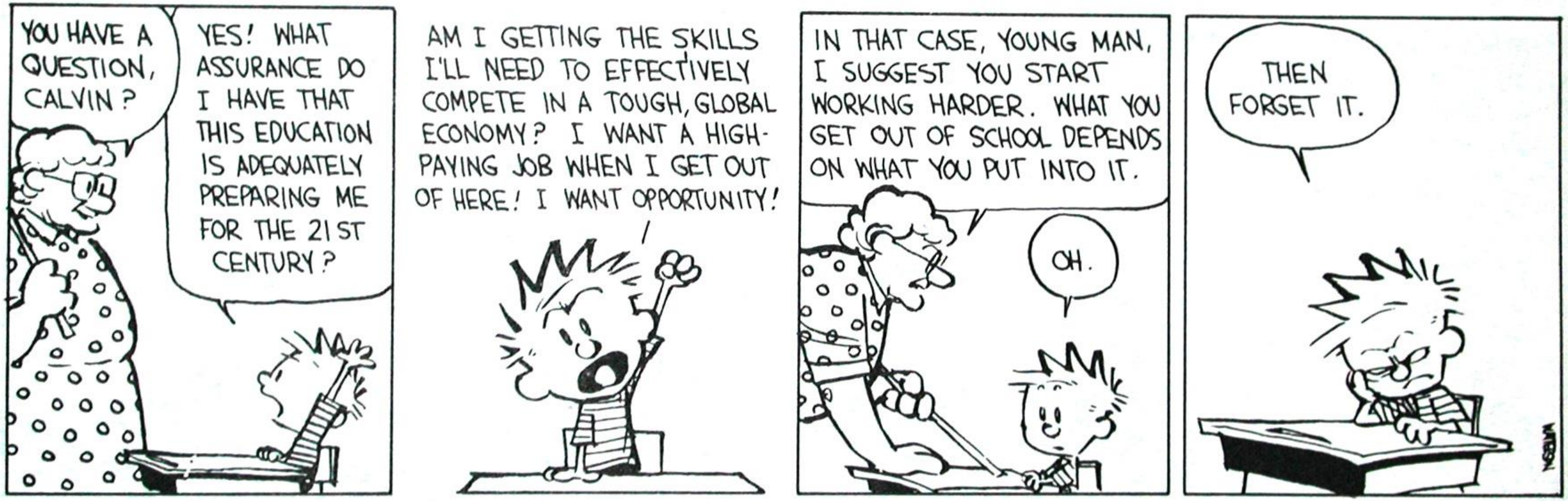
Introduction to Embedded Systems

These lecture notes created by Mr. James B. (Jim) Carlson, NCSU

ECE 306 – Introduction to Embedded Systems


YOU HAVE A QUESTION, CALVIN?
YES! WHAT ASSURANCE DO I HAVE THAT THIS EDUCATION IS ADEQUATELY PREPARING ME FOR THE 21ST CENTURY?
AM I GETTING THE SKILLS I'LL NEED TO EFFECTIVELY COMPETE IN A TOUGH, GLOBAL ECONOMY? I WANT A HIGH-PAYING JOB WHEN I GET OUT OF HERE! I WANT OPPORTUNITY!
IN THAT CASE, YOUNG MAN, I SUGGEST YOU START WORKING HARDER. WHAT YOU GET OUT OF SCHOOL DEPENDS ON WHAT YOU PUT INTO IT.
OH.
THEN FORGET IT.
WATTERSON

WE'RE SUPPOSED TO HAVE THIS WHOLE STUPID BOOK READ BY TOMORROW.
FLIP-IP-IP-IP-IP-IP-IP!
WATTERSON
THERE! IT'S GOOD TO GET THAT OUT OF THE WAY!
READING GOES FASTER IF YOU DON'T SWEAT COMPREHENSION.
WHERE'S THE FRISBEE?

Today

- Course Syllabus
- What Are Embedded Systems?
- Why Are We ….?
- Course Coverage Overview
- Introduction to Microcontroller-based Circuit Design

Why Are We…?

- Using C instead of Java?
  - C is the de facto standard for embedded systems because of
    - Precise control over what the processor is doing.
    - Predictable behavior, no OS (e.g. Garbage Collection) preemption
    - Modest requirements for ROM, RAM, and MIPS, so much cheaper system
- Learning assembly language?
  - The compiler translates C into assembly language. To understand whether the compiler is doing a reasonable job, you need to understand what it has produced.
  - Sometimes we may need to improve performance by writing assembly versions of functions.
- Required to have a microcontroller board?
  - The best way to learn is hands-on.
  - You will keep these boards after the semester ends for possible use in other projects (e.g. Senior Design, Embedded System Design, Mechatronics, etc.)

Definition of an Embedded Computer

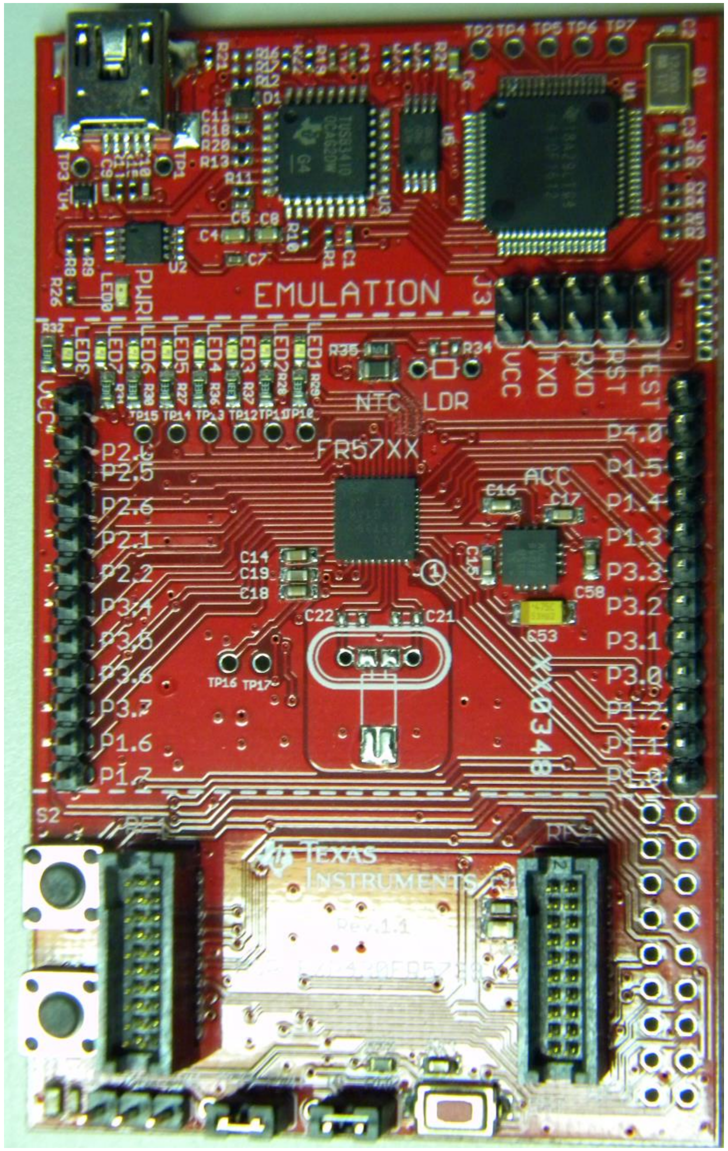
- Computer purchased as part of some other piece of equipment
  - Typically dedicated software (may be user-customizable)
  - Often replaces previously electromechanical components
  - Often no “real” keyboard
  - Often limited display or no general-purpose display device
- But, every system is unique -- there are always exceptions

A Customer View

- Reduced Cost
- Increased Functionality
- Improved Performance
- Increased Overall Dependability

Microcontroller and Starter Kit

- Texas Instruments
- MSP430 family of microcontrollers
- MSP-EXP430FR5739 Experimenter Board



MSP430 Hardware / Software

- Set / Clear LEDs
  - Set and Clear Specific LEDs?
- Test Switch Functionality
  - What happens when switches are pressed?
- Evaluate Clock
  - Is the clock configured how you expect it to be?
- Test the Display
  - Is your assembly and soldering working?

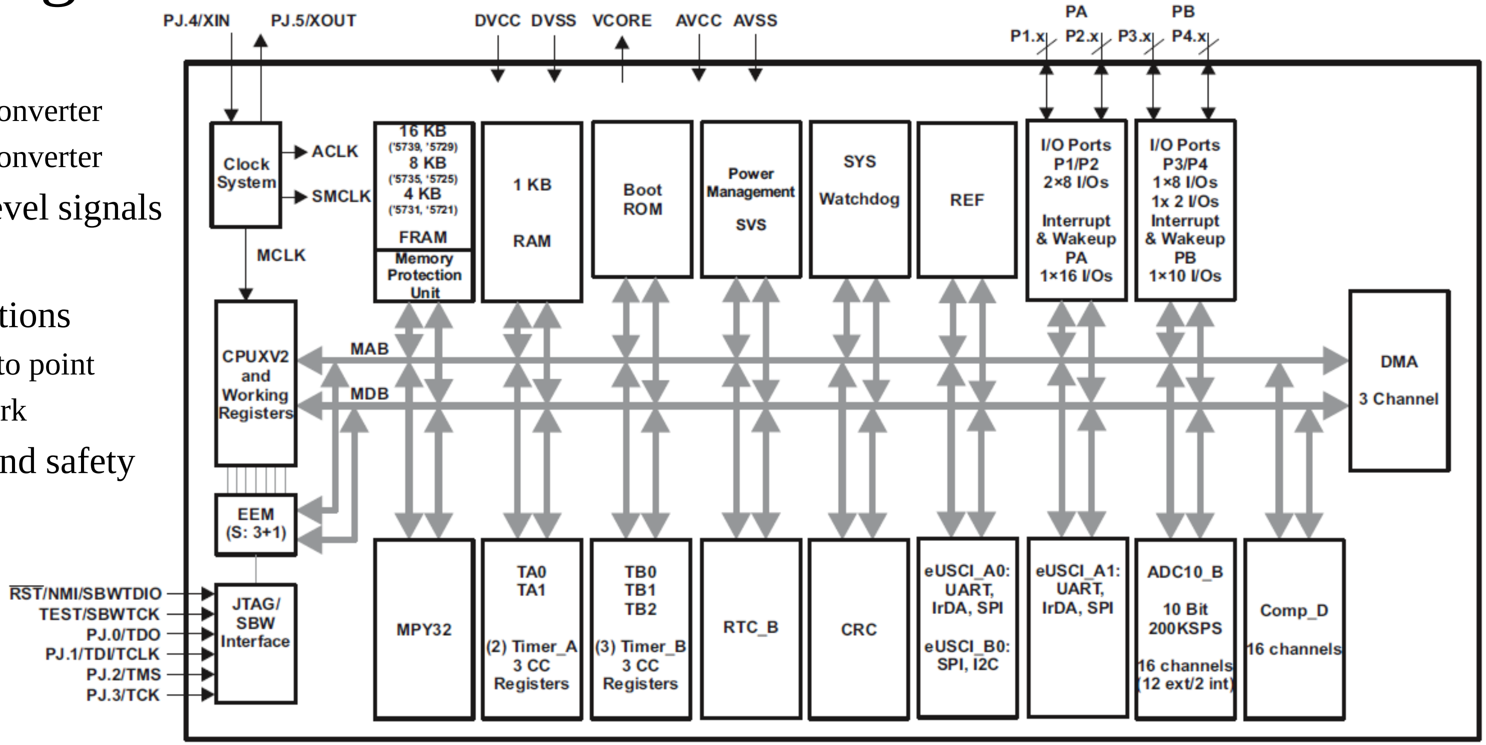
Course Overview

- Introduction to Embedded Systems
- MSP430 Processor
  - MSP430 Instruction Set Architecture
  - Circuit Design
- Programming
  - Assembly Language Programming
  - C Programming Review
  - C and the Compiler
- Software Development
  - Debugging
  - Simulation Design and Debugging
- Interfacing
  - Using and Programming Interrupts
  - Digital I/O Peripherals: General Purpose, T/C and PWM
  - Analog I/O Peripherals
  - Serial Communications and Peripherals
- Optimizations
  - Performance Analysis
  - Power Analysis
- Multithreaded Systems
  - Threads, Tasks and Simple Scheduling
  - Real-Time Operating Systems
  - Threaded Program Design

Microcontroller vs. Microprocessor

Microcontroller has peripherals for embedded interfacing and control

- Analog
  - ADC Converter
  - DAC Converter
- Non-logic level signals
- Timing
- Communications
  - point to point
  - network
- Reliability and safety



Designing a Microcontroller into a System

Power supply

Clock signal generator

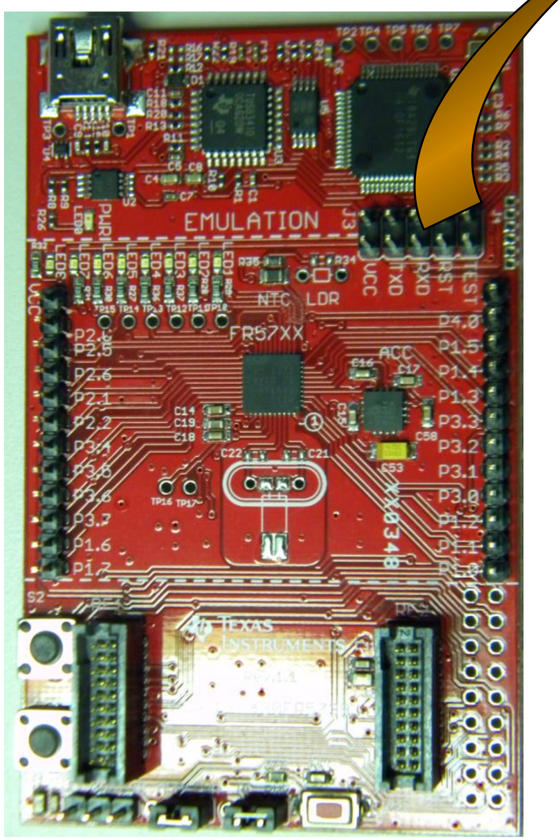
Reset controller

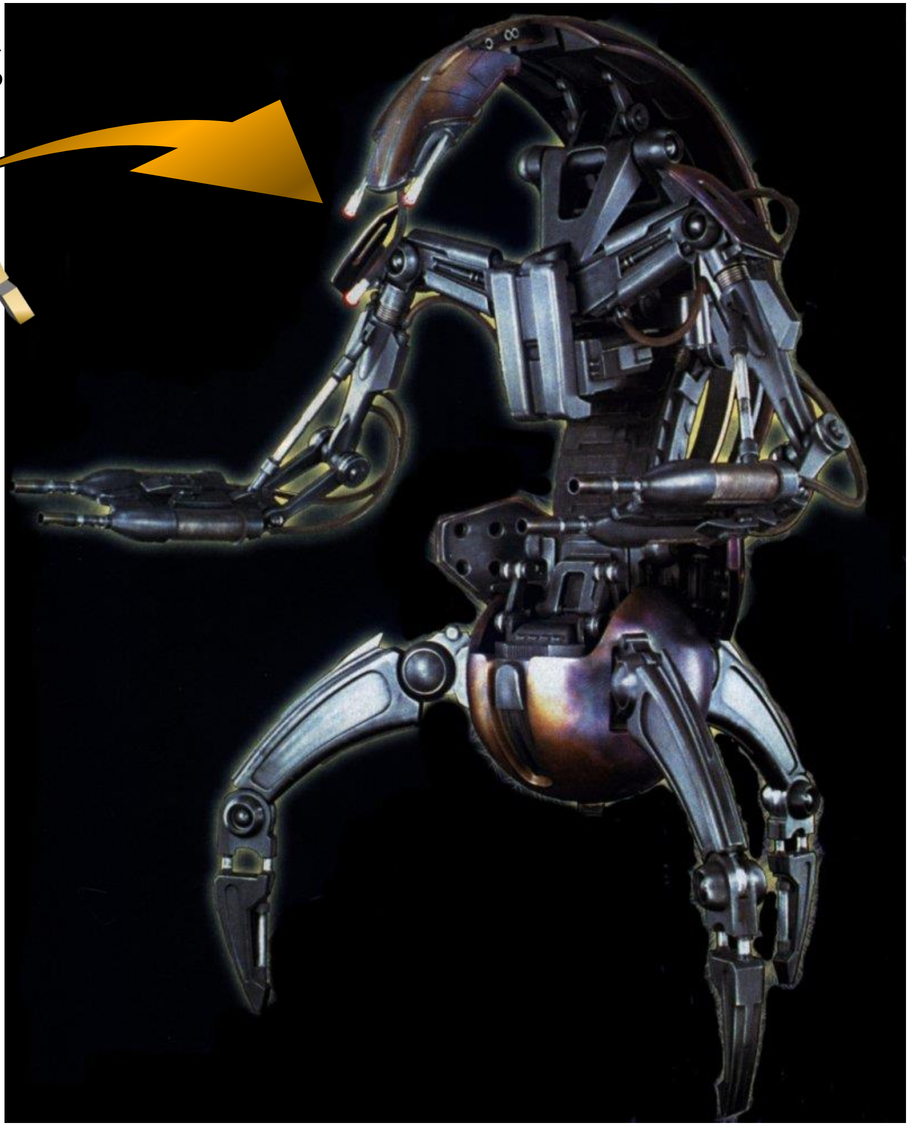
Memory

Digital interfacing

Analog interfacing

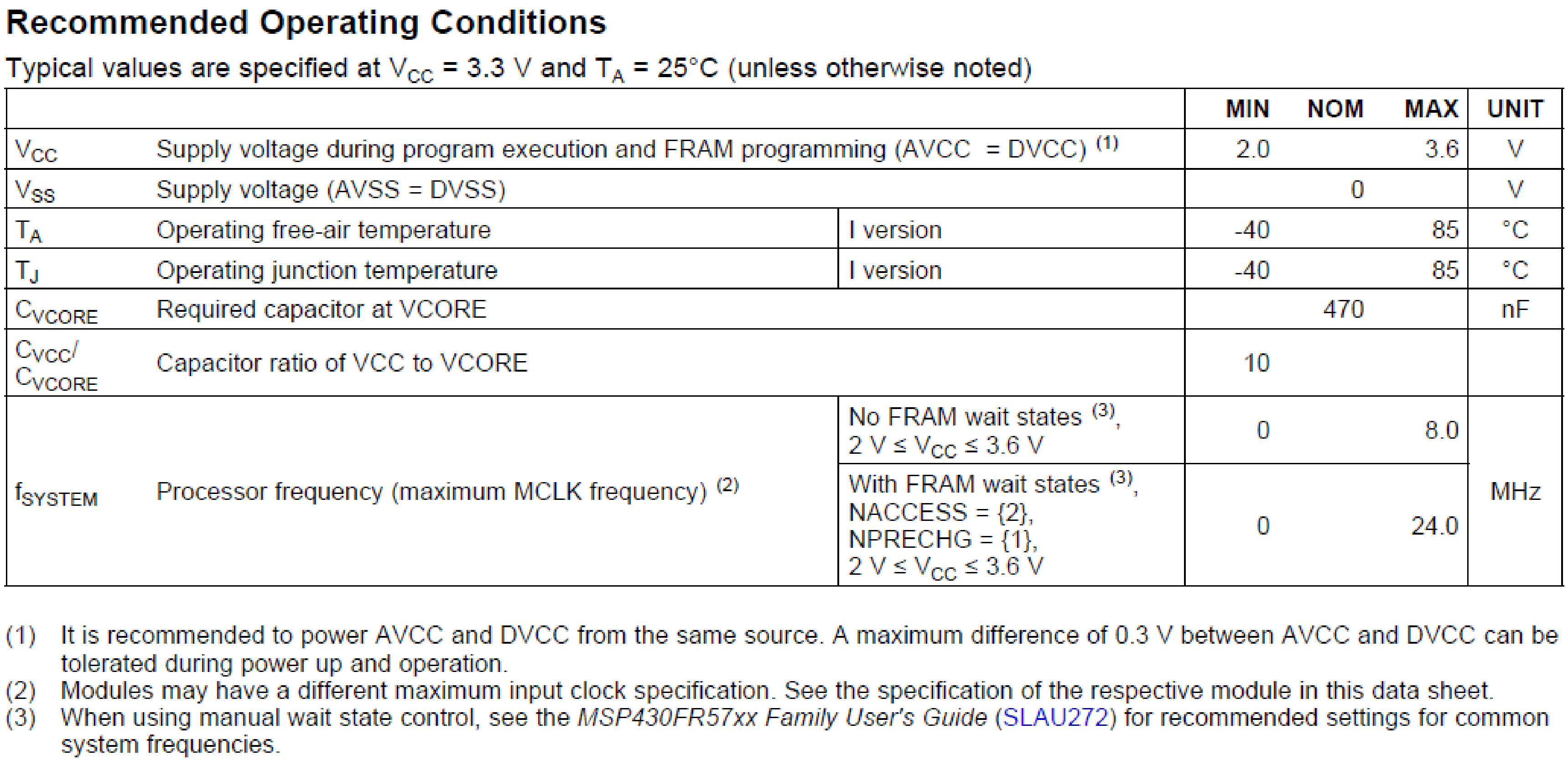
Communications





Power Supply

- What do we need (voltage and current)? Look at manual for voltage and current for microcontroller and other circuits



Recommended Operating Conditions

Typical values are specified at V_{CC} = 3.3 V and T_A = 25°C (unless otherwise noted)

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage during program execution and FRAM programming (AVCC = DVCC) [1]		2.0		3.6	V
V_{SS}	Supply voltage (AVSS = DVSS)			0		V
T_A	Operating free-air temperature	I version	-40		85	°C
T_J	Operating junction temperature	I version	-40		85	°C
C_{VCORE}	Required capacitor at VCORE			470		nF
C_{VCC}/ C_{VCORE}	Capacitor ratio of VCC to VCORE		10			
f_{SYSTEM}	Processor frequency (maximum MCLK frequency) [2]	No FRAM wait states [3], 2 V ≤ V_{CC} ≤ 3.6 V	0		8.0	MHz
		With FRAM wait states [3], NACCESS = {2}, NPRECHG = {1}, 2 V ≤ V_{CC} ≤ 3.6 V	0		24.0	

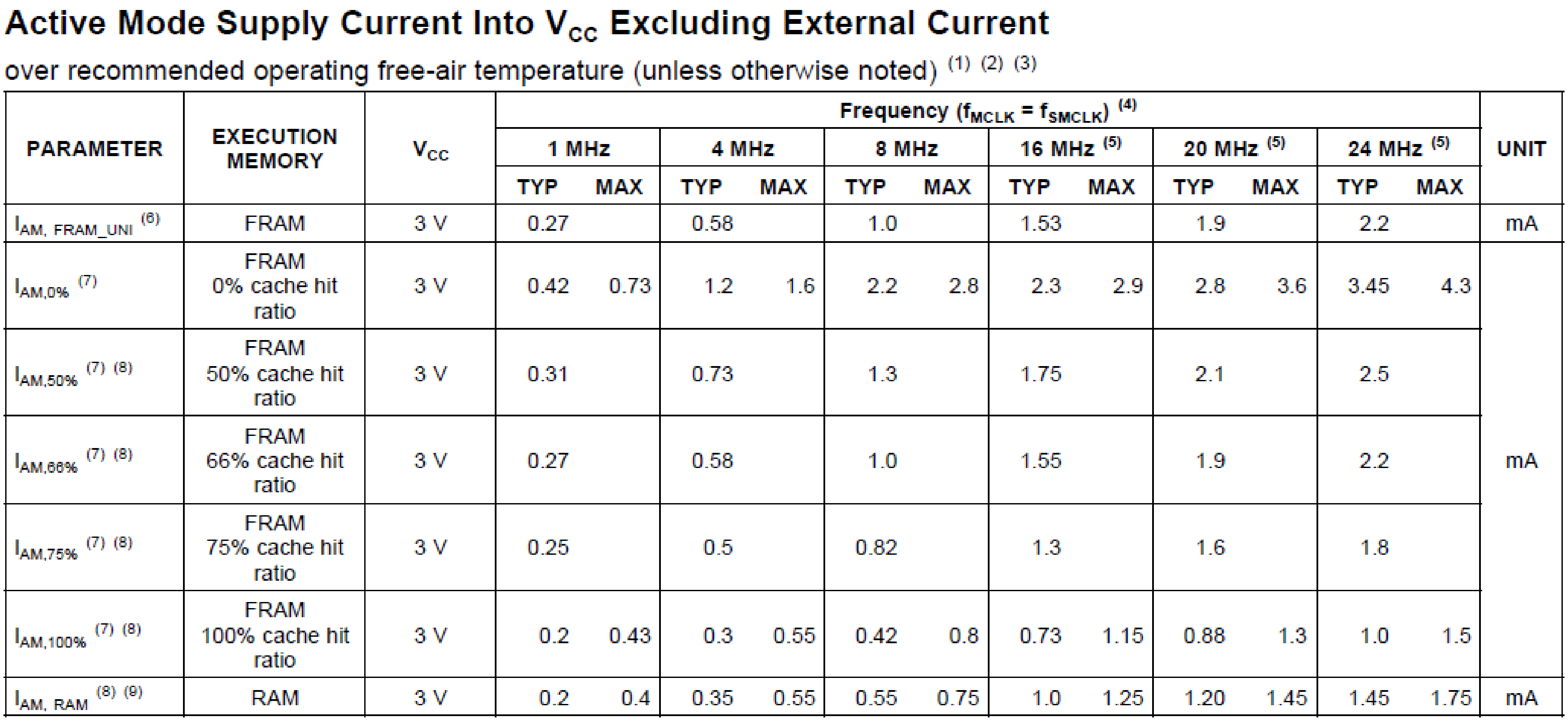
[1] It is recommended to power AVCC and DVCC from the same source. A maximum difference of 0.3 V between AVCC and DVCC can be tolerated during power up and operation.

[2] Modules may have a different maximum input clock specification. See the specification of the respective module in this data sheet.

[3] When using manual wait state control, see the *MSP430FR57xx Family User's Guide* (SLAU272) for recommended settings for common system frequencies.

Power Supply

- What do we need (voltage and current)? Look at manual for voltage and current for microcontroller and other circuits



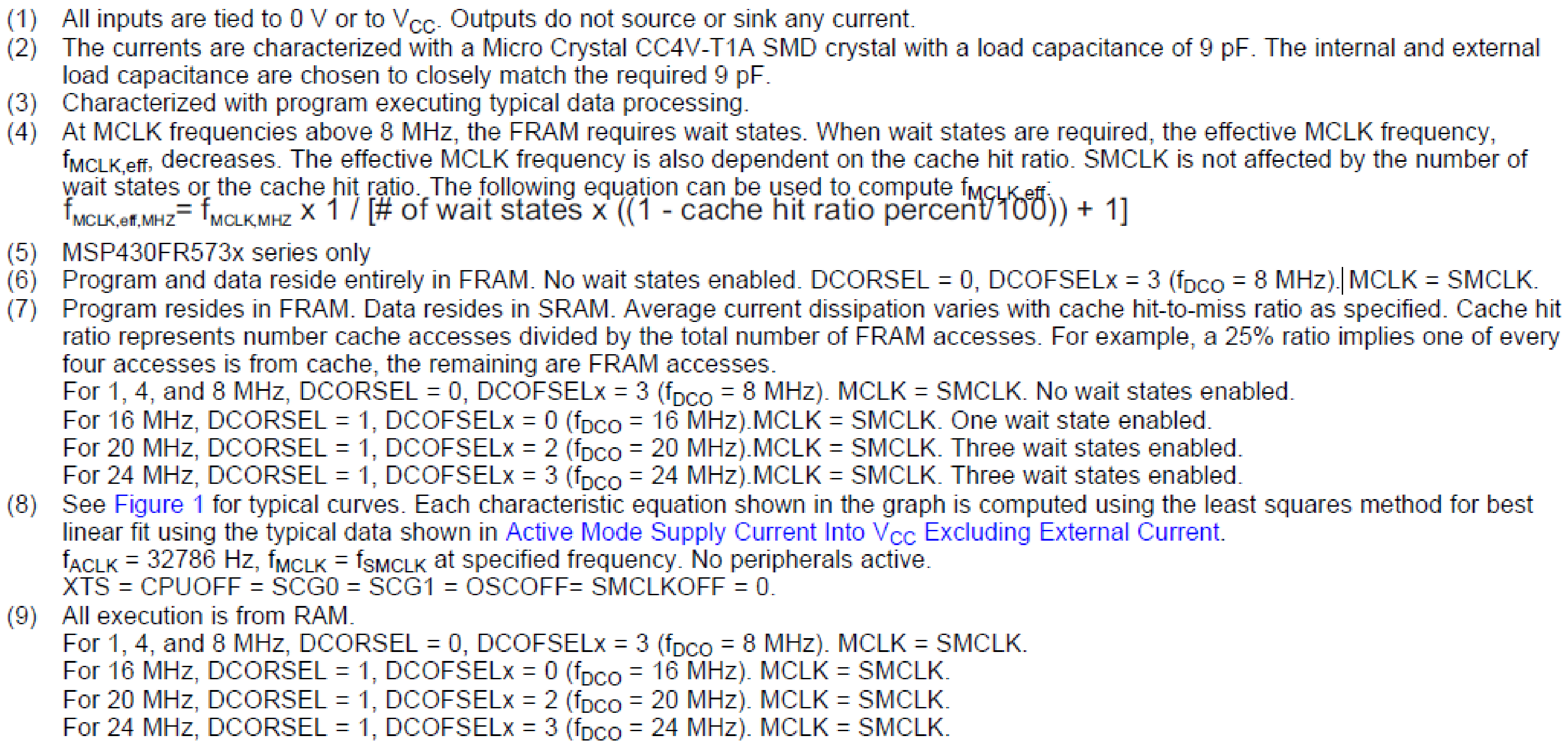
Active Mode Supply Current Into V_{CC} Excluding External Current

over recommended operating free-air temperature (unless otherwise noted) [1] [2] [3]

PARAMETER	EXECUTION MEMORY	V_{CC}	Frequency (f_{MCLK} = f_{SMCLK}) [4]												UNIT
			1 MHz		4 MHz		8 MHz		16 MHz [5]		20 MHz [5]		24 MHz [5]		
			TYP	MAX	TYP	MAX	TYP	MAX	TYP	MAX	TYP	MAX	TYP	MAX	
$I_{AM, FRAM\_UNI}$ [6]	FRAM	3 V	0.27		0.58		1.0		1.53		1.9		2.2		mA
$I_{AM,0\%}$ [7]	FRAM 0% cache hit ratio	3 V	0.42	0.73	1.2	1.6	2.2	2.8	2.3	2.9	2.8	3.6	3.45	4.3	mA
$I_{AM,50\%}$ [7] [8]	FRAM 50% cache hit ratio	3 V	0.31		0.73		1.3		1.75		2.1		2.5		
$I_{AM,66\%}$ [7] [8]	FRAM 66% cache hit ratio	3 V	0.27		0.58		1.0		1.55		1.9		2.2		
$I_{AM,75\%}$ [7] [8]	FRAM 75% cache hit ratio	3 V	0.25		0.5		0.82		1.3		1.6		1.8		
$I_{AM,100\%}$ [7] [8]	FRAM 100% cache hit ratio	3 V	0.2	0.43	0.3	0.55	0.42	0.8	0.73	1.15	0.88	1.3	1.0	1.5	
$I_{AM, RAM}$ [8] [9]	RAM	3 V	0.2	0.4	0.35	0.55	0.55	0.75	1.0	1.25	1.20	1.45	1.45	1.75	mA

Power Supply

- What do we need (voltage and current)? Look at manual for voltage and current for microcontroller and other circuits



(1) All inputs are tied to 0 V or to V_{CC}. Outputs do not source or sink any current.

(2) The currents are characterized with a Micro Crystal CC4V-T1A SMD crystal with a load capacitance of 9 pF. The internal and external load capacitance are chosen to closely match the required 9 pF.

(3) Characterized with program executing typical data processing.

(4) At MCLK frequencies above 8 MHz, the FRAM requires wait states. When wait states are required, the effective MCLK frequency, $f_{MCLK,eff}$, decreases. The effective MCLK frequency is also dependent on the cache hit ratio. SMCLK is not affected by the number of wait states or the cache hit ratio. The following equation can be used to compute $f_{MCLK,eff}$:
$f_{MCLK,eff,MHZ}= f_{MCLK,MHZ}$ x 1 / [# of wait states x ((1 - cache hit ratio percent/100)) + 1]

(5) MSP430FR573x series only

(6) Program and data reside entirely in FRAM. No wait states enabled. DCORSEL = 0, DCOFSELx = 3 (f_{DCO} = 8 MHz). MCLK = SMCLK.

(7) Program resides in FRAM. Data resides in SRAM. Average current dissipation varies with cache hit-to-miss ratio as specified. Cache hit ratio represents number cache accesses divided by the total number of FRAM accesses. For example, a 25% ratio implies one of every four accesses is from cache, the remaining are FRAM accesses.
For 1, 4, and 8 MHz, DCORSEL = 0, DCOFSELx = 3 (f_{DCO} = 8 MHz). MCLK = SMCLK. No wait states enabled.
For 16 MHz, DCORSEL = 1, DCOFSELx = 0 (f_{DCO} = 16 MHz).MCLK = SMCLK. One wait state enabled.
For 20 MHz, DCORSEL = 1, DCOFSELx = 2 (f_{DCO} = 20 MHz).MCLK = SMCLK. Three wait states enabled.
For 24 MHz, DCORSEL = 1, DCOFSELx = 3 (f_{DCO} = 24 MHz).MCLK = SMCLK. Three wait states enabled.

(8) See Figure 1 for typical curves. Each characteristic equation shown in the graph is computed using the least squares method for best linear fit using the typical data shown in Active Mode Supply Current Into V_{CC} Excluding External Current.
f_{ACLK} = 32786 Hz, f_{MCLK} = f_{SMCLK} at specified frequency. No peripherals active.
XTS = CPUOFF = SCG0 = SCG1 = OSCOFF= SMCLKOFF = 0.

(9) All execution is from RAM.
For 1, 4, and 8 MHz, DCORSEL = 0, DCOFSELx = 3 (f_{DCO} = 8 MHz). MCLK = SMCLK.
For 16 MHz, DCORSEL = 1, DCOFSELx = 0 (f_{DCO} = 16 MHz). MCLK = SMCLK.
For 20 MHz, DCORSEL = 1, DCOFSELx = 2 (f_{DCO} = 20 MHz). MCLK = SMCLK.
For 24 MHz, DCORSEL = 1, DCOFSELx = 3 (f_{DCO} = 24 MHz). MCLK = SMCLK.

Power Supply

- Where do we get the power?
- Plug into wall
  - Wall - Need to drop 120 VAC to Low Voltage VDC
    - Transformer-based
    - Switching
- Use batteries
  - Battery Voltage changes with discharge
    - Conversion from variable voltage to constant voltage
      - Regulator
      - Switcher

Clock Signal Generator

- Why? To make the logic run!
- The clock system module supports low system cost and low power consumption.
- Using three internal clock signals, the user can select the best balance of performance and low power consumption.
- The clock module can be configured to operate without any external components, with one or two external crystals, or with resonators, under full software control.

Clock Signal Generator

- The clock system module includes several possible clock sources:
  - XT1CLK: Low-frequency or high-frequency oscillator that can be used with low-frequency 32.768-KHz watch crystals, standard crystals, resonators, or external clock sources in the 4 MHz to 24 MHz range. When optional XT2 is present, the XT1 high-frequency mode may or may not be available, depending on the device configuration.
  - VLOCLK: Internal very-low-power low-frequency oscillator with 10-kHz typical frequency
  - DCOCLK: Internal digitally controlled oscillator (DCO) with three selectable fixed frequencies
  - XT2CLK: Optional high-frequency oscillator that can be used with standard crystals, resonators, or external clock sources in the 4 MHz to 24 MHz range.

Clock Signal Generator

- Four system clock signals are available from the clock module:
  - ACLK: Auxiliary clock. The ACLK is software selectable as XT1CLK, VLOCLK, DCOCLK, and when available, XT2CLK. ACLK can be divided by 1, 2, 4, 8, 16, or 32. ACLK is software selectable by individual peripheral modules.
  - MCLK: Master clock. MCLK is software selectable as XT1CLK, VLOCLK, DCOCLK, and when available, XT2CLK. MCLK can be divided by 1, 2, 4, 8, 16, or 32. MCLK is used by the CPU and system.
  - SMCLK: Subsystem master clock. SMCLK is software selectable as XT1CLK, VLOCLK, DCOCLK, and when available, XT2CLK. SMCLK is software selectable by individual peripheral modules.
  - MODCLK: Module clock. MODCLK is used by various peripheral modules and is sourced by MODOSC.

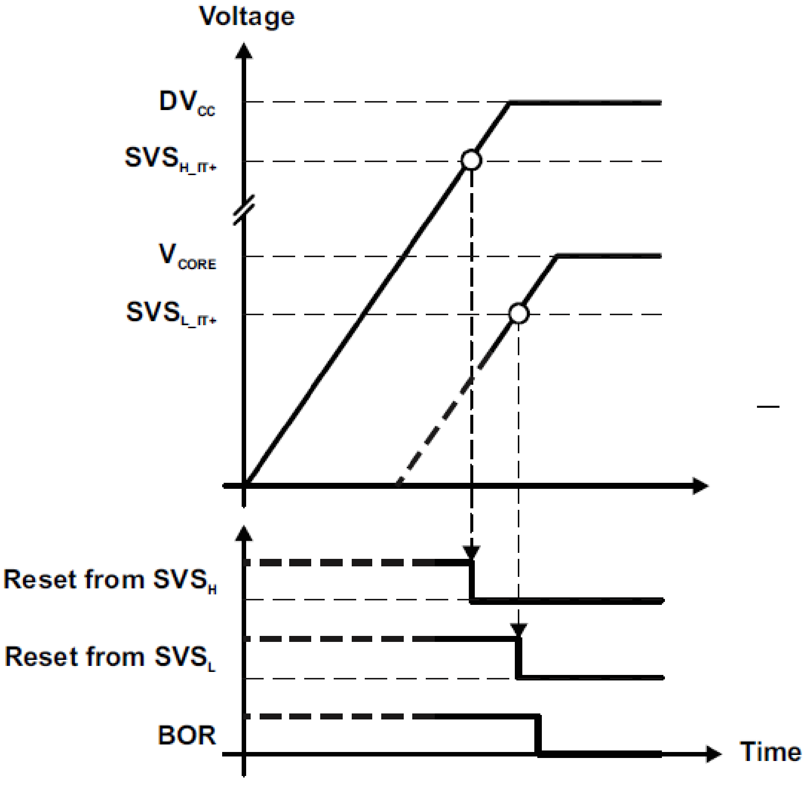
Reset Controller

- Why? So the processor starts off in a predictable state (e.g. program start address, operating modes…)
  - Reset processor whenever
    1. Power supply voltage drops below a threshold
    2. Something catastrophic happens

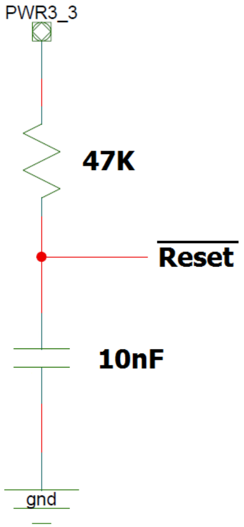
- **Power Management Module (PMM)**
  - The PMM includes an integrated voltage regulator that supplies the core voltage to the device. The PMM also includes supply voltage supervisor (SVS) and brownout protection. The brownout circuit is implemented to provide the proper internal reset signal to the device during power-on and power-off. The SVS circuitry detects if the supply voltage drops below a user-selectable safe level. SVS circuitry is available on the primary and core supplies.

Reset Controller

- **System Module (SYS)**
  - The SYS module handles many of the system functions within the device. These include power-on reset (POR) and power-up clear (PUC) handling, NMI source selection and management, reset interrupt vector generators, bootstrap loader entry mechanisms, and configuration management (device descriptors).
  - When the device is powering up, the SVS_H and SVS functions are enabled by default. Initially, DV_{CC} is low, and therefore the PMM holds the device in BOR reset. When both the SVS and SVS_L levels are met, the reset is released.



PMM Action at Device Power-Up



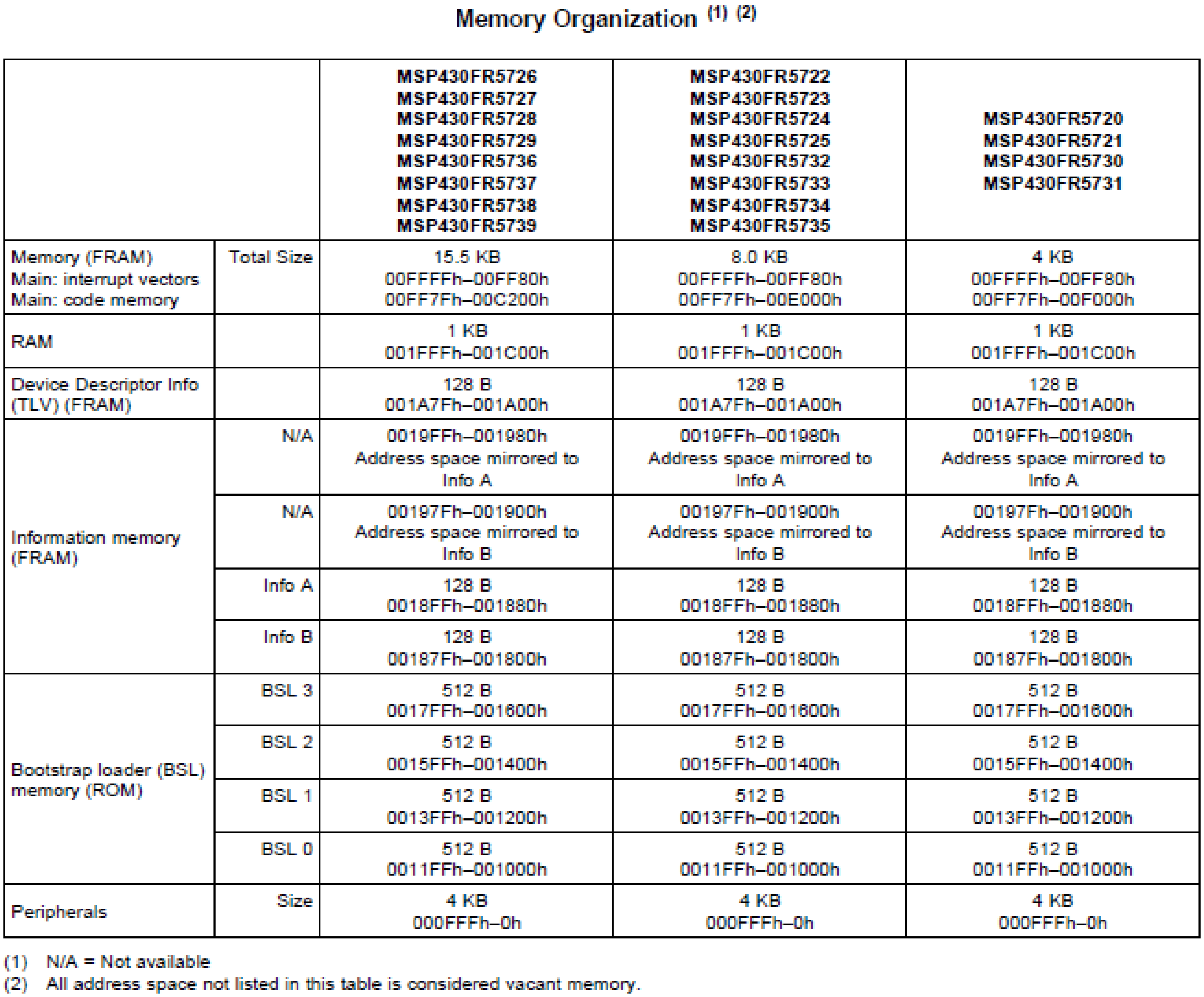
Result of Reset

- System control registers are initialized to predefined values
- Listed in full in **MSP430FR57xx Family User's Guide**
    - Texas Instruments Literature Number: SLAU272B

Memory

- Internal Memory / External Memory
  - Internal Memory
    - MAP contains more than just memory
    - Broken into system function registers
    - Program Store / Data
  - External memory can be accessed in Various ways
    - Serial Access [SPI / I2C]
    - Byte Access using 1 port
    - Word Access using 2 ports

Memory



Memory Organization [1] [2]

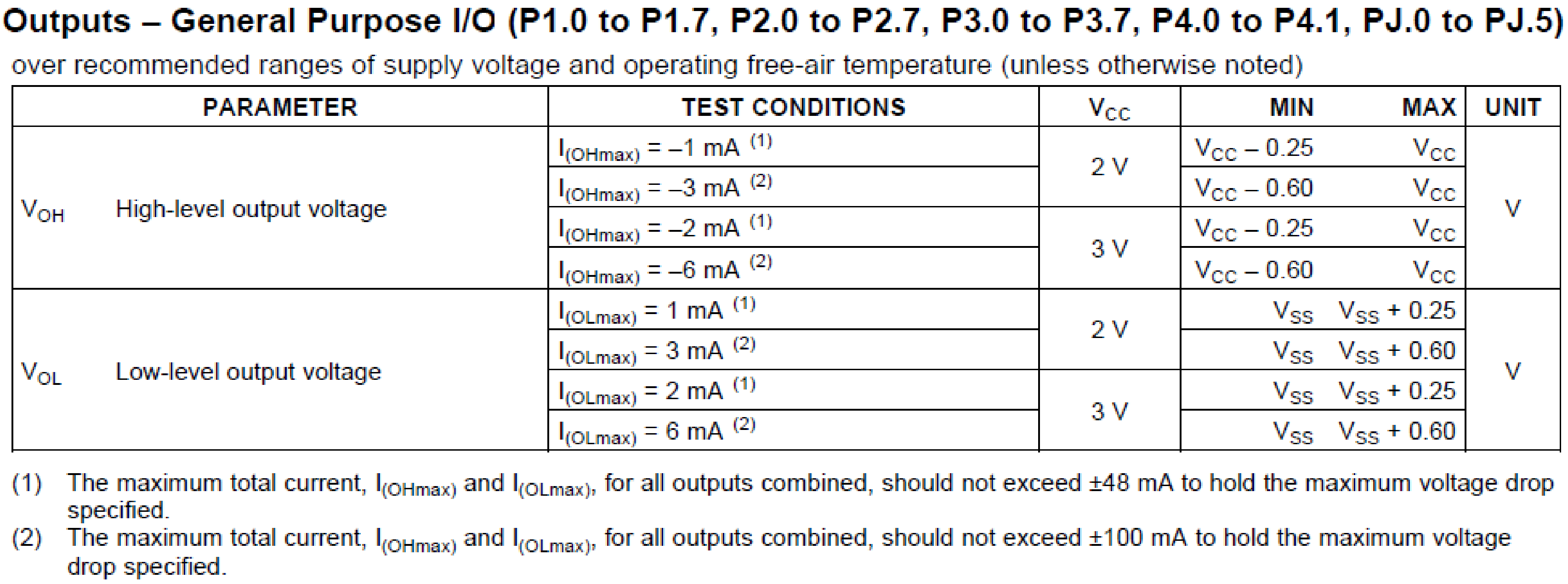
		MSP430FR5726 MSP430FR5727 MSP430FR5728 MSP430FR5729 MSP430FR5736 MSP430FR5737 MSP430FR5738 MSP430FR5739	MSP430FR5722 MSP430FR5723 MSP430FR5724 MSP430FR5725 MSP430FR5732 MSP430FR5733 MSP430FR5734 MSP430FR5735	MSP430FR5720 MSP430FR5721 MSP430FR5730 MSP430FR5731
Memory (FRAM) Main: interrupt vectors Main: code memory	Total Size	15.5 KB 00FFFFh–00FF80h 00FF7Fh–00C200h	8.0 KB 00FFFFh–00FF80h 00FF7Fh–00E000h	4 KB 00FFFFh–00FF80h 00FF7Fh–00F000h
RAM		1 KB 001FFFh–001C00h	1 KB 001FFFh–001C00h	1 KB 001FFFh–001C00h
Device Descriptor Info (TLV) (FRAM)		128 B 001A7Fh–001A00h	128 B 001A7Fh–001A00h	128 B 001A7Fh–001A00h
Information memory (FRAM)	N/A	0019FFh–001980h Address space mirrored to Info A	0019FFh–001980h Address space mirrored to Info A	0019FFh–001980h Address space mirrored to Info A
	N/A	00197Fh–001900h Address space mirrored to Info B	00197Fh–001900h Address space mirrored to Info B	00197Fh–001900h Address space mirrored to Info B
	Info A	128 B 0018FFh–001880h	128 B 0018FFh–001880h	128 B 0018FFh–001880h
	Info B	128 B 00187Fh–001800h	128 B 00187Fh–001800h	128 B 00187Fh–001800h
Bootstrap loader (BSL) memory (ROM)	BSL 3	512 B 0017FFh–001600h	512 B 0017FFh–001600h	512 B 0017FFh–001600h
	BSL 2	512 B 0015FFh–001400h	512 B 0015FFh–001400h	512 B 0015FFh–001400h
	BSL 1	512 B 0013FFh–001200h	512 B 0013FFh–001200h	512 B 0013FFh–001200h
	BSL 0	512 B 0011FFh–001000h	512 B 0011FFh–001000h	512 B 0011FFh–001000h
Peripherals	Size	4 KB 000FFFh–0h	4 KB 000FFFh–0h	4 KB 000FFFh–0h

[1] N/A = Not available

[2] All address space not listed in this table is considered vacant memory.

Digital Interfacing

- Communicate with Simple Sensors (Switches), Actuators (LEDs, Motors) and other digital logic (Real-Time Clock, GPS)
- Problem: Logic level outputs are often not what sensors provide or actuators need
  - Voltage may be out of range for inputs



Outputs – General Purpose I/O (P1.0 to P1.7, P2.0 to P2.7, P3.0 to P3.7, P4.0 to P4.1, PJ.0 to PJ.5)

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	MAX	UNIT
V_{OH}	High-level output voltage	$I_{(OHmax)}$ = –1 mA [1]	2 V	V_{CC} – 0.25	V_{CC}	V
		$I_{(OHmax)}$ = –3 mA [2]		V_{CC} – 0.60	V_{CC}	
		$I_{(OHmax)}$ = –2 mA [1]	3 V	V_{CC} – 0.25	V_{CC}	
		$I_{(OHmax)}$ = –6 mA [2]		V_{CC} – 0.60	V_{CC}	
V_{OL}	Low-level output voltage	$I_{(OLmax)}$ = 1 mA [1]	2 V	V_{SS}	V_{SS} + 0.25	V
		$I_{(OLmax)}$ = 3 mA [2]		V_{SS}	V_{SS} + 0.60	
		$I_{(OLmax)}$ = 2 mA [1]	3 V	V_{SS}	V_{SS} + 0.25	
		$I_{(OLmax)}$ = 6 mA [2]		V_{SS}	V_{SS} + 0.60	

[1] The maximum total current, $I_{(OHmax)}$ and $I_{(OLmax)}$, for all outputs combined, should not exceed ±48 mA to hold the maximum voltage drop specified.

[2] The maximum total current, $I_{(OHmax)}$ and $I_{(OLmax)}$, for all outputs combined, should not exceed ±100 mA to hold the maximum voltage drop specified.

Digital Interfacing

- Solution covered in **General Purpose Digital I/O** class

Analog Interfacing

- Communicate with *analog* sensors and actuators
  - Many devices use analog signals, not digital
    - microphone, thermometer, speaker, video …
- Input: Analog to Digital Converter (ADC)
  - Produces multibit binary number AD = $(2^{Nbits}-1) * V_{In}/V_{Ref}$
  - Nbits = 10 for our chip
  - Takes a finite amount of time (conversion speed),
- Output: Digital to Analog Converter (DAC) [not on FRAM]
  - Converts n-bit binary number to equivalent voltage
$V_{Out} = V_{Ref} * n/255$
  - Typically need to buffer this signal to increase drive current
- Solution covered in detail in **Analog I/O Peripherals**

More Analog Interfacing

- Can use a comparator to detect when a voltage exceeds a given threshold
- Some microcontrollers have built-in comparators

Communications

- Communicate with smart components on networks, other processors and devices (e.g. GPS)
- Use dedicated protocol controller chips which translate bytes of data into streams of bits with extra features for
  - Error detection and/or correction
  - Addressing
  - Requesting data
  - Message content, format and priority
- Solutions covered in more detail in **Serial Communications** class

Miscellaneous

- Leftover Pins
  - Port pins: either
    - Configure for input and connect directly to V_{SS}
    - Configure for output and leave disconnected
  - X_{OUT} (if using external clock on X_{IN}): Leave disconnected
  - AV_{CC} : Connect to V_{CC}
  - AV_{SS} : Connect to V_{SS}.
  - V_{REF}: ADC reference voltage: Connect to V_{CC} unless other reference level prefered
- Connect a bypass capacitor (>= 0.1 µF) between V_{CC} and V_{SS} pins close to the MCU for noise. De-couple the processor power from the power source.