SJSU SAN JOSÉ STATE UNIVERSITY

Charles W. Davidson College of Engineering

Department of Computer Engineering

Real-Time Embedded System
Co-Design
CMPE 146 Section 1
Fall 2024

Power Management

Power Control Manager (PCM)

- Manages power-mode change requests
- Clock System (CS) and the Power Supply System (PSS) settings are the two primary elements that control the power settings of the device
- Various power modes available for optimization of power in different execution conditions

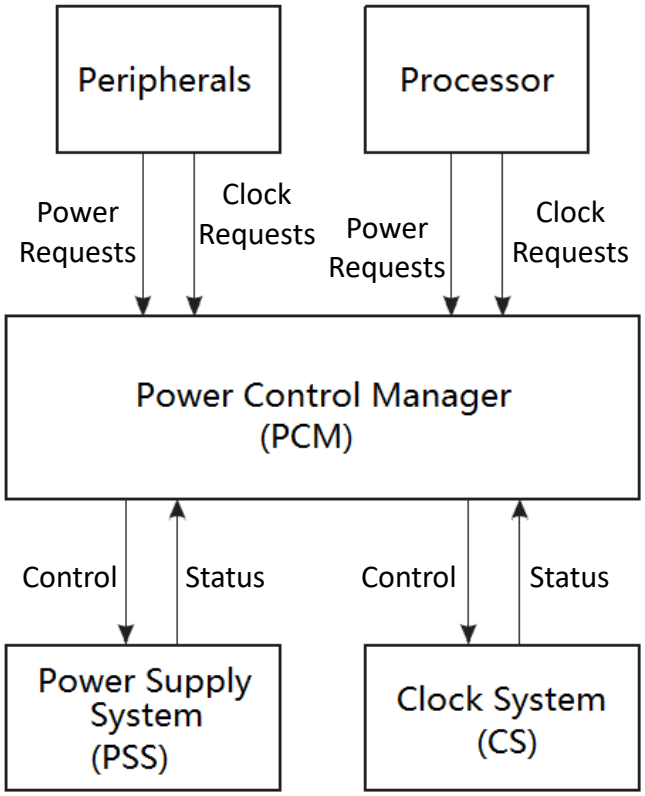

Diagram source: TI MSP432P4xx SimpleLink Microcontrollers Technical Reference Manual

PCM input events

Most common events to cause a power change request to PCM:

- PCM control register (PCMCTL0)
  - Can be modified directly by the application execution to request that a particular power state to be entered
- Interrupts
  - Interrupts in low-power modes cause operation to automatically return to active mode
- Reset events
  - Events cause the power mode to be set back to its default setting
- Debug events
  - Power mode settings are adapted to support debug hardware requirements
  - For example, during debugging, sleep mode must be exited in order to restore system clock for access to peripherals or memory
    - When done, resume the sleep mode

Power Modes

- Switching operations are based on adjusting voltage and frequency
  - Commonly known as Dynamic Voltage Frequency Scaling (DVFS)
- Two main power operating modes
  - Active Mode (AM): Program execution is possible
  - Low Power Mode (LPM): Program execution is suspended
- In AM, there are six submodes that PCM supports on MSP432, based on usages of
  - Frequency: Normal or low frequency (LF, ≤ 128 kHz)
  - Voltage regulator: LDO (less efficient) or DCDC (more efficient)
  - Core voltage level: VCORE0 (lower) or VCORE1 (higher)

Operating State	Frequency	Regulator	Voltage
AM_LDO_VCORE0	Normal	LDO	VCORE0
AM_LDO_VCORE1	Normal	LDO	VCORE1
AM_DCDC_VCORE0	Normal	DCDC	VCORE0
AM_DCDC_VCORE1	Normal	DCDC	VCORE1
AM_LF_VCORE0	Low	LDO	VCORE0
AM_LF_VCORE1	Low	LDO	VCORE1

Power Modes (cont'd)

- In LPM, there are 12 submodes in three general categories

Power Mode	Operating State	Main Features
LPM0 (Sleep)	LPM0_LDO_VCORE0	Same as the corresponding AM state except that program execution is suspended
	LPM0_LDO_VCORE1	
	LPM0_DCDC_VCORE0	
	LPM0_DCDC_VCORE1	
	LPM0_LF_VCORE0	
	LPM0_LF_VCORE1	
LPM3 (Deep Sleep)	LPM3_LDO_VCORE0	Only RTC and WDT functional at 32 kHz max All other peripherals and SRAM under SRPG
	LPM3_LDO_VCORE1	
LPM4 (Deep Sleep)	LPM4_LDO_VCORE0	All peripherals under SRPG
	LPM4_LDO_VCORE1	
LPM3.5 (Stop or Shut Down)	LPM3.5_LDO_VCORE0	RTC and WDT can be functional at 32 kHz max Only SRAM Bank 0 under SRPG
LPM4.5 (Stop or Shut Down)	LPM4.5_VCORE_OFF	All devices are powered down Device I/O pin states are retained

AM-LPM Mode Transitions

- LPM must be entered from AM through program control

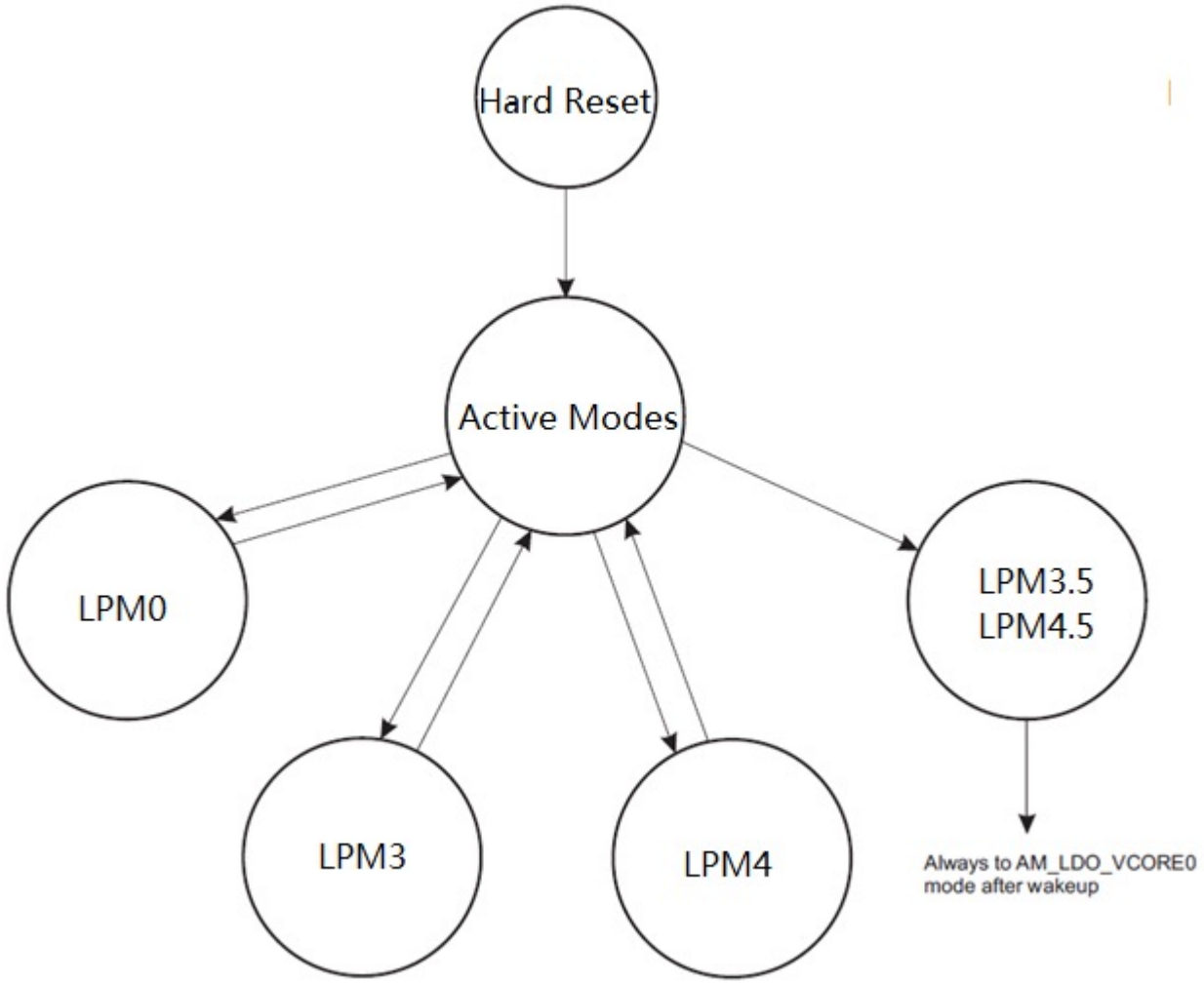

Diagram source: TI MSP432P4xx SimpleLink Microcontrollers Technical Reference Manual

AM Mode State Transitions

- Device enters AM_LDO_VCORE0 state after reset
- Valid state transitions and latencies:

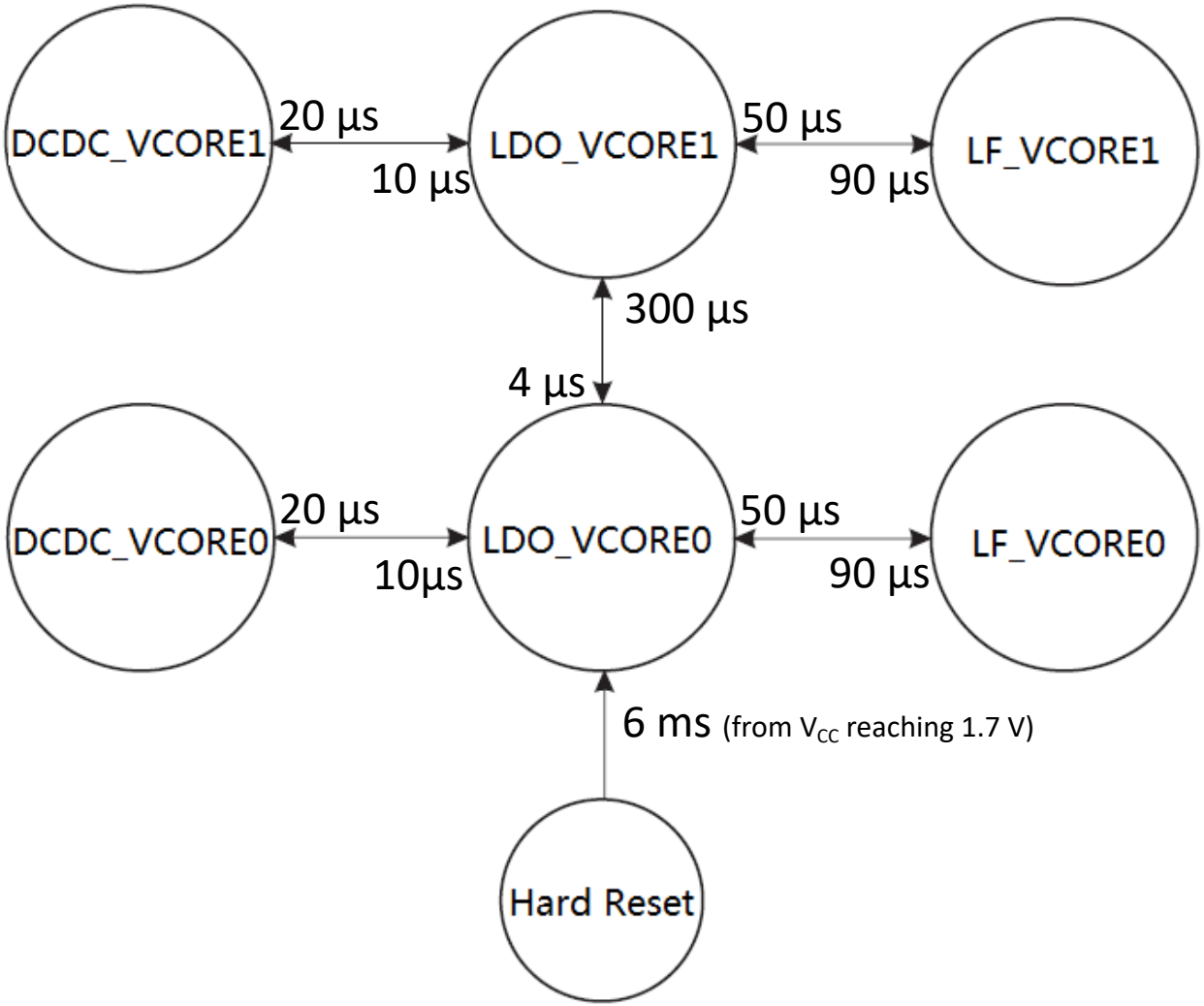

Diagram source: TI MSP432P4xx SimpleLink Microcontrollers Technical Reference Manual

LPM0 State Transitions

- Entry to LPM0 is possible from any AM
- Can only transition to/from corresponding AM DCDC/LDO and VCOREx states, i.e., voltage regulator and level remain the same
- Valid state transitions and latencies:

Originating State	Final State	Latency (MCLK cycles)
AM_LDO_VCOREx	LPM0_LDO_VCOREx	1
LPM0_LDO_VCOREx	AM_LDO_VCOREx	3
AM_DCDC_VCOREx	LPM0_DCDC_VCOREx	1
LPM0_DCDC_VCOREx	AM_DCDC_VCOREx	3
AM_LF_VCOREx	LPM0_LF_VCOREx	1
LPM0_LF_VCOREx	AM_LF_VCOREx	3

x = 0 or 1

LPM3/4 State Transitions

- Entry to LPM3/4 is possible from AM with LDO voltage regulator
  - DCDC voltage regulator cannot be used in LPM3/4
- Can only transition to/from corresponding VCOREx state
  - Transitions at low frequencies take much longer
- Valid state transitions and latencies:

Originating State	Final State	Latency (µs)
AM_LDO_VCORE0	LPM3/4_LDO_VCORE0	22
LPM3/4_LDO_VCORE0	AM_LDO_VCORE0	8[1] or 9[2]
AM_LDO_VCORE1	LPM3/4_LDO_VCORE1	21
LPM3/4_LDO_VCORE1	AM_LDO_VCORE1	7.5[1] or 8[2]
AM_LF_VCOREx	LPM3/4_LDO_VCOREx	240 (128 kHz) or 880 (32 kHz)
LPM3/4_LDO_VCOREx	AM_LF_VCOREx	45 (128 kHz) or 150 (32 kHz)

x = 0 or 1

[1] Wake up from I/O without glitch filter

[2] Wake up from I/O with glitch filter

LPM3.5/4.5 State Transitions

- Entry to LPM3.5/4.5 is possible from any AM
- Uses only VCORE0 in LPM3.5 and LPM4.5
- Exiting from LPM3.5 and LPM4.5 causes a POR event
- Valid state transitions and latencies:

Originating State	Final State	Latency (µs)
AM_LDO_VCOREx	LPM3.5	25
AM_DCDC_VCOREx	LPM3.5	35
AM_LF_VCOREx	LPM3.5	225
AM_LDO_VCOREx	LPM4.5	25
AM_DCDC_VCOREx	LPM4.5	35
AM_LF_VCOREx	LPM4.5	250
LPM3.5	AM_LDO_VCORE0	700
LPM4.5	AM_LDO_VCORE0	800

x = 0 or 1

Current Consumptions

Some typical current consumption values on MSP432

State	Current	Battery* Run Time
High-frequency AM	3.2 mA at 40 MHz (80 µA/MHz)	15.6 hrs.
Low-frequency AM	83 µA at 128 kHz	251 days
High-frequency LPM0	0.87 mA at 40 MHz	2.4 days
Low-frequency LPM0	66 µA at 128 kHz	315 days
LPM3 (with RTC)	660 nA	8.6 yrs.
LPM3.5 (with RTC)	630 nA	9.1 yrs.
LPM4	500 nA	11.4 yrs.
LPM4.5	25 nA	228 yrs.

*Coin-size battery: Nominal voltage of 3 V, ~50 mA-hr

Enter Sleep Modes

- Two instructions on Cortex-M3/M4 processor
- WFI (Wait For Interrupt)
  - Triggers sleep mode immediately
  - Processor can be woken up by interrupts, reset or debug operation
- WFE (Wait For Event)
  - Triggers sleep mode immediately if no pending event
  - Processor can be woken by an event, which can be an interrupt, debug operation, reset or a pulse signal at an external input pin
- Sleep-on-Exit feature
  - If enabled, processor automatically enters sleep mode upon exiting from an interrupt handler
  - Allows processor to be active only when an interrupt request is to be serviced

DriverLib

- Simplifies management of power states
- Hide details from users on switching between power states
  - There are specific constraints/requirements of state transitions
- Functions to manipulate modes and states
  - PCM_setPowerMode()
  - PCM_setPowerState()
- Low Power Mode entry functions
  - PCM_gotoLPM0()
  - PCM_gotoLPM3()
  - PCM_gotoLPM4()
  - PCM_shutdownDevice()
    - For transitioning to LPM3.5 or LPM4.5
  - Except for LPM3.5 or LPM4.5, original state of the device before low power mode entry is retained
    - After the device wakes up from low power mode, the original power mode is restored

Memory Access

- It is beneficial to optimize memory accesses to reduce as much power consumption as possible during code execution
- Power consumption for SRAM is lower than flash memory
  - Copying program to SRAM to execute would help
  - SRAM also runs at the same clock frequency as the processor
- Executing out of ROM yields both higher performance (0 wait state access) and better power consumption (much lower than flash execution, and even better than SRAM)
  - Use the DriverLib residing in ROM
- Enable the read buffering feature in flash controller
  - Reduce power consumption and improve performance across predominantly contiguous memory accesses

Application Considerations

- LPM0 is useful to save power when processor execution is not required, yet very fast wake-up time is necessary
- LPM3 and LPM4 modes are useful for relatively infrequent processor activity followed by long periods of low-frequency activity, better known as low-duty-cycle applications
- The wake-up time from LPM3 and LPM4 is longer than wake-up times from LPM0, but the average power consumption is significantly lower
- Reducing frequency may reduce overall system throughput but could increase the overall energy consumption
  - Device stays in active operating mode for more time and in low-power mode for less time
- Always operating at high frequency to quickly complete an active mode task may not be the best strategy
  - Peak current requirements during short bursts can drain the battery faster, thus shortening battery life

Application Considerations (cont'd)

- An application can program the DCO (Digitally Controlled Oscillator) for the best combination of optimal power consumption and required accuracy
- Tools like TI's EnergyTrace can help developers construct a detailed energy profile of the system
- Use accelerators or built-in FPU to reduce power consumption
- RTOS (Real-Time Operating System) can make the power management transparent to the programmer