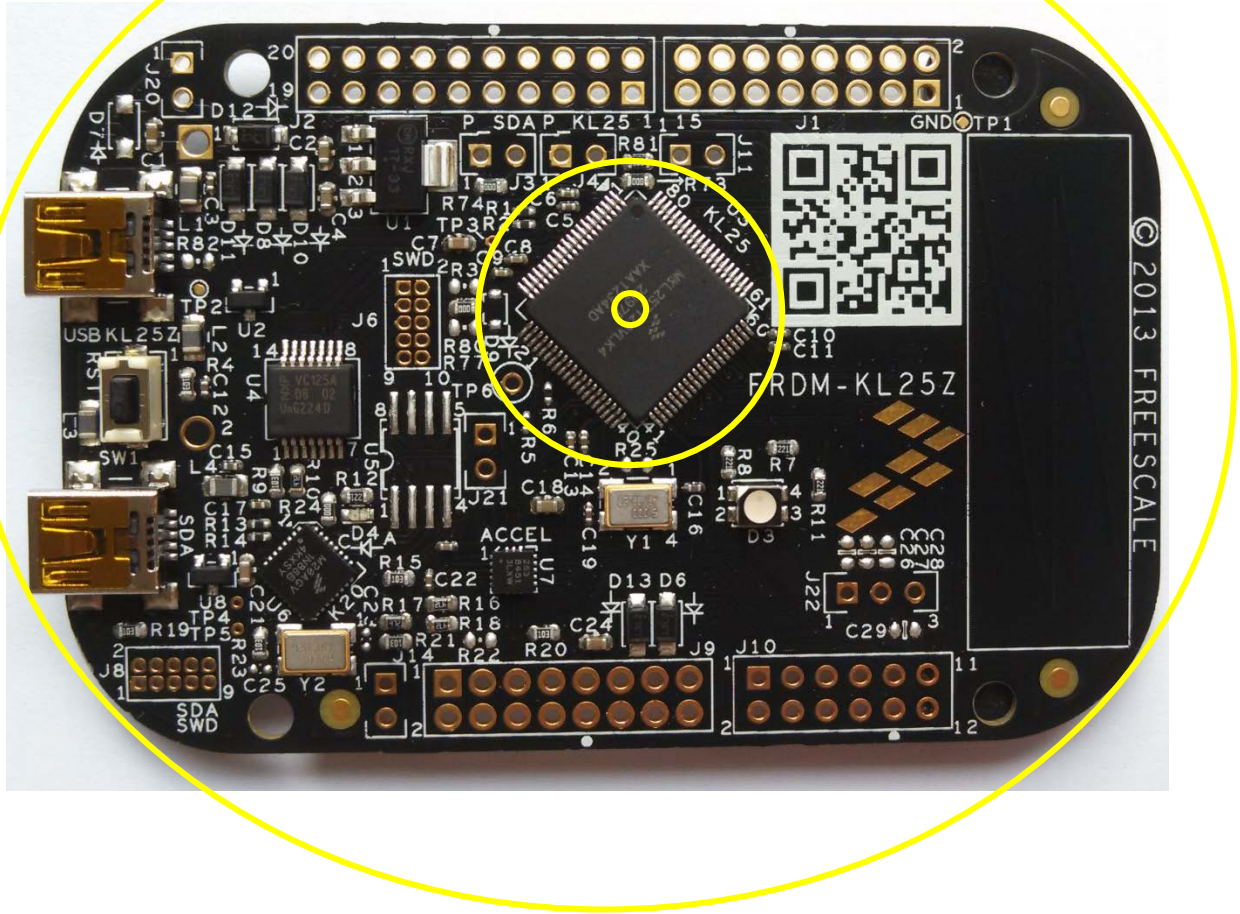
CPU, MCU and Board Features for Low Power and Energy

Overview

- M0+ CPU Core
- KL25Z MCU and Peripherals
- FRDM-KL25Z Development Board



ARM

Information Source – Reference Manual

- Chapter 7: Power Management
  - Describes different modes available
- Chapter 13: System Mode Controller
  - Switches system between different power modes
- Chapter 5: Clock Distribution
  - Clock sources
  - Controls 1.0 V band-gap reference (for ADC, CMP)
- Chapter 12: System Integration Module
  - Selection of clock sources for configurable peripherals
- Chapter 15: Low-Leakage Wakeup Unit
- Chapter 24: Multipurpose Clock Generator
- Chapter 33: Low-Power Timer
- Chapter 14: Power Management Controller
  - Detects low operating voltage
    - Low Voltage Warning interrupts
    - Power-On Reset

ARM

Application Notes

- AN4503
- KLQRUG

Freescale Semiconductor
Application Note

Document Number:AN4503
Rev. 1, 11/2012

Power Management for Kinetis and ColdFire+ MCUs

When and how to use low-power modes

by: Philip Drake

Contents

1 Introduction

Applications strive for high performance within constrained energy budgets, which continue to play a significant role in determining embedded designs. Increasing requirements do not allow for compromises on performance and continue to push for low energy budgets.

The Kinetis and ColdFire+ microcontroller families include internal power management features that can be used to control the microcontroller's power usage. This application note discusses how to use the power management systems, provides use case examples, and shows real-time current measurement results for these use cases.

Also included is a discussion of the differences between power management features on the various microcontrollers, along with drivers demonstrating the low-power features. Tips are given for using each of the power modes.

Power management methods discussed here do not include

Freescale Semiconductor
Users Guide

KLQRUG
Rev. 0, 09/2012

Kinetis L Peripheral Module Quick Reference

A Compilation of Demonstration Software for Kinetis L Series Modules

This collection of code examples, useful tips, and quick reference material has been created to help you speed the development of your applications. Most chapters in this document contain examples that can be modified to work with Kinetis MCU Family members. When you're developing your application, consult your device data sheet and reference manual for part-specific information, such as which features are supported on your device.

Sample code can be found at KL25_SC.exe, available from:
www.freescale.com/files/32bit/software/KL25_SC.exe

Information about the ARM core can be found in the help center at ARM.com

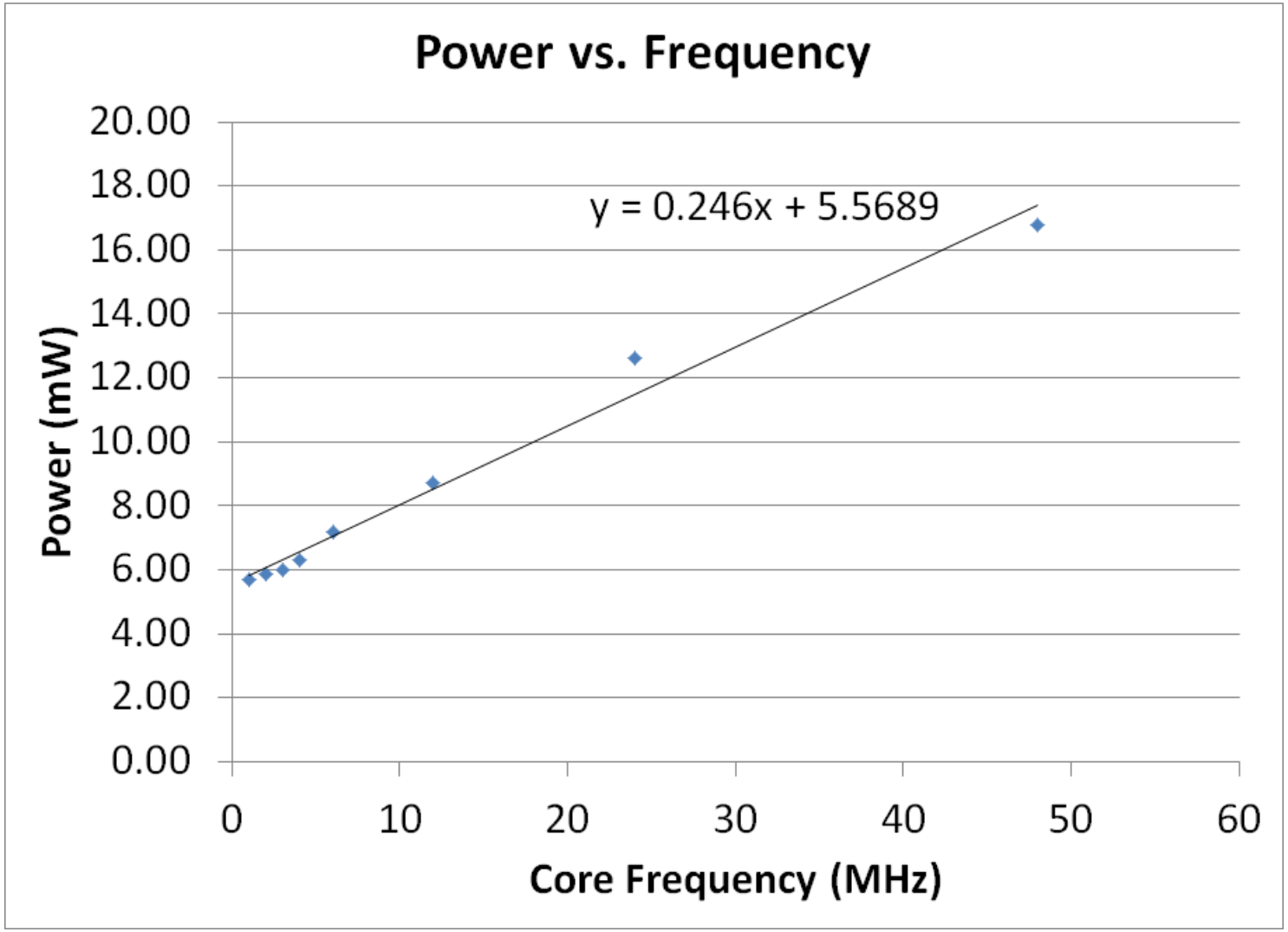
The most up-to-date revisions of our documents are on the Web. Your printed copy may be an earlier revision. To verify that you have the latest information available, refer to freescale.com

Cortex M0+ CPU Core

Setting the MCU Clock Frequency

KL25Z MCU Power at 3.0 V

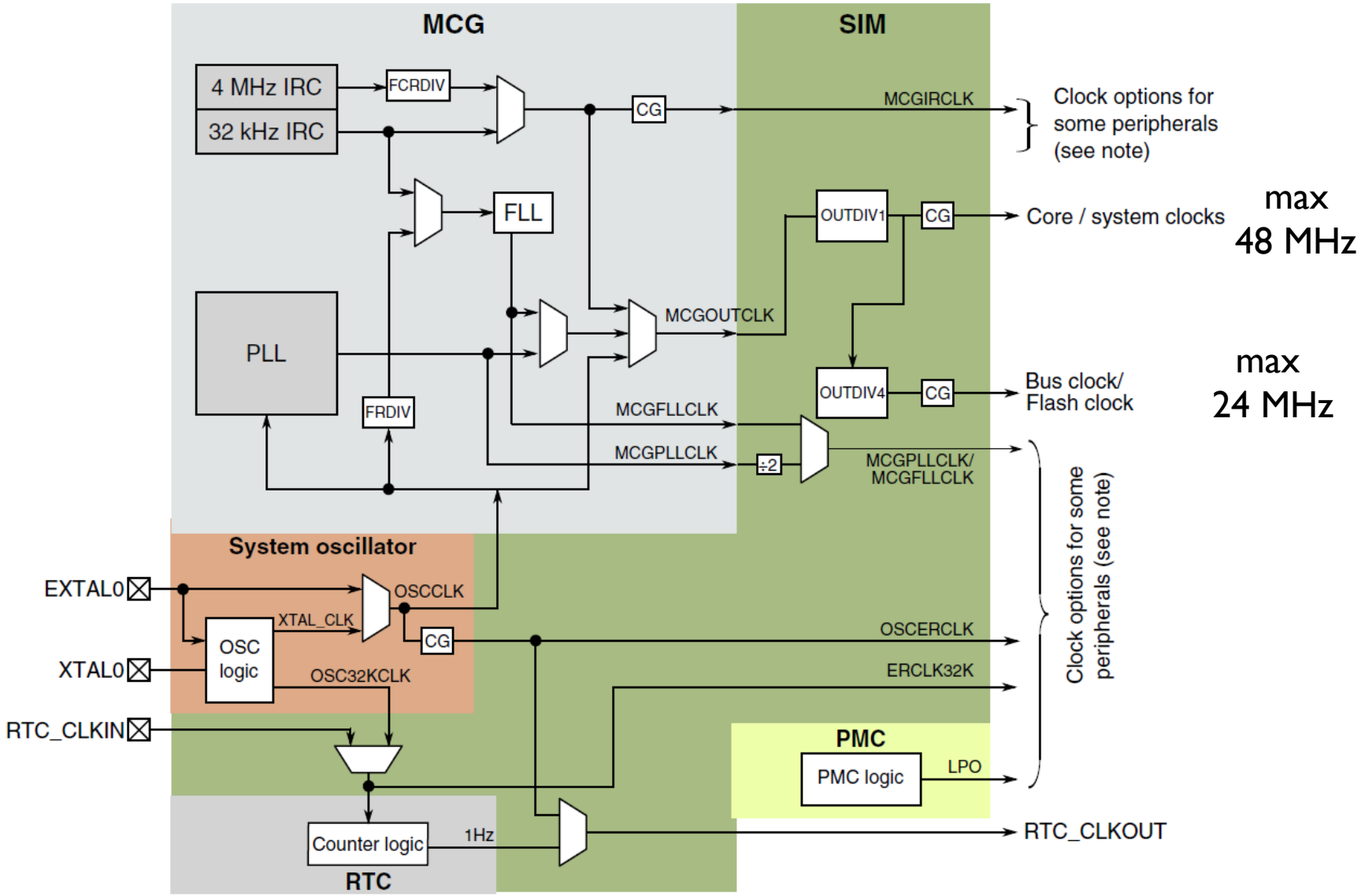
- Characteristics
  - Static power is 5.5689 mW
  - Dynamic power is roughly proportional to frequency: 0.246 mW/MHz
- Can reduce power consumption by reducing CPU core frequency



ARM

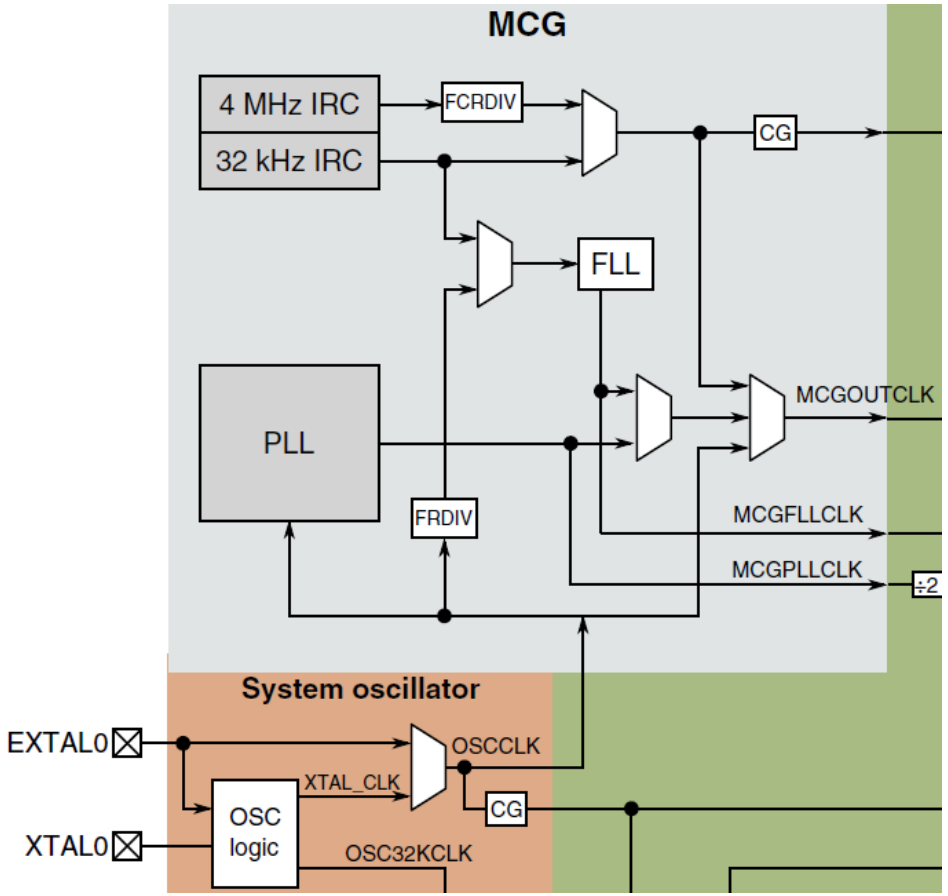
KL25 Clock System

- Clock system contains multiple subsystems
  - MCG – multi-purpose clock generator, internal to MCU
  - System oscillator – uses external crystal or clock signal
  - SIM – System Integration Module – Divides and routes clock signals
  - RTC – Real-Time Clock – Hardware counters to track time-of-day and date
  - PMC – Power Management Controller – provides clock for some low power peripherals



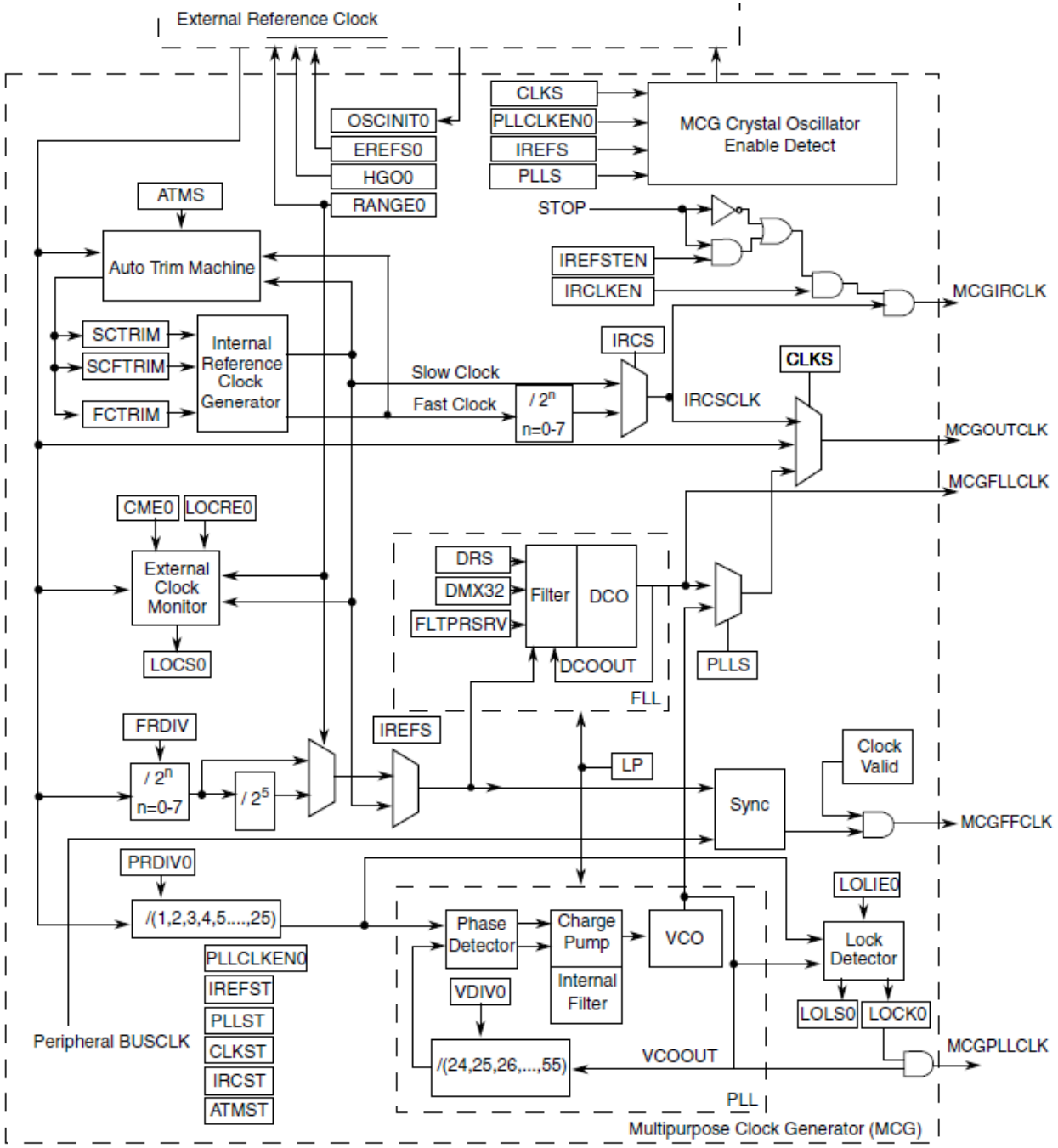
What Clock Sources are Available?

- Multipurpose Clock Generator (MCG)
- Internal Reference Clocks
  - Fast IRC: 4 MHz
  - Slow IRC: 32 kHz
- External Reference Clock
  - From system oscillator and external crystal
- Frequency-Locked Loop
  - Used as Digitally-Controlled Oscillator (DCO)
  - Can multiply input frequency by an integer
- Phase-Locked Loop



MCG Control Registers

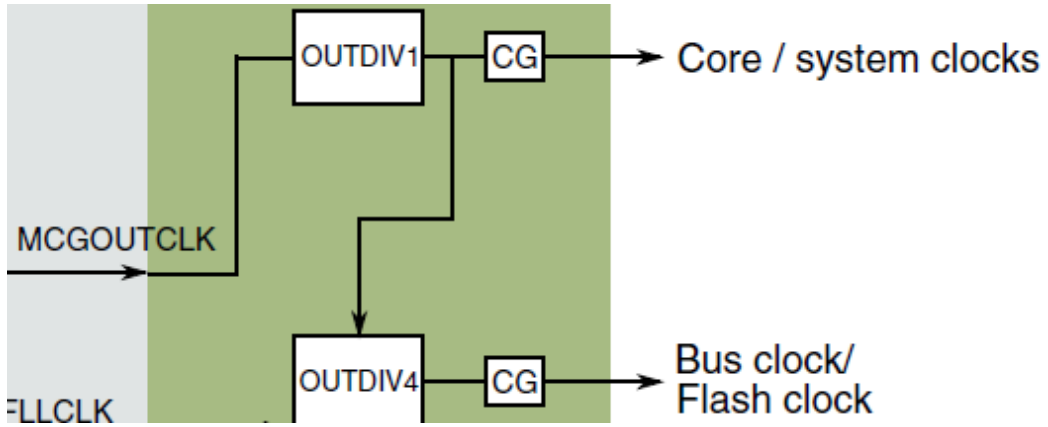
- Extensive configuration options available!
- See Chapter 24 of reference manual for details



ARM

How Do We Divide the Clock Speed?

- Dividers are located in SIM (CLKDIV1)
- Core and System clock
  - Up to 48 MHz
  - Input is MCGOUTCLK
  - Divided by OUTDIV1+1 (four bits)
- Bus and Flash clock
  - Up to 24 MHz
  - Input is core and system clock
  - Divided by OUTDIV4+1 (three bits)

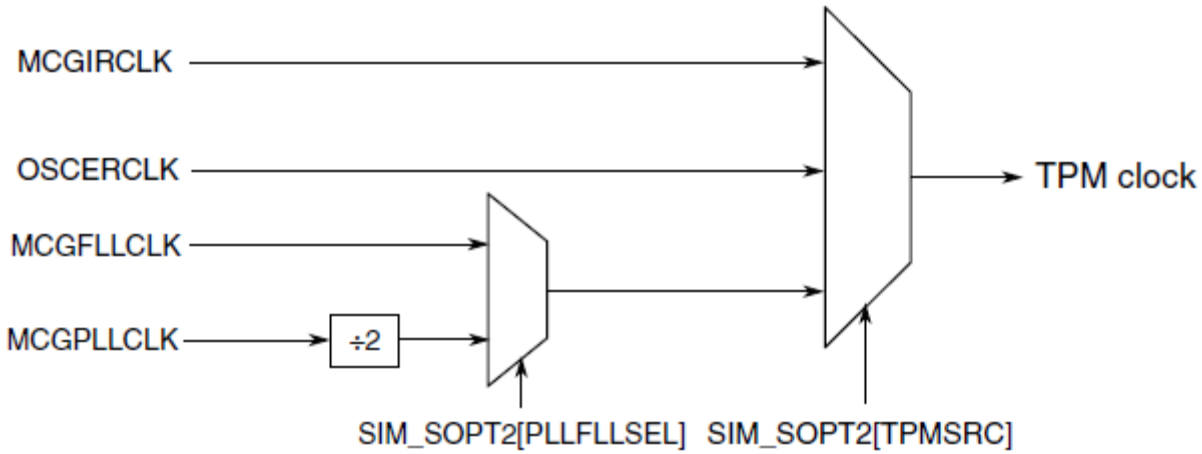


- SIM register CLKDIV1 controls both dividers

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTDIV1				0									OUTDIV4			0															
W																																
Reset	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ARM

Side Effects



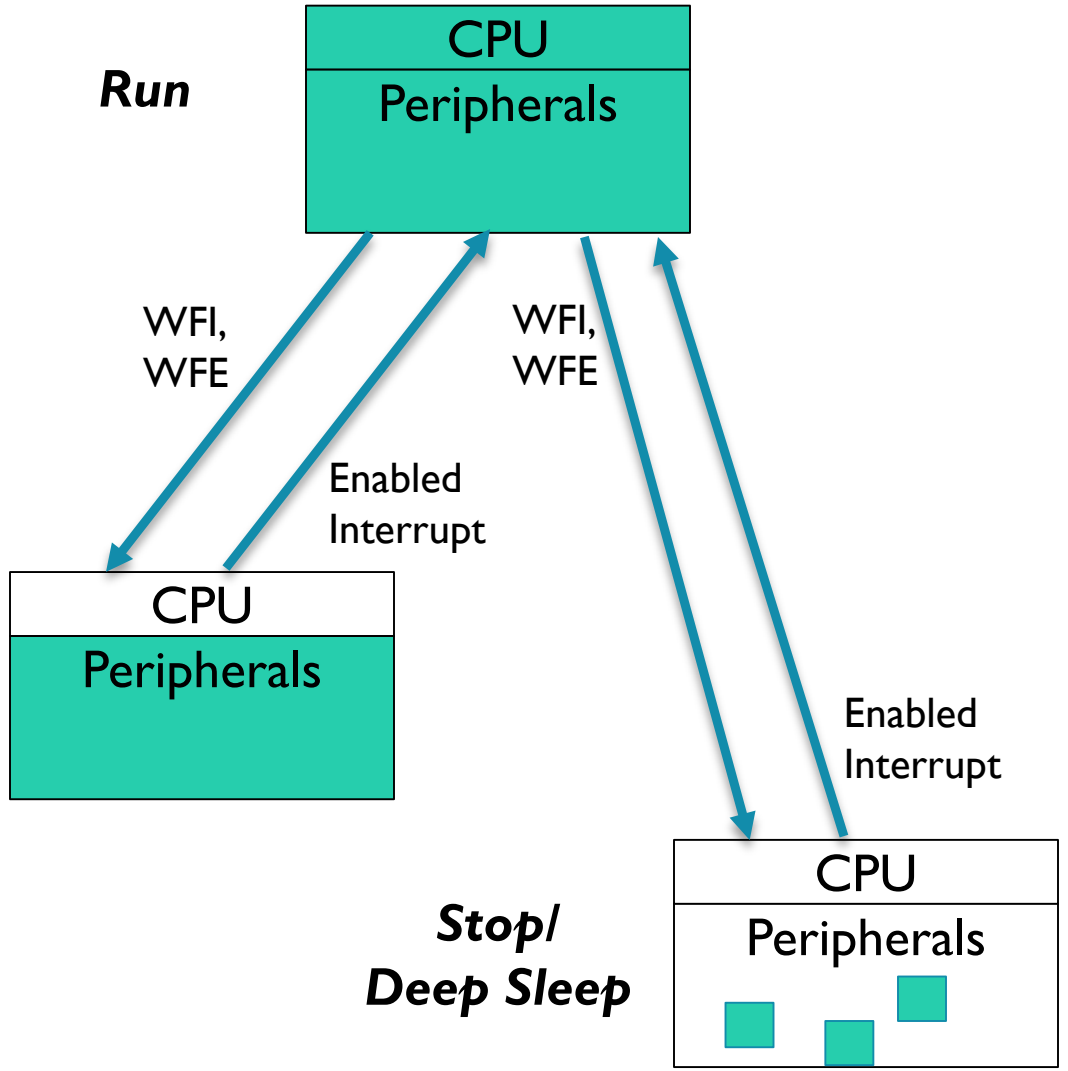
- PIT and other peripherals are clocked by bus clock
  - Changing bus frequency will change PIT period
  - May not be what you want
- TPM uses other clock inputs which don't change with bus clock
  - Can select FLL clock, PLL clock, internal reference clock, external reference clock, or oscillator clock
  - Use SIM SOPT2 register to select clock source

ARM

CPU and MCU Operating Modes

Classes of Power Modes

- Run modes – CPU executes instructions
  - Regular – runs at up to 48 MHz
  - Very Low Power – runs at up to 4 MHz
- Wait modes (ARM "sleep") – CPU doesn't execute instructions
  - Select **sleep** as low-power mode – SCR_SLEEPDEEP = 0
  - Enter by executing WFI instruction (wait for interrupt) or WFE (wait for event)
  - Peripherals operate
  - Exit when enabled interrupt occurs (limited set)
- Stop modes (ARM "deep sleep") – CPU doesn't execute instructions
  - Select **deep sleep** as low-power mode – SCR_SLEEPDEEP = 1
  - Enter by executing WFI instruction
  - Peripherals don't operate
  - Exit when enabled interrupt from AWIC occurs



Typical Currents in Different Modes

	Current (microamps) at V_{DD} = 3V			
Mode	**Normal**	**LL**	**VLL**	**VLP**
Run	5000			250
Wait	3700			135
Stop	345	1.9		4.4
Stop 3			1.4	
Stop 1			0.77	
Stop 0			0.38	

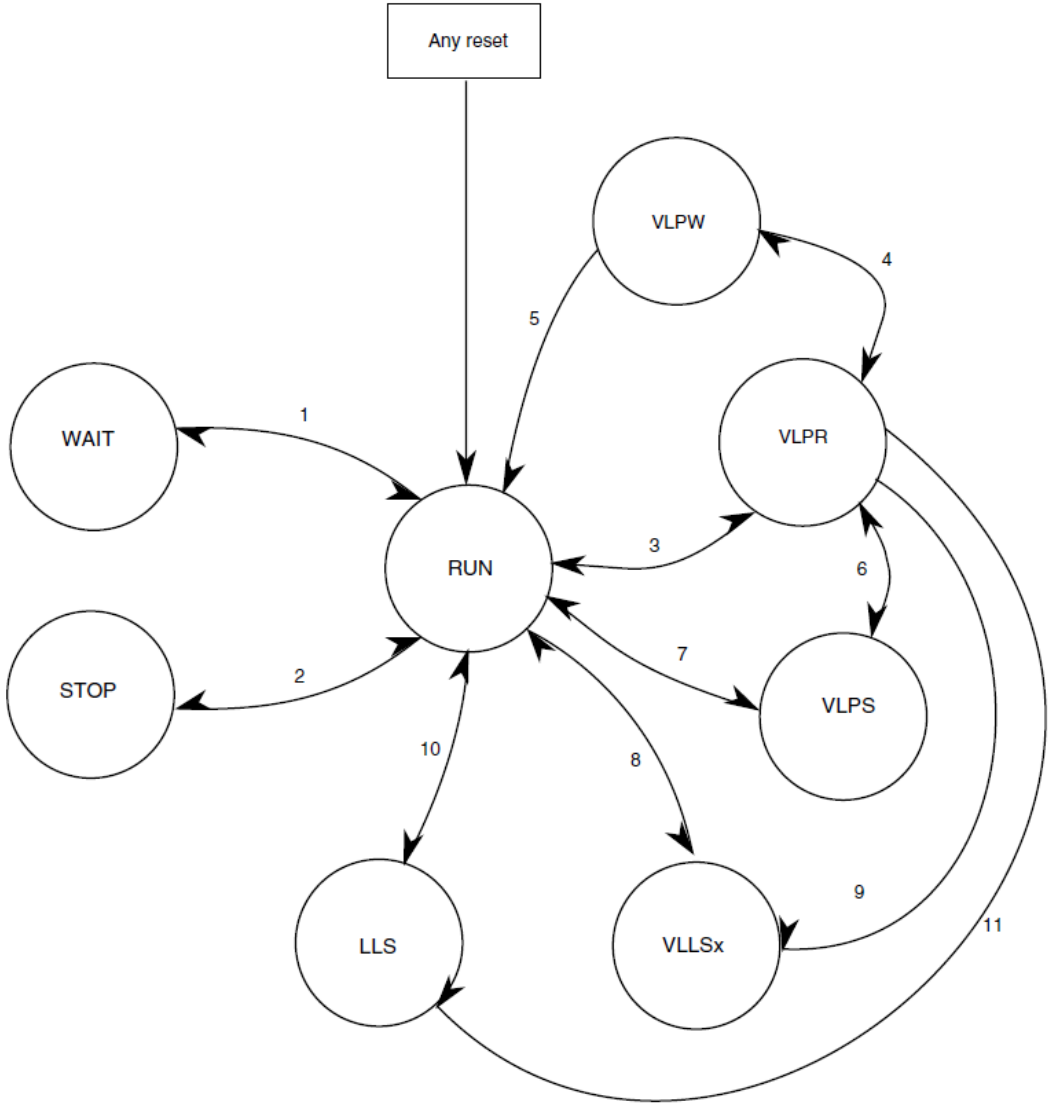
ARM

Stop Modes

- STOP: The basic stop mode
  - Core and system clocks are gated off
- VLPS: Very-low-power stop
  - Core and system clocks are gated off
- LLS: Low-leakage stop
  - Core and system clocks are gated off.
  - Internal logic supply voltage is reduced but state is retained.
- VLLS3: Very-low-leakage stop 3
  - RAM and I/O states are retained.
  - Internal logic is powered down and lost.
- VLLS2: Not available on Cortex-M0+
- VLLS1: Very-low-leakage stop 1
  - I/O states are retained.
  - RAM and internal logic are powered down and lost.
- VLLS0: Very-low-leakage stop 0
  - I/O states are retained.
  - RAM and internal logic are powered down and lost.
  - 1 kHz LPO is disabled
  - Power on reset circuit can be enabled or disabled

Valid Transitions Between Power Modes

- Reset takes MCU to RUN mode (runs at up to 48 MHz)
- From RUN mode, can get into most other modes
  - Wait
  - Stop
  - LL Stop
  - VLL Stop
- From VLPR (Very Low Power Run) mode (up to 4 MHz), can get into only some of other modes
  - VLP Wait
  - VLP Stop
  - LL Stop



ARM

How Do We Select The Low Power Mode?

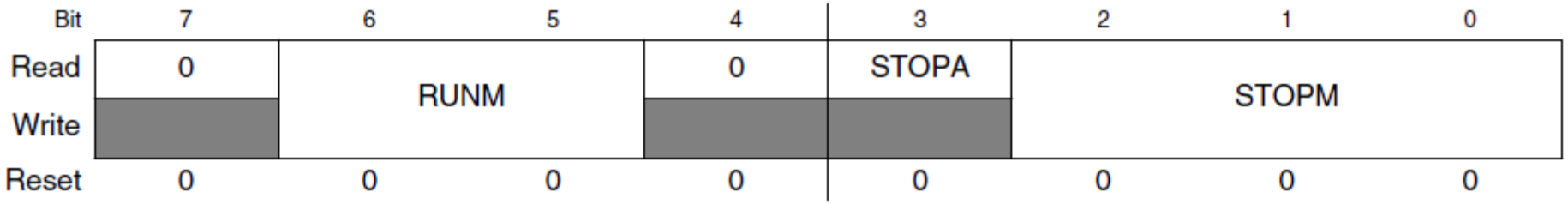
Address: 4007_E000h base + 0h offset = 4007_E000h

Bit	7	6	5	4	3	2	1	0
Read	0		AVLP	0	ALLS	0	AVLLS	0
Write								
Reset	0	0	0	0	0	0	0	0

- First need to allow the modes in Power Mode Protection register (SMC_PMPROT)
  - AVLP: allow very-low-power modes
  - ALLS: allow low-leakage stop mode
  - AVLLS: allow very-low-leakage stop mode
- Can only write PMPROT once after reset

ARM

Power Mode Control Register SMC_PMCTRL

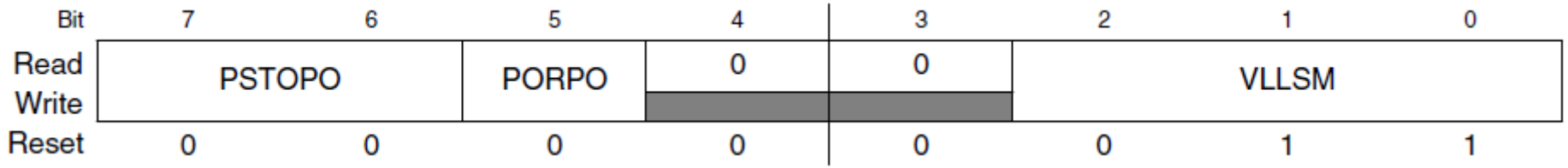


Bit	7	6	5	4	3	2	1	0
Read	0	RUNM		0	STOPA	STOPM		
Write								
Reset	0	0	0	0	0	0	0	0

- RUNM: Run Mode control
  - 00 Normal Run
  - 10 Very Low-power run
  - Other values reserved
- STOPM: Stop Mode control
  - 000 Normal Stop
  - 010 Very-low-power stop
  - 011 Low-leakage stop
  - 100 Very-low-leakage stop
  - Other values reserved
- STOPA: Stop aborted
  - Previous stop mode entry sequence did not complete

ARM

Stop Mode Control Register

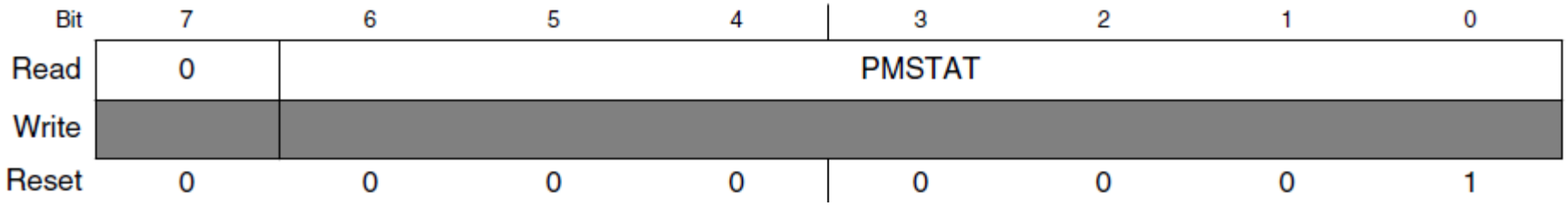


Bit	7	6	5	4	3	2	1	0
Read	PSTOPO		PORPO	0	0	VLLSM		
Write								
Reset	0	0	0	0	0	0	1	1

- PSTOPO – Allows stop to cause only a partial stop
  - 00: STOP: Normal stop mode
  - 01: PSTOP1: Partial stop with both system and bus clocks disabled
  - 10: PSTOP2: Partial stop with both system and bus clocks enabled
- PORPO - Controls power-on-reset circuit activity in VLLS0 mode
  - 0: POR enabled
  - 1: POR disabled
- VLLSM – Selects which VLLS Mode to enter if STOPM is VLLS
  - 000:VLLS0
  - 001:VLLS1
  - 010:VLLS2
  - 011:VLLS3

ARM

Power Mode Status Register – SMC_PMSTAT



- Indicates the MCU's power mode at present time
- Read-only

000_0001	Current power mode is RUN
000_0010	Current power mode is STOP
000_0100	Current power mode is VLPR
000_1000	Current power mode is VLPW
001_0000	Current power mode is VLPS
010_0000	Current power mode is LLS
100_0000	Current power mode is VLLS

ARM

KL25Z MCU and Peripherals

Power Only the Necessary Peripherals

I_{CMP}	CMP peripheral adder measured by placing the device in VLLS1 mode with CMP enabled using the 6-bit DAC and a single external input for compare. Includes 6-bit DAC power consumption.	22	22	22	22	22	22	µA
I_{RTC}	RTC peripheral adder measured by placing the device in VLLS1 mode with external 32 kHz crystal enabled by means of the RTC_CR[OSCE] bit and the RTC ALARM set for 1 minute. Includes ERCLK32K (32 kHz external crystal) power consumption.	432	357	388	475	532	810	nA

- Peripherals do use power
  - Example: Table 6 in KL25Z Data Sheet (excerpt above)
  - Microamps, nanoamps – they may matter for your application!
- Save power by disabling clock to unused peripherals
- Use SIM System Clock Gating Control Registers (SIM_SCGC4-7) to disable those clock signals

- May also want to disable USB voltage regulator if not needed (control with SIM_SOPT1)

ARM

Low Power Modes vs. Module Operation

- Details in Section 7.5 of KL25 Sub-Family Reference Manual
- Key
  - FF: full functionality
  - static: non-functional, but state information in registers and memory is retained
  - powered, low power: non-functional, but memory is retained
  - OFF: not powered, is reset state on wake-up

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
Core modules						
NVIC	FF	FF	static	static	static	OFF
System modules						
Mode Controller	FF	FF	FF	FF	FF	FF
LLWU[1]	static	static	static	static	FF	FF[2]
Regulator	low power	low power	ON	low power	low power	low power in VLLS3, OFF in VLLS0/1
LVD	disabled	disabled	ON	disabled	disabled	disabled
Brown-out Detection	ON	ON	ON	ON	ON	ON in VLLS1/3, optionally disabled in VLLS0[3]
DMA	FF	FF	Async operation	Async operation	static	OFF
Watchdog	FF	FF	static	static	static	OFF

ARM

Low Power Modes vs. Module Operation

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
Memory and memory interfaces						
Flash	1 MHz max access - no program	low power	low power	low power	OFF	OFF
SRAM_U and SRAM_L	low power	low power	low power	low power	low power	low power in VLLS3, OFF in VLLS0/1
Clocks						
1kHz LPO	ON	ON	ON	ON	ON	ON in VLLS1/3, OFF in VLLS0
System oscillator (OSC)	OSCERCLK max of 16MHz crystal	OSCERCLK max of 16MHz crystal	OSCERCLK optional	OSCERCLK max of 16MHz crystal	limited to low range/low power	limited to low range/low power in VLLS1/3, OFF in VLLS0
MCG	4 MHz IRC	4 MHz IRC	static - MCGIRCLK optional; PLL optionally on but gated	static - MCGIRCLK optional	static - no clock output	OFF
Core clock	4 MHz max	OFF	OFF	OFF	OFF	OFF
System clock	4 MHz max	4 MHz max	OFF	OFF	OFF	OFF
Bus clock	1 MHz max	1 MHz max	OFF	OFF	OFF	OFF

ARM

Low Power Modes vs. Peripheral Module Operation

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
Timers						
TPM	FF	FF	FF with clocks	FF with clocks	static	OFF
PIT	FF	FF	static	static	static	OFF
LPTMR	FF	FF	FF	FF	FF	FF[4]
RTC	FF	FF	FF	FF	FF	FF[5]
Analog						
16-bit ADC	FF	FF	ADC internal clock only	ADC internal clock only	static	OFF
CMP[6]	FF	FF	HS or LS compare	HS or LS compare	LS compare	LS compare in VLLS1/3, OFF in VLLS0
6-bit DAC	FF	FF	static	static	static	static, OFF in VLLS0
12-bit DAC	FF	FF	static	static	static	static
Human-machine interfaces						
GPIO	FF	FF	wakeup	wakeup	static, pins latched	OFF, pins latched
TSI	FF	wakeup[7]	wakeup[7]	wakeup[7]	wakeup[7]	wakeup[7]

ARM

Low Power Modes vs. Peripheral Module Operation

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
Communication Interfaces						
USB FS/LS	static	static	static	static	static	OFF
USB Voltage Regulator	optional	optional	optional	optional	optional	optional
UART0	1 Mbps	1 Mbps	FF with clocks	FF with clocks	static	OFF
UART1 , UART2	62.5 kbps	62.5 kbps	static, wakeup on edge	static, wakeup on edge	static	OFF
SPI0	master mode 500 kbps, slave mode 250 kbps	master mode 500 kbps, slave mode 250 kbps	static, slave mode receive	static, slave mode receive	static	OFF
SPI1	master mode 2 Mbps, slave mode 1 Mbps	master mode 2 Mbps, slave mode 1 Mbps	static, slave mode receive	static, slave mode receive	static	OFF
I^2C0	50 kbps	50 kbps	static, address match wakeup	static, address match wakeup	static	OFF
I^2C1	50 kbps	50 kbps	static, address match wakeup	static, address match wakeup	static	OFF

ARM

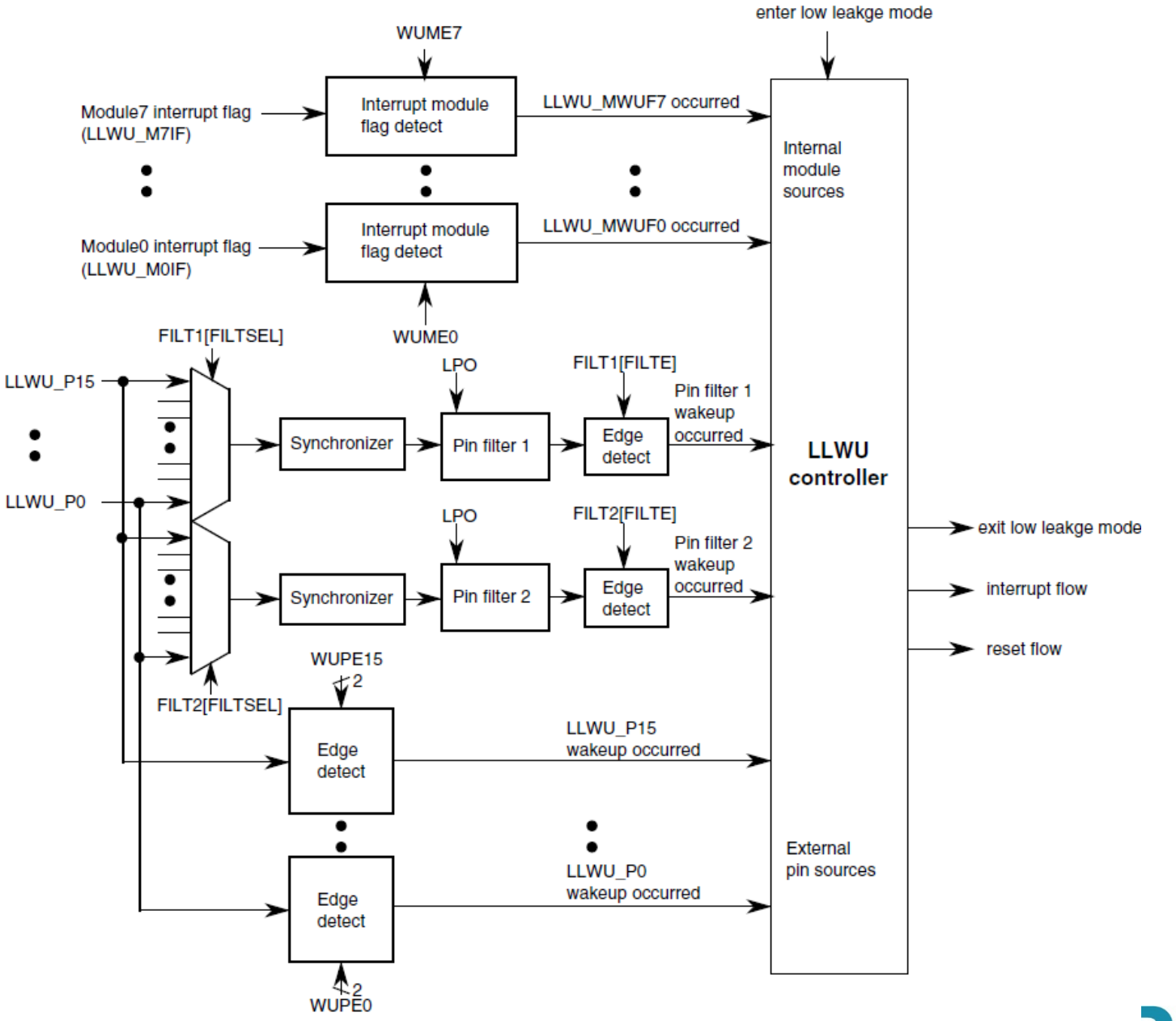
Using Peripherals to Wake Up

How Do We Wake Up?

- It depends on the sleep mode
  - NVIC is awake only in run and wait modes, not in stop modes
  - If not awake, need to use low-leakage wakeup unit (LLWU)
    - LLWU has own interrupt vector
    - LLWU will wake up NVIC
- Low Power Timer (LPTMR) operates in all modes
  - Be sure to enable LPTMR0 flag in LLWU to allow waking from LLS mode
- After waking up, may want to go to sleep immediately
  - If MCU only needs to execute code in ISRs
  - Set SLEEPONEXIT bit in SCR

Low-Leakage Wakeup Unit

- LLWU can wake up MCU or NVIC, uses very little power
- Is activated only in low-leakage power modes
- Can monitor external pins for edges or changes (LLWU_P0-P15)
  - Can filter out noise to prevent false wakeup
- Can monitor internal MCU modules for interrupt request (LLWU_M0IF-M7IF)
- Different wake-up methods
  - If in LLS mode, interrupt is generated
  - If in VLLS mode, reset is generated



LLWU Wakeup Sources

External

LLWU Source	Port Bit
LLWU_P5	PTB0
LLWU_P6	PTC1
LLWU_P7	PTC3
LLWU_P8	PTC4
LLWU_P9	PTC5
LLWU_P10	PTC6
LLWU_P14	PTD4
LLWU_P15	PTD6

Internal

LLWU Source	Module
LLWU_M0IF	LPTMR0
LLWU_M1IF	CMP0
LLWU_M2IF	Reserved
LLWU_M3IF	Reserved
LLWU_M4IF	TSI0
LLWU_M5IF	RTC Alarm
LLWU_M6IF	Reserved
LLWU_M7IF	RTC Seconds

ARM

Transition Delays

Table 4. Power mode transition operating behaviors

Symbol	Description	Min.	Typ.	Max.	Unit	Notes
t_{POR}	After a POR event, amount of time from the point V_{DD} reaches 1.8 V to execution of the first instruction across the operating temperature range of the chip.	—	—	300	µs	
	• VLLS0 → RUN	—	95	115	µs	
	• VLLS1 → RUN	—	93	115	µs	
	• VLLS3 → RUN	—	42	53	µs	
	• LLS → RUN	—	4	4.6	µs	
	• VLPS → RUN	—	4	4.4	µs	
	• STOP → RUN	—	4	4.4	µs	

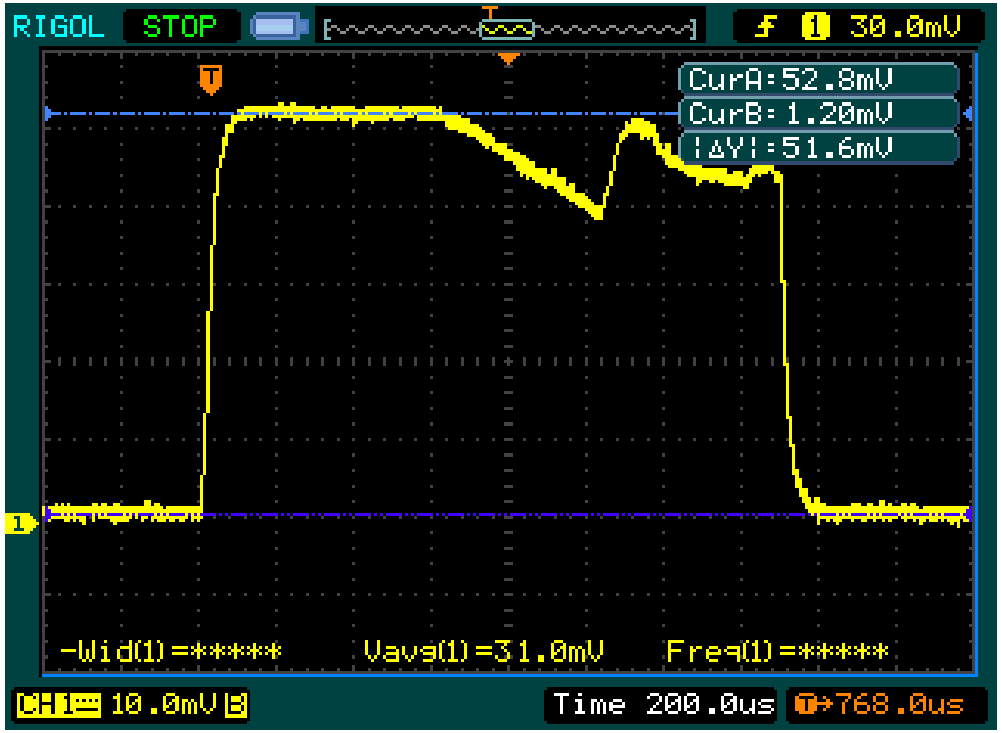
- Times to transition between different power modes vary

- ~ 100 microseconds to come out of VLLS0 or VLLS1
- ~ 50 microseconds to come out of VLLS3
- ~4 microseconds to come out of LLS, VLPS or STOP

ARM

Power Profiling

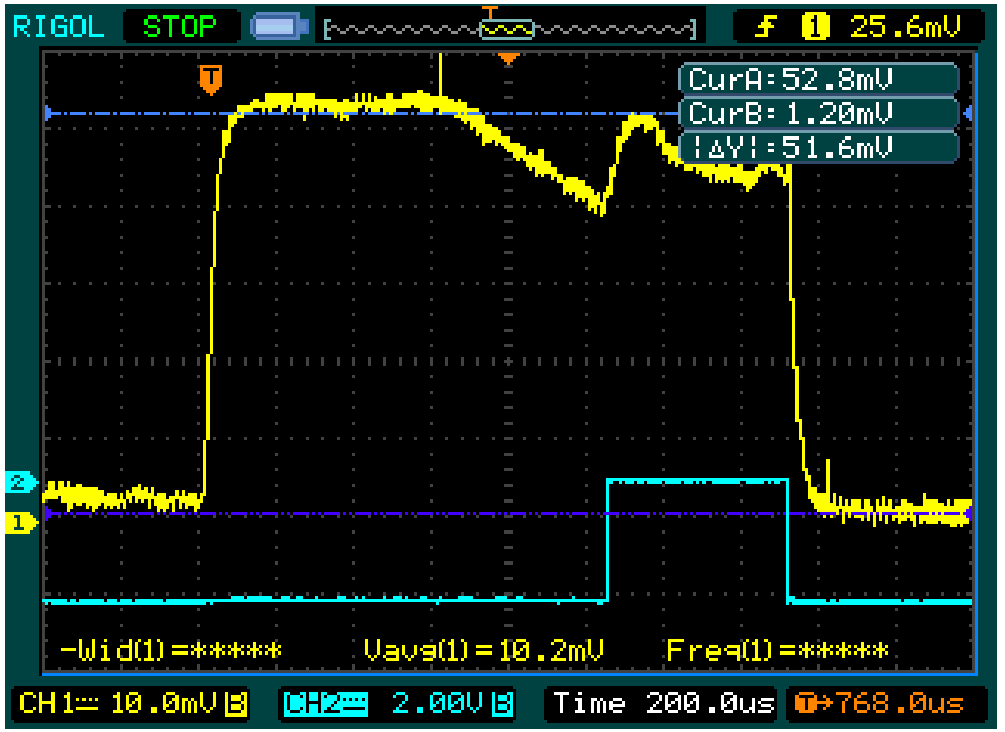
- What is happening here?
- Duration is about 1.5 ms
- Let's find out – add an output bit to indicate when processor is awake (1), when it is asleep (0)
  - Set to 1 on reset, entry to ISR, immediately after WFI instruction
  - Clear to 0 immediately before WFI



ARM

The Plot Thickens

- What is happening before our ISR starts running?
- It is taking a very long time (about 1 ms) compared to the ISR duration, leading to a lot of extra power consumption
- *Extra credit exercise*

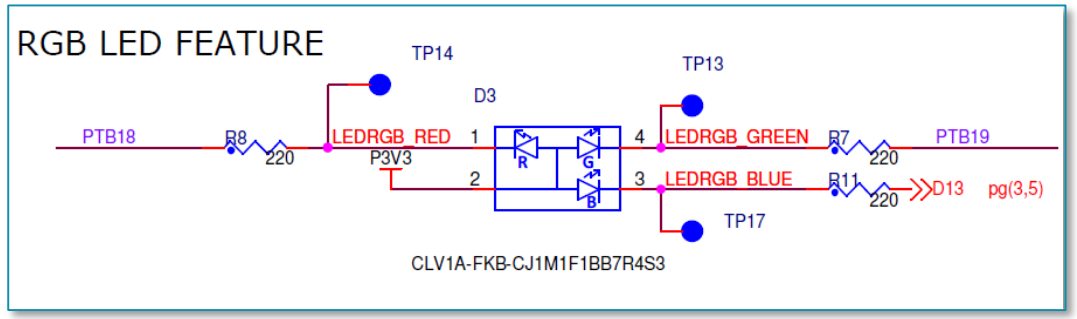


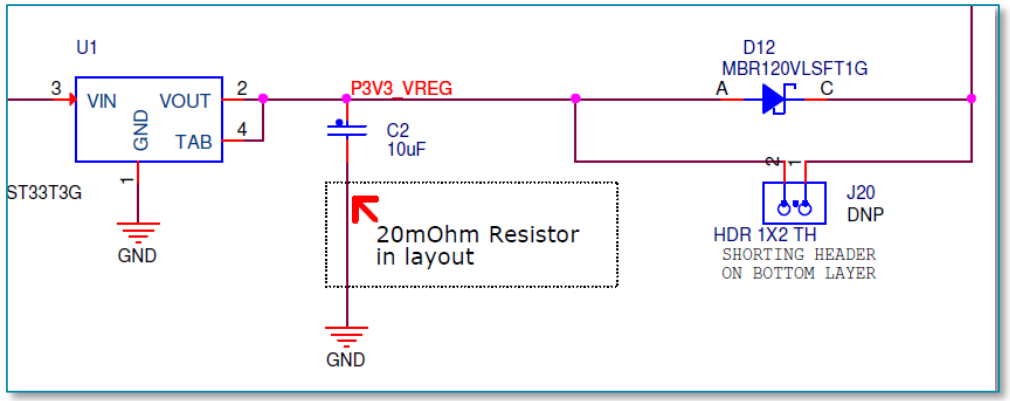
ARM

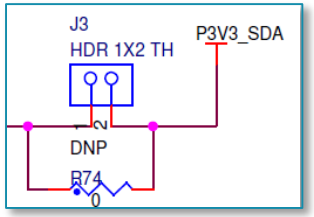
FRDM-KL25Z Development Board

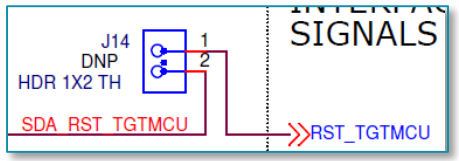
FRDM-KL25Z Features for Low-Power or Low-Energy

- High-Efficiency LEDs
  - Don't need much current (~2 mA) to be very bright
  - Can pulse-width modulate them for dimming
    - Can even reduce series resistors!
- Accelerometer
  - Low-power device
  - Adjustable sampling rate
  - Sleep mode
  - Auto-sleep mode
- Linear Voltage Regulator – don't use it!
  - Cut trace (J20) which shorts diode D12 on output, can power P3V3 rail separately now
- Disconnect SDA
  - Disconnect power (J3, P_SDA) and reset line (J14)





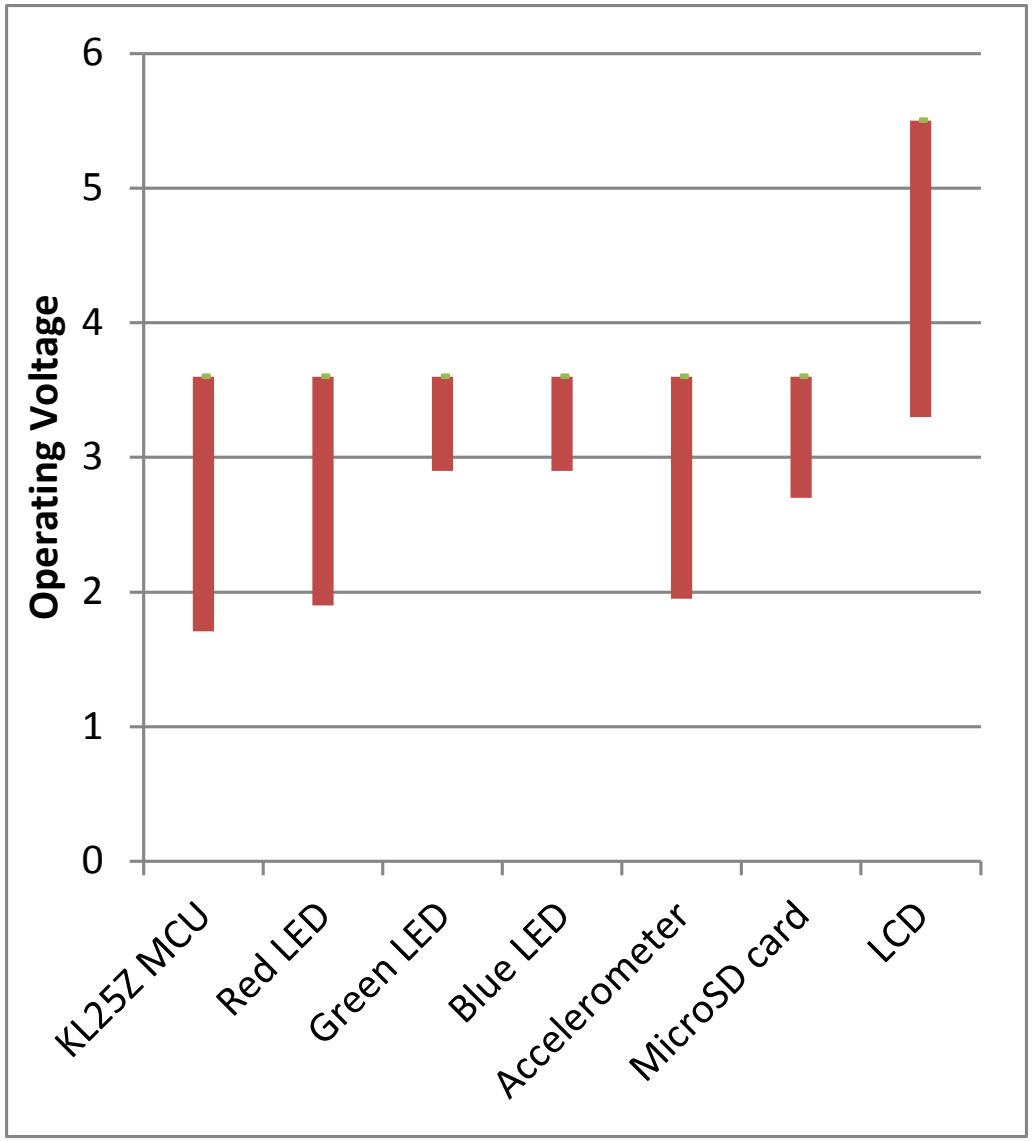




Voltage Scaling for the KL25

How about Lowering the Operating Voltage?

- KL25Z MCU rated to operate from 1.71V to 3.6V
- P α V^2 so this will have a **BIG** impact
- Dropping from 3.3 to 1.8V would reduce power and energy by a factor of $3.3^2/1.8^2 = 3.36x$
- How low can we go? What stops working on the Freedom board or other system as we lower the supply voltage?



ARM

Minimum Data Retention Voltage

Table 1. Voltage and current operating requirements (continued)

Symbol	Description	Min.	Max.	Unit	Notes
V_{HYS}	Input hysteresis	0.06 × V_{DD}	—	V	
I_{ICDIO}	Digital pin negative DC injection current — single pin • $V_{IN} < V_{SS}$-0.3V	-5	—	mA	1
I_{ICAIO}	Analog[2] pin DC injection current — single pin • $V_{IN} < V_{SS}$-0.3V (Negative current injection) • $V_{IN} > V_{DD}$+0.3V (Positive current injection)	-5 —	— +5	mA	3
I_{ICcont}	Contiguous pin DC injection current —regional limit, includes sum of negative injection currents or sum of positive injection currents of 16 contiguous pins • Negative current injection • Positive current injection	-25 —	— +25	mA	
V_{RAM}	V_{DD} voltage required to retain RAM	1.2	—	V	

- Datasheet says minimum RAM retention voltage = 1.2V
  - *Extra Credit Project - Verify it works!*
- What about processor registers? 1.2V also, or something higher?
- How low can we take the MCU supply voltage and have all RAM and registers retain their values?

ARM