POWER MANAGEMENT IN EMBEDDED SYSTEMS

Pravir Kumar(200601086)
Dorav Kumar(200601206)
Team O
FLOW OF PRESENTATION

- Introduction
- Motivation Behind Power Management
- Why efficiency in Power Consumption is More Important for Embedded Devices?
- Approaches To Power Management
- Power management during Design of the system
- Power management at run-time
- Other measures
- Power Management ICs
- Major areas of concern
- References
Pi – I/O devices
uC – microcontroller
PM - Power Management
INTRODUCTION (2/2)

- Power Management: A set of H/W and S/W techniques used to minimize the power consumption of a device while meeting its performance requirements.
Motivation Behind Power Management[1]

- To enhance battery life
- Lower heat dissipation to increase system stability
- Reduced cooling requirements
- Thus reduces operating cost for energy and cooling
- Reduces the impact on the environment
- Reduced noise
Why efficiency in power consumption is more important for embedded devices?

- Embedded applications generally have constraint over size.
  - This in turn puts a limit over the battery size.
  - Lesser surface area implies less amount of heat dissipation so more heating
- Cost constraint forces to have lower cooling budget
- For speed we have to switch to lower CMOS design technology (like 90nm, 45nm), increase static power dissipation due to leakage current
- Many embedded applications are installed in remote areas, making it difficult to change battery frequently
APPROACHES TO POWER MANAGEMENT[2]

- Power management during Design of the system
- Power management at run-time
- Other measures
Power Management during Design of System

- Don’t always go after very accurate component but select one which satisfies the requirement and also consumes less power.
  - If it is not a time critical system then select VLO(low power oscillator) in place of RTC. [3]

- Either select all components working at common input voltage range or effectively partition of separate voltage and clock domains according to need of components working at different input range[4]

- Minimize number of separate clock generator by using shared high fan-out clocks between the components

- Try to minimize output capacitance and loading on output pins.
POWER MANAGEMENT AT RUN-TIME (1/9)

- Turn off clock when not needed
- While booting a system if some components can be left un-powered until they are actually needed then it minimizes the power consumption
- Utilize interrupts at different I/O devices instead of continuous polling by the processor
Try to use sleep modes as much as possible such that all functions which are not needed should be disabled [5]

- Lowers static power consumption
  \[ P_{\text{static}} = V_{dd} \times I_q \]
- Power Down: Everything is shut down, including the clock source except for external interrupts and watch dog
- Power Save: External clock is turned on to keep track of time
- Idle: Processor is halted but both external and internal interrupts are enabled
Efficient use of Brown Out Detector[3]:-

- Generally BOD is always kept ON for continuous measurement of input voltage level, even if the microcontroller (MCU) is in sleep mode.

- For low power consumption disable the BOD after entering into the sleep mode and enable it before the MCU is awake.
Dynamic Power management[6]:

- Now a days many embedded applications such as video and audio playback and gaming run for a longer duration of time. The ratio of processor’s running time to idle time is significantly larger in these applications. So, for these types of applications, we need such techniques which save power during run time.
POWER MANAGEMENT AT RUN-TIME (5/9)

- We need to adopt such type of technique in our embedded application, which can make processor run at different clock frequency depending on the requirement. An MPEG video player needs higher performance than an MP3 audio player. So, the same processor which is used to run both the application can be adjusted to work at a lower frequency while playing audio without compromising for quality of sound.

- Follows the principle “completing a task before its deadline is an inefficient use of energy”.

- Based on the fact that frequency of a processor implemented in CMOS is proportional to the supply voltage.
POWER MANAGEMENT AT RUN-TIME (6/9)

- $P_{\text{dynamic}} = CV_{DD}^2 f_c$
  
  Reducing the supply voltage to slow the clock frequency reduces the dynamic power consumption in a quadratic fashion

- But also increases the run time of the application

- If increased run time isn’t a problem, we can adopt this technique.
POWER MANAGEMENT AT RUN-TIME (7/9)

- Different techniques to scale down operating frequency and supply voltage[2]:

  1) Static Benchmarking technique: While booting the system, scale down frequency and voltage to such a level which enables the application to fully meet its requirements, but minimizes excess capacity of the system.
POWER MANAGEMENT AT RUN-TIME (8/9)

2) Interval Based Scheduling:

- When an application is running, periodically take samples of processor’s utilization and accordingly adjust the frequency and voltage.
- Uses the dynamic variability of the application’s processing needs
- Scores better over static benchmarking technique
- Scheduling based upon current and previous processor’s utilization by the application
- Not able to anticipate future needs of application
- Not acceptable for systems with hard real-time deadlines
POWER MANAGEMENT AT RUN-TIME (9/9)

3) Dynamic scheduling:

- Variation in the processor’s frequency and voltage according to the predicted workload
- Uses comparison of processor’s utilization time for task completed with the worst-case execution time and deadline of the next task to predict workload
OTHER MEASURES

- Use of power aware high level language compilers
  - These compilers try to minimize the operating system calls.

- Efficient use of code compression techniques to reduce the memory access.
POWER MANAGEMENT SOLUTIONS FROM INDUSTRY

- Texas Instruments Power Management IC
- AVR Pico-Power technology

Extensive research is going on to decrease power consumption at device level.
Major Areas of Concern (1/2)

- Trade off between static (SPM) and dynamic power management (DPM) [8]
  - DPM slows down the processor’s speed to lower dynamic power consumption of the processor.
  - This results in more static power consumption of the processor and more energy consumption of peripheral devices since active period of peripheral devices are also lengthened to reduce time duration for sleep mode.
  - To reduce static power consumption, we may want to shorten the active task as much as possible, resulting in more dynamic power consumption.

Conclusion - There is a need to find the balance between SPM and DPM.
MAJOR AREAS OF CONCERN (2/2)

- Trade off between processing speed and power consumption
  - With increasing need of higher processing speed, the designer will have to move to lower CMOS design technology.
  - But with reduction in size of technology the leakage current increases significantly [9].
REFERENCES

3. Innovative Techniques for Extremely Low Power Consumption with 8-bit Microcontrollers  Arne Martin Holberg, AVR Project Manager and Asmund Saetre, AVR Marketing Manager (Pp. 5-6).
7. Low Power Intel® Architecture for Small Form Factor Devices, an article by  by Ram Chary, Pat A. Correia, Raviprakash Nagaraj, and James Song.
9. Lec_03_CMOS_Inverter accessible at(\\daiictpdc\\Lecture\\Sushanta Kumar\\SEM_2008-09\\BTECH\\EL322_Introduction_To_VLSI\\Lecture Notes).
THANK YOU