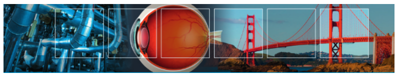
Chapter 3: Node Architecture



Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture

Node Architecture

- Wireless sensor nodes are the essential building blocks in a wireless sensor network
  - *sensing*, *processing*, and *communication*
  - *stores* and *executes* the communication protocols as well as data processing algorithms
- The node consists of *sensing*, *processing*, *communication*, and *power subsystems*
  - trade-off between flexibility and efficiency – both in terms of energy and performance

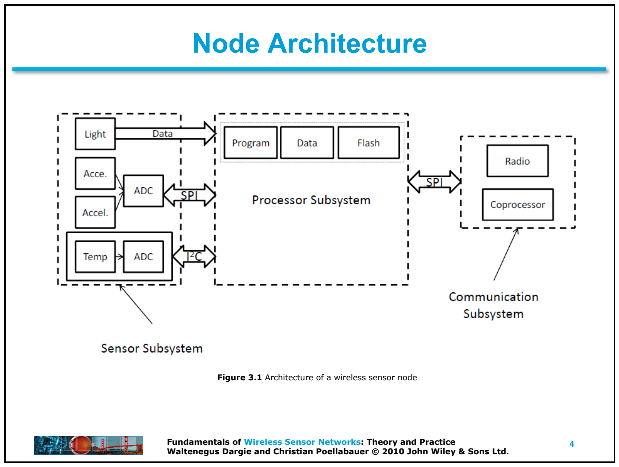
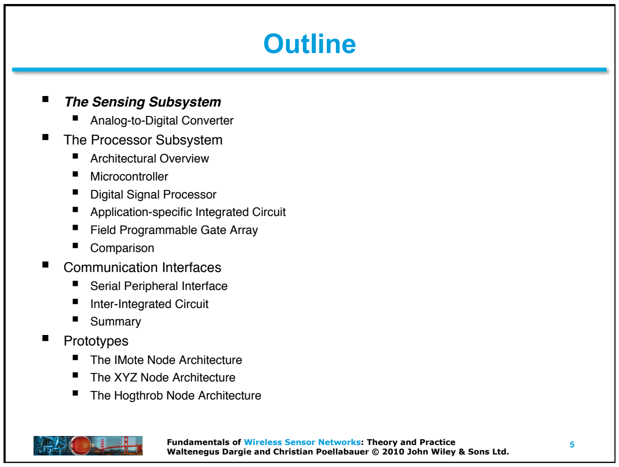
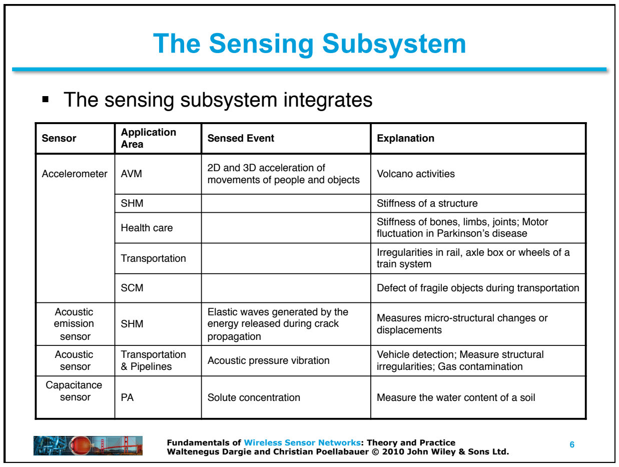

Node Architecture

Figure 3.1 Architecture of a wireless sensor node

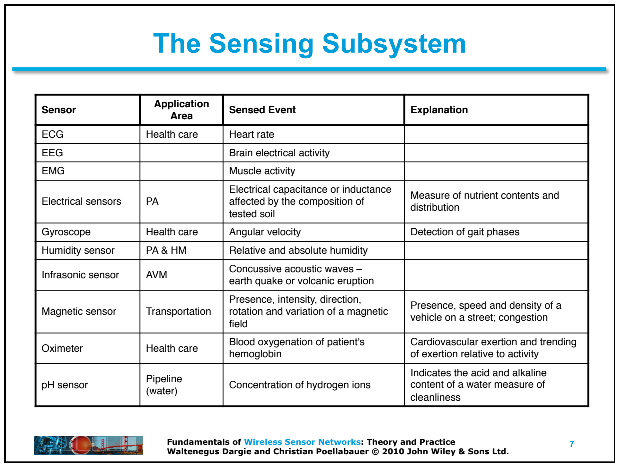

Outline

- ***The Sensing Subsystem***
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture


The Sensing Subsystem

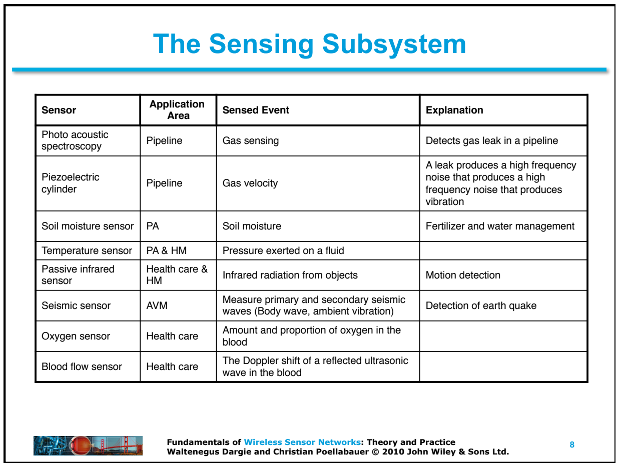
- The sensing subsystem integrates

Sensor	Application Area	Sensed Event	Explanation
Accelerometer	AVM	2D and 3D acceleration of movements of people and objects	Volcano activities
	SHM		Stiffness of a structure
	Health care		Stiffness of bones, limbs, joints; Motor fluctuation in Parkinson's disease
	Transportation		Irregularities in rail, axle box or wheels of a train system
	SCM		Defect of fragile objects during transportation
Acoustic emission sensor	SHM	Elastic waves generated by the energy released during crack propagation	Measures micro-structural changes or displacements
Acoustic sensor	Transportation & Pipelines	Acoustic pressure vibration	Vehicle detection; Measure structural irregularities; Gas contamination
Capacitance sensor	PA	Solute concentration	Measure the water content of a soil



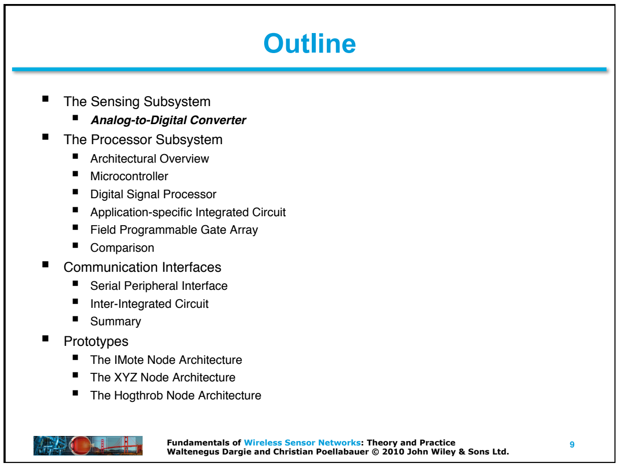
The Sensing Subsystem

Sensor	Application Area	Sensed Event	Explanation
ECG	Health care	Heart rate	
EEG		Brain electrical activity	
EMG		Muscle activity	
Electrical sensors	PA	Electrical capacitance or inductance affected by the composition of tested soil	Measure of nutrient contents and distribution
Gyroscope	Health care	Angular velocity	Detection of gait phases
Humidity sensor	PA & HM	Relative and absolute humidity	
Infrasonic sensor	AVM	Concussive acoustic waves – earth quake or volcanic eruption	
Magnetic sensor	Transportation	Presence, intensity, direction, rotation and variation of a magnetic field	Presence, speed and density of a vehicle on a street; congestion
Oximeter	Health care	Blood oxygenation of patient's hemoglobin	Cardiovascular exertion and trending of exertion relative to activity
pH sensor	Pipeline (water)	Concentration of hydrogen ions	Indicates the acid and alkaline content of a water measure of cleanliness



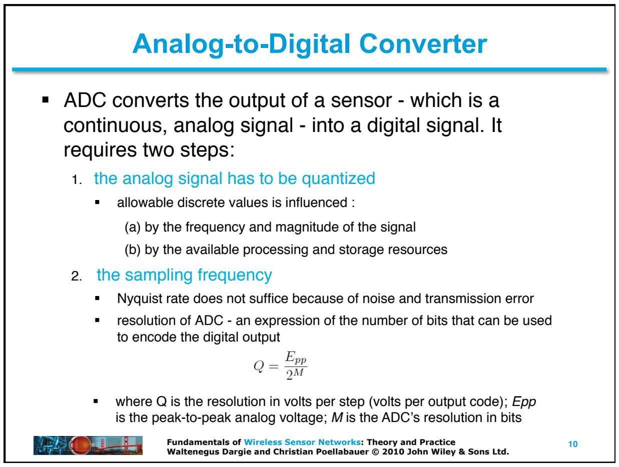
The Sensing Subsystem

Sensor	Application Area	Sensed Event	Explanation
Photo acoustic spectroscopy	Pipeline	Gas sensing	Detects gas leak in a pipeline
Piezoelectric cylinder	Pipeline	Gas velocity	A leak produces a high frequency noise that produces a high frequency noise that produces vibration
Soil moisture sensor	PA	Soil moisture	Fertilizer and water management
Temperature sensor	PA & HM	Pressure exerted on a fluid	
Passive infrared sensor	Health care & HM	Infrared radiation from objects	Motion detection
Seismic sensor	AVM	Measure primary and secondary seismic waves (Body wave, ambient vibration)	Detection of earth quake
Oxygen sensor	Health care	Amount and proportion of oxygen in the blood	
Blood flow sensor	Health care	The Doppler shift of a reflected ultrasonic wave in the blood	



Outline

- The Sensing Subsystem
  - ***Analog-to-Digital Converter***
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



Analog-to-Digital Converter

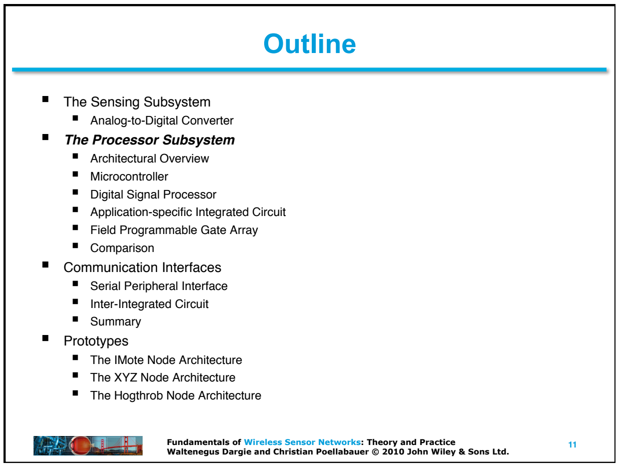
- ADC converts the output of a sensor - which is a continuous, analog signal - into a digital signal. It requires two steps:
  1. the analog signal has to be quantized
     - allowable discrete values is influenced :

       (a) by the frequency and magnitude of the signal

       (b) by the available processing and storage resources
  2. the sampling frequency
     - Nyquist rate does not suffice because of noise and transmission error
     - resolution of ADC - an expression of the number of bits that can be used to encode the digital output

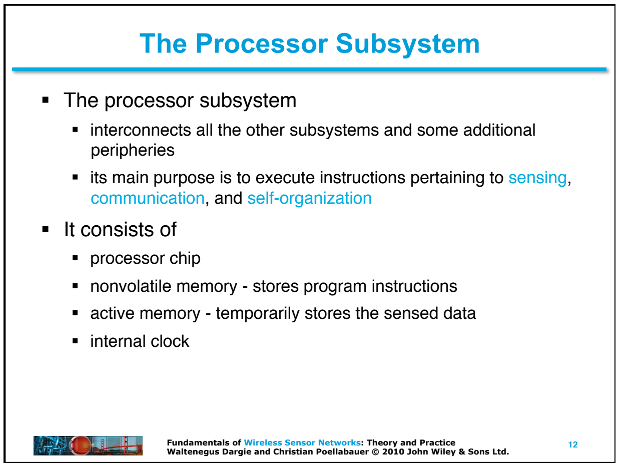
$$Q = \frac{E_{pp}}{2^M}$$

- where Q is the resolution in volts per step (volts per output code); *Epp* is the peak-to-peak analog voltage; *M* is the ADC's resolution in bits



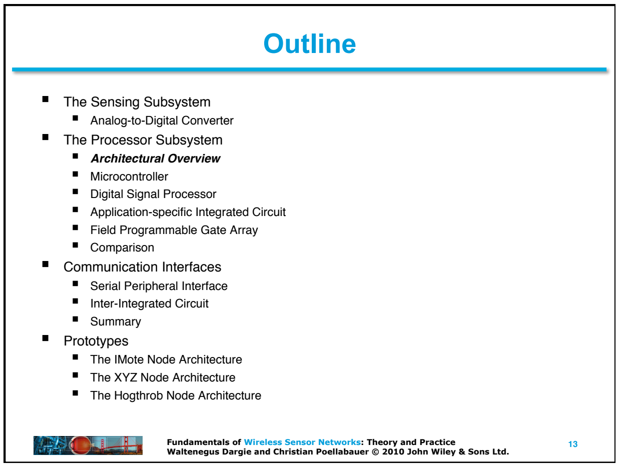
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- ***The Processor Subsystem***
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



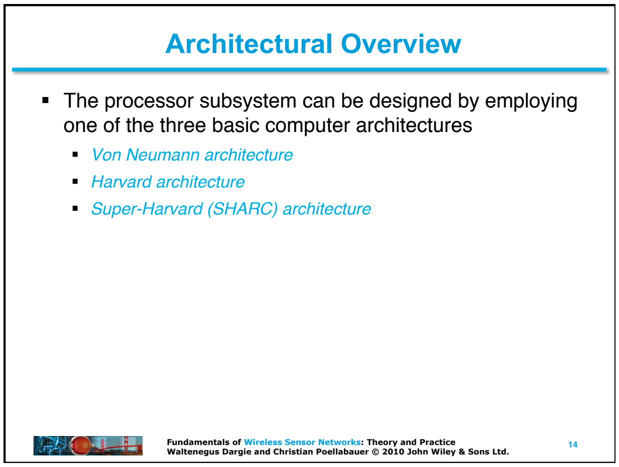
The Processor Subsystem

- The processor subsystem
  - interconnects all the other subsystems and some additional peripheries
  - its main purpose is to execute instructions pertaining to sensing, communication, and self-organization
- It consists of
  - processor chip
  - nonvolatile memory - stores program instructions
  - active memory - temporarily stores the sensed data
  - internal clock



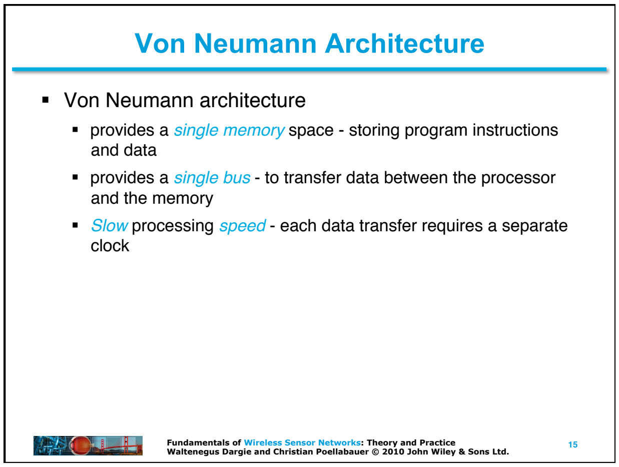
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - ***Architectural Overview***
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



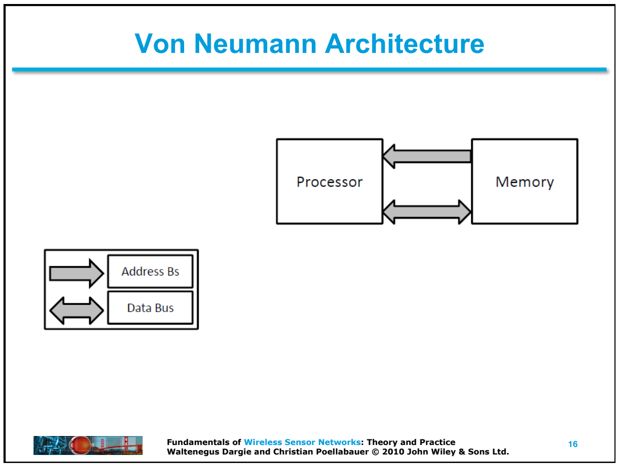
Architectural Overview

- The processor subsystem can be designed by employing one of the three basic computer architectures
  - *Von Neumann architecture*
  - *Harvard architecture*
  - *Super-Harvard (SHARC) architecture*

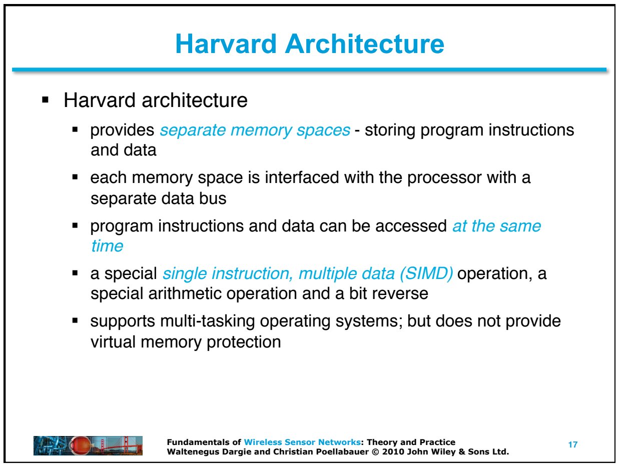


Von Neumann Architecture

- Von Neumann architecture
  - provides a *single memory* space - storing program instructions and data
  - provides a *single bus* - to transfer data between the processor and the memory
  - *Slow* processing *speed* - each data transfer requires a separate clock

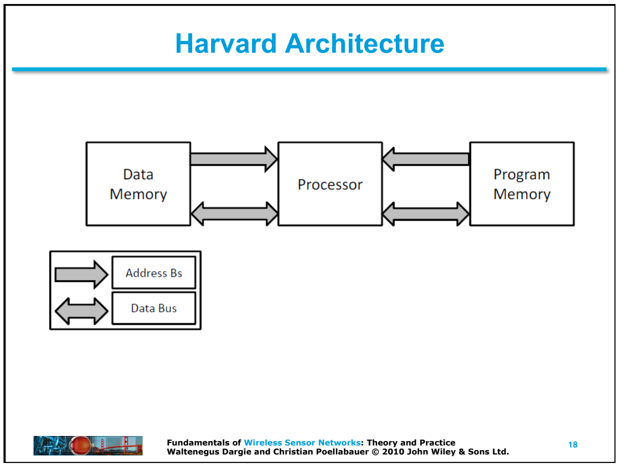


Von Neumann Architecture



Harvard Architecture

- Harvard architecture
  - provides *separate memory spaces* - storing program instructions and data
  - each memory space is interfaced with the processor with a separate data bus
  - program instructions and data can be accessed *at the same time*
  - a special *single instruction, multiple data (SIMD)* operation, a special arithmetic operation and a bit reverse
  - supports multi-tasking operating systems; but does not provide virtual memory protection

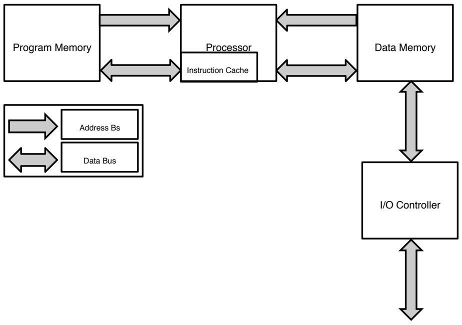


Harvard Architecture

Super-Harvard Architecture

- Super-Harvard architecture (best known: SHARC)
  - an *extension of the Harvard architecture*
  - *adds two components* to the Harvard architecture:
    - internal instruction cache - temporarily store frequently used instructions - enhances performance
    - an underutilized program memory can be used as a temporary relocation place for data
    - Direct Memory Access (DMA)
      - costly CPU cycles can be invested in a different task
      - program memory bus and data memory bus accessible from outside the chip

Super-Harvard Architecture



Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - ***Microcontroller***
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture

Microcontroller

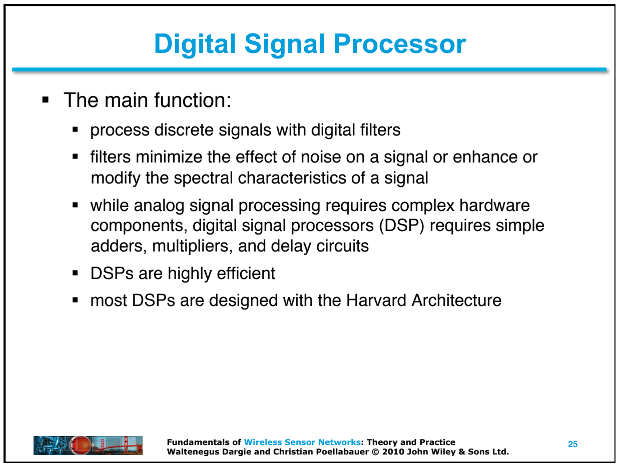
- **Structure of microcontroller**
  - integrates the following components:
    - CPU core
    - volatile memory (RAM) for data storage
    - ROM, EPROM, EEPROM, or Flash memory
    - parallel I/O interfaces
    - discrete input and output bits
    - clock generator
    - one or more internal analog-to-digital converters
    - serial communications interfaces

Microcontroller

- Advantages:
  - suitable for building computationally less intensive, standalone applications, because of its *compact construction*, *small size*, *low-power consumption*, and *low cost*
  - *high speed* of the programming and eases debugging, because of the use of higher-level programming languages
- Disadvantages:
  - *not* as *powerful* and as *efficient* as some custom-made processors (such as DSPs and FPGAs)
  - some applications (simple sensing tasks but large scale deployments) may prefer to use architecturally simple but energy- and cost-efficient processors

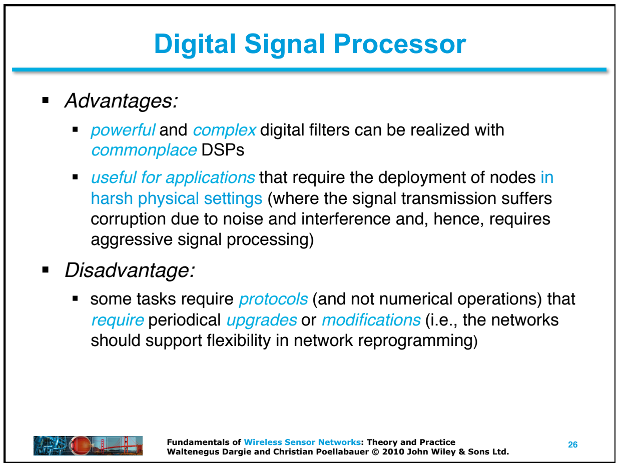
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - ***Digital Signal Processor***
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



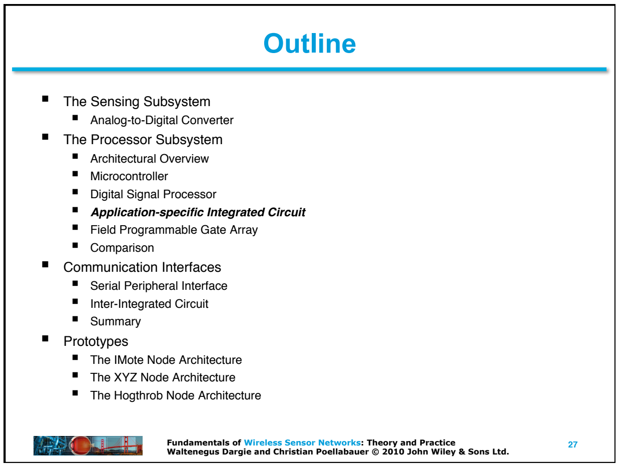
Digital Signal Processor

- The main function:
  - process discrete signals with digital filters
  - filters minimize the effect of noise on a signal or enhance or modify the spectral characteristics of a signal
  - while analog signal processing requires complex hardware components, digital signal processors (DSP) requires simple adders, multipliers, and delay circuits
  - DSPs are highly efficient
  - most DSPs are designed with the Harvard Architecture



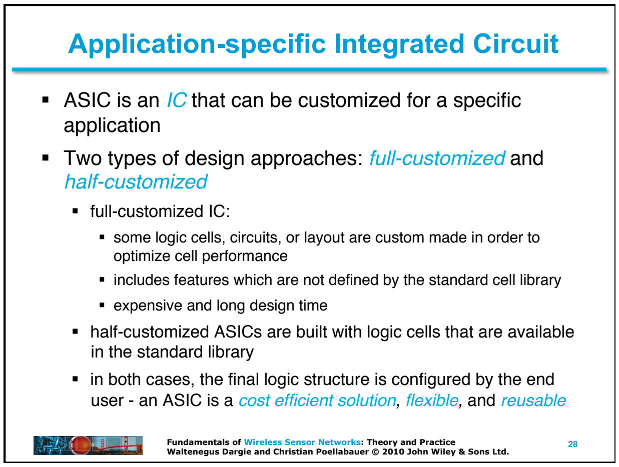
Digital Signal Processor

- *Advantages:*
  - *powerful* and *complex* digital filters can be realized with *commonplace* DSPs
  - *useful for applications* that require the deployment of nodes in harsh physical settings (where the signal transmission suffers corruption due to noise and interference and, hence, requires aggressive signal processing)
- *Disadvantage:*
  - some tasks require *protocols* (and not numerical operations) that *require* periodical *upgrades* or *modifications* (i.e., the networks should support flexibility in network reprogramming)



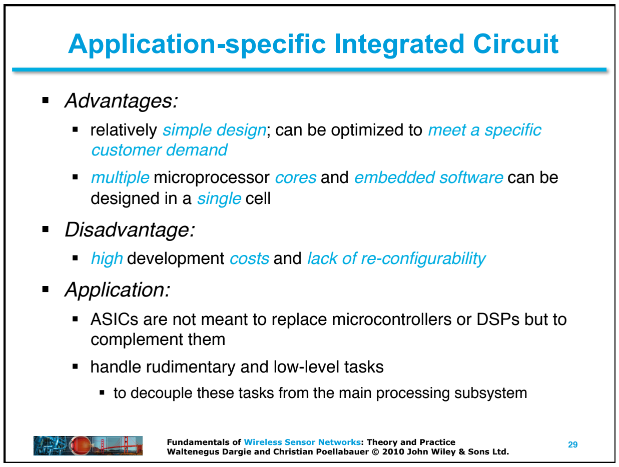
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - ***Application-specific Integrated Circuit***
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



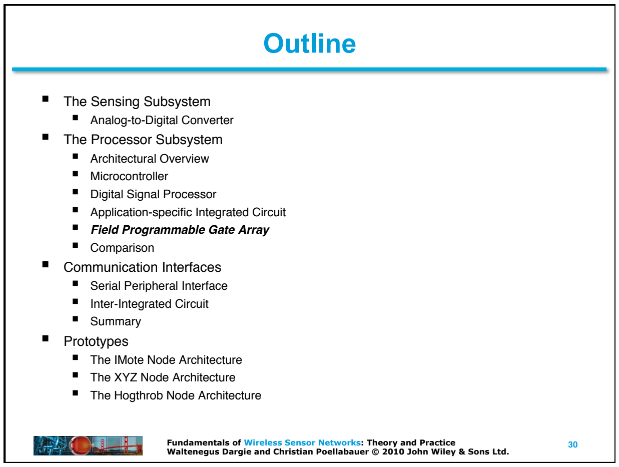
Application-specific Integrated Circuit

- ASIC is an *IC* that can be customized for a specific application
- Two types of design approaches: *full-customized* and *half-customized*
  - full-customized IC:
    - some logic cells, circuits, or layout are custom made in order to optimize cell performance
    - includes features which are not defined by the standard cell library
    - expensive and long design time
  - half-customized ASICs are built with logic cells that are available in the standard library
  - in both cases, the final logic structure is configured by the end user - an ASIC is a *cost efficient solution*, *flexible*, and *reusable*



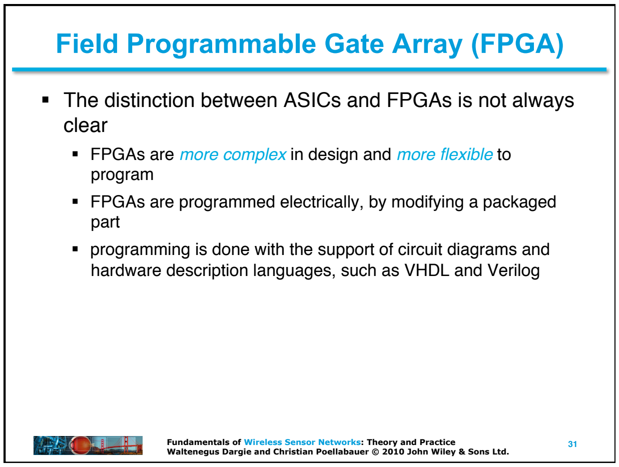
Application-specific Integrated Circuit

- *Advantages:*
  - relatively *simple design*; can be optimized to *meet a specific customer demand*
  - *multiple* microprocessor *cores* and *embedded software* can be designed in a *single* cell
- *Disadvantage:*
  - *high* development *costs* and *lack of re-configurability*
- *Application:*
  - ASICs are not meant to replace microcontrollers or DSPs but to complement them
  - handle rudimentary and low-level tasks
    - to decouple these tasks from the main processing subsystem



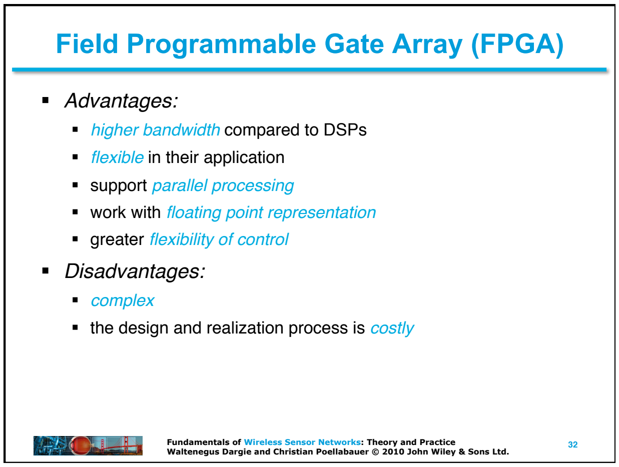
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - ***Field Programmable Gate Array***
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



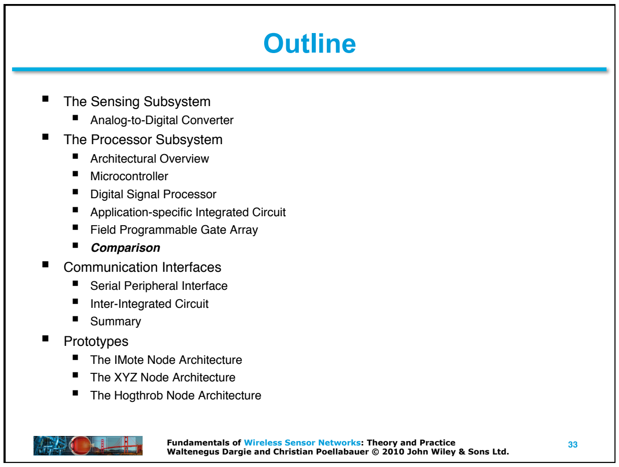
Field Programmable Gate Array (FPGA)

- The distinction between ASICs and FPGAs is not always clear
  - FPGAs are *more complex* in design and *more flexible* to program
  - FPGAs are programmed electrically, by modifying a packaged part
  - programming is done with the support of circuit diagrams and hardware description languages, such as VHDL and Verilog



Field Programmable Gate Array (FPGA)

- *Advantages:*
  - *higher bandwidth* compared to DSPs
  - *flexible* in their application
  - support *parallel processing*
  - work with *floating point representation*
  - greater *flexibility of control*
- *Disadvantages:*
  - *complex*
  - the design and realization process is *costly*



Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - ***Comparison***
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture

Comparison

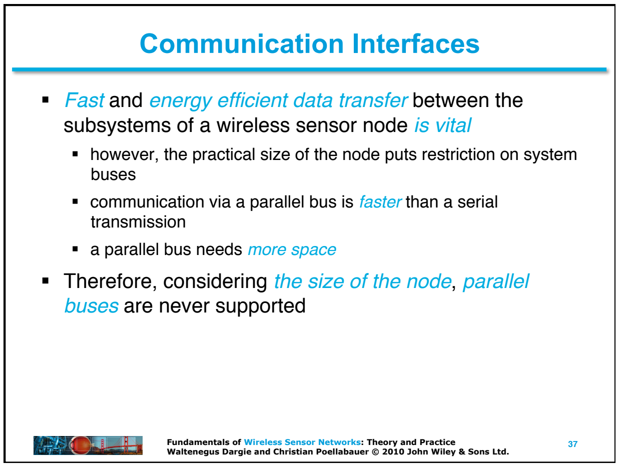
- Working with a *micro-controller* is *preferred* if the design goal is to achieve *flexibility*
- Working with the other mentioned options is preferred if power consumption and computational efficiency is desired
- *DSPs* are expensive, large in size and less flexible; they are *best for signal processing*, with specific algorithms
- *FPGAs are faster* than both microcontrollers and digital signal processors and support *parallel computing*; but their production cost and the programming difficulty make them *less suitable*

Comparison

- *ASICs* have *higher bandwidths*; they are *the smallest* in size, *perform much better*, and *consume less power* than any of the other processing types; but have a *high cost* of production owing to the complex design process

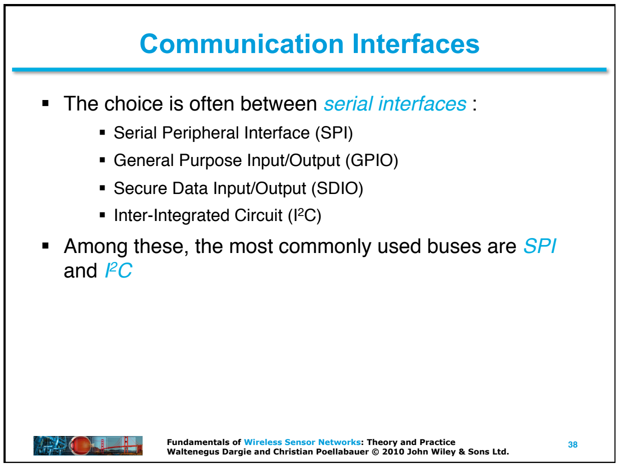
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- ***Communication Interfaces***
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



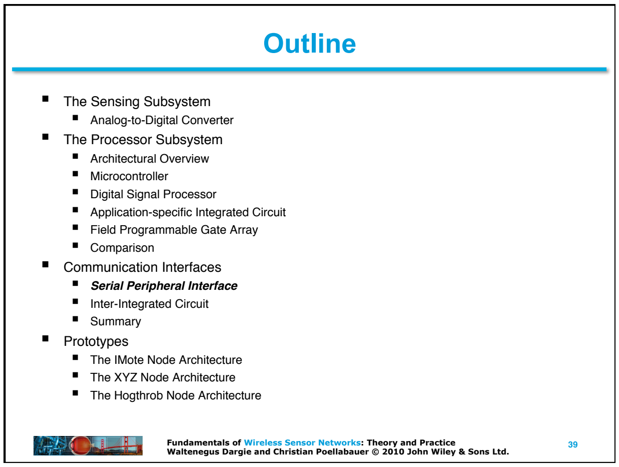
Communication Interfaces

- *Fast* and *energy efficient data transfer* between the subsystems of a wireless sensor node *is vital*
  - however, the practical size of the node puts restriction on system buses
  - communication via a parallel bus is *faster* than a serial transmission
  - a parallel bus needs *more space*
- Therefore, considering *the size of the node*, *parallel buses* are never supported



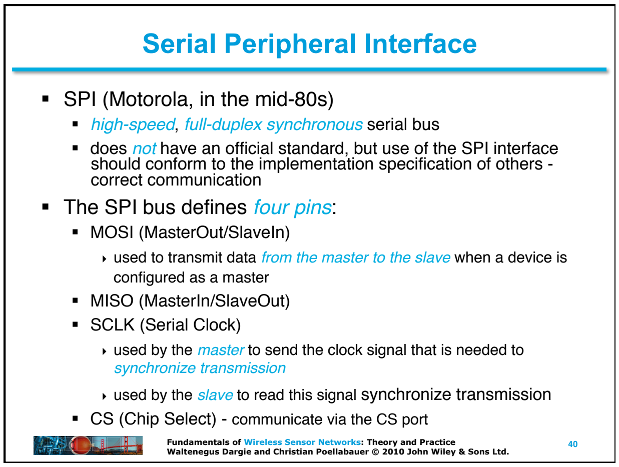
Communication Interfaces

- The choice is often between *serial interfaces* :
  - Serial Peripheral Interface (SPI)
  - General Purpose Input/Output (GPIO)
  - Secure Data Input/Output (SDIO)
  - Inter-Integrated Circuit (I^{2}C)
- Among these, the most commonly used buses are *SPI* and *I^{2}C*



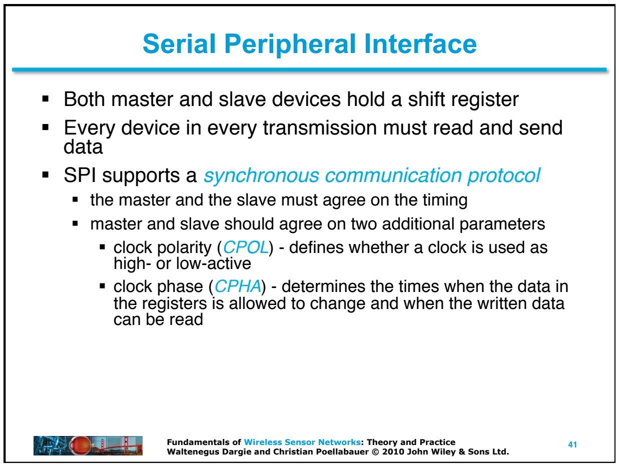
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - ***Serial Peripheral Interface***
  - Inter-Integrated Circuit
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



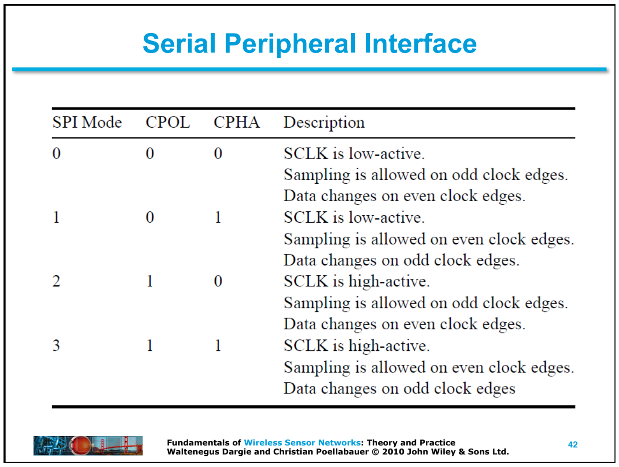
Serial Peripheral Interface

- SPI (Motorola, in the mid-80s)
  - *high-speed*, *full-duplex synchronous* serial bus
  - does *not* have an official standard, but use of the SPI interface should conform to the implementation specification of others - correct communication
- The SPI bus defines *four pins*:
  - MOSI (MasterOut/SlaveIn)
    - used to transmit data *from the master to the slave* when a device is configured as a master
  - MISO (MasterIn/SlaveOut)
  - SCLK (Serial Clock)
    - used by the *master* to send the clock signal that is needed to *synchronize transmission*
    - used by the *slave* to read this signal synchronize transmission
  - CS (Chip Select) - communicate via the CS port



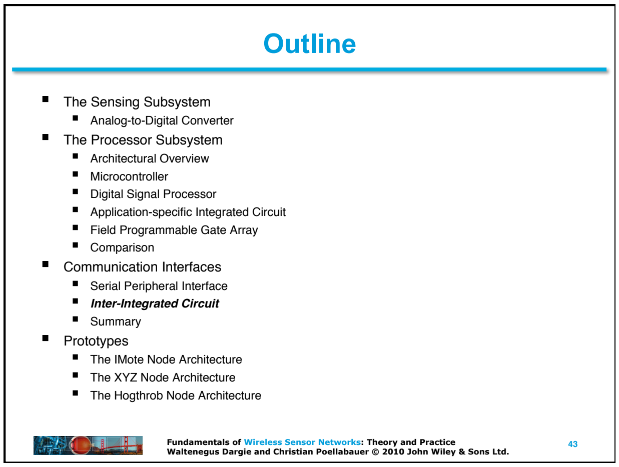
Serial Peripheral Interface

- Both master and slave devices hold a shift register
- Every device in every transmission must read and send data
- SPI supports a *synchronous communication protocol*
  - the master and the slave must agree on the timing
  - master and slave should agree on two additional parameters
    - clock polarity (*CPOL*) - defines whether a clock is used as high- or low-active
    - clock phase (*CPHA*) - determines the times when the data in the registers is allowed to change and when the written data can be read



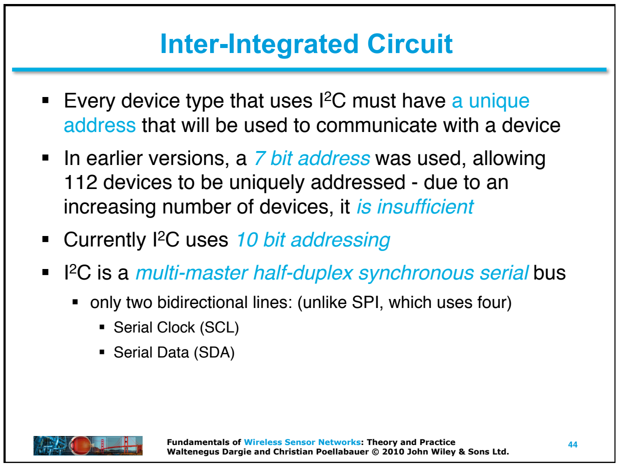
Serial Peripheral Interface

SPI Mode	CPOL	CPHA	Description
0	0	0	SCLK is low-active. Sampling is allowed on odd clock edges. Data changes on even clock edges.
1	0	1	SCLK is low-active. Sampling is allowed on even clock edges. Data changes on odd clock edges.
2	1	0	SCLK is high-active. Sampling is allowed on odd clock edges. Data changes on even clock edges.
3	1	1	SCLK is high-active. Sampling is allowed on even clock edges. Data changes on odd clock edges



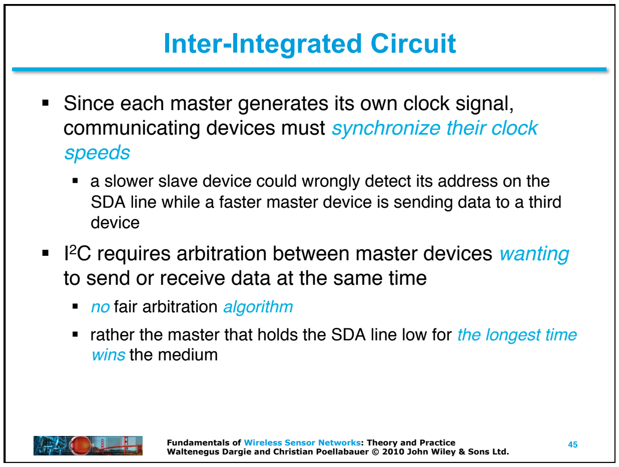
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - ***Inter-Integrated Circuit***
  - Summary
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



Inter-Integrated Circuit

- Every device type that uses I^{2}C must have a unique address that will be used to communicate with a device
- In earlier versions, a *7 bit address* was used, allowing 112 devices to be uniquely addressed - due to an increasing number of devices, it *is insufficient*
- Currently I^{2}C uses *10 bit addressing*
- I^{2}C is a *multi-master half-duplex synchronous serial* bus
  - only two bidirectional lines: (unlike SPI, which uses four)
    - Serial Clock (SCL)
    - Serial Data (SDA)



Inter-Integrated Circuit

- Since each master generates its own clock signal, communicating devices must *synchronize their clock speeds*
  - a slower slave device could wrongly detect its address on the SDA line while a faster master device is sending data to a third device
- I^{2}C requires arbitration between master devices *wanting* to send or receive data at the same time
  - *no* fair arbitration *algorithm*
  - rather the master that holds the SDA line low for *the longest time wins* the medium

Inter-Integrated Circuit

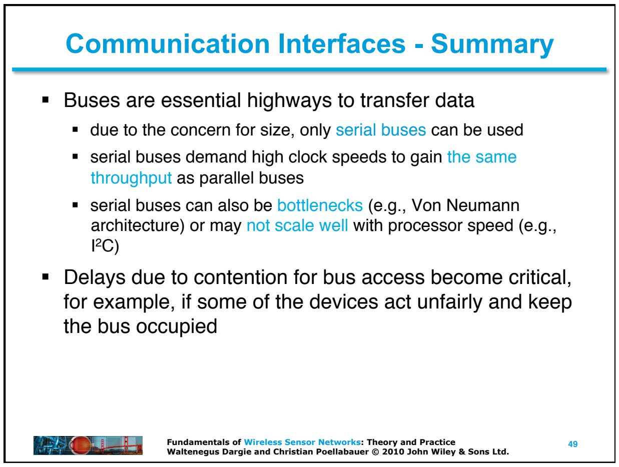
- I^{2}C enables a device to read data *at a byte level* for a fast communication
  - the device can hold the SCL low until it completes reading or sending the next byte - called *handshaking*
- The *aim* of I^{2}C is *to minimize costs* for connecting devices
  - accommodating lower transmission speeds
- I^{2}C defines two speed modes:
  - *a fast-mode* - a bit rate of up to 400Kbps
  - *high-speed mode* - a transmission rate of up to 3.4 Mbps
  - they are downwards compatible to ensure communication with older components

Comparison

SPI	I^{2}C
4 lines enable full-duplex transmission	2 lines reduce space and simplify circuit layout; Lowers costs
No addressing is required due to CS	Addressing enables multi-master mode; Arbitration is required
Allowing only one master avoids conflicts	Multi-master mode is prone to conflicts
Hardware requirement support increases with an increasing number of connected devices -- costly	Hardware requirement is independent of the number of devices using the bus
The master's clock is configured according to the slave's speed but speed adaptation slows down the master.	Slower devices may stretch the clock -- latency but keeping other devices waiting
Speed depends on the maximum speed of the slowest device	Speed is limited to 3.4 MHz
Heterogeneous registers size allows flexibility in the devices that are supported.	Homogeneous register size reduces overhead
Combined registers imply every transmission should be read AND write	Devices that do not read or provide data are not forced to provide potentially useless bytes
The absence of an official standard leads to application specific implementations	Official standard eases integration of devices since developers can rely on a certain implementation

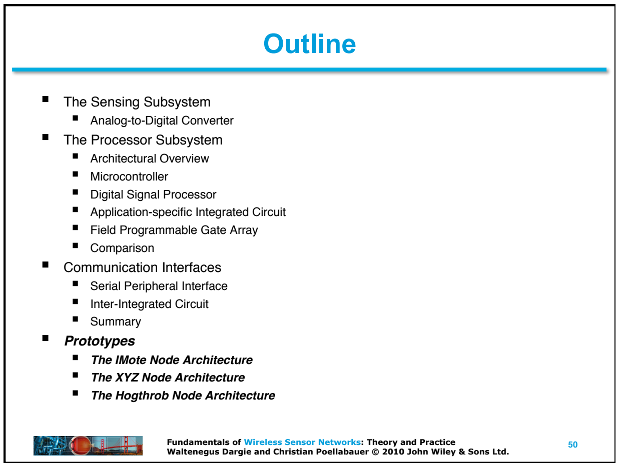
Outline

- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - ***Summary***
- Prototypes
  - The IMote Node Architecture
  - The XYZ Node Architecture
  - The Hogthrob Node Architecture



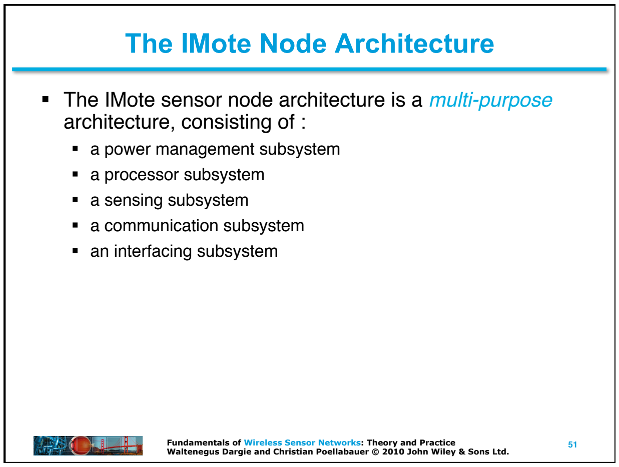
Communication Interfaces - Summary

- Buses are essential highways to transfer data
  - due to the concern for size, only serial buses can be used
  - serial buses demand high clock speeds to gain the same throughput as parallel buses
  - serial buses can also be bottlenecks (e.g., Von Neumann architecture) or may not scale well with processor speed (e.g., I^{2}C)
- Delays due to contention for bus access become critical, for example, if some of the devices act unfairly and keep the bus occupied



Outline

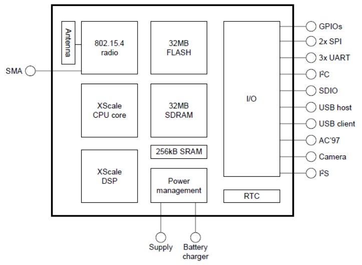
- The Sensing Subsystem
  - Analog-to-Digital Converter
- The Processor Subsystem
  - Architectural Overview
  - Microcontroller
  - Digital Signal Processor
  - Application-specific Integrated Circuit
  - Field Programmable Gate Array
  - Comparison
- Communication Interfaces
  - Serial Peripheral Interface
  - Inter-Integrated Circuit
  - Summary
- ***Prototypes***
  - ***The IMote Node Architecture***
  - ***The XYZ Node Architecture***
  - ***The Hogthrob Node Architecture***



The IMote Node Architecture

- The IMote sensor node architecture is a *multi-purpose* architecture, consisting of :
  - a power management subsystem
  - a processor subsystem
  - a sensing subsystem
  - a communication subsystem
  - an interfacing subsystem

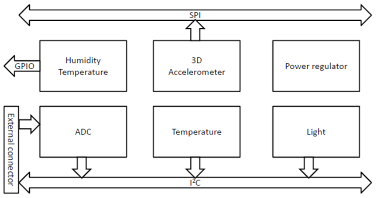
The IMote Node Architecture



The IMote Node Architecture

- A multiple-sensor board contains :
  - a 12-bit, four channels ADC
  - a high-resolution temperature/humidity sensor
  - a low-resolution digital temperature sensor
  - a light sensor
  - the I^{2}C bus is used to connect *low* data rate sources
  - the SPI bus is used to interface *high* data rate sources

The IMote Node Architecture



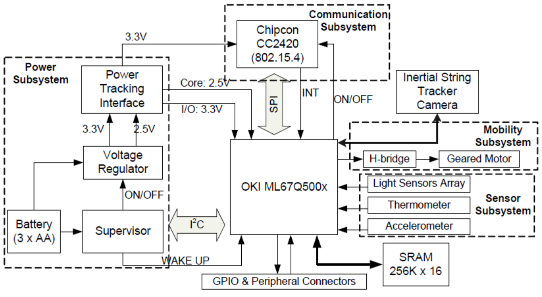
The IMote Node Architecture

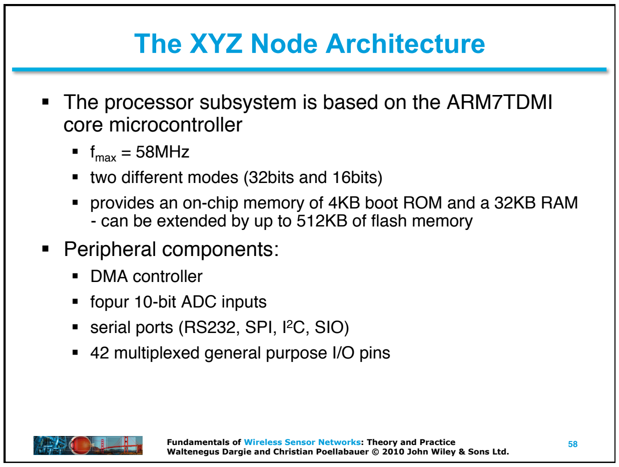
- The processing subsystem provides
  - *main processor* (microprocessor)
    - operates in low voltage (0.85V) and low frequency (13MHz) mode
    - Dynamic Voltage Scaling (104MHz - 416MHz)
    - sleep and deep sleep modes
    - thus enabling *low power operation*
  - *coprocessor* (a DSP)
    - accelerates multimedia operations - computation intensive

The XYZ Node Architecture

- Consists of the four subsystems:
  - power subsystem
  - communication subsystem
  - mobility subsystem
  - sensor subsystem

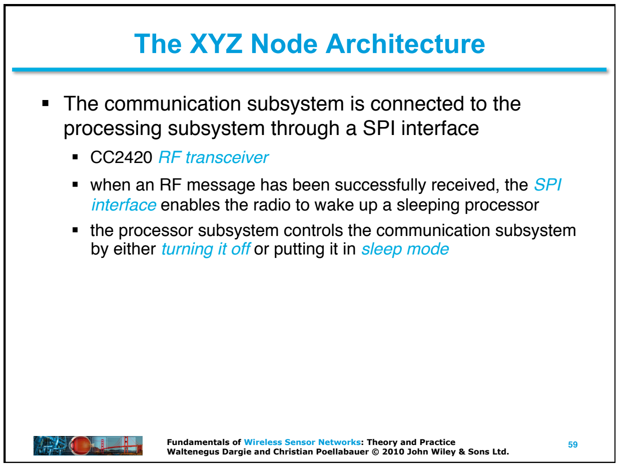
The XYZ Node Architecture





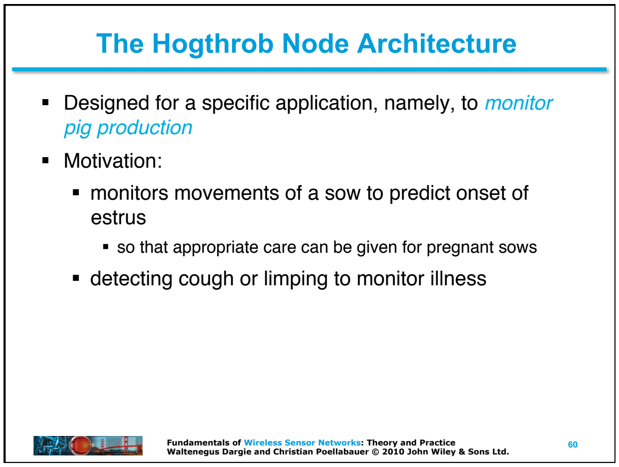
The XYZ Node Architecture

- The processor subsystem is based on the ARM7TDMI core microcontroller
  - f_{max} = 58MHz
  - two different modes (32bits and 16bits)
  - provides an on-chip memory of 4KB boot ROM and a 32KB RAM - can be extended by up to 512KB of flash memory
- Peripheral components:
  - DMA controller
  - fopur 10-bit ADC inputs
  - serial ports (RS232, SPI, I^{2}C, SIO)
  - 42 multiplexed general purpose I/O pins



The XYZ Node Architecture

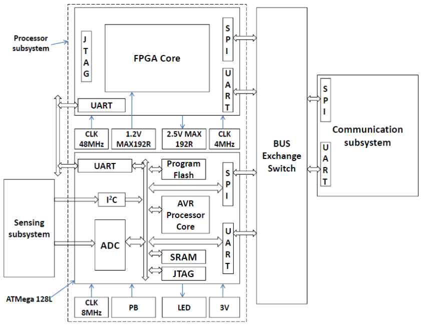
- The communication subsystem is connected to the processing subsystem through a SPI interface
  - CC2420 *RF transceiver*
  - when an RF message has been successfully received, the *SPI interface* enables the radio to wake up a sleeping processor
  - the processor subsystem controls the communication subsystem by either *turning it off* or putting it in *sleep mode*



The Hogthrob Node Architecture

- Designed for a specific application, namely, to *monitor pig production*
- Motivation:
  - monitors movements of a sow to predict onset of estrus
    - so that appropriate care can be given for pregnant sows
  - detecting cough or limping to monitor illness

The Hogthrob Node Architecture



The Hogthrob Node Architecture

- The processing subsystems consists of :
  - *microcontroller*
    - performs *less complex*, less energy intensive tasks
    - *initializes the FPGA and functions* as an external timer and an ADC converter to it
  - *Field Programmable Gate Array*
    - executes the sow monitoring application
    - coordinates the functions of the sensor node

The Hogthrob Node Architecture

- There are a number of interfaces supported by the processing subsystem, including
  - the I²C interface for the sensing subsystem
  - the SPI interface for the communication subsystem
  - the JTAG interface for in-system programmability and debugging
  - the serial (RS232) interface for interaction with a PC