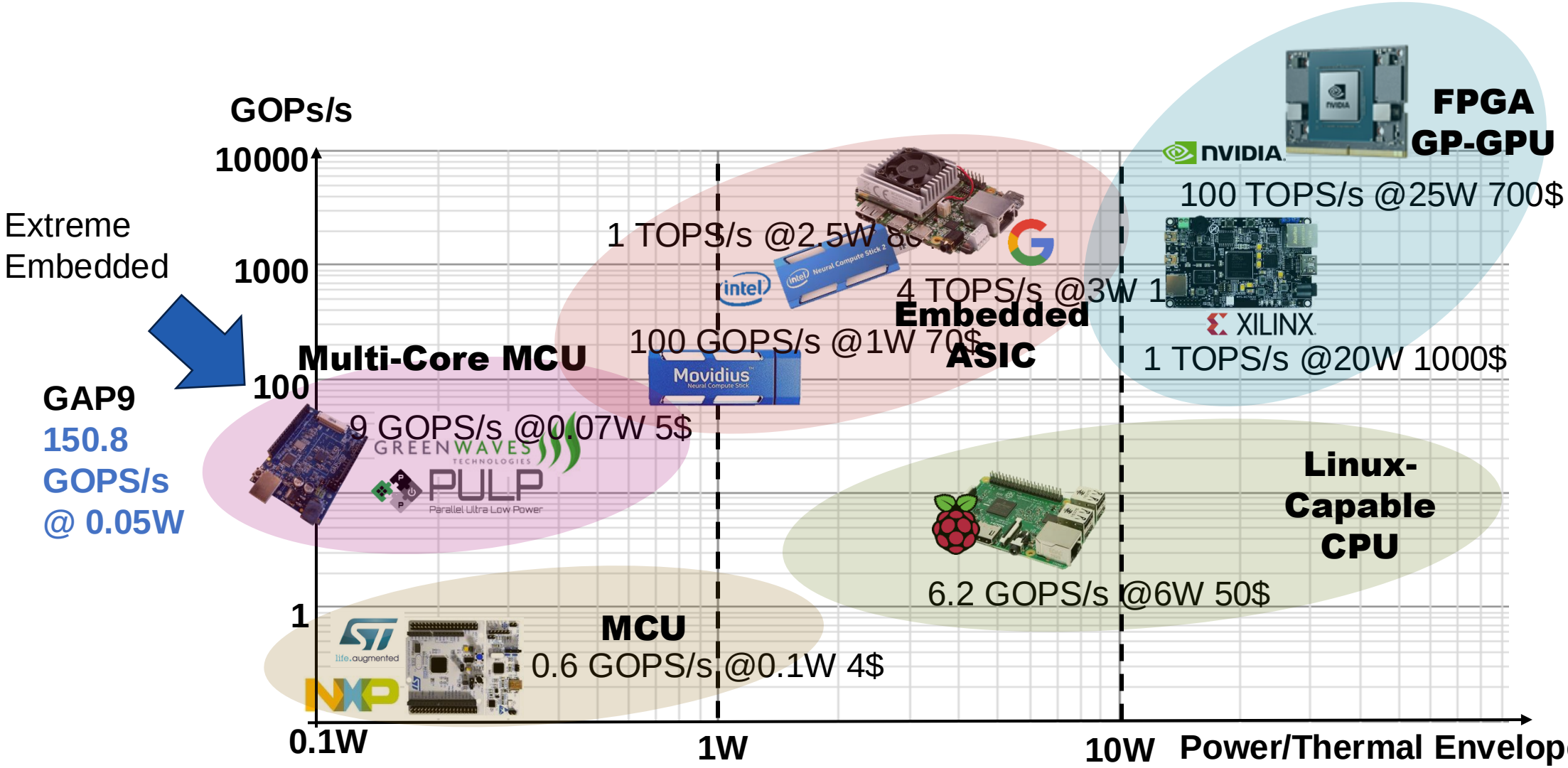
- It is necessary to *optimize hardware and software*
- Use of *heterogeneous architectures* to adapt for required performance and to application classes
- Apply *specialization techniques*
- All of them have optimized efficiency over time



GOP/J: Giga-Operations per Joule

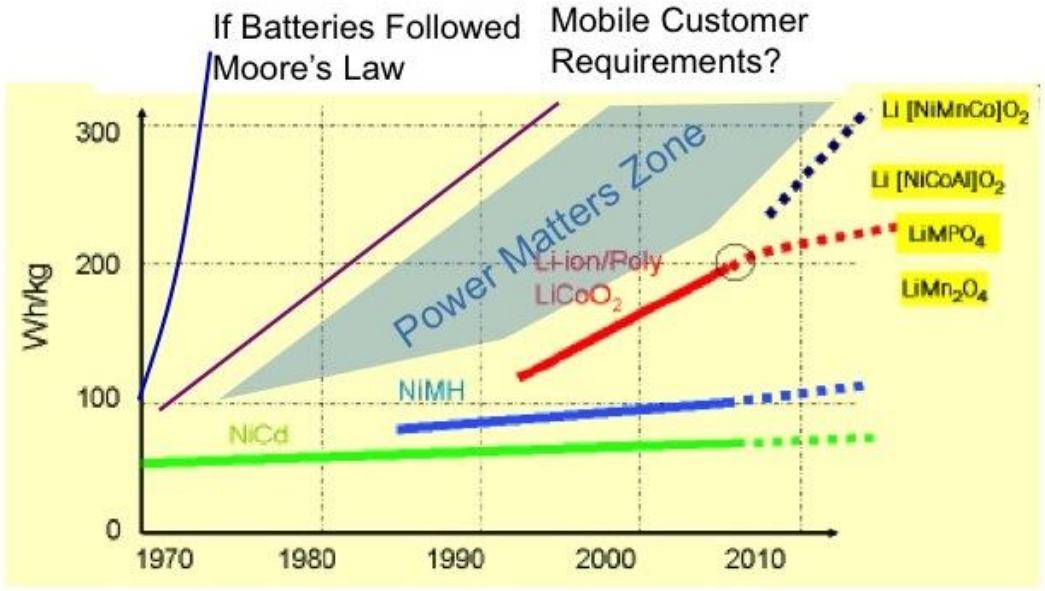
Embedded Hardware Platforms

Jetson AGX Orin 64GB
275 TOPS/s @ 15-60W



Embedded Systems Nightmare – Power Availability

The ubiquitous computing dream of *embedded systems everywhere* is accompanied by the nightmare of *battery replacement and disposal*



Source: Avicenne

Battery Technology is Stuck!

No Moore's Law in batteries:
2-3%/year growth

Embedded systems' lifetime such as wearables depends on battery life!

Solution

Design ultra low power smart systems and harvest energy from environment (heat, light, radio, or vibrations…)

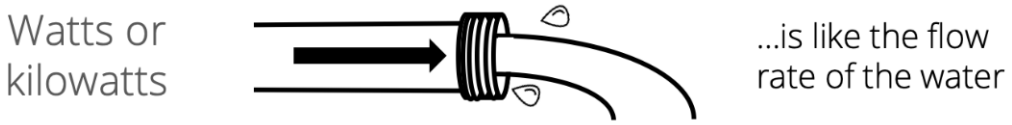
Power and Energy

Power and Energy

Electric power refers to the rate at which electrical energy is consumed or dissipated by an electrical device. It is given by the following expressions:

$$\boldsymbol{P} = I \times U = \frac{U^2}{R} = I^2 \times R$$

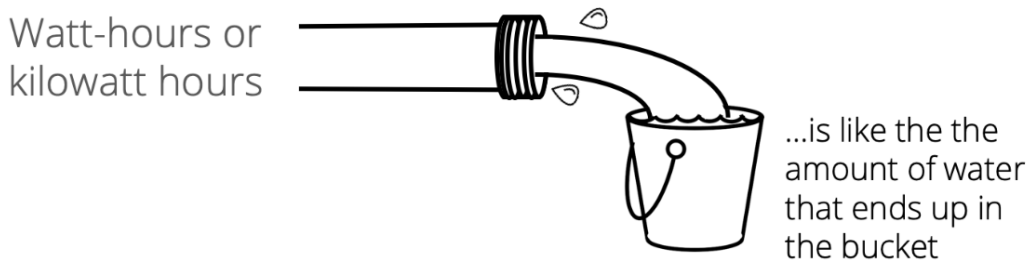
Unit: Watt (W) or Joule/second (J/s)



The work done or energy supplied by the source in maintaining the flow of electric current is called *electrical energy* and is given by:

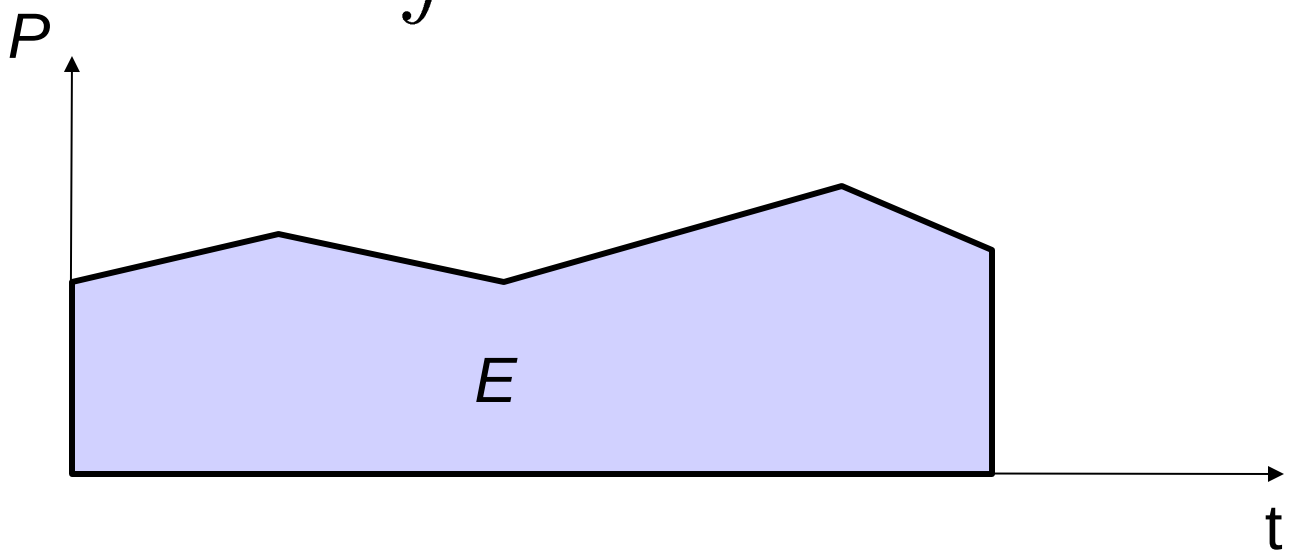
$$\boldsymbol{E} = \int P(t)\, dt$$

Unit: Watt hour (Wh) or Joule (J)



Power and Energy

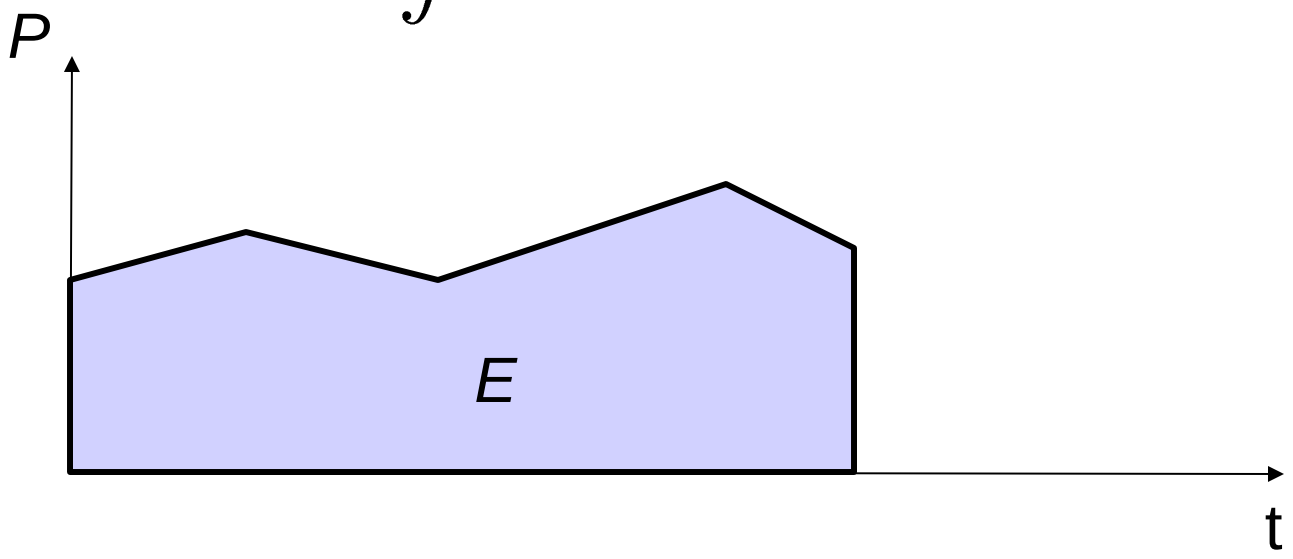
$$E = \int P(t)dt$$



Example energy consumption

Power and Energy

$$E = \int P(t)dt$$

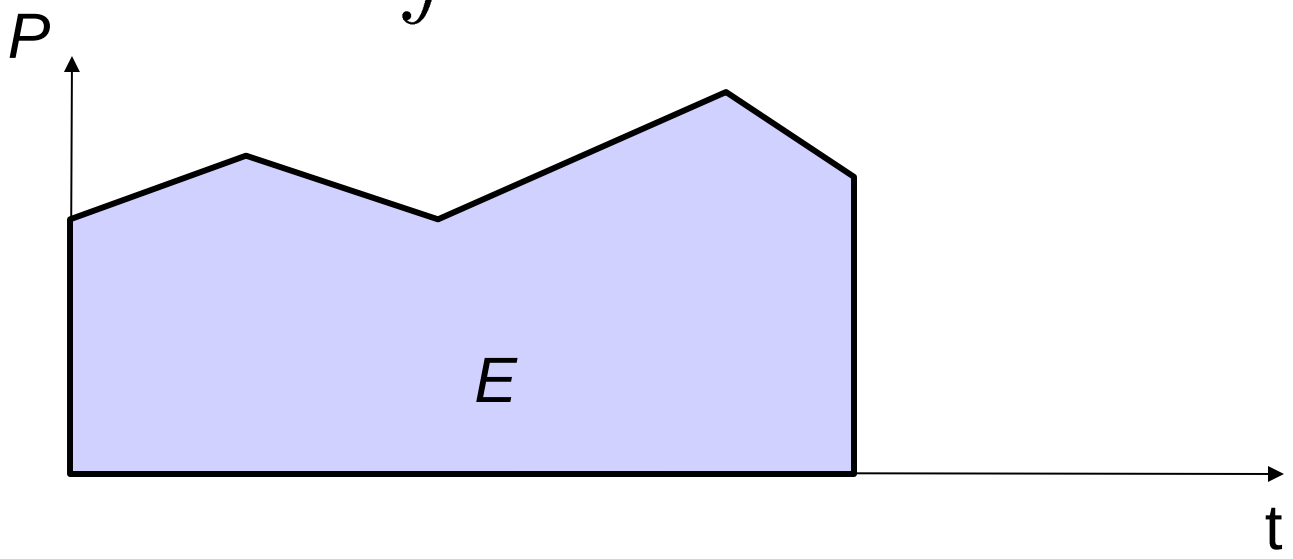


Example energy consumption

- Fast execution with same power profile means less energy

Power and Energy

$$E = \int P(t)dt$$



Example energy consumption

- Fast execution with same power profile means less energy
- Often, power must be increased to increase speed. This then also increases the energy consumption

Low Power vs. Low Energy

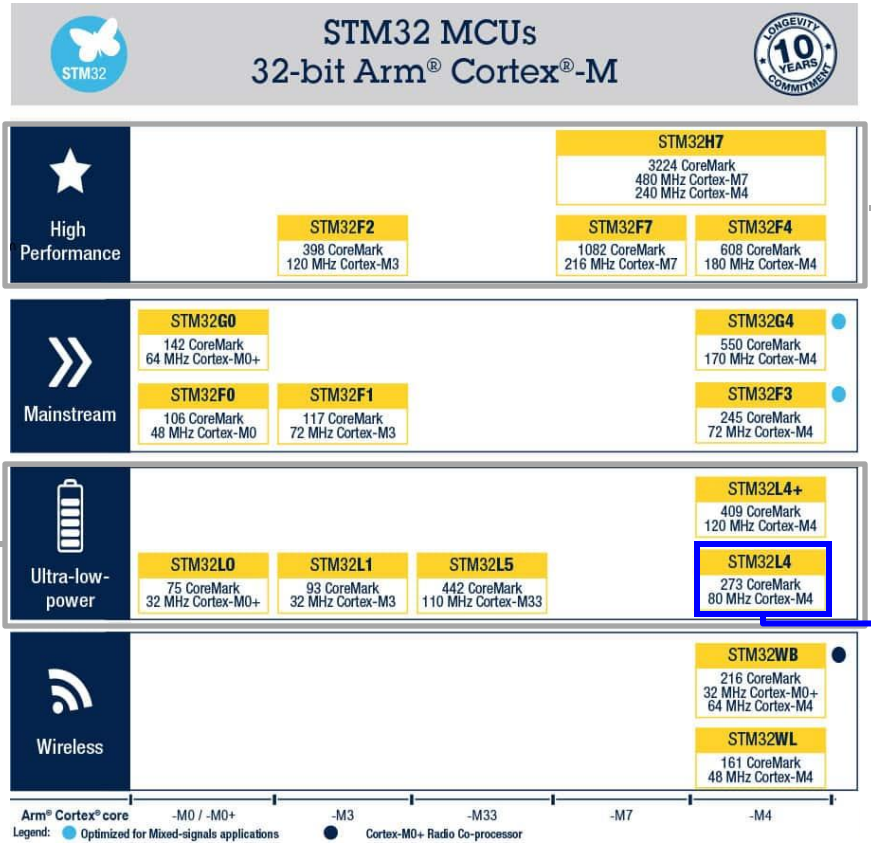
- Minimizing the *power consumption (voltage * current)* is important for:
  - The design of the power supply and voltage regulators
  - The dimensioning of interconnect between power supply and components
  - Cooling (short term cooling)
    - high cost and limited space

- Minimizing the *energy consumption* is important due to:
  - Restricted availability of energy (mobile systems)
  - Limited battery capacities (only slowly improving)
  - Very high costs of energy (energy harvesting, solar panels)
  - Reduce maintenance, omit battery replacement (expensive)
  - Achieve long lifetimes

Low- and High-Power Embedded Systems

Low Power Embedded Systems

- Battery operated devices



High Power Embedded Systems

- Flight Controller

MCU used in Exercises

ASICS

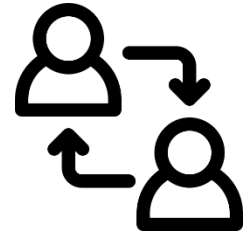
FPGA

General Purpose Processor

Flexibility

Power Efficiency

Clicker: Low Power Modes



What is the primary *benefit of utilizing low power modes* in embedded systems?

- ❑ Increased heat dissipation
- ❑ Extended battery life and reduced power consumption
- ❑ Faster execution of code
- ❑ Enhanced sensor performance

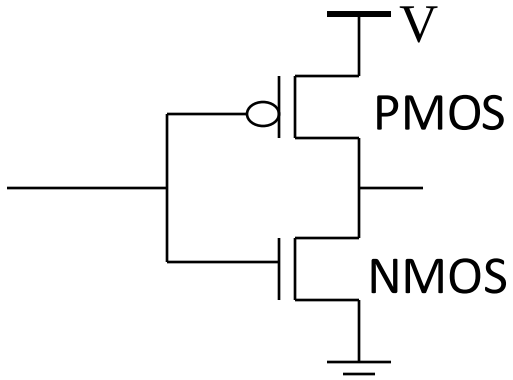
Low Power Design Techniques

Low Power Design

Power consumption can be reduced on different system levels:

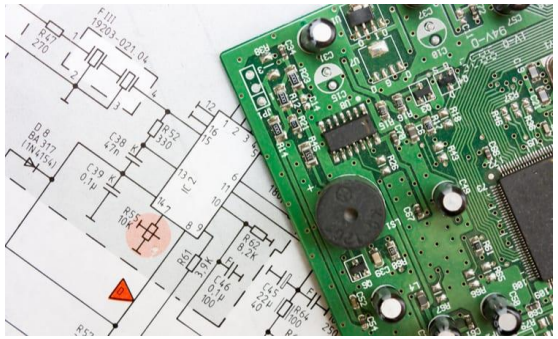
CMOS Level:

Reduce power dissipation



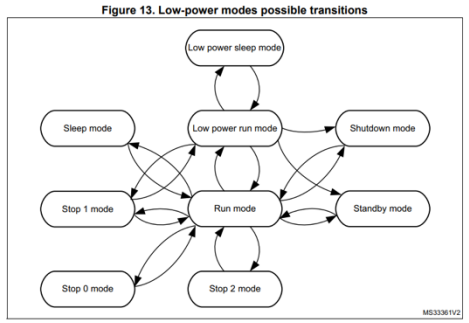
System Level:

low-power hardware selection



Software Level:

power-aware firmware



Low Power Design – CMOS Level

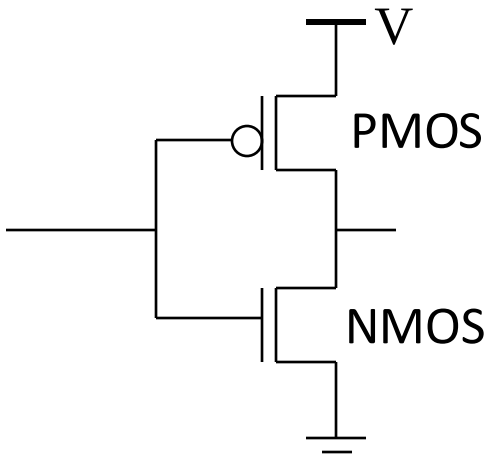
Power Dissipation

Power dissipation: The process in which computer processors consume electrical energy that is dissipated in from of heat due to the resistance in the electronic circuits.

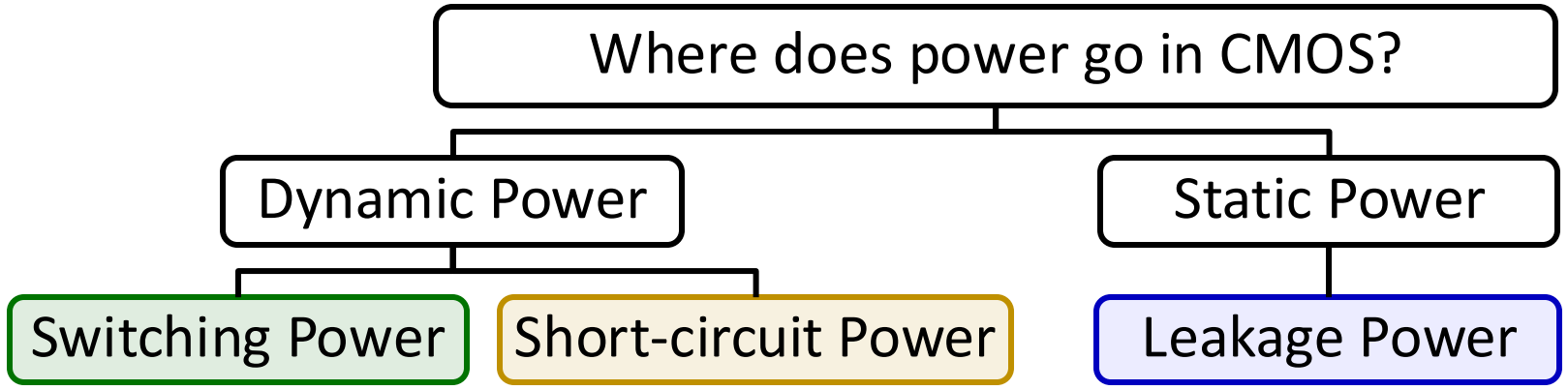
- For high performance systems, power dissipation is one of the most important aspects to minimize.
- High power dissipation comes with:
  - Lower reliability/durability due to overheating
  - Needs cooling, which can increase size and cost of system
  - Need for larger batteries, reduces the battery-lifetime
  - Increased maintenance

Power Dissipation – CMOS

The Smallest building block of a digital processor is the *CMOS (Complementary metal-oxide semiconductor)* circuit



CMOS circuit

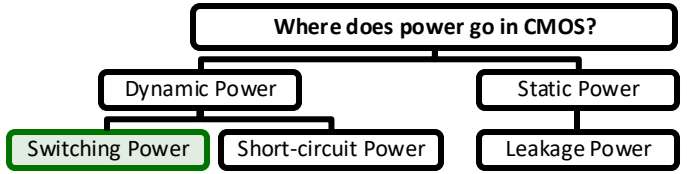


$$P = ACV^2 f + \tau AVI_{short} f + VI_{leak}$$

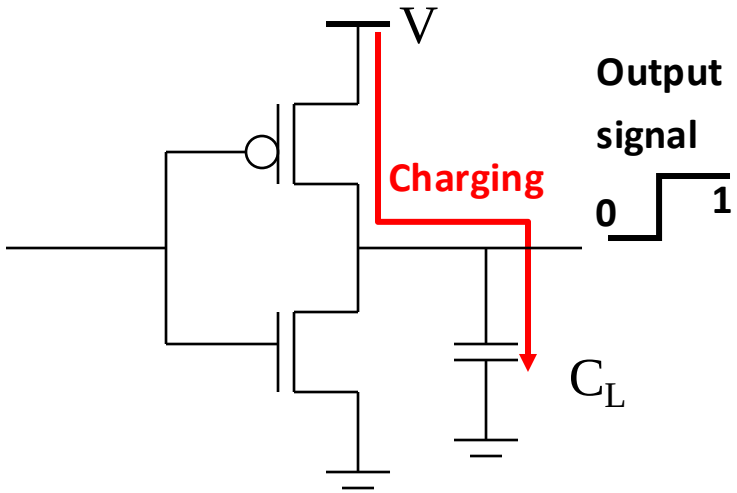
[1]

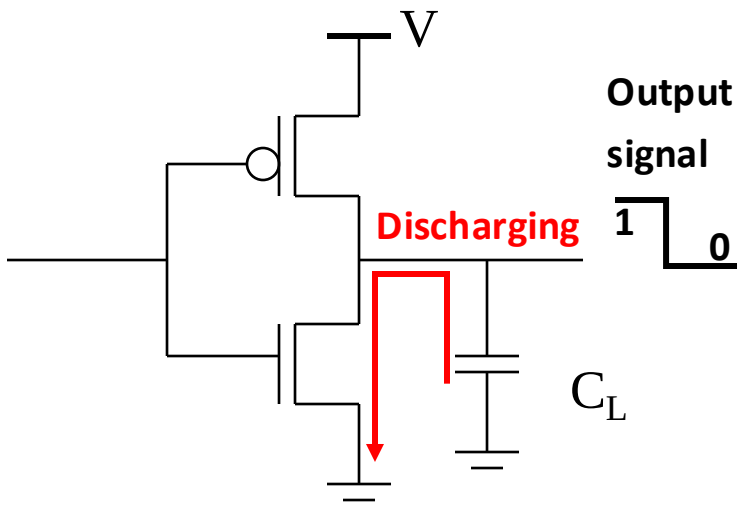
[1] Ng, Len Luet, et al. "Power Consumption in CMOS Circuits. *Electromagnetic Field in Advancing Science and Technology*. IntechOpen, 2022.

Switching Power Consumption



The power consumed when a transistor is switching from one logic state to another to charge/discharge a load capacitance C_L is called *switching power consumption.*

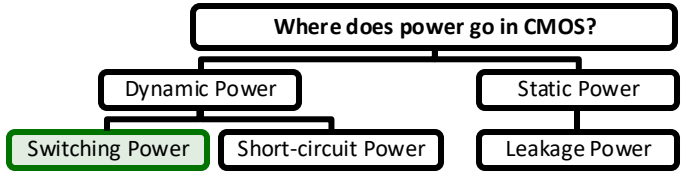




- Energy dissipated during one switching cycle (0 → 1 → 0):
  - $\frac{1}{2} * C_L * V^2 + \frac{1}{2} * C_L * V^2 = C_L * V^2$
- Power consumption is dependent on the switching frequency f_{switch}:
  - $f_{switch} * C_L * V^2$

[1] Ng, Len Luet, et al. "Power Consumption in CMOS Circuits. *Electromagnetic Field in Advancing Science and Technology*. IntechOpen, 2022.

Switching Power – Activity Factor

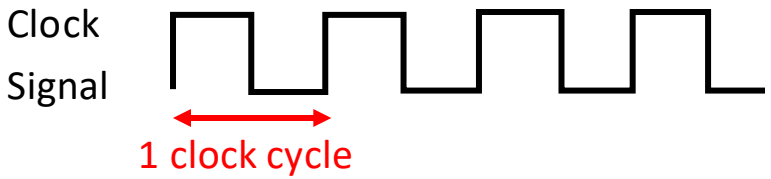


The switching power consumption depends in the systems *switching frequency* which again depends in the application and usage (activity).

$$f_{switch} = A * f_{clk}$$

A represents the *activity factor*; the probability of the circuit node switching in one clock cycle.

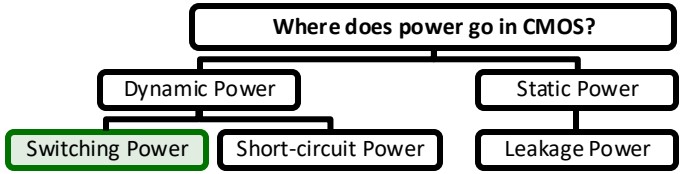
$$P = A * C_L * V^2 * f_{clk}$$



$$A = 1$$
$$f_{switch} = 1 f_{clk}$$

$$A = 0.5$$
$$f_{switch} = 0.5 f_{clk}$$

Switching Power Summary



C – Total capacitance seen by the gate's outputs:
Function of wire lengths, transistor sizes, …

V – Supply voltage
Trend: has been dropping

$$P_{switch} = A * C * V^2 * f$$

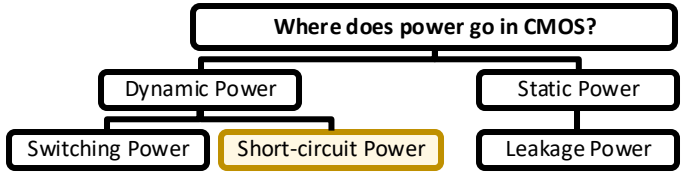
A - Activity of gate; probability of the CMOS node switching in one clock cycle

f – clock frequency
Trend: increasing

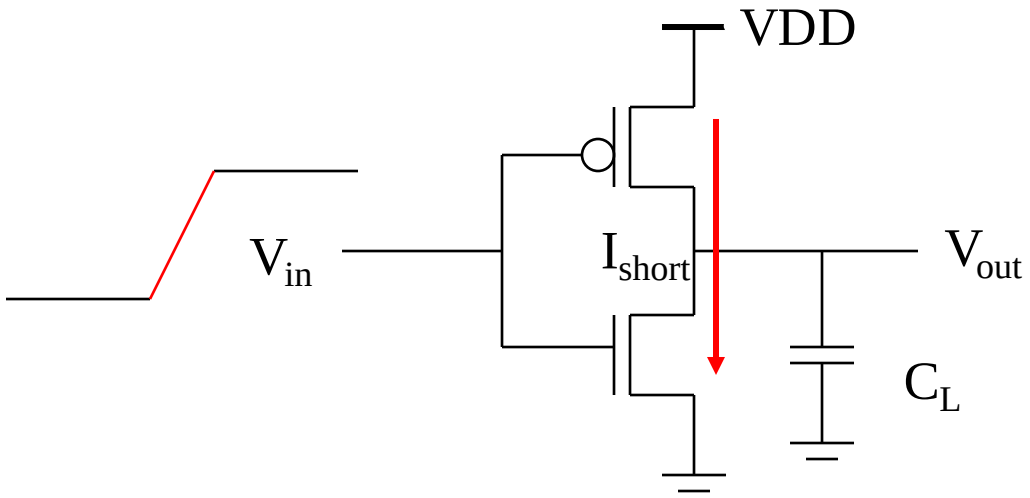
Reducing Strategy

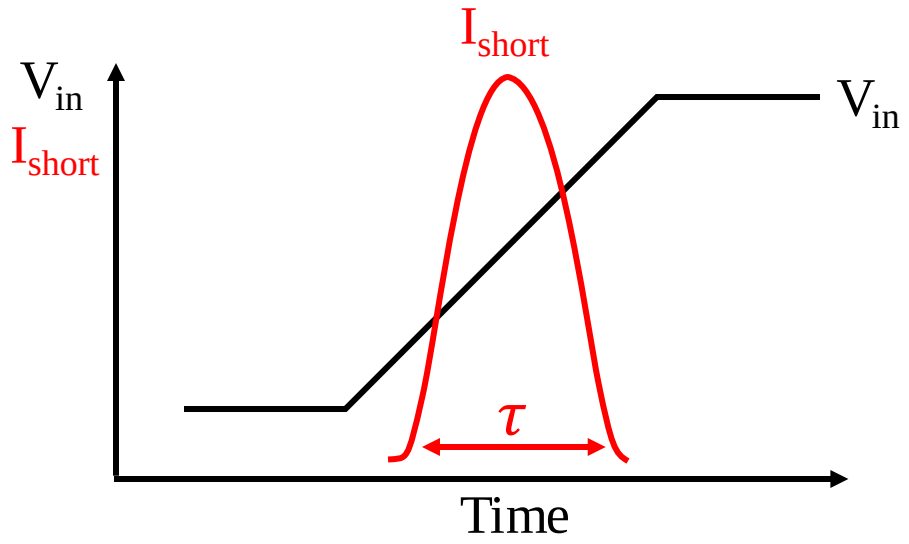
- Reducing V has quadratic effect; Limits?
- Lower C - shrink structures, shorten wires
- Reduce switching activity - Turn off unused parts or use design techniques to minimize number of transitions

Short-Circuit Power Consumption



During a state transition, both transistors are conducting for a short period of time. The very low impedance path from VDD to GND leads to a current spike (I_{short}).



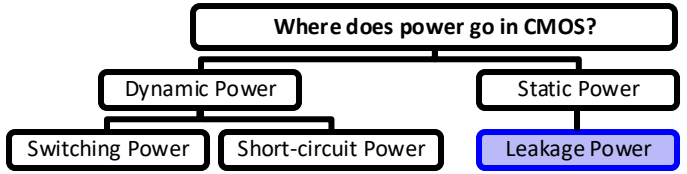


Reducing Strategy

$$P_{short} = \tau A V I_{short} f$$

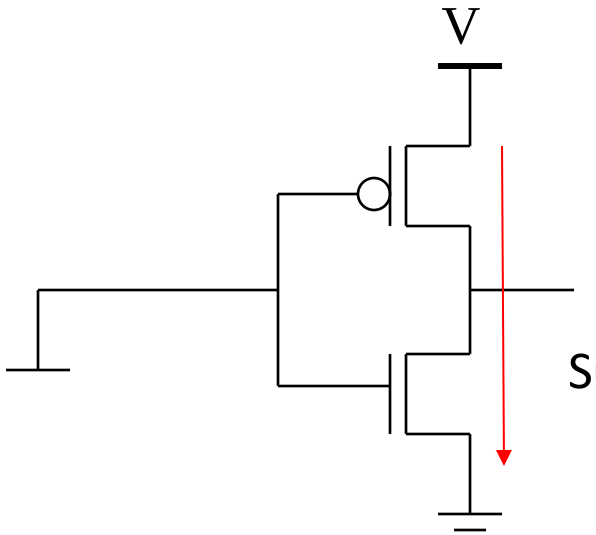
- Lower the supply voltage V_{in}
- Slope engineering – match the rise/fall time of the input and output signals

Leakage Power

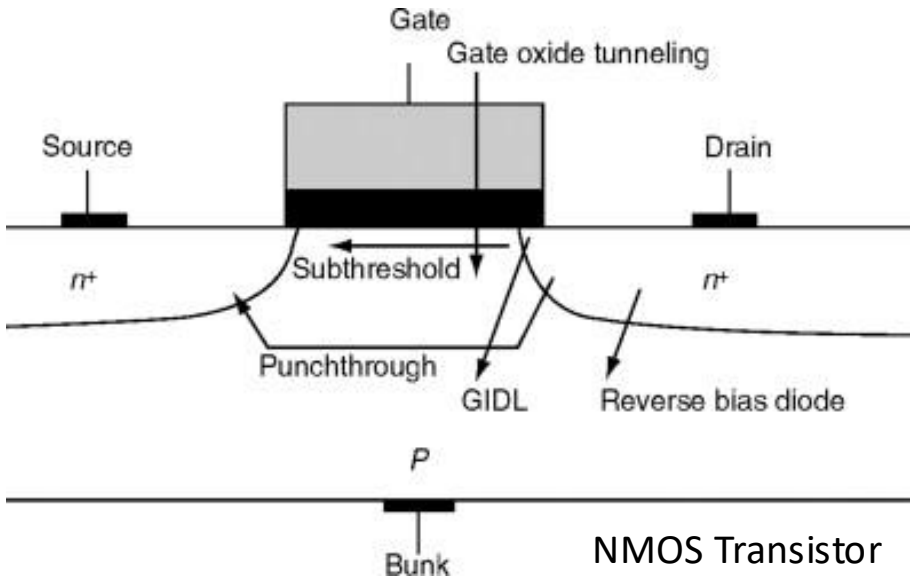


The *static power leakage* is caused by sub-threshold current flows. It grows exponentially with increasing temperature and decreases in V_{TH}.

$$P_{leak} = VI_{leak}$$

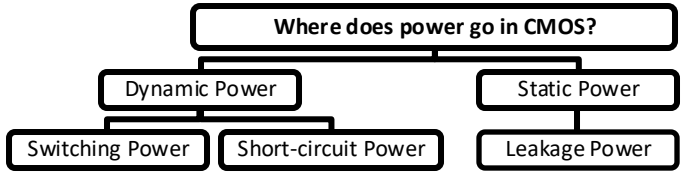


Sub-threshold current



NMOS Transistor

Impact of Clock Frequency on Power



- Increasing the clock frequency directly raises both the *Switching Power* and *Short-circuit Power*
- Raising the maximum clock frequency requires a voltage increase, which significantly impacts the *Switching Power* (quadratically), the *Short-circuit Power* and the *Leakage Power*

$$P = ACV^2 f + \tau AVI_{short} f + VI_{leak}$$

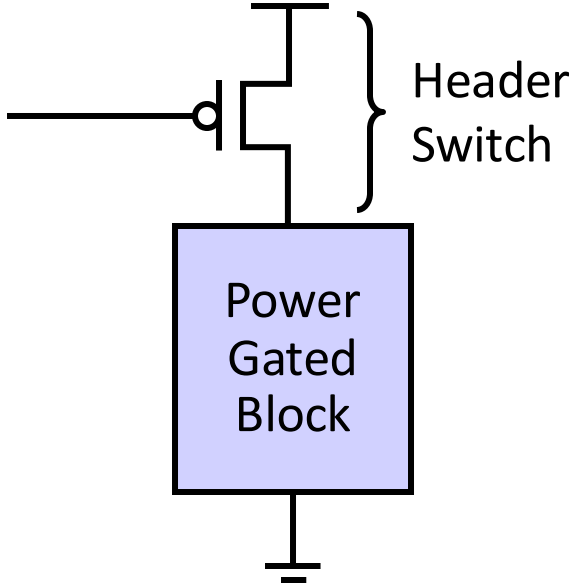
Switching Power | Short-circuit Power | Leakage Power

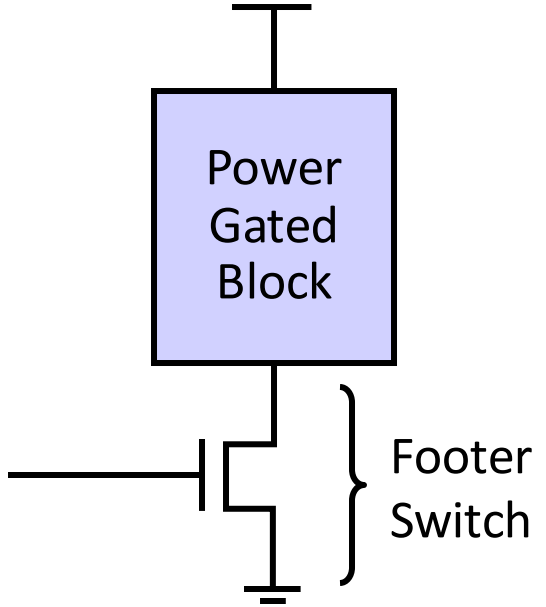
Low Power Design – System Level

Power Gating

Power gating is one of the most effective ways to minimize static power consumption at the system level.

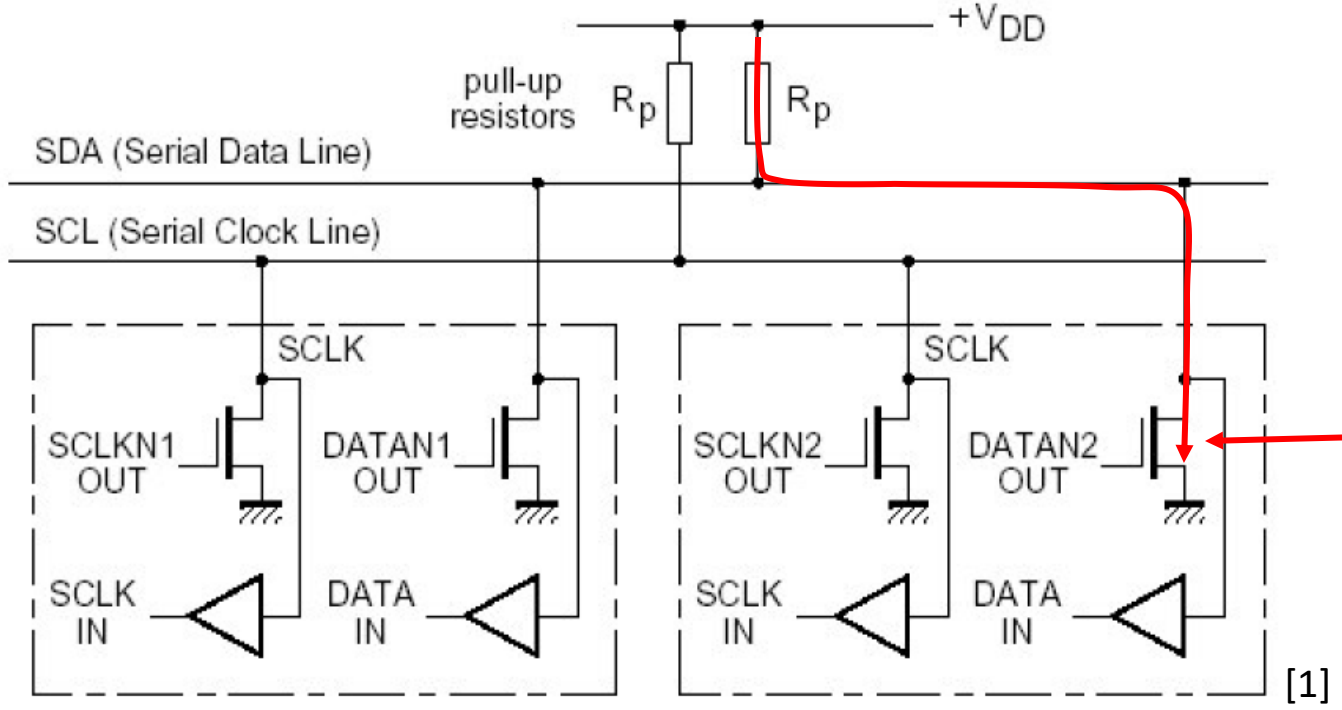
- Use power gating switches to cut-off the power supply of unused sub-systems
- Sensors and ICs often have enable (EN) functionalities. Disable components if they are not in use





Large Pull-ups for Open-Drain Topologies

Large pull-up resistors *reduce the current draw* during the active phases of I2C (when SDA or/and SCL are pulled low). However, large pull-ups also reduce the bus speed.



[1]

During zero states, a constant current flows through Rp to ground.

To represent digital zeros, the SCL/SDA lines are pulled down by the digital I2C front-end

[1] THE I2C-BUS SPECIFICATION VERSION 2.1

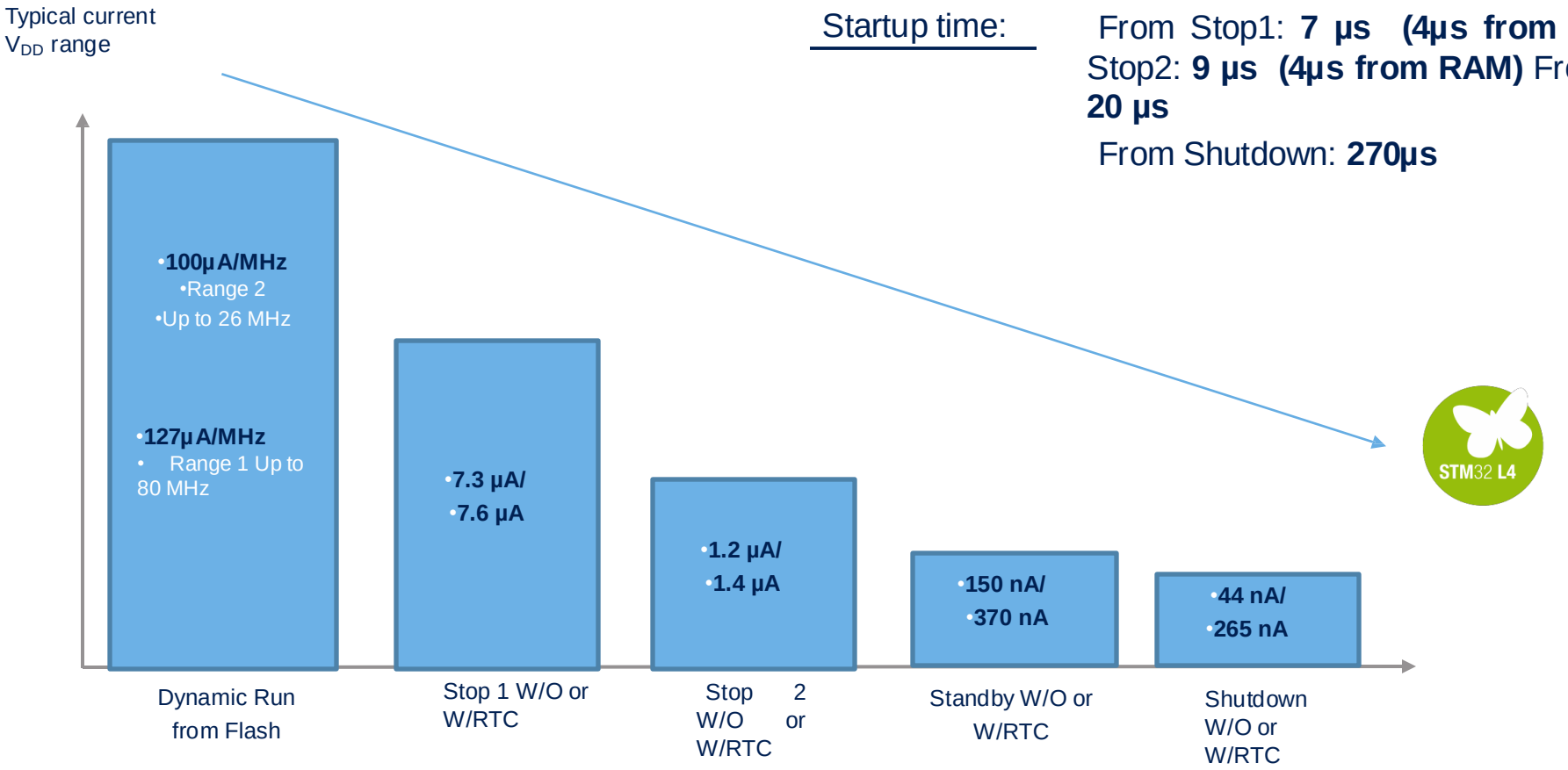
Component Selection

With an *appropriate component selection* at system level, the power consumption can be reduced.

- Sensors and integrated circuits (ICs) often enable pins, and deactivating these pins when not in use can further conserve energy
- In ultra-low-power designs, the self-discharge rate of batteries becomes a critical factor. Therefore, it is essential to choose the appropriate battery size and chemistry
- Capacitors do not provide perfect insulation. Small DC currents can leak through the dielectric, causing the capacitor to discharge continuously
- Use inductors with low series resistance to enhance the efficiency of DC-DC converters
- Reduce the number of components needed for a sophisticated design. Less components use less power

Low Power Design – Hardware-Software Level

Low Power modes on Embedded Processors: STM32L4



Startup time:

From Stop1: **7 µs (4µs from RAM)** From Stop2: **9 µs (4µs from RAM)** From Standby: **20 µs**

From Shutdown: **270µs**

Range 2 => Vcore = 1.0V

Range 1 => Vcore = 1.28V

Key features

- 7 low-power modes with fast wakeup
  - Down to 30 nA with I/O wake-up
  - Down to 350 nA with 32 KB RAM retained
  - Wake-up from high number of peripherals
- Down to 100 µA / MHz in Run mode
- Battery backup mode with RTC and backup registers
- Independent power supplies

•Application benefits

- High flexibility to lower power consumption depending on active peripherals, required performance and needed wakeup sources
- Increase battery life

Low power modes in moderns processors: STM32L4

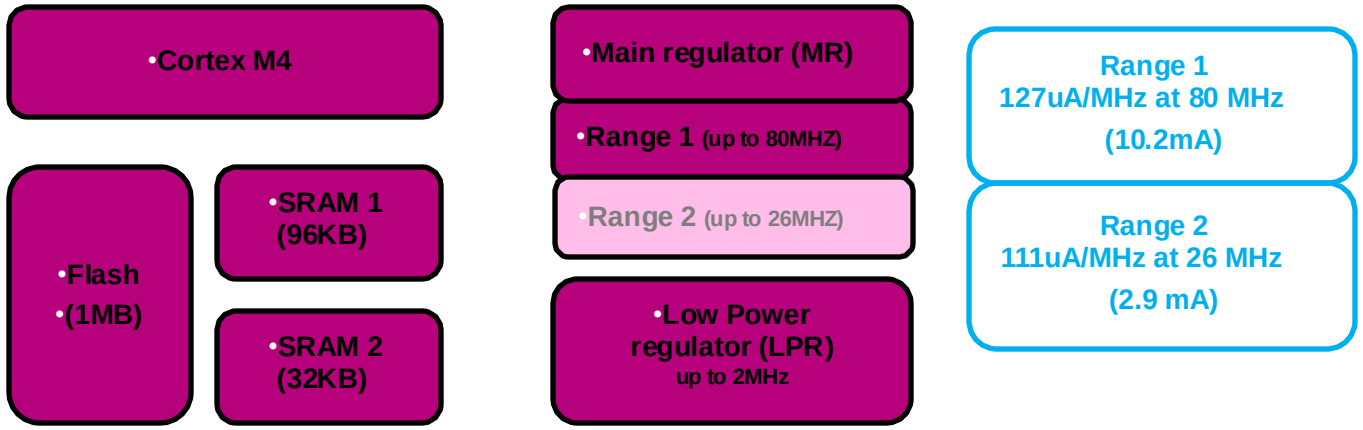
Typ @ VDD =1.8 V @ 25 °C

** : with RTC*

*** : from SRAM1*

STM32L4 Power Mode: Run mode

Run Mode Range 1
:Execution from Flash



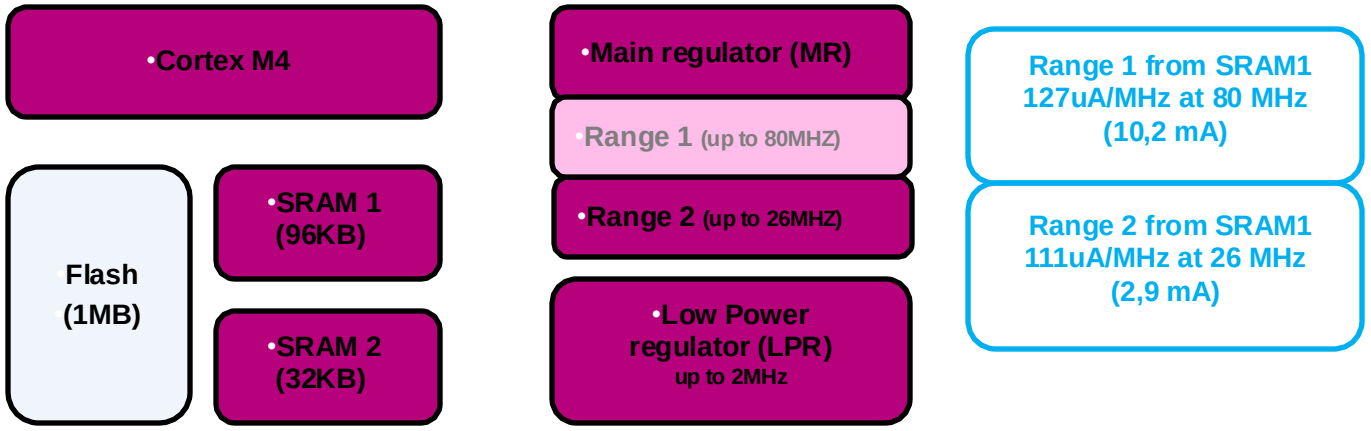
Available Clock

- HSI
- HSE
- LSI
- LSE
- MSI

- Active cell
- Frozen cell
- Cell in power-down
- Available Periph and clock

STM32L4 Power Mode:Run mode

Run Mode Range 2
: Execution from SRAM

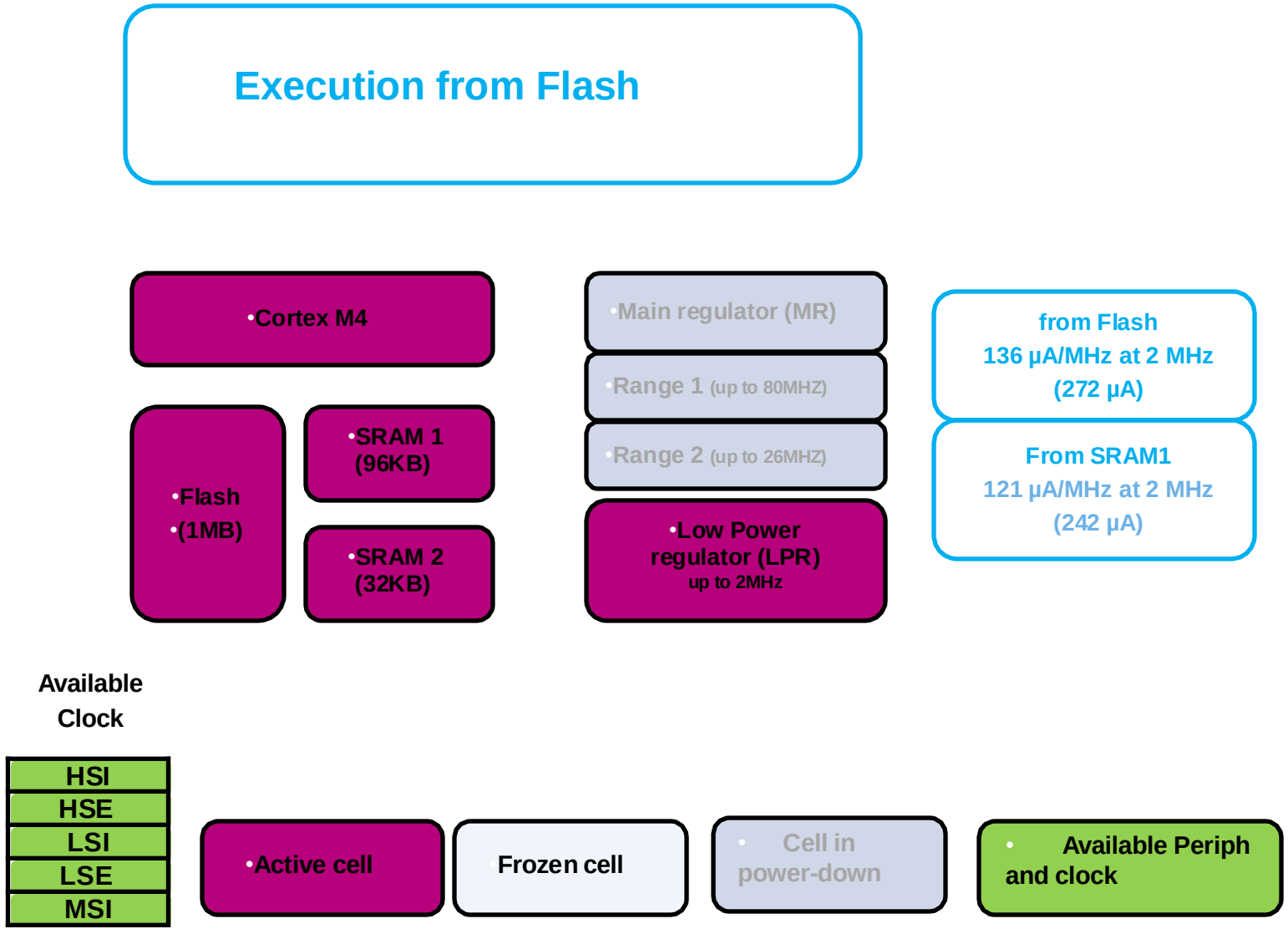


Available Clock

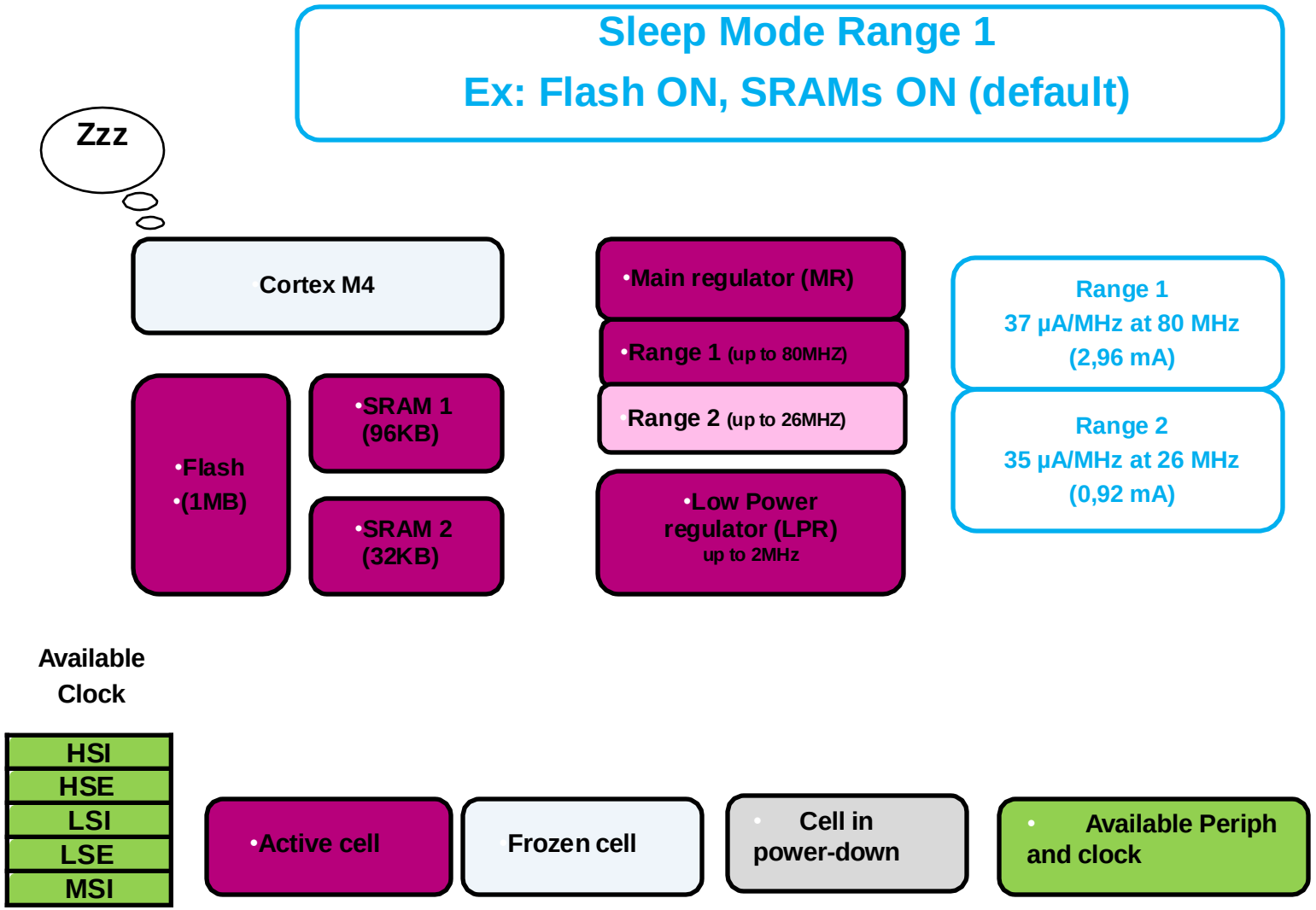
- HSI
- HSE
- LSI
- LSE
- MSI

- Active cell
- Frozen cell
- Cell in power-down
- Available Periph and clock

STM32L4 Power Mode: Low-power run mode



STM32L4 Power Mode: Sleep mode



Reduce Power – Dynamic Power Management

Table 4. STM32L476xx modes overview

Mode	Regulator (1)	CPU	Flash	SRAM	Clocks	DMA & Peripherals(2)	Wakeup source	Consumption(3)	Wakeup time
Run	MR range 1	Yes	ON(4)	ON	Any	All	N/A	112 µA/MHz	N/A
	SMPS range 2 High							40 µA/MHz(5)	
	MR range2					All except OTG_FS, RNG		100 µA/MHz	
	SMPS range 2 Low							39 µA/MHz(6)	
LPRun	LPR	Yes	ON(4)	ON	Any except PLL	All except OTG_FS, RNG	N/A	136 µA/MHz	to Range 1: 4 µs to Range 2: 64 µs
Sleep	MR range 1	No	ON(4)	ON(7)	Any	All	Any interrupt or event	37 µA/MHz	6 cycles
	SMPS range 2 High							13 µA/MHz(5)	
	MR range2					All except OTG_FS, RNG		35 µA/MHz	6 cycles
	SMPS range 2 Low							15 µA/MHz(6)	
LPSleep	LPR	No	ON(4)	ON(7)	Any except PLL	All except OTG_FS, RNG	Any interrupt or event	40 µA/MHz	6 cycles

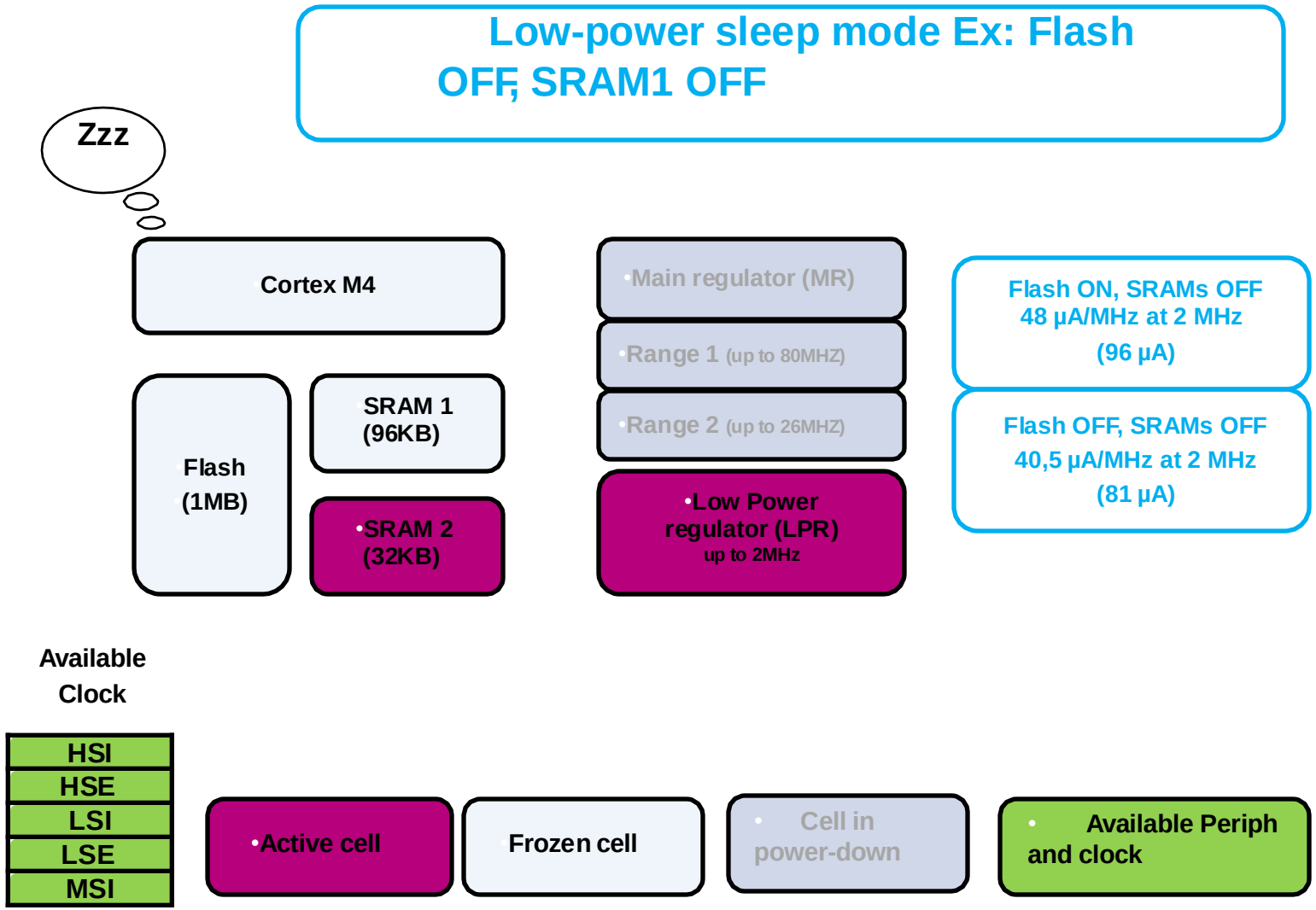
[1]

Sleep Mode:

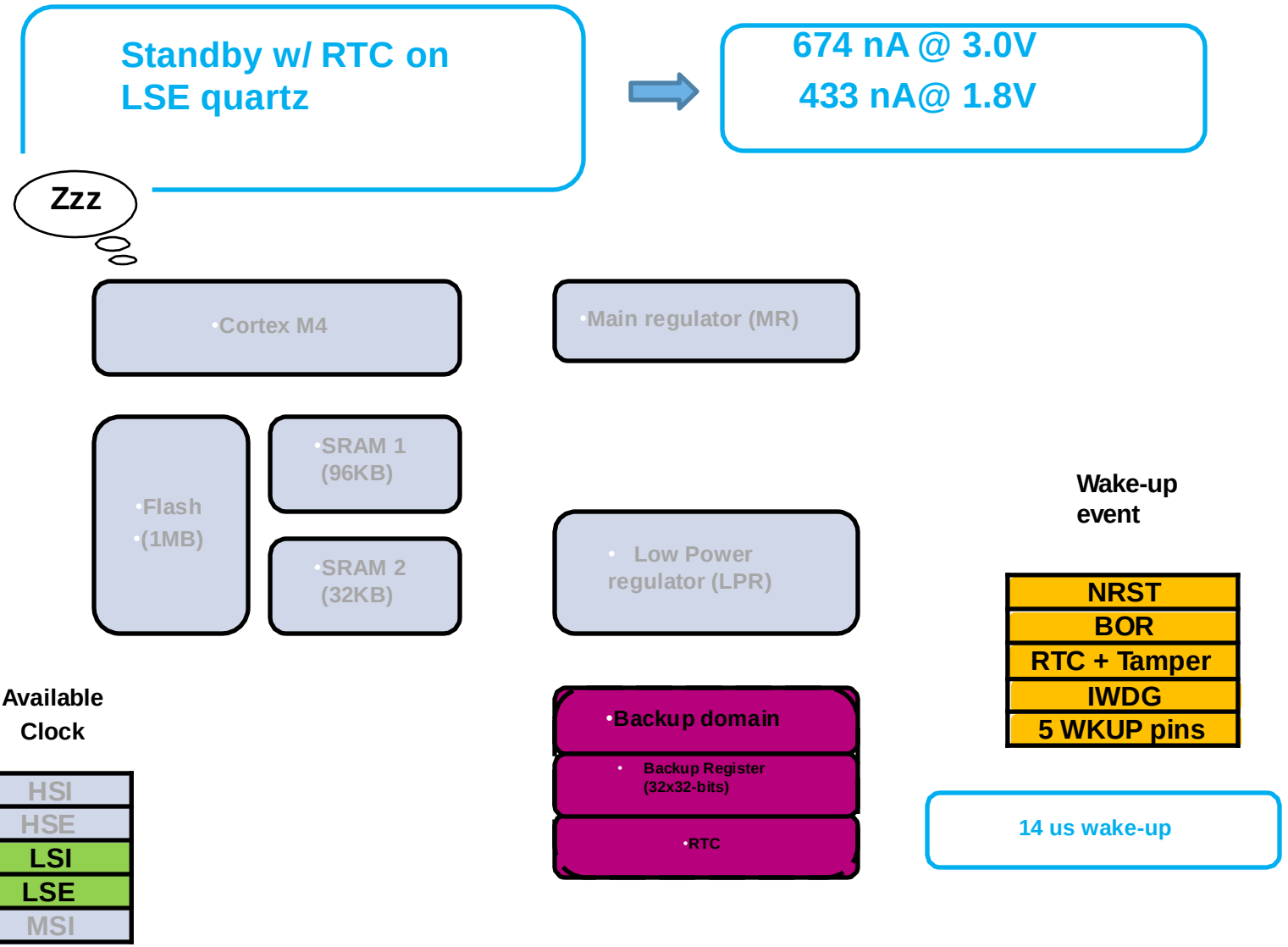
- only the CPU is stopped, all peripherals continue to operate and can wake up the CPU when an interrupt/event occurs

[1]Data Sheet STM32L47xxx

STM32L4 Power Mode: Low-power sleep mode



STM32L4 Power Mode: Standby Mode



STM32L4 Power Mode: Stop Mode

Stop 1 w/ RTC on LSE quartz → **7.9 µA @3.0V**
7.6 µA @1.8V

Cortex M4

Flash (1MB)

SRAM 1 (96KB)

SRAM 2 (32KB)

Main regulator (MR)

Low Power regulator (LPR)

Backup domain

Backup Register (32x32-bits)

RTC

Available Clock

HSI
HSE
LSI
LSE
MSI

I/Os kept, and configurable

Wake-up event

NRS
BOR
PVD
PVM
RTC + Tamper
LCD
USB OTG
USART
LP UART I2C 1 / I2C 2
I2C 3
SWPMI
COMP
LPTIM 1
LPTIM 2
IWDG
GPIOs

6us wake-up from Flash 4us wake-up from RAM

Available Peripheral

GPIO
DMA
FSMC
QSPI
BOR
PVD, PVM
LCD
USB OTG
USART
LP UART
I2C 1 / I2C 2
I2C 3
SPI
CAN
SDMMC
SWPMI
SAI
DFSDM
ADC
DAC
OPAMP
COMP
Temp
Sensor
Timers
LPTIM 1
LPTIM 2
IWDG
WWDG
Systick
Timer Touch
Sens RNG
AES

Reduce Power – Dynamic Power Management

Table 4. STM32L476xx modes overview (continued)

Mode	Regulator (1)	CPU	Flash	SRAM	Clocks	DMA & Peripherals(2)	Wakeup source	Consumption(3)	Wakeup time
Stop 0	Range 1(8) Range 2(8)	No	Off	ON	LSE LSI	BOR, PVD, PVM RTC, LCD, IWDG COMPx (x=1,2) DAC1 OPAMPx (x=1,2) USARTx (x=1...5)(9) LPUART1(9) I2Cx (x=1...3)(10) LPTIMx (x=1,2) *** All other peripherals are frozen.	Reset pin, all I/Os BOR, PVD, PVM RTC, LCD, IWDG COMPx (x=1..2) USARTx (x=1...5)(9) LPUART1(9) I2Cx (x=1...3)(10) LPTIMx (x=1,2) OTG_FS(11) SWPMI1(12)	108 µA	0.7 µs in SRAM 4.5 µs in Flash
Stop 1	LPR	No	Off	ON	LSE LSI	BOR, PVD, PVM RTC, LCD, IWDG COMPx (x=1,2) DAC1 OPAMPx (x=1,2) USARTx (x=1...5)(9) LPUART1(9) I2Cx (x=1...3)(10) LPTIMx (x=1,2) *** All other peripherals are frozen.	Reset pin, all I/Os BOR, PVD, PVM RTC, LCD, IWDG COMPx (x=1..2) USARTx (x=1...5)(9) LPUART1(9) I2Cx (x=1...3)(10) LPTIMx (x=1,2) OTG_FS(11) SWPMI1(12)	6.6 µA w/o RTC 6.9 µA w RTC	4 µs in SRAM 6 µs in Flash

Stop Modes:

- Achieves the lowest power consumption while retaining the content of SRAM and registers

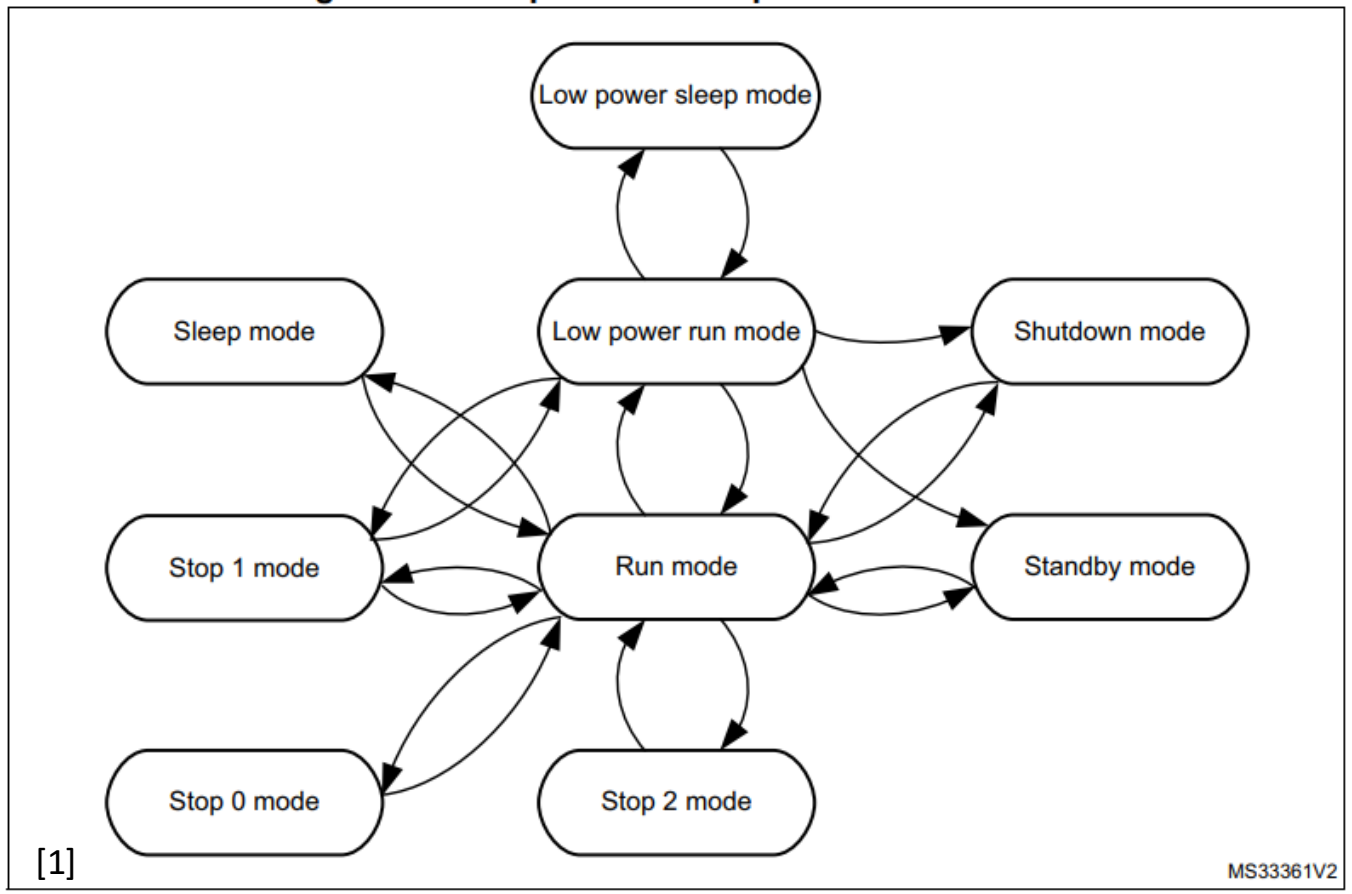
[1]

[1]Reference Manual STM32L47xxx

Reduce Power – Dynamic Power Management

Dynamic power management (DPM) tries to assign optimal power saving states during program execution. DPM requires hardware and software support

Figure 13. Low-power modes possible transitions



[1]

[1] Reference Manual STM32L47xxx

Reduce Power – Dynamic Power Management

Table 4. STM32L476xx modes overview (continued)

Mode	Regulator (1)	CPU	Flash	SRAM	Clocks	DMA & Peripherals(2)	Wakeup source	Consumption(3)	Wakeup time
Stop 2	LPR	No	Off	ON	LSE LSI	BOR, PVD, PVM RTC, LCD, IWDG COMPx (x=1..2) I2C3(10) LPUART1(9) LPTIM1 *** All other peripherals are frozen.	Reset pin, all I/Os BOR, PVD, PVM RTC, LCD, IWDG COMPx (x=1..2) I2C3(10) LPUART1(9) LPTIM1	1.1 µA w/o RTC 1.4 µA w/RTC	5 µs in SRAM 7 µs in Flash
Standby	LPR	Powered Off	Off	SRAM2 ON	LSE LSI	BOR, RTC, IWDG *** All other peripherals are powered off. *** I/O configuration can be floating, pull-up or pull-down	Reset pin 5 I/Os (WKUPx)(13) BOR, RTC, IWDG	0.35 µA w/o RTC 0.65 µA w/ RTC	14 µs
	OFF			Powered Off				0.12 µA w/o RTC 0.42 µA w/ RTC	
Shutdown	OFF	Powered Off	Off	Powered Off	LSE	RTC *** All other peripherals are powered off. *** I/O configuration can be floating, pull-up or pull-down(14)	Reset pin 5 I/Os (WKUPx)(13) RTC	0.03 µA w/o RTC 0.33 µA w/ RTC	256 µs

Shutdown Mode:

- Lowest power consumption
- RTC can remain active
- SRAM and registers are lost

[1]

[1]Reference Manual STM32L47xxx

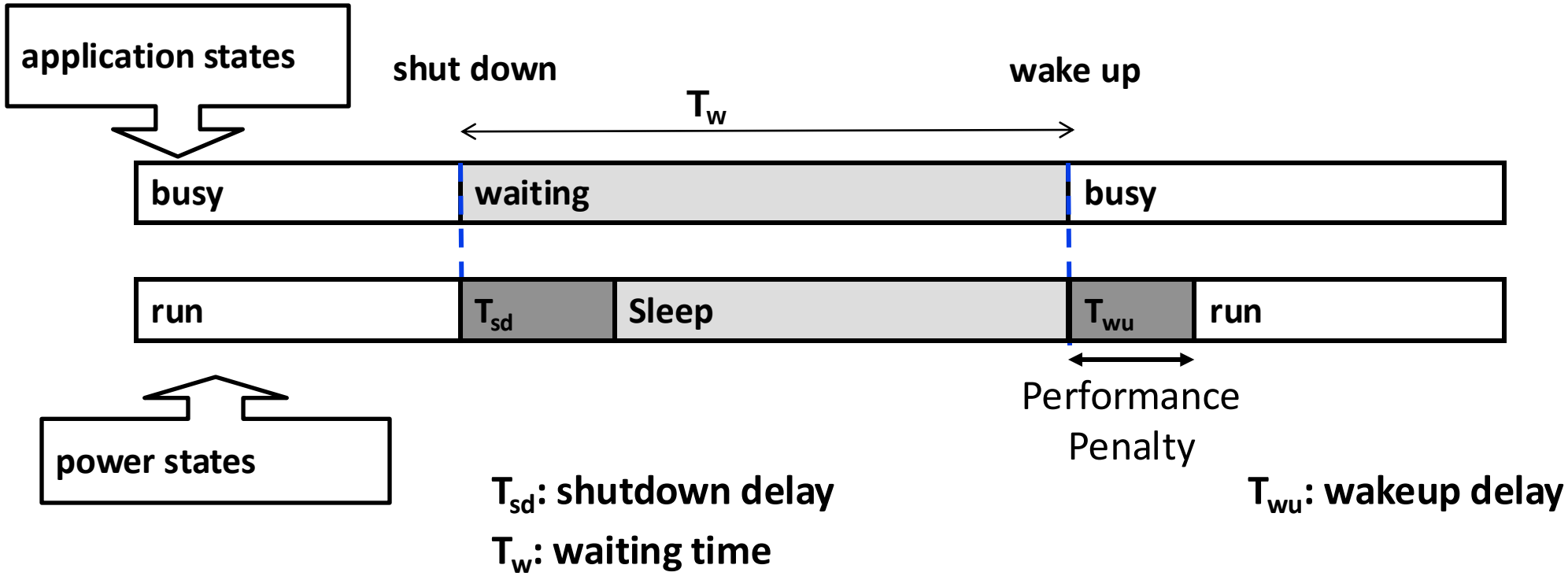
Reduce Power – Dynamic Power Management

Low power mode summary

Mode	Power Consumption	CPU State	Peripheral State	SRAM and Register Retention	Wake-Up Sources	Wake-Up Time
Sleep	Low	Stopped	Active	Retained	Interrupts from active peripherals	Very fast (ns range)
Stop	Lower	Stopped	Some Off	Retained	External interrupts, RTC alarm, etc.	Moderate (µs range)
Standby	Lowest	Off	Mostly Off	Not retained	Reset, RTC alarm, WKUP pins	Longest (µs range)

→ *For more details, refer to the microcontroller's **Reference Manual** and/or Low-Power Mode **Application Note***

Reduce Power – Dynamic Power Management

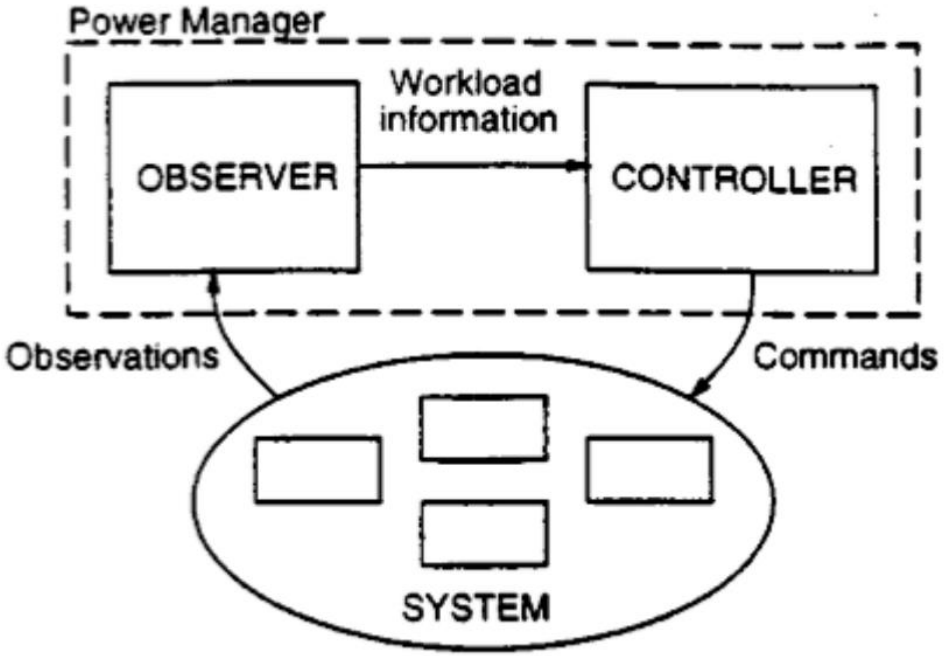


Desired: Shutdown only during long waiting times. This leads to a trade-off between energy saving and overhead.

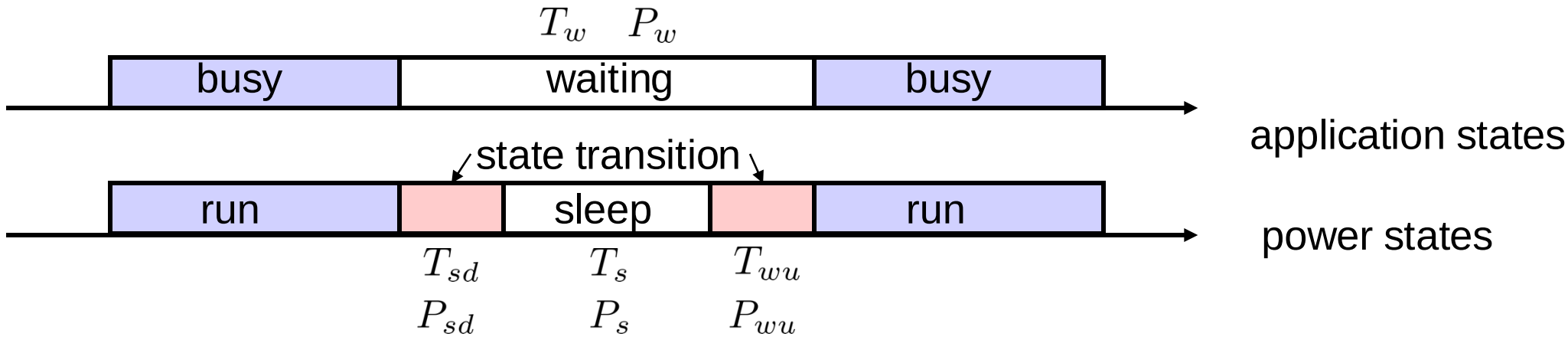
Reduce Power – Break-Even Time

The *Break-Even Time* is the minimum waiting time required to compensate the cost overhead of entering an inactive (sleep) state.

- Enter a low-power state is beneficial only if the waiting time is longer than the break-even time
- Assumptions for the calculation:
  - No performance penalty is tolerated.
  - An ideal power manager has the *full* knowledge of the future workload trace. On the previous slide, we supposed that the power manager has *no* knowledge about the future.



Reduce Power – Break-Even Time



Scenario 1 (no transition): $E_1 = T_w \cdot P_w$

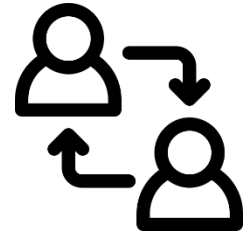
Scenario 2 (state transition): $E_2 = T_{sd} \cdot P_{sd} + T_{wu} \cdot P_{wu} + \overbrace{(T_w - T_{sd} - T_{wu})}^{T_s} \cdot P_s$

Break-even time: Limit for T_w such that $E_2 \leq E_1$

Break-even constraint: $$T_w \geq \frac{T_{sd} \cdot (P_{sd} - P_s) + T_{wu} \cdot (P_{wu} - P_s)}{P_w - P_s}$$

Time constraint $T_w \geq T_{sd} + T_{wu}$

Clicker: Low Power Modes



Which of the following statements best *defines the low-power mode* in a microcontroller?

- ❑ In low-power mode, the microcontroller operates at its maximum clock frequency
- ❑ In low-power mode, the microcontroller reduces the power consumption by turning of certain internal components
- ❑ In low-power mode, the power consumption is temporarily increased for faster data processing
- ❑ The low-power mode is only applicable to high-performance tasks

What Did You Learn?

- ✓ Power and Energy
- ✓ Power Dissipation
  - ✓ Switching, Short-Circuit and Leakage
- ✓ Power Gating
- ✓ Dynamic Power Management
  - ✓ Break-Even Time