Introduction to Embedded Systems (CSE 337/537)

Amarjeet Singh

January 10, 2011

Partly adopted from EE202A, UCLA Slides by Mani Srivastava

Logistics

- Website updated - http://www.iiitd.edu.in/~amarjeet/EmSys.html
- Orders for microcontroller kits
- Registration to course mailing list
  - Those on the list will be assumed to be registered to the course
  - I will add them to the blog - start contributing soon: http://cse537-2011.blogspot.com/
  - Will use as the sole mode of communication for class purpose
- Introductory programming class on Saturday
  - Preferable time?

Embedded Systems - Typical Challenges

- Limited processing speed, storage capacity and communication bandwidth
  - Substantial processing capability in aggregate
- Limited energy
  - But must operate for long periods of time with wireless communication
- Close interaction with environments
  - Indeterminacy in execution - e.g. waiting for events from multiple sources
  - Physical environment is delay intolerant - can't put it on wait with an hour glass icon!
- Multiple concurrent activities - sensing, processing, communication
  - Handling timing constraints are crucial

Embedded Systems - Timing Constraints

- Time is a major discriminator relative to general purpose computing systems
  - Held to higher reliability and predictability standard
    - You do not expect your TV to crash and reboot!
  - Correctness of system depends not only on the logical result of the computation but also on the time at which the results are produced
  - The reaction of the system to external events must occur during their evolution
    - System's internal time must be measured using the same time scale as is used for measuring the time in the environment (external time)

Embedded Systems - Timing Constraints

- Real time != Fast
  - Fast is relative, and does not capture the real constraint
  - Concept of time is not an intrinsic property of an embedded system
  - Rather, it is strictly related to environment in which the system operates
  - E.g. doesn't make sense to design a flight control system without considering the timing characteristics of the aircraft

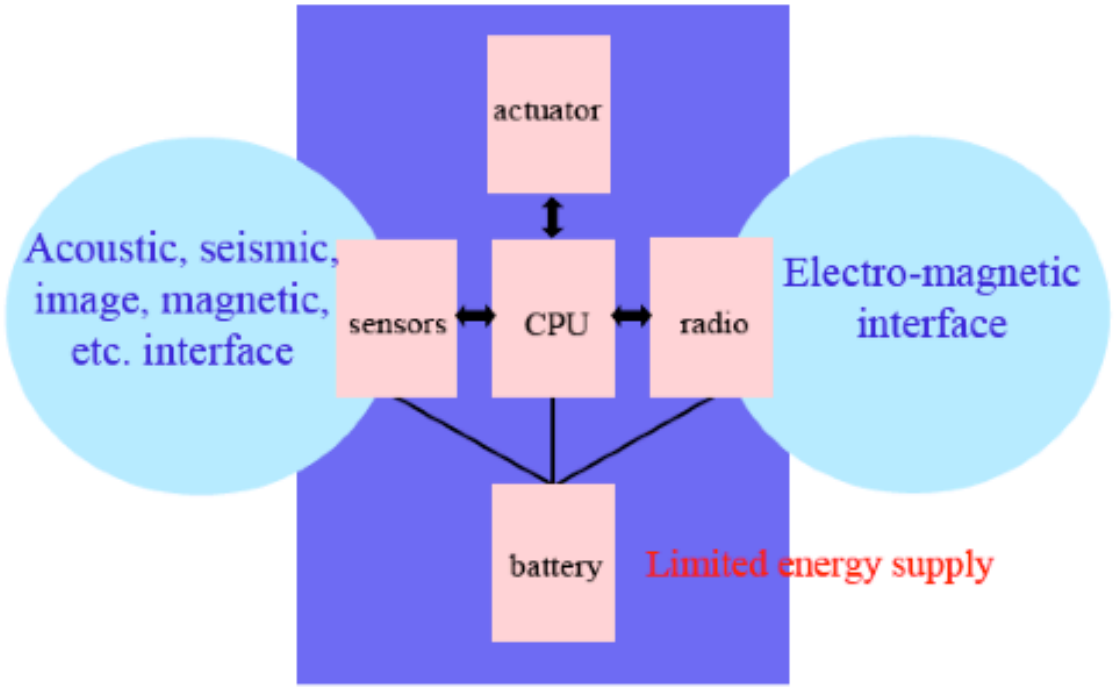
- Erroneous belief: faster hardware will solve the problem
  - Faster hardware improves system throughput but does not imply that individual timing constraints will be met

Embedded Systems - Timing Constraints

- Fast vs. predictability: However short the average response time may be, without a scientific methodology one cannot guarantee that individual timing requirements are met under all possible circumstances
  - There was a man who drowned crossing a stream with an average depth of six inches! :)

- Worst case performance is often difficult to compute correctly
  - Complex processor architecture (pipelines, memory hierarchy, and parallelism)
  - Stochastic techniques used by compilers

Example Embedded System



Often a far cry from general purpose computers

- CPUs
  - 8 b / 8 Mhz to 32 b / 400 MHz
- Storage
  - 2 KB - 64 MB RAM
  - 1 MB - 1 GB Flash
  - No disk
- Radio
  - 38 kbps / 10 m to 802.11b / 100 m
- Power
  - 3 mW - 3W

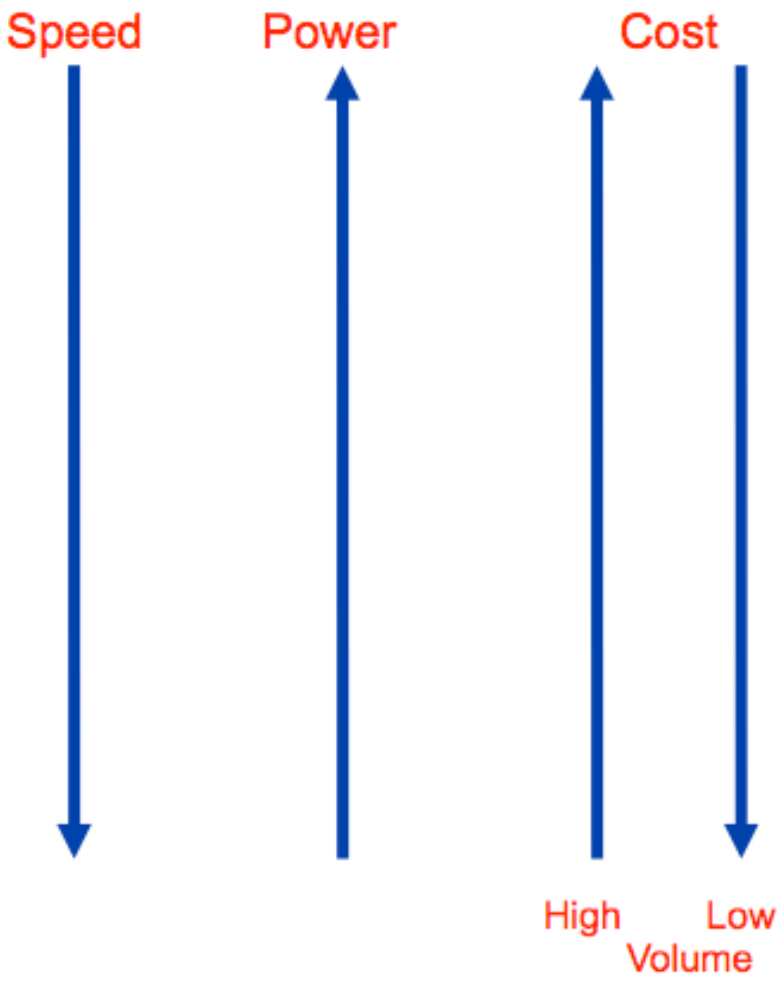
Embedded Platform Design Space

- Processor
  - Energy, computation
- Network Interface
  - Wired, RF, Acoustic, Optical
  - Energy, range, bandwidth, interference-robustness
- Energy supply
  - Wired, battery, scavenging
  - Lifetime, size
- User Interface
  - Type, energy, complexity
- Sensing
  - Type, energy, range, accuracy, resolution, frequency, fidelity
- Actuation
  - Type, energy, range
- Storage
  - Energy, capacity, bandwidth
- Packaging
  - Form factor, weight, weather-proof
- Overall
  - Cost

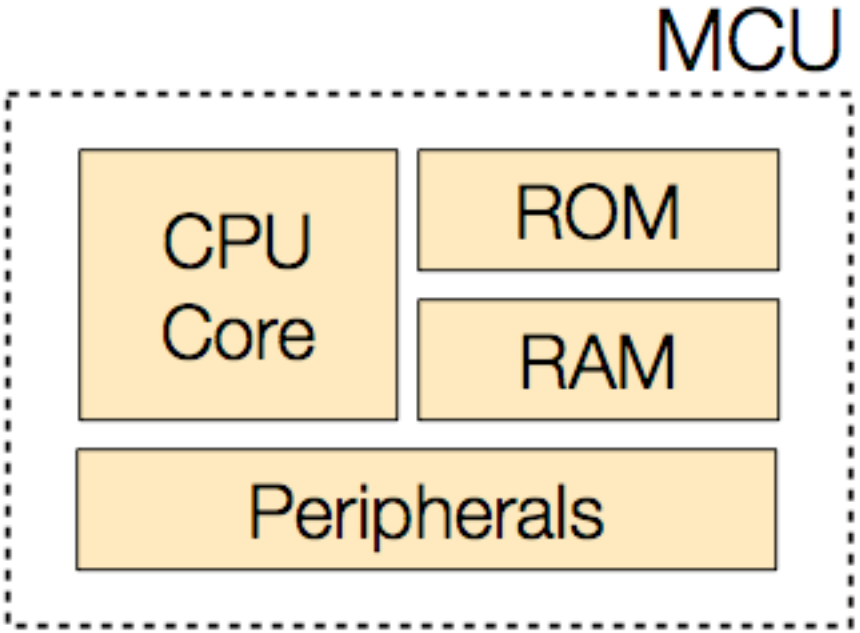
Key Issues are **Size, Energy** and **Performance**

Processing Choices

- Microprocessors
- Domain-specific processors
  - DSP
  - Network processors
  - Microcontrollers
- ASIPs
- Reconfigurable SoC
- FPGA
- Gatearray
- ASIC



Microcontroller Architecture



- Chip vendors either develop their own CPU Core or License IP
- Specific chips usually targeted towards a small set of applications

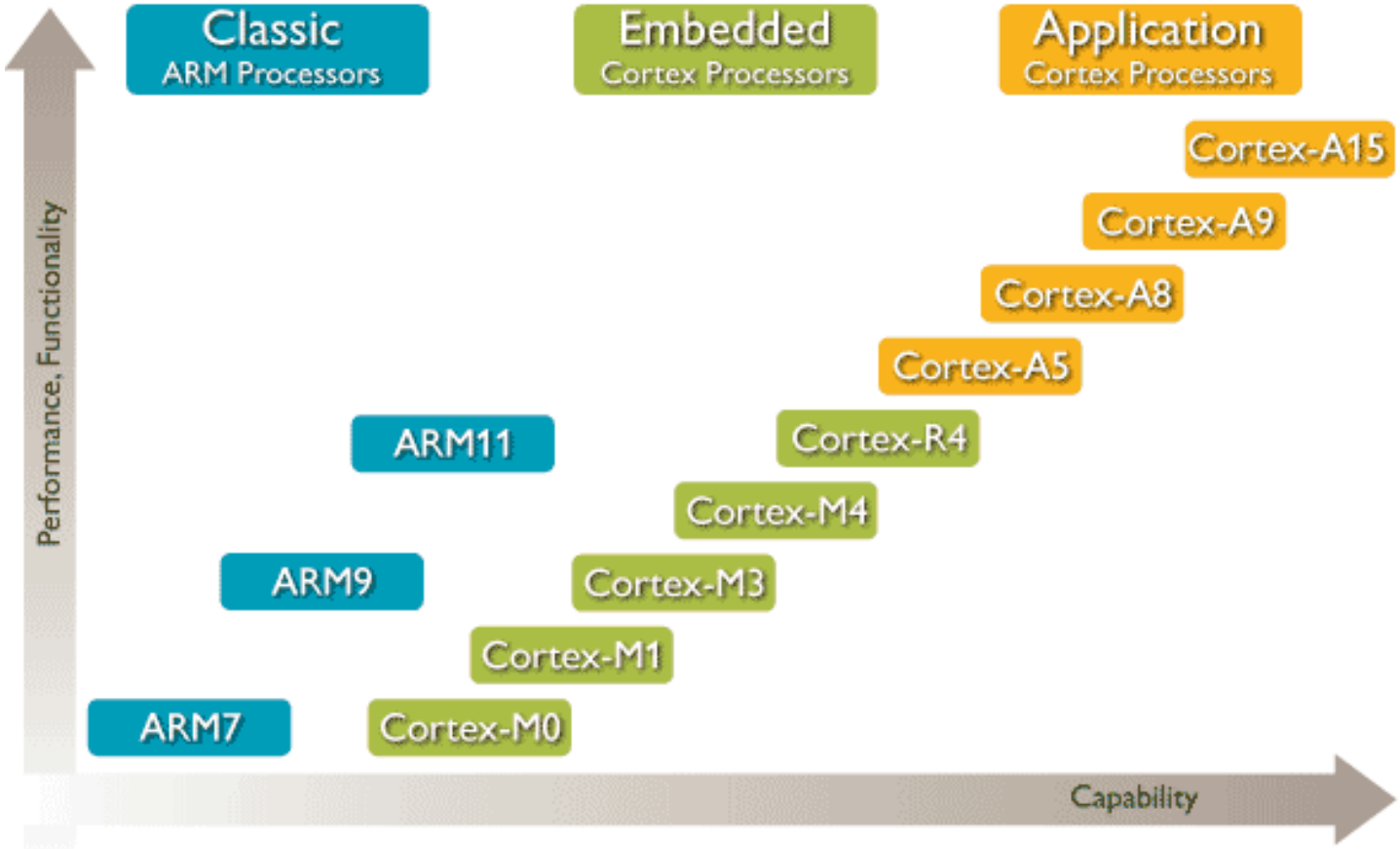
Microcontrollers

- Different from the regular desktop CPU
  - Smaller in size
  - Reduced instruction set
  - Less power consumption
  - Lower frequencies

- Within microcontrollers, there is a huge variation

Bus Width	CPU Speeds	RAM	ROM
8-bit	1-8 MHz	128-1K	512 to 10K
16-bit	4-25 Mhz	1K to 10K	10K to 128K
32-bit	10-100 Mhz	10K to 64M	128K to 512M

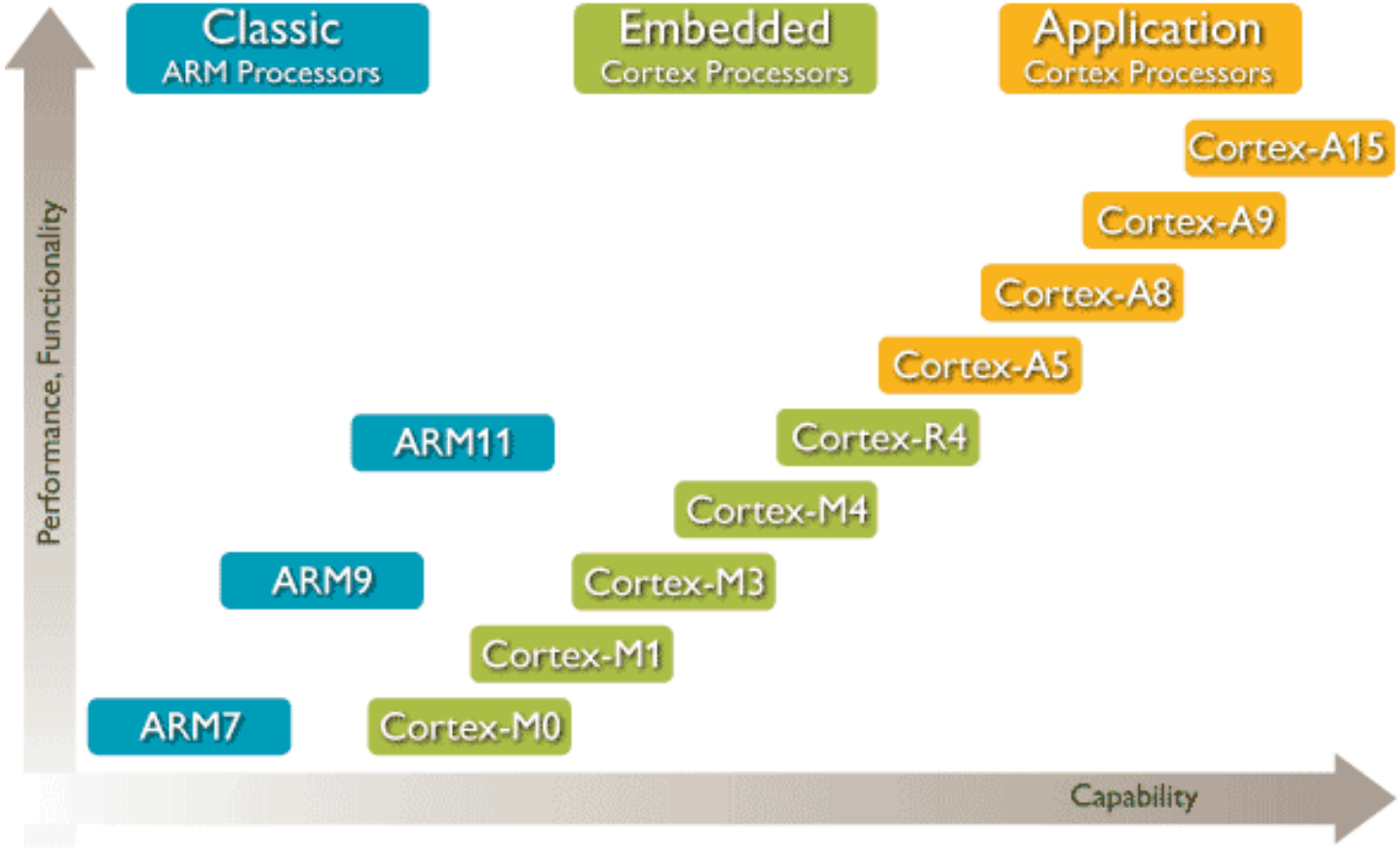
ARM Processors - 32 bit family



- Cortex A Series: High performance of up to 2 GHz +
  - Applications include smartphones, netbooks, eBook Readers Digital TV, Home Gateways

Source: www.arm.com

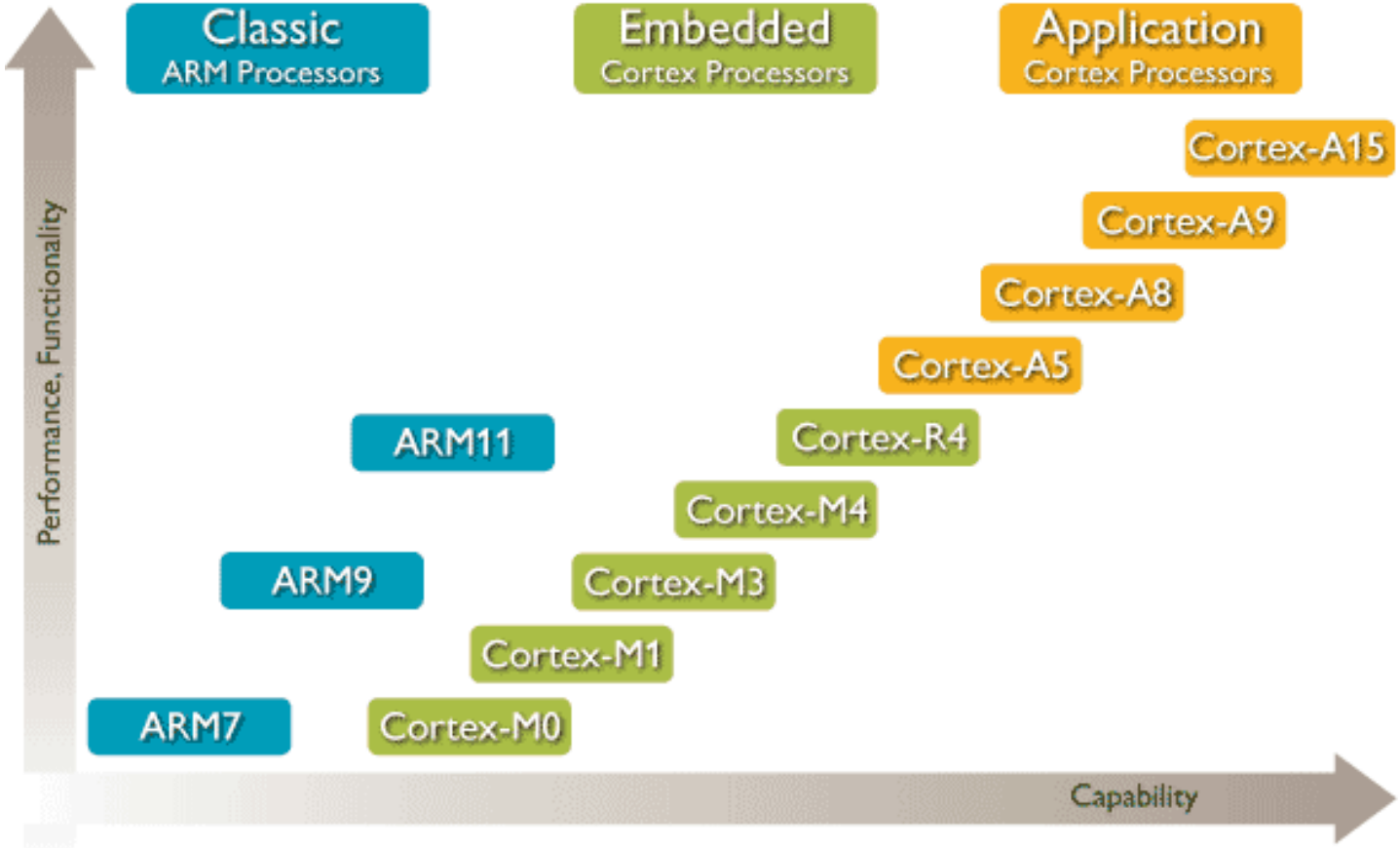
ARM Processors - 32 bit family



- Cortex R Series: Real time applications
  - Applications include automotive braking systems, mass storage controller, networking, printing

Source: www.arm.com

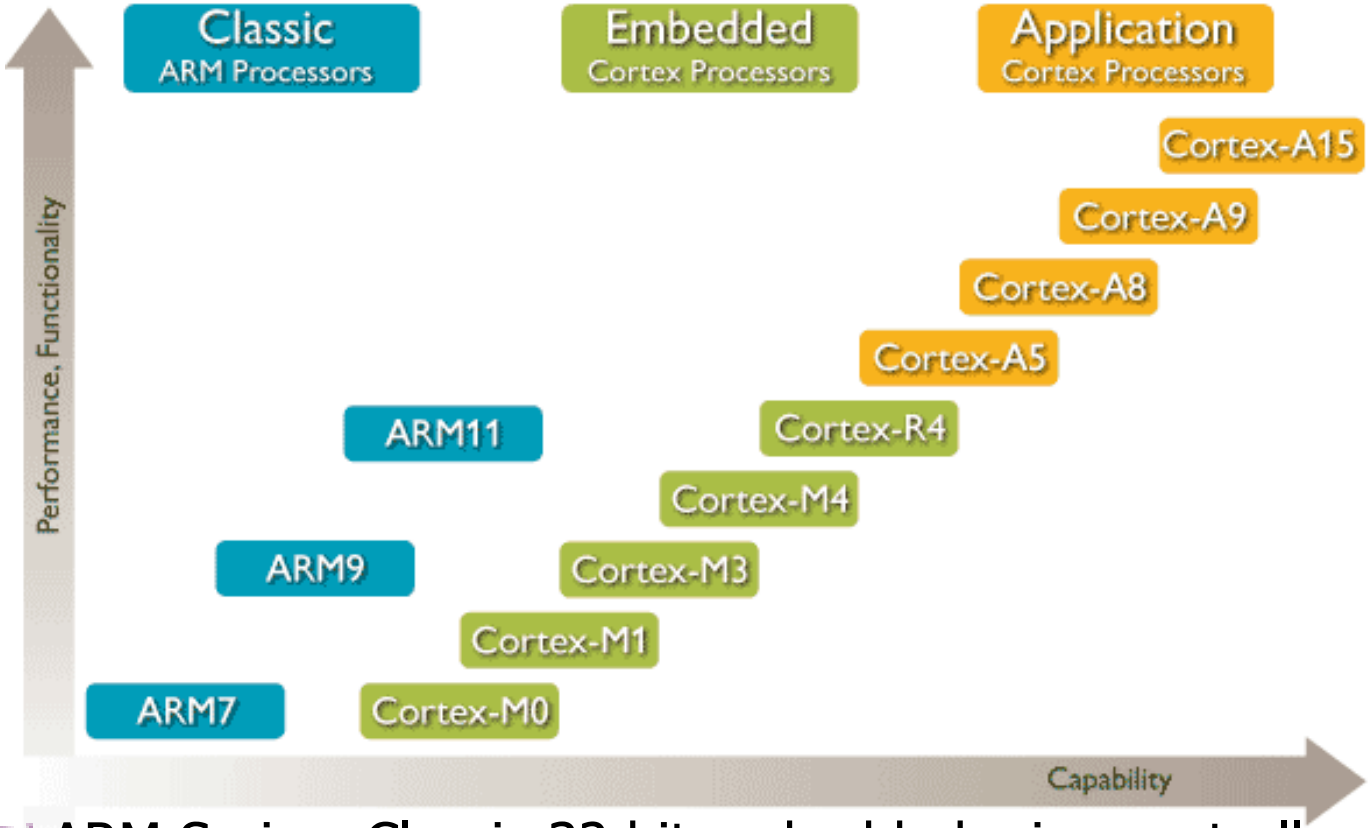
ARM Processors - 32 bit family



- Cortex M Series: Embedded microcontrollers
  - Applications include mixed signal devices, smart sensors, automotive body electronics and airbags

Source: www.arm.com

ARM Processors - 32 bit family

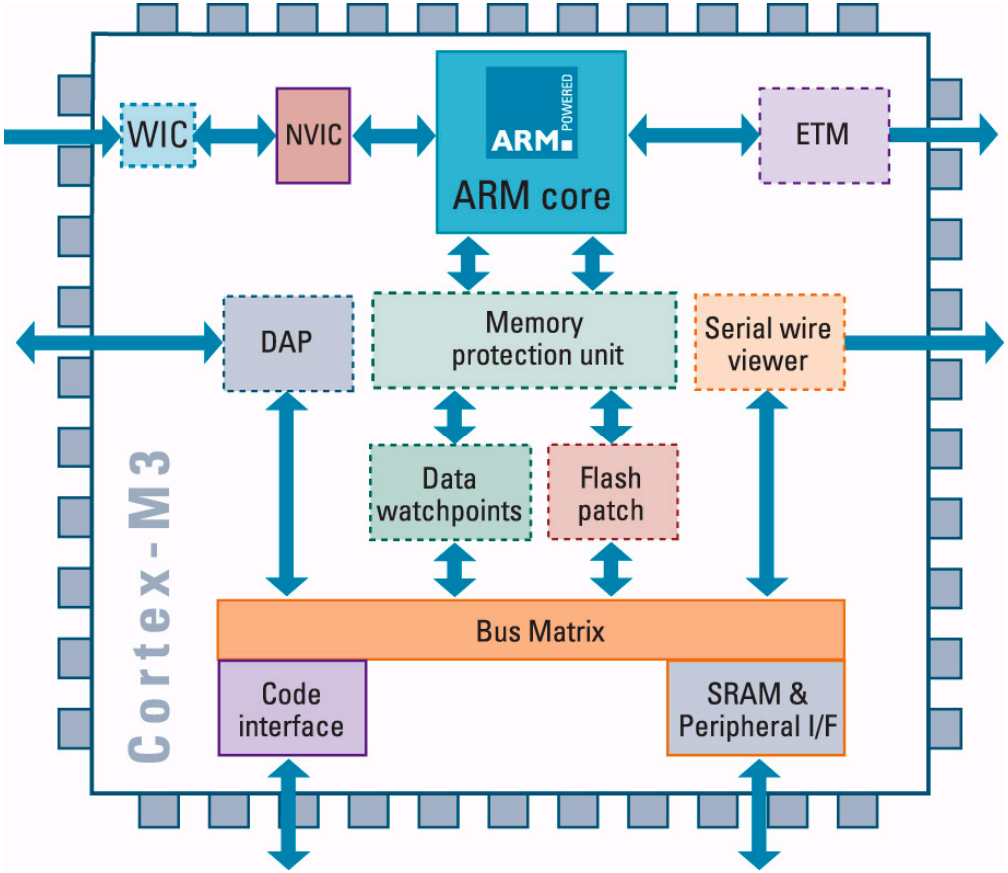


- ARM Series: Classic 32 bit embedded microcontrollers
  - ARM7 discontinued and replaced with CortexM series
  - ARM9 can be extended to CortexR and Cortex-A series
  - More than 15 billion shipped so far!

Source: www.arm.com

Cortex M3

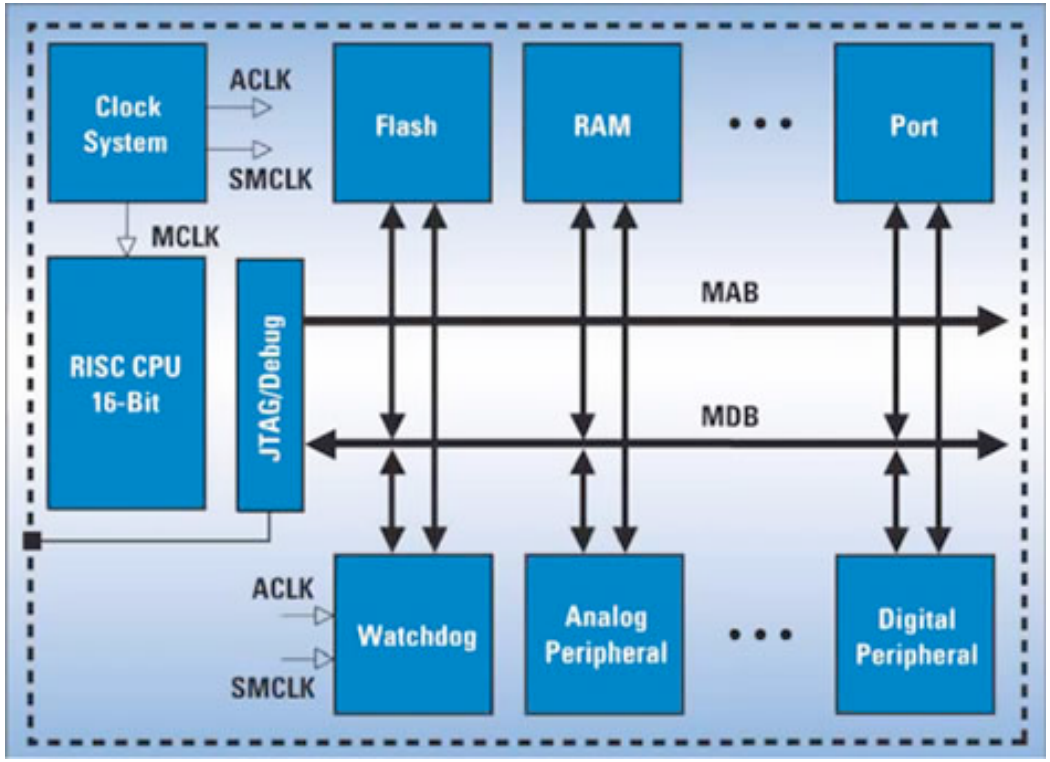
- RISC Processor Core
  - 32-bit CPU
- Executes Thumb-2 ISA
  - Blend of 16/32 bit instructions
- Low power modes
  - Multiple power domains
- Nested Vector Interrupt Controller (NVIC)
  - Vector table containing address of the function to be executed for a particular interrupt handler



Source: www.arm.com

TI-MSP430 - 16 bit family

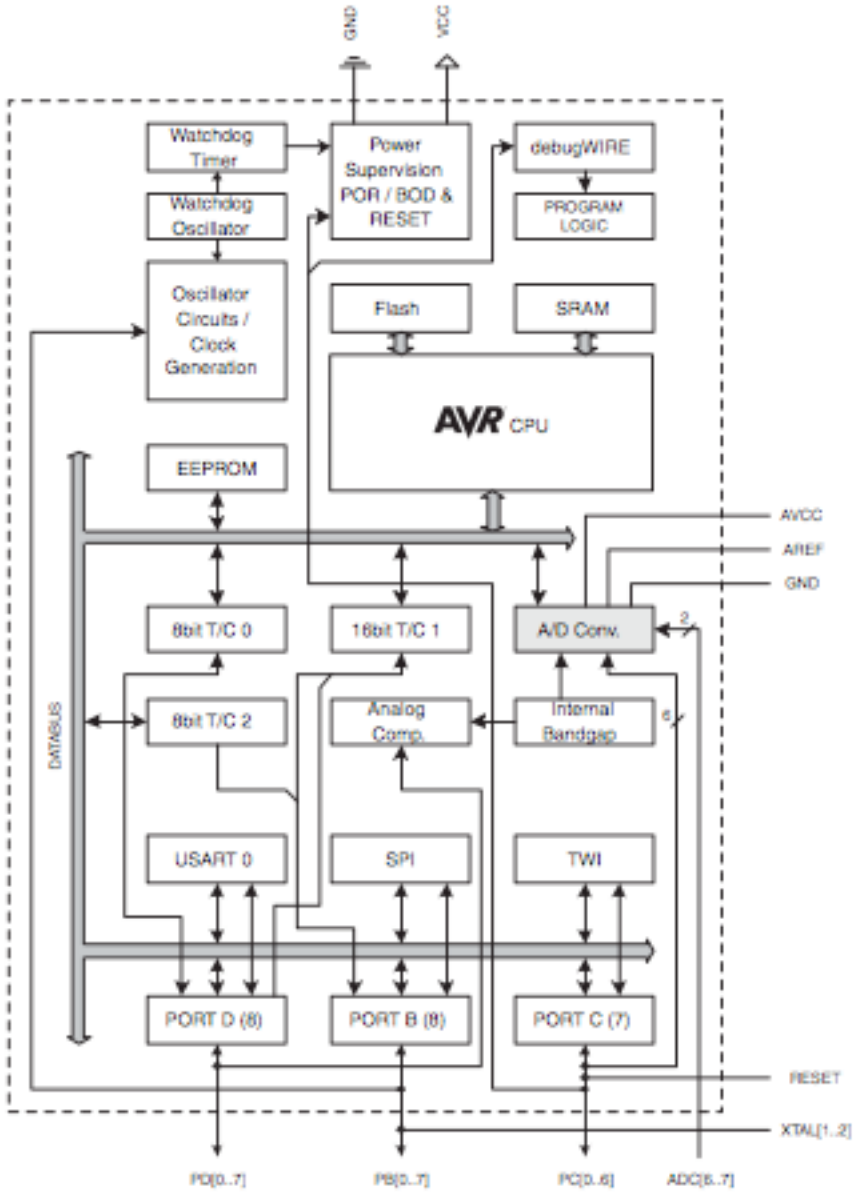
- RISC Processor Core
  - 16-bit CPU
  - Up to 25 MHz clock
- Ultra Low Power Consumption
  - ~200 uA/MHz - Active
  - ~1.5 uA - Standby
  - 1.1 uA - Off mode
  - 0.18 uA - Shutdown
- Wake up from standby in less than 5 uS
- Over 200 devices in MSP family



Source: www.ti.com

Atmel ATMEGA328p - 8 bit family

- RISC Processor Core
  - 8-bit CPU
  - Upto 20 MHz
- Low power consumption (@ 1 MHz)
  - Active mode - 0.2 mA
  - Power down mode - 0.1 uA
  - Power save mode - 0.75 uA
- 32x8 General Purpose Registers
- 4/8/16/32 KB Flash Memory
- 256/512/1K Bytes EEPROM
- 512/1K/2K Bytes SRAM
- 8/16-bit Timers/Counters



Source: www.atmel.

MCU Comparison

- 8/16 bit

Mfg	Device	Year	RAM (kB)	Flash (kB)	Active (mA)	Sleep (μA)	Wake (μs)	DMA (y/n)
Atmel	ATmega1281	2005	8	128	0.9	1	6	no
	ATmega165P	2007	1	16	0.33	0.65	n/a	no
	ATtiny13A	2008	0.064	1	0.19	0.15	n/a	no
TI	MSP430F1611	2004	10	48	0.5	2.6	6	yes
	MSP430F437	2004	1	32	0.3	1.1	6	yes
	MSP430F412	2001	0.256	4	0.2	0.7	6	no
	MSP430F2002	2005	0.128	1	0.22	0.5	1	no
Microchip	PIC24F16KA102	2009	1.536	8	0.195	0.54	1	no

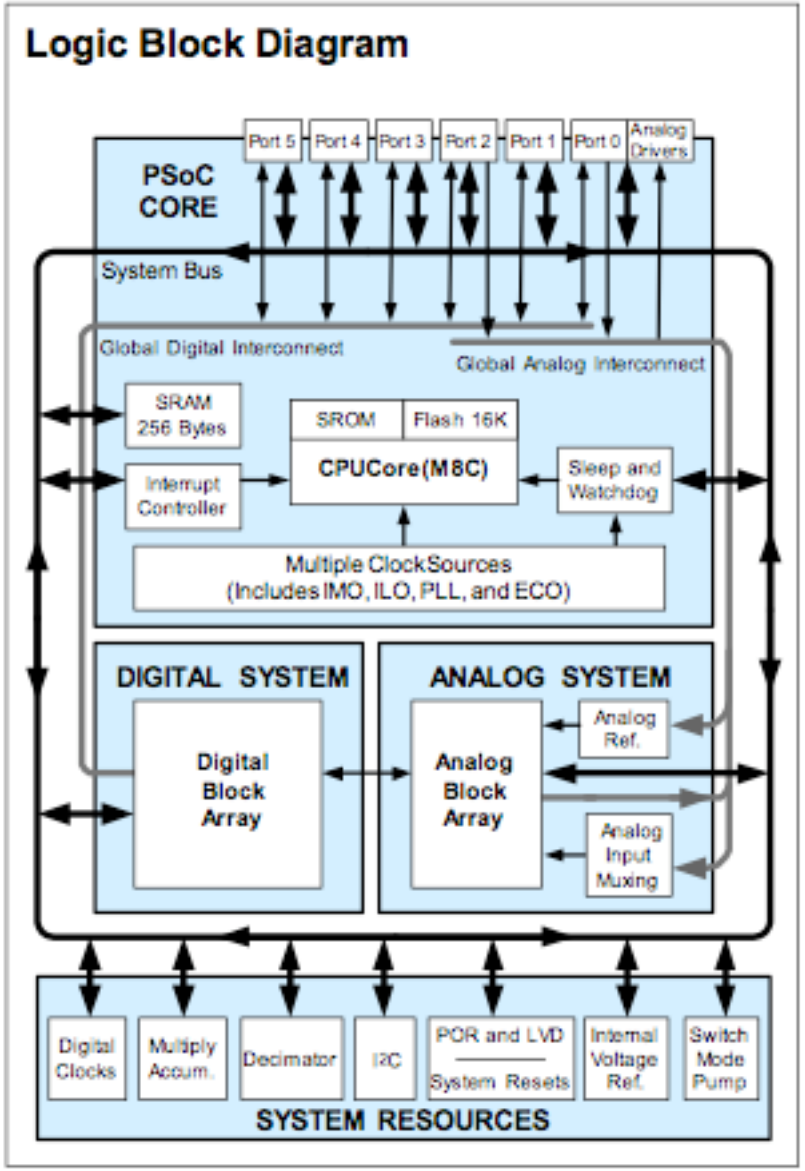
Active power consumption @ 1MHz

- 8/16/32 bit

Mfg	Device	Arch (bit)	VCC (V)	RAM (kB)	Flash (kB)	Active (mA) high	low	Stop (μA)	Standby (μA)	Wake (μs) stop	standby
Atmel	ATmega1281	8	1.8-5.5	8	128	n/a	10 @ 8MHz	n/a	7.5	n/a	6
	AVR32UCB0256	32	3.0-3.6	32	128	23.5 @ 60MHz	5.5 @ 12MHz	21.5	n/a	n/a	n/a
	SAM3U4E	32	1.6-3.6	52	2x128	44 @ 84MHz	18.3 @ 24MHz	9.2	0.37	<10	<500
STM	STM32F103RC	32	2.0-3.6	48	256	51 @ 72MHz	7.2 @ 8MHz	24	3.4	5.4	50
TI	MSP430F1611	16	1.8-3.6	10	48	n/a	0.5 @ 8MHz	n/a	2.6	n/a	6
	MSP430F5437	16	1.8-3.6	16	256	n/a	1.3 @ 8MHz	8	1.7	5	5

Cypress PSoC

- Programmable System on Chip
- Core
  - M8C, 8051, Cortex-M3
  - Flash Memory, SRAM
  - Watchdog, multiple clocks
- Configurable Analog and Digital blocks
  - Similar to CPLD/FPGA
  - Blocks can be combined for
    - ADC
    - Counters
    - Amplifiers
- Programmable routing and interconnects
- TiVo, Capacitive sensing of iPoD



Digital Signal Processors

- Similar to MCUs in architecture
- CPU Core optimized for complex numeric tasks
  - Basic execution unit: Multiply + Accumulate
  - Data intensive operations
  - Usually used as co-processor
  - Signal filtering
  - Video compression/Decompression
- Example: Analog Devices Blackfin
  - Runs embedded OS (uCLinux) - Does not need host CPU
  - 600 MHz and below
- Example: TI C64X
  - Used primarily for data encoding/decoding
  - Coupled with ARM Cortex-A8 (as host CPU)
  - 1 GHz and below

Communication Interfaces in Embedded Systems

- Communication is an important aspect of Embedded Systems
- Often contain specialized communication chips
- Wired
  - Interfacing with sensors and other system components
    - CAN - Controller Area Network
    - I^2C - Inter-Integrated Circuit
    - SPI - Serial Peripheral Interface
    - UART - Universal Asynchronous Receiver Transmitter
    - USB - Universal Serial Bus
  - Communication with other embedded systems
    - Ethernet
- Wireless
  - Ever more useful due to lower cost and ease of installation
  - Many standards for short,mid, long range communication

Wireless Technologies

- Short range
  - IEEE 802.15.4, ZigBee Alliance
    - Home Automation
    - Sensor Networks
  - z-Wave
    - Home Automation
  - Bluetooth
    - Short range
    - Audio headsets
- Mid range
  - 802.11
  - WiMax
- Long range
  - Cellular - GSM/CDMA
  - Satellite
  - Proprietary point to point links
- Infrared
  - Remote control
- Proprietary transceivers

Radio Choices for Wireless Embedded Nodes

Mfg	Device	Year	Wake (ms)	VCC (V)	RxSens (dBm)	TxPwr (dBm)	Rx (mA)	Tx (mA)	Sleep (μA)	FIFO (Rx/Tx)	SCLK (MHz)	SFD (y/n)	CCA (y/n)	AES (y/n)	Area (mm^2)
Atmel	RF230	2006	1.1	1.8-3.6	-101	+3	15.5	16.5	.02	128	8.0	no	no	no	25
Ember	EM260	2006	1	2.1-3.6	-99	+2.5	28	28	1.0	128	5	yes	yes	yes	36
Freescale	MC13192	2004	7-20	2.0-3.4	-92	+4	37	30	1.0	128/256	8.0	yes	yes	yes	25
	MC13202	2007	7-20	2.0-3.4	-92	+4	37	30	1.0	128/256	8.0	yes	yes	yes	25
	MC13212	2005	7-20	2.0-3.4	-92	+3	37	30	1.0	128/256	8.0	yes	yes	yes	81
Jennic	JN5121	2005	>2.5	2.2-3.6	-93	+1	38	28	<5.0	16	16.0	yes	yes	yes	64
	JN5139	2007	>2.5	2.2-3.6	-95.5	+0.5	37	37	2.8	16	16.0	yes	yes	yes	64
TI	CC2420	2003	0.58	2.1-3.6	-95	0	18.8	17.4	1	128/128	10	yes	yes	yes	49
	CC2430	2005	0.65	2.0-3.6	-92	0	17.2	17.4	0.5	128/128	4	yes	yes	yes	49
	CC2520	2008	0.50	1.8-3.8	-98	+5	18.5	25.8	.03	128/128	8.0	yes	yes	yes	25

$$P(\text{in dBm}) = 10\log\frac{P(\text{in mW})}{1\text{mW}}$$

+20dBm=100mW; 0 dBm = 1 mW

-3 dBm = 0.5mW; -10 dBm = 0.1 mW

	802.11g	802.15.4
Chipset	Atheros 5006XS	CC2420
Output Power	16dbm	0dbm
Rx Sensitivity	-78dbm@36Mbps	-90dbm@250Kbps
Tx Power (Max Output)	1320mW	57.42mW
Rx Power	924mW	65.01mW
Total Power	2.24W	122.43mW
Effective Throughput	20Mbps	125Kbps
Efficiency (nJ/bit)	112	979

Peripherals in Embedded Systems

- Diverse set of peripherals to provide specific functionality
- Timers and Counters
  - Generate events at specific time or determine duration between two external events
  - Count pulses on some input signals
  - Watchdog timer - Failure to receive a signal in time will produce system reset
- Pulse Width Modulators (PWM)
  - Allows controlling time period and duty cycle
- Analog to Digital Converters (ADC)
  - Physical environment around the system produces analog values while the system understands digital values
  - Number of encoding bits and maximum voltage of input signal decides the resolution ($V_{max}/(2^n - 1)$)

Memory Interface in Embedded Systems

- Typically small (few KBs) on-chip memory
  - MSP430 - Flash up to 32 KB; RAM - 6 KB
  - AVR - Flash up to 32 KB; EEPROM up to 1 KB; RAM up to 2 KB
  - LPC1768 (Cortex-M3 Based) - Flash up to 512 KB; RAM up to 64 KB

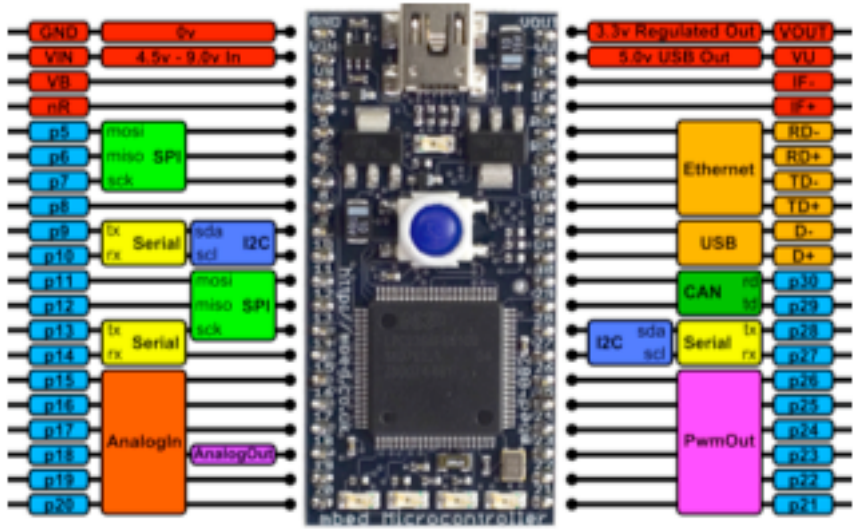
- DMA Controller allows movement of data from one memory address to another without CPU intervention
  - Increases the throughput of peripheral modules
  - Reduce system power consumption (CPU can be in sleep mode while the data moves to/from peripheral)

- Typically each device comes with different compositions of on-chip memory

Device	Flash	EEPROM	RAM
ATmega168PA	16K Bytes	512 Bytes	1K Bytes
ATmega328	32K Bytes	1K Bytes	2K Bytes
ATmega328P	32K Bytes	1K Bytes	2K Bytes

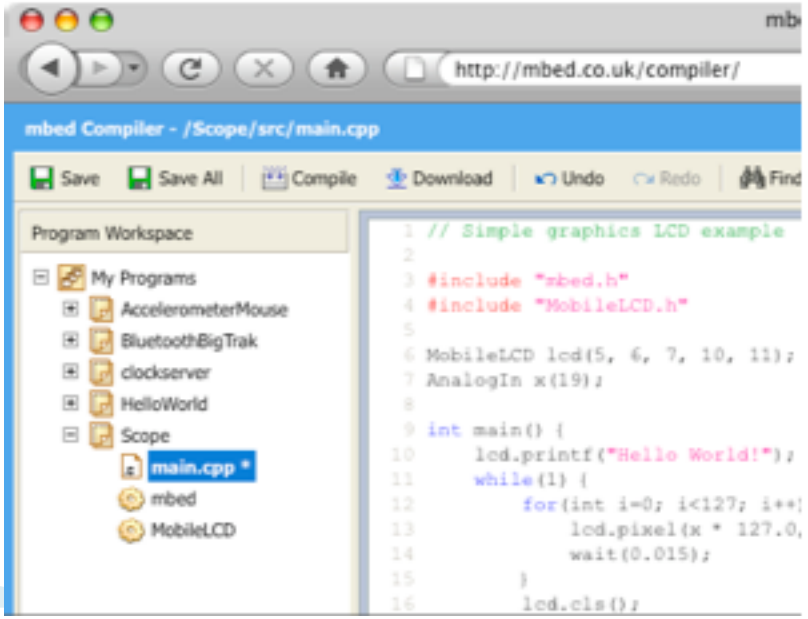
Example Platforms - mBed

- Cortex-M3 Core running at 96MHz, with 512KB FLASH, 64KB RAM and a load of interfaces including Ethernet, USB Device, CAN, SPI, I2C and other I/O



- One of the cheapest commercially available ARM based platform out there

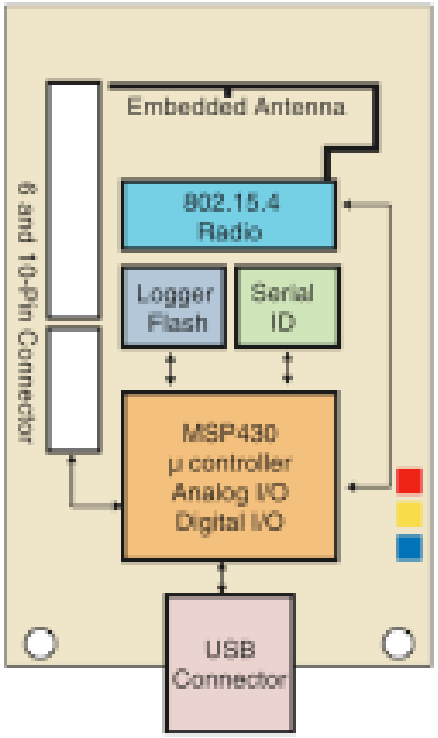
- "Cloud" compiler: Web based tool chain, lots of libraries and documentation and good community support



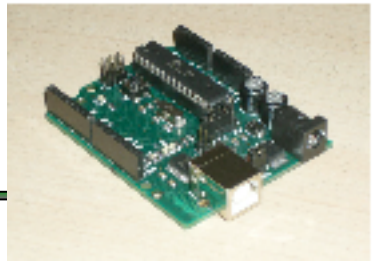
mBed Design Challenge

Example Platforms - TelosB

- TI MSP-430 Microcontroller with 10 KB RAM
- IEEE 802.15.4 compliant
  - 250 kbps radio
  - Integrated on-board antenna
  - Receive mode - 23 mA; Idle mode - 21 uA; Sleep mode - 1 uA
- Data collection and programming via USB
- Open source operating system (TinyOS)
- Optional integrated temperature, light and humidity sensor
- 1 MB external flash
- Active mode - 1.8 mA; Sleep mode - 5.1 uA
- Runs on 2xAA Batteries



Example Platforms - AVR Based



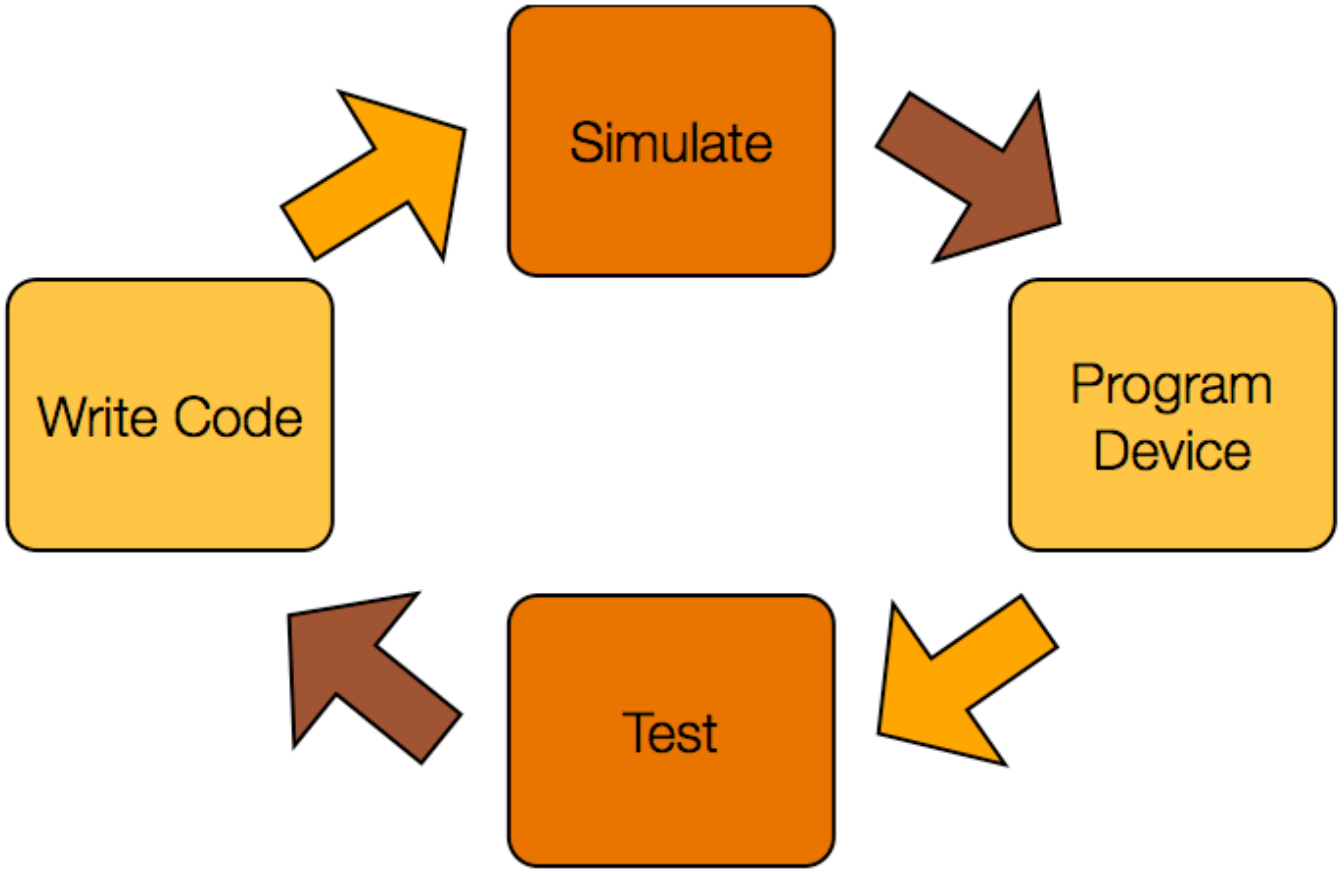
- ATMEGA328P Microcontroller
- Pre-programmed boot loader for automatically downloading the hex program
  - Works as a proxy in-circuit programmer to directly download program from computer to the microcontroller
  - Jumper setting to lock the microcontroller in bootloader mode permanently
- Programming in C
- More details about the AVR programming in the next class
- Similar to the commercially available Arduino board

Other Platforms We Have

- uLEAP: Developed at UCLA
  - MSP430 Microcontroller
  - Energy sensing circuitry on board
  - Bluetooth communication
  - Programming in uC Linux

- Atmel MEGA32u4 based platform
  - Same cost as the the ATMEGA328P
  - Not so standard pinout for connecting breakout boards
  - USB OTG for connecting USB slave devices externally

Embedded System Development



- Often for smaller code it is easier to just do the programming and test the code on the actual hardware itself

Programming Languages and Compilers

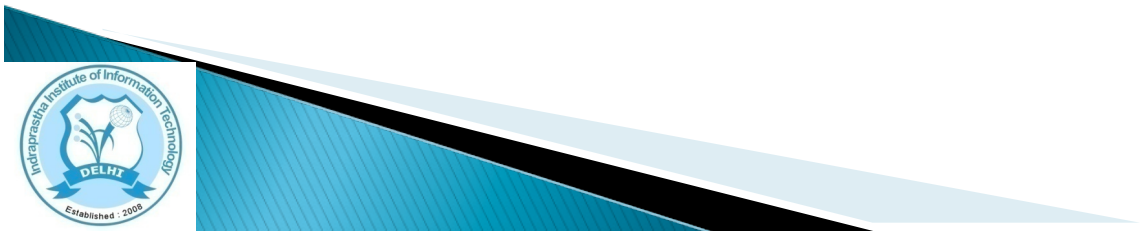
- Many possible languages
  - Assembler (MCU specific)
  - C/C++, NesC
  - Java
  - Verilog / VHDL
  - Python

- Cross compiler translates high level code to MCU specific code instructions
  - Cross compiler usually runs on your development (host) system
  - Generated binary does not usually execute on host system
  - Needs an emulator/simulator to interpret

Programming Embedded Devices

- Each class of devices is programmed slightly differently
- 8/16-bit class devices: ATiny, ATmega, MSP430, Cortex-M3
  - Have a small on-chip hardware unit called a bootstrap loader (UISP, BSL)
  - An external programming device communicates with the BSL through multiplexed pins, eg. the serial interface, USB
  - These devices can also be programmed via JTAG pins
  - Typically, the BSL will erase and overwrite the entire flash memory
- 32-bit class devices: Cortex-M3, ARM7, PXA27x
  - Have a software bootloader that is programmed once using JTAG
  - Bootloader always starts up first and loads the required OS kernel
  - Bootloader has board specific drivers that can communicate with the host machine through a serial port or network connection
  - Typically, only parts of the flash are erased and overwritten
  - Once OS is booted, can use utilities like NFS to load applications into RAM

QUESTIONS?


Indraprastha Institute of Information Technology
DELHI
Established : 2008