Chapter 8: Power Management

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Power Management

- Energy is a scarce resource in WSNs for the following reasons:
  1. the *nodes are very small in size* to accommodate high-capacity power supplies compared to the complexity of the task they carry out
  2. it is *impossible to manually change*, replace, or recharge batteries - WSNs consist of a large number of nodes
  3. *the size of nodes* is still a *constraining* factor for renewable energy and self-recharging *mechanisms*
  4. *the failure of a few nodes* may cause the entire network to fragment prematurely

Power Management

- The problem of power consumption can be approached from *two angles*:
  - develop energy-efficient communication protocols
    - self-organization, medium access, and routing protocols
  - identify activities in the networks that are both wasteful and unnecessary then mitigate their impact
- Most inefficient activities are results of *non-optimal configurations* in hardware and software components:
  - e.g., a considerable amount of energy is wasted by an idle processing or a communication subsystem
  - a radio that aimlessly senses the media or overhears while neighboring nodes communicate with each other consumes a significant amount of power

Power Management

- Wasteful and unnecessary activities can be described as *local* or *global*
  - *e.g.,* some nodes exhausted their batteries prematurely because of *unexpected overhearing* of traffic that caused the communication subsystem to become operational for a longer time than originally intended
  - some nodes exhausted their batteries prematurely because they aimlessly *attempted to establish links with a network* that had become no longer accessible to them

Power Management

- A *dynamic power management (DPM)* strategy ensures that power is consumed economically
  - the strategy can have a *local* or *global* scope, or *both*
  - *a local DPM strategy* aims to
    - minimize the power consumption of *individual nodes*
      - by providing each subsystem with the amount of power that is sufficient to carry out a task at hand
      - when there is no task to be processed, the DPM strategy forces some of the subsystems to operate at the *most economical power mode* or puts them into a *sleeping mode*
  - *a global DPM strategy* attempts to
    - minimize the power consumption of *the overall network* by defining a *network-wide sleeping state*

Power Management

- Synchronous sleeping schedule
  - let individual nodes *define* their own sleeping schedules
  - *share* these schedules with their neighbors to enable a coordinated sensing and an efficient inter-node communication
  - the *problem* is that neighbors need to synchronize time as well as schedules and the process is energy intensive
- Asynchronous sleeping schedule
  - let individual nodes *keep* their sleeping schedules to themselves
  - a node that initiates a communication should *send a preamble* until it receives an acknowledgment from its receiving partner
  - avoids the needs to synchronize schedules
  - it can have *a latency side-effect* on data transmission

Power Management

- In both approaches, individual nodes *wake up periodically*
  - to determine whether there is a node that wishes to communicate with them
  - to process tasks waiting in a queue

Power Management

- Fundamental premises about Embedded systems:
  - predominantly event-driven
  - experience non-uniform workload during operation time
- DPM refers to selectively shutting-off and/or slowing-down system components that are idle or underutilised
- A policy determines the type and timing of power transitions based on system history, workload and performance constraints

Power Management

- It has been described in the literature as a linear optimisation problem
  - the objective function is the expected performance
    - related to the expected waiting time and the number of jobs in the queue
  - the constraint is the expected power consumption
    - related to the power cost of staying in some operation state and the energy consumption for the transfer from one server state to the next

Outline

- ***Local Power Management Aspects***
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Local Power Management Aspects

- The first step is the *understanding of how power is consumed* by the different subsystems of a wireless sensor node, this knowledge enables
  - wasteful activities to be avoided and to frugally budget power
  - one to estimate the overall power dissipation rate in a node and how this rate affects the lifetime of the entire network
- In the following subsections, a mode detail observation into the different subsystems of a node is made

Outline

- Local Power Management Aspects
  - ***Processor Subsystem***
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Processor Subsystem

- Most existing processing subsystems employ microcontrollers, notably
  - Intel's StrongARM and Atmel's AVR
- These microcontrollers can be configured to operate at *various power modes*
  - *e.g.,* the ATmega128L microcontroller has *six different power modes*:
    - idle, ADC noise reduction, power save, power down, standby, and extended standby

Processor Subsystem

Sleep Mode	Active clock domains					Oscillators		Wake up sources					
	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Osc Enabled	INT7	TWI Addr. Match	Timer	EEPROM Ready	ADC	Other I/O
Idle			X	X	X	X	X	X	X	X	X	X	X
ADC noise red.				X	X	X	X	X	X	X	X	X	
power down								X	X				
Power save					x		x	x	x	x			
standby						x		x	x				
Ext. standby						x	x	x	x	x			

Source: ATMEL, Atmega 128: 2008

Processor Subsystem

- The *idle* mode stops the CPU
  - while allowing the SRAM, Timer/Counters, SPI port and interrupt system to continue functioning
- The *power down* mode saves the registers' content
  - while freezing the oscillator and disabling all other chip functions until the next interrupt or Hardware Reset
- In the *power-save* mode, the asynchronous timer continues to run
  - allowing the user to maintain a timer base while the remaining components of the device enter into a sleep mode

Processor Subsystem

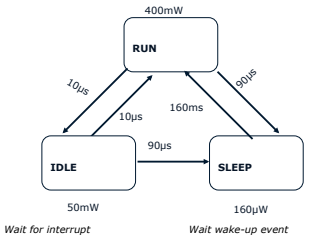
- The *ADC noise reduction* mode stops the CPU and all I/O modules
  - *except* the asynchronous *timer* and the *ADC*
  - the aim is to minimize switching noise during ADC conversions
- In *standby* mode, a crystal/resonator oscillator runs while the remaining hardware components enter into a sleep mode
  - this allows very fast start-up combined with low power consumption
- In *extended standby* mode, both the main oscillator and the asynchronous timer continue to operate

Processor Subsystem

- Additional to the above configurations, the processing subsystem can operate with different supply voltages and clock frequencies
- *Transiting* from one power mode to another also has its own power and latency cost

Processor Subsystem

- Power state machine for the StrongARM-1100 processor



Source: Benini, 2000

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - ***Communication Subsystem***
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Communication Subsystem

- The *power consumption* of the communication subsystem can *be influenced by several aspects*:
  - the modulation type and index
  - the transmitter's power amplifier and antenna efficiency
  - the transmission range and rate
  - the sensitivity of the receiver
- These aspects can be *dynamically reconfigured*

Communication Subsystem

- Determining the most efficient active state operational mode is *not a simple decision*
  - *e.g.,* the power consumption of a transmitter may *not* necessarily be reduced by simply reducing the transmission rate or the transmission power
    - the reason is that there is a *tradeoff* between *the useful power required for data transmission* and the *power dissipated in the form of heat* at the power amplifier
  - usually, the dissipation power (heat energy) *increases* as the transmission power *decreases*
  - in fact most commercially available transmitters operate efficiently at one or two transmission power levels
    - below a certain level, the efficiency of the power amplifier *falls drastically*

Communication Subsystem

- In some cheap transceivers, even when at the maximum transmission power mode, more than *60%* of the supply DC power is dissipated in the form of useless *heat*
- For example, the *Chipcon CC2420* transceiver has eight programmable output power levels ranging from –24 dBm to 0 dBm

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - ***Bus Frequency and RAM Timing***
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Bus Frequency and RAM Timing

- The processor subsystem consumes power when it interacts with the other subsystems via *the internal high-speed buses*
- The specific amount depends on the *frequency* and *bandwidth* of the communication
- These two parameters can be optimally configured depending on the *interaction type*, but *bus protocol timings* are usually optimized for particular bus frequencies
- Moreover, bus controller drivers require to be notified *when bus frequencies change* to ensure optimal performance

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - ***Active Memory***
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Active Memory

- It is made up of *capacitor-transistor pairs (DRAM)*
  - arranged in rows and columns, each row being a single memory bank
  - have to be recharged periodically in order to store data
- The *refresh interval*
  - a measure of the number of rows that must be refreshed
  - a *low refresh interval* corresponds to a *high clock frequency*
  - a *higher refresh interval* corresponds to a *low clock frequency*

Active Memory

- Consider two typical values: 2K and 4K
  - 2K: *refreshes more cells* at a low interval and completes the process faster, thus it *consumes more power*
  - 4K: *refreshes less cells* at a slower frequency, but it *consumes less power*
- A DRAM memory unit can be configured to operate in one of the following *power modes*:
  - temperature-compensated self-refresh mode
  - partial array self-refresh mode
  - power down mode

Active Memory

- *Temperature-compensated self-refresh mode*
  - the standard refresh rate of a memory unit can be adjusted according to its ambient *temperature*
- *Partial array self-refresh mode*
  - the self-refresh rate can be *increased* if the entire memory array is not needed to store data
  - the refresh operation can be *limited* to the portion of the memory array in which data will be stored
- *Power down mode*
  - if no actual data storage is required, the supply voltage of most or the entire on-board memory array can be *switched off*

Active Memory

- The *RAM timing* is another parameter that affects the power consumption of the memory unit
  - it refers to the *latency* associated with accessing the memory unit
    - before a processor subsystem accesses a particular cell in a memory, it should first *determine* the particular *row* or bank
    - then *activate* the row with a *row access strob (RAS)* signal
    - the activated row can be accessed until the data is exhausted
    - the time required to activate a row in a memory is t_{RAS}, which is relatively small but could *impact the system's stability* if set incorrectly

Active Memory

- The *delays* between the activation of a row (a cell) and the writing of data into or reading of data from the cell is given as t_{RCD}
- This time can be short or long, depending on how the memory cell is accessed
- If it is accessed sequentially, it is insignificant
- If it is accessed in a random fashion, the current active row must first be *deactivated* before a new row is activated
- In this case, t_{RCD} can cause significant *latency*

Active Memory

- A memory cell is activated through a *column access strob (CAS)*
  - the delay between the CAS signal and the availability of valid data on the data pins is called *CAS latency*
  - *low CAS latency* means *high performance* but also *high power consumption*
  - the time required to terminate one row access and begin the next row access is t_{RP}
  - the time required to switch rows and select the next cell for reading, writing, or refreshing is expressed as $t_{RP} + t_{RCD}$
  - the duration of time required between the active and precharge commands is called t_{RAS}
    - it is a measure of how long the processor must wait before the next memory access can begin

Active Memory

Parameter	Description
RAS	Row Address Strobe or Row Address Select
CAS	Column Address Strobe or Column Address Select
t_{RAS}	A time delay between the precharge and activation of a row
t_{RCD}	The time required between RAS and CAS access
t_{CL}	CAS latency
t_{RP}	The time required to switch from one row to the next row
t_{CLK}	The duration of a clock cycle
Command rate	The delay between Chip Select (CS)
Latency	The total time required before data can be written to or read from memory

Table 8.2 Parameters of RAM timing

Active Memory

- When a RAM is accessed by *clocked logic*, the times are generally rounded up to the *nearest clock cycle*
  - for example, when accessed by a 100-MHz processor (with 10 ns clock duration), a 50-ns SDRAM can perform the first read in 5 clock cycles and additional reads within the same page every 2 clock cycles
  - this is generally described as "5 – 2 – 2 – 2" timing

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - ***Power Subsystem***
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Power Subsystem

- The power subsystem supplies power to all the other subsystems
- It consists of
  - *the battery*
  - *the DC – DC converter*
    - it provides the right amount of supply voltage to each individual hardware component
    - by transforming the main DC supply voltage into a suitable level
    - the transformation can be a *step-down (buck)*, a *step-up (boost)*, or an *inversion (flyback)* process, depending on the requirements of the individual subsystem

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - ***Battery***
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Battery

- A wireless sensor node is powered by *exhaustible batteries*
  - the main factor affect the quality of these batteries is *cost*
- Batteries are specified by a rated current capacity, *C*, expressed in *ampere-hour*
  - this quantity describes the rate at which a battery discharges without significantly affecting the prescribed supply voltage
  - as the discharge rate *increases*, the rated capacity *decreases*
  - most portable batteries are rated at 1*C*
    - this means a 1000 mAh battery provides 1000mA for 1 hour, if it is discharged at a rate of 1*C*
      - *e.g.*, at a rate of 0.5*C*, providing 500mA for 2 hours
      - at a rate of 2*C*, 2000mA for 30 minutes

Battery

- In reality, batteries perform at *less than the prescribed rate*. Often, the *Peukert Equation* is applied to quantifying the capacity offset

$$t = \frac{C}{I^n} \qquad \text{Equation (8.1)}$$

  - where *C* is the theoretical capacity of the battery expressed in ampere-hours
  - *I* is the current drawn in Ampere (A)
  - *t* is the time of discharge in seconds
  - *n* is the Peukert number, a constant that directly relates to the internal resistance of the battery

Battery

- The value of the Peukert number indicates how well a battery performs under continuous heavy currents
  - a value close to *1 indicates* that the *battery performs well*
  - the higher the number, the more capacity is lost when the battery is discharged at high currents
- Figure 8.3 shows how the effective battery capacity can be reduced at high and continuous discharge rates
  - by intermittently using the battery, it is possible during quiescent periods to increase the diffusion and transport rates of active ingredients and to match up the depletion created by excessive discharge
  - because of this potential for recovery, the capacity reduction can be undermined and the operating efficiency can be enhanced

Battery

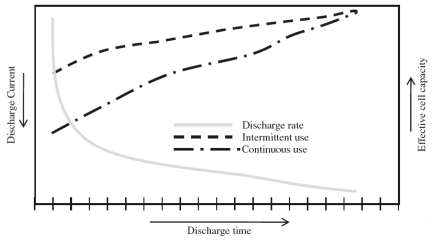


Figure 8.3 The Peukert curve displaying the relationship between the discharging rate and the effective voltage. The *x*-axis is a time axis

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - ***DC – DC Converter***
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

DC – DC Converter

- The DC – DC converter transforms one voltage level into another
- *The main problem* is its *conversion efficiency*
- A typical DC – DC converter consists of
  - a power supply
  - a switching circuit
  - a filter circuit
  - a load resistor

DC – DC Converter

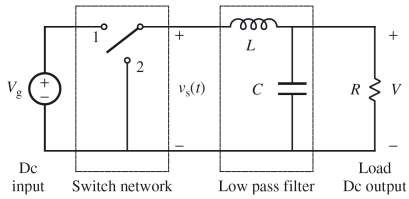


Figure 8.4 A DC – DC converter consisting of a supply voltage, a switching circuit, a filter circuit, and a load resistance

DC – DC Converter

- In the figure 8.4, the circuit consists of *a single-pole, double-throw (SPDT) switch*
  - SPDT is connected to a DC supply voltage, V_g
  - considering the inductor, L, as a short circuit
  - the capacitor, C, as an open circuit for the DC supply voltage
  - the switch's output voltage, $V_s(t) = V_g$ when the switch is in position 1
  - $V_s(t) = 0$ When it is in position 2
  - varying the position of the switch at a frequency, f_s yields a periodically varying square wave, $v_s(t)$, that has a period $T_s = 1/f_s$
  - $v_s(t)$ can be expressed by a duty cycle D
  - D describes the fraction of time that the switch in position 1, ($0 \leq D \leq 1$)

DC – DC Converter

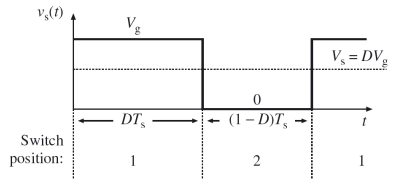


Figure 8.5 The output voltage of a switching circuit of a DC – DC converter

DC – DC Converter

- A DC – DC converter is realized
  - by employing active switching components
  - such as diodes and power MOSFETs
- Using the *inverse Fourier transformation*
  - the DC component of $v_s(t)$ (V_s) is described as:

$$V_s = \frac{1}{T_s}\int_0^{T_s} v_s(t)dt = DV_g \qquad \text{Equation (8.2)}$$

  which is the average value of $v_s(t)$
- In other words, the integral value represents the area under the waveform of Figure 8.5 for a single period, or the height of V_g multiplied by the time T_s
- It can be seen that the switching circuit reduces the DC component of the supply voltage by a factor that equals to the duty cycle, *D* Since $0 \leq D \leq 1$ holds, the expression: $V_s \leq V_g$ is true

DC – DC Converter

- The switching circuit consumes power
  - due to the existence of a resistive component in the switching circuit, there is power dissipation
  - the efficiency of a typical switching circuit is between 70 and 90%
- In addition to the desired DC voltage, $v_s(t)$ also contains undesired harmonics of the switching frequency, f_s
  - these harmonics must be removed so that the converter's output voltage $v(t)$ is essentially equal to the DC component $V = V_s$
  - for this purpose, a DC – DC converter employs a *lowpass filter*

DC – DC Converter

- In Figure 8.4, a first-order *LC* lowpass filter is connected to the switching circuit
  - the filter's cutoff frequency is given by:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \qquad \text{Equation (8.3)}$$

  - the cutoff frequency, f_c, should be sufficiently *less* than the switching frequency, f_s
  - so that the lowpass filter allows only the DC component of $v_s(t)$ to pass
- In an ideal filter, there is no power dissipation
  - because the passive components (inductors and capacitors) are energy storage components
- Subsequently, the DC–DC converter produces a DC output voltage
  - its magnitude is controlled by the duty cycle, D, using circuit elements that (ideally) do not dissipate power

DC – DC Converter

- The conversion ratio, $M(D)$, is defined as the ratio of the DC output voltage, V , to the DC input voltage, V_g, under a steady-state condition:

$$M(D) = \frac{V}{V_g} \qquad \text{Equation (8.4)}$$

- For the buck converter shown in Figure 8.4, $M(D) = D$
- Figure 8.6 illustrates the linear relationship between the input DC voltage, V_g and the switching circuit's duty cycle, D

DC – DC Converter

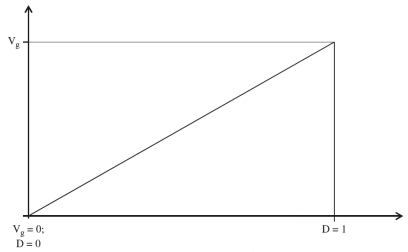


Figure 8.6 A linear relationship between a DC supply voltage and the duty cycle of a switching circuit

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- ***Dynamic Power Management***
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Dynamic Power Management

- Once the design time parameters are fixed, a *dynamic power management (DPM)* strategy attempts to
  - minimize the power consumption of the system by dynamically defining the most economical operation conditions
  - this condition takes the requirements of *the application*, *the topology of the network*, and *the task arrival rate of the different subsystems* into account.
- Different approaches to a DPM strategy can be categorized:
  1. *dynamic operation modes*
  2. *dynamic scaling*
  3. *energy harvesting*

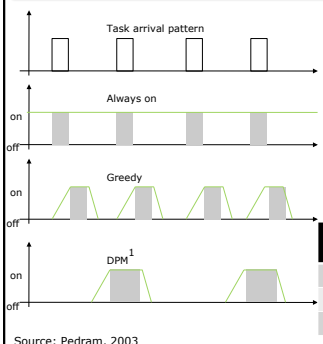
Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - ***Dynamic Operation Modes***
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Dynamic Operation Modes

- In general, a subcomponent or a part it can have n different power modes
  - if there are x hardware components that can have n distinct power consumption levels, a DPM strategy can define $x \times n$ different power mode configurations, P_n
- The *task* of the DPM strategy is:
  - select the optimal configuration that matches the activity of a wireless sensor node
- Two associated *challenges*:
  1. transition between the different power configurations costs extra power
  2. a transition has an associated delay and the potential of missing the occurrence of an interesting event

Selective Switching



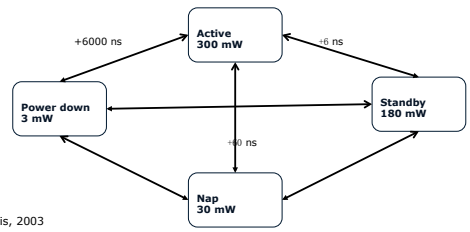
Parameter	Value
P_{on}	10 W
P_{off}	0 W
$P_{on \to off}$	10 W
$P_{off \to on}$	40 W
$t_{on \to off}$	1 s
$t_{off \to on}$	2 s
t_R	25 s

Policy	Energy	Avg. Latency
Always on	250 J	1 s
Reactive greedy	240 J	3 s
Power-aware	140 J	2.5 s

Source: Pedram, 2003

Dynamic Operation Modes

- Memory access



Source: Ellis, 2003

Selective Switching

Power Mode	StrongARM	Memory	MEMS & ADC	RF
P_0	Sleep	Sleep	Off	Off
P_1	Sleep	Sleep	On	Off
P_2	Sleep	Sleep	On	RX
P_3	Idle	Sleep	On	RX
P_4	Active	Active	On	TX, RX

Source: Sinha and Chandrakasan, 2001

Dynamic Operation Modes

Configuration	Processor	Memory	Sensing subsystem	Communication subsystem
P_0	Active	Active	On	Transmitting/receiving
P_1	Active	On	On	On (transmitting)
P_2	Idle	On	On	Receiving
P_3	Sleep	On	On	Receiving
P_4	Sleep	On	On	Off
P_5	Sleep	On	Off	Off

Table 8.3 Power saving configurations
DPM strategy with six different power modes: {*P0*, *P1*, *P2*, *P3*, *P4*, *P5*}

Dynamic Operation Modes

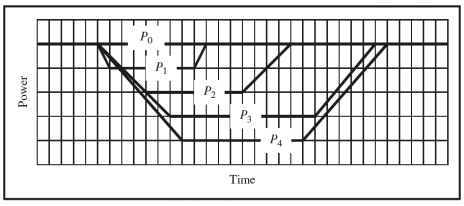


Figure 8.7 Transition between different power modes and the associated transition costs

Dynamic Operation Modes

- The decision for a particular power mode depends on
  - the anticipated task in the queues of the different hardware components
- Failure to realistically estimate future tasks can cause a node to miss interesting events or to delay in response
- In a WSN, the events outside of the network cannot be modeled as deterministic phenomena
  - e.g., a leak in a pipeline; a pestilence in a farm
  - no need for setting up a monitoring system
- An accurate event arrival model enables a DPM strategy to decide for the right configuration that has a *long duration* and *minimal power consumption*

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - ***Transition Costs***
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Transition Costs

- Suppose:
  - each subsystem of a wireless sensor node operates in just two different power modes only, it can be either *on* or *off*
  - moreover, assume that the transition from *on* to *off* does not have an associated power cost
  - but the reverse transition (from *off* to *on*) has a cost in terms both of power and a time delay
  - these costs are justified if the power it saves in the *off* state is large enough
  - in other words, the amount of the *off* state power is considerably large and the duration of the *off* state is long
  - it is useful to quantify these costs and to set up a transition threshold

Transition Costs

- Suppose:
  - the minimum time that a subsystem stays in an *off* state is t_{off}
  - the power consumed during this time is P_{off}
  - the transition time is $t_{\text{off,on}}$
  - the power consumed during the transition is $p_{\text{off,on}}$
  - the power consumed in an *on* state is P_{on}. Hence:

$$P_{off} \cdot t_{off} + P_{off,on} \cdot t_{off,on} \geq P_{on} \cdot \left(t_{off} + t_{off,on} \right) \quad \text{Equation (8.5)}$$

  - therefore, t_{off} is justified if:

$$t_{off} \geq \max\left(0, \frac{\left(P_{on} - P_{off,on} \right) t_{off,on}}{P_{on} - P_{off}} \right) \quad \text{Equation (8.6)}$$

Transition Costs

- Equations (8.5) and (8.6) can describe a subsystem with n distinct operational power modes
  - in this case a transition from any state i into j is described as $t_{i,j}$
  - hence, the transition is justified if Equation (8.7) is satisfied

$$t_j \geq \max\left(0, \frac{\left(P_i - P_{j,k} \right) t_{i,j}}{P_i - P_j} \right) \quad \text{Equation (8.7)}$$

  - where t_j is the duration of the subsystem in state j

Transition Costs

- If the transition cost from a higher power mode (*on*) to a lower power mode (*off*) is not negligible
  - the energy that can be saved through a power transition (from state i to state j , $E_{\text{saved}_{ij}}$) is expressed as:

$$E_{saved,j} = P_i \cdot \left(t_j + t_{i,j} + t_{j,i} \right) - \left(P_{i,j} \cdot t_{i,j} + p_{j,i} \cdot t_{j,i} + p_j \cdot t_j \right)$$

Equation (8.8)

- If the transition from state i to state j costs the same amount of power and time delay as the transition from state j to state i, it can be expressed as:

$$E_{saved,j} = P_i \cdot \left(t_j + t_{i,j} + t_{j,i} \right) - \left(\frac{P_i + P_j}{2} \right) \left(t_{i,j} + t_{j,i} \right) - \left(P_i + P_j \right) t_j$$

Equation (8.9)

Transition Costs

- Obviously, the transition is justified if $E_{\text{saved},j} > 0$. This can be achieved in three different ways, by:
  1. increasing the gap between P_i and P_j
  2. increasing the duration of state j, (t_j)
  3. decreasing the transition times, $t_{j,i}$

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - ***Dynamic Scaling***
  - Task Scheduling
- Conceptual Architecture
  - Architectural Overview

Dynamic Scaling

- Dynamic voltage scaling (*DVS*) and dynamic frequency scaling (*DFS*) aim to:
  - adapt the performance of the processor core when it is in the active state
- In most cases, the tasks scheduled to be carried out by the processor core do *not require its peak performance*
- Some tasks are completed ahead of their deadline and the processor enters into a low-leakage idle mode for the remaining time
- In Figure 8.8, even though the two tasks are completed ahead of their schedule, the processor still runs at peak frequency and supply voltage - *wasteful*

Dynamic Scaling

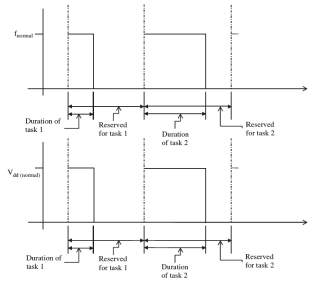


Figure 8.8 A processor subsystem operating at its peak performance

Dynamic Scaling

- In Figure 8.9 the performance of the processing subsystem is *adapted* (reduced) according to the criticality of the tasks it processes
  - each task is stretched to its planned schedule while the supply voltage and the frequency of operation are reduced
- The basic building blocks of the processor subsystem are *transistors*
  - they are classified into *analog* and *digital (switching) transistors*
  - depending on their operation regions (namely, cut-off, linear, and saturation)

Dynamic Scaling

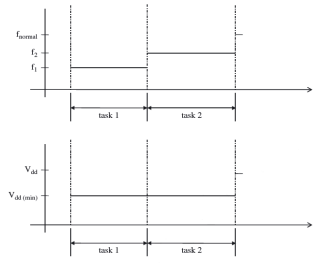


Figure 8.9 Application of dynamic voltage and frequency scaling

Dynamic Scaling

- An *analog transistor* (amplifier)
  - operates in the linear amplification region
  - there is a linear relationship between the input and the output of the transistor. This is expressed as:

$$v_{out} = \frac{A}{1 - AB} v_{in} \qquad \text{Equation (8.10)}$$

  - where A is the gain of the amplifier
  - B is a term that determines the portion of the output that should be fed back to the input in order to stabilize the amplifier

Dynamic Scaling

- A *digital (switching) transistor*
  - operates in either the cutoff or the saturation region
  - makes the relationship between the input and the output voltage nonlinear
  - that is how the *zeros* and *ones* of a digital system are generated, represented or processed
  - the transition duration from the cutoff to the saturation region determines how good a transistor is as a switching element
    - in an ideal switching transistor, the transition takes place in no time In practical transistors, the duration is greater than zero
    - the quality of the processor depends on *the switching time*

Dynamic Scaling

- The *switching time* in turn depends on
  - the cumulative capacitance effect created between the three joints of the transistors
  - Figure 8.10 displays a typical NAND gate made up of CMOS transistors
- A capacitor is created by two conductors
  - two conductors are separated by a dielectric material
  - there is a potential difference between the two conductors
- The capacitance of a capacitor is
  - positive proportional to the cross-sectional area of the conductors
  - inversely proportional to the separating distance

Dynamic Scaling

- In a switching transistor
  - a capacitance is created at the contact points of the source, gate and drain
  - affecting the transistor's switching response
  - the switching time can be approximated by the following equation:

$$t_{delay} = \frac{C_s \cdot V_{dd}}{I_{d_{sat}}} \qquad \text{Equation (8.11)}$$

  - where C_s is the source capacitance, V_{dd} is the biasing voltage of the drain, and $I_{d_{sat}}$ is the saturation drain current

Dynamic Scaling

- Switching costs energy and the magnitude of the energy depends
  - the operating frequency and the biasing voltage
  - Sinha and Chandrakasan (2001) provide a first-order approximation that can be expressed as:

$$E(r) = CV_0 2T_s f_{ref} r\left[\frac{V_t}{V_0} + \frac{r}{2} + \sqrt{r\frac{V_t}{V_0} + \left(\frac{r}{2}\right)^2}\right] \qquad \text{Equation (8.12)}$$

    - where, C is the average switching capacitance per cycle
    - T_s is the sampling period; f_{ref} is the operating frequency at V_{ref}
    - r is the normalized processing rate ($r = f/f_{ref}$)
    - $V_0 = (V_{ref} - V_t)^2 / V_{ref}$ with V_t being the threshold voltage
- It can be deduced that
  - reducing the operating frequency linearly reduces the energy cost
  - reducing the biasing voltage reduces the energy cost quadratically

Dynamic Scaling

- However, these two quantities cannot be reduced beyond a certain limit
  - for example, the minimum operating voltage for a CMOS logic to function properly was first derived by Swanson and Meindl (1972)
  - it is expressed as:

$$V_{dd,\text{limit}} = 2 \cdot \frac{kT}{q} \cdot \left[1 + \frac{C_{fs}}{C_{ox} + C_d}\right] \cdot \ln\left(1 + \frac{C_d}{C_{ox}}\right) \qquad \text{Equation (8.13)}$$

    - where $C_f s$ is the surface state capacitance per unit area
    - C_{ox} is the gate-oxide capacitance per unit area
    - C_d is the channel depletion region capacitance per unit area
  - finding the optimal voltage limit requires a tradeoff between the switching energy cost and the associated delay

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - ***Task Scheduling***
- Conceptual Architecture
  - Architectural Overview

Task Scheduling

- In a dynamic voltage and frequency scaling, the DPM strategy *aims* to
  - autonomously determine the magnitude of the biasing voltage (V_{dd})
  - the clock frequency of the processing subsystem
- The decision for a particular voltage or frequency is based on:
  - *the application latency requirement*
  - *the task arrival rate*
  - ideally, these two parameters are adjusted so that a task is completed "just in time" - the processor does not remain idle and consume power unnecessarily

Task Scheduling

- Practically, Idle cycles cannot be completely avoided
  - the processor's workload cannot be known *a priori*
  - the estimation contains error
- Comparison between an ideal and real dynamic voltage scaling strategies is shown in Figure 8.11

Task Scheduling

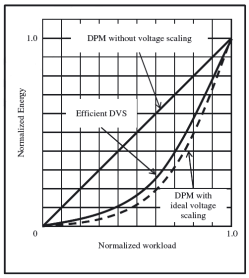


Figure 8.11 Application of dynamic voltage scaling based on workload estimation (Sinha and Chandrakasan (2001)

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- ***Conceptual Architecture***
  - Architectural Overview

Conceptual Architecture

- A conceptual architecture for enabling a DPM strategy in a wireless sensor node should address three essential concerns:
  1. in attempting to optimize power consumption, how much is the extra workload that should be produced by the DPM itself?
  2. should the DPM be a centralized or a distributed strategy?
  3. if it is a centralized approach, which of the subcomponents should be responsible for the task?

Conceptual Architecture

- A typical DPM strategy:
  - monitors the activities of each subsystem
  - makes decisions concerning the most suitable power configuration
    - optimizes the overall power consumption
    - this decision should take the application requirements
- An accurate DPM strategy requires bench marking to estimate the *task arrival and processing rate*

Conceptual Architecture

- A DPM strategy can be
  - *central approach*
  - *distributed approach*
- Advantage of a centralized approach
  - it is easier to achieve a global view of the power consumption of a node and to implement a comprehensible adaptation strategy
  - a global strategy can add a computational overhead on the subsystem that does the management
- Advantage of a distributed approach
  - scales well by authorizing individual subsystems to carry out local power management strategies

Conceptual Architecture

- Local strategies may *contradict* with global goals
- Given the relative simplicity of a wireless sensor node and the quantifiable tasks that should be processed, *most existing power management strategies advocate a centralized solution*

Conceptual Architecture

- In case of a centralized approach, the main question is
  - which subsystems is responsible for handling the task ---- the processor subsystem or the power subsystem
- The *power subsystem*
  - has *complete information* about the energy reserve of the node
  - the *power budget* of each subsystem
  - *but* it requires *vital information* from the processing subsystems
    - the task arrival rate
    - priority of individual tasks
  - it needs to have some *computational capability*
    - presently available power subsystems do not have these characteristics

Conceptual Architecture

- Most existing architectures
  - place the processor subsystem at the center
  - all the other subsystems communicate with each other through it
  - t he operating system runs on the processing subsystem, managing, prioritizing and scheduling tasks
- Subsequently, *the processing subsystem*
  - have more comprehensive knowledge about the activities of all the other subsystems
  - these characteristics make it appropriate place for executing a DPM

Outline

- Local Power Management Aspects
  - Processor Subsystem
  - Communication Subsystem
  - Bus Frequency and RAM Timing
  - Active Memory
  - Power Subsystem
    - Battery
    - DC – DC Converter
- Dynamic Power Management
  - Dynamic Operation Modes
    - Transition Costs
  - Dynamic Scaling
  - Task Scheduling
- Conceptual Architecture
  - ***Architectural Overview***

Architectural Overview

- The DPM strategy should not affect the system's stability
- The application requirements should be satisfied
  - the quality of sensed data and latency
- A WSN is deployed for a specific task
  - that task does not change, or changes only gradually
- The designer of a DPM has at his or her disposal the architecture of the wireless sensor node, the application requirements, and the network topology to devise a suitable strategy

Architectural Overview

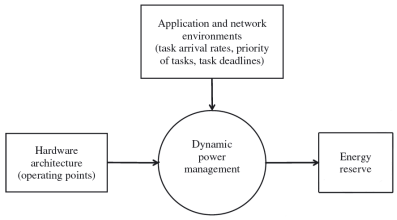


Figure 8.12 Factors affecting a dynamic power management strategy

Architectural Overview

- The system's hardware architecture
  - it is the basis for defining multiple operational power modes and the possible transitions between them
- A local power management strategy
  - it defines rules to describe the behavior of the power mode transition
  - according to a change in the activity of the node; or
  - based on a request from a global power management scheme; or
  - based on a request from the application
- This (see Figure 8.13) can be described as a circular process consisting of three basic operations
  - energy monitoring
  - power mode estimation
  - task scheduling

Architectural Overview

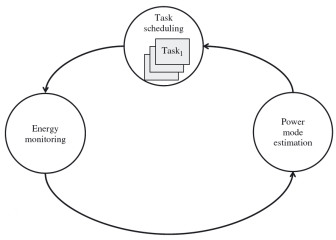


Figure 8.13 An abstract architecture for a dynamic power management strategy

Architectural Overview

- Figure 8.13 illustrates
  - how dynamic power management can be thought of as a machine that moves through different states in response to different types of events
  - tasks are scheduled in a task queue, and the execution time and energy consumption of the system are monitored
  - depending on how fast the tasks are completed, a new power budget is estimated and transitions in power modes
  - the DPM strategy decides the higher level of operating power mode
    - in case of a deviation in the estimated power budget from the power mode

Architectural Overview

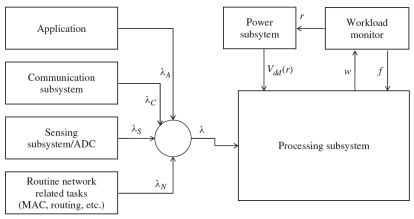


Figure 8.14 A conceptual architecture of a dynamic voltage scaling.
(This architecture is the modified version of the one proposed by Sinha and Chandrakasan in (Sinha and Chandrakasan 2001))

Architectural Overview

- Figure 8.14 shows
  - an implementation of the abstract architecture of Figure 8.13 to support dynamic voltage scaling
  - the processing subsystem
    - receives tasks from the application, the communication subsystem, and the sensing subsystem
    - it handles internal tasks pertaining to network management
      - such as managing a routing table and sleeping schedules
  - each of these sources produces a task at a rate of λ_i
  - the overall task arrival rate, λ, is the summation of the individual tasks arrival rates, $\lambda = \sum \lambda_i$
  - the *workload monitor* observes λ for a duration of τ seconds and predicts the task arrival rate for the next β seconds

Architectural Overview

- The estimated task arrival rate is represented by r in the figure
- Based on the newly computed task arrival rate r, the processing subsystem estimates the supply voltage and the clock frequency it requires to process upcoming tasks