Networked Embedded Systems Lab
Prof. Marco Zimmerling

Introduction to Embedded Systems – WS 2022/23

Exercise 5: Low Power I

Task 1: Dynamic Voltage Scaling and Dynamic Power Management

Suppose that the power consumption $P(f)$ of a given CMOS processor at frequency f is:

$$P(f) = \left(10\left(\frac{f}{100\,\text{MHz}}\right)^3 + 20\right)\ \text{mW}$$

To reduce the power consumption, the execution frequency of the processor is adjusted using dynamic voltage scaling. The maximum (minimum) supported frequency $f_{\max}$ ($f_{\min}$) is 1000 MHz (50 MHz). Assume that frequency switching has negligible overhead, and that the processor can operate at any frequency between 50 MHz and 1000 MHz.

In addition, dynamic power management is applied to further reduce the power consumption. Assume that in sleep mode the processor does not consume any power (0 mW) and that modes can be switched without delays. Changing from run mode to sleep mode does not require any energy (0 J). However, going from sleep mode to run mode requires additional energy, namely 30 µJ.

The system has three jobs to execute:

	arrival time	deadline	execution cycles
τ_1	0 ms	2 ms	100000
τ_2	2 ms	6 ms	100000
τ_3	6 ms	7 ms	80000

Initially (at 0 ms) the processor is in the run mode. The processor is also required to be in run mode at time 7 ms.

(a) What does the constant term in the power consumption $P(f)$ represent? Where does this term come from?

(b) The energy consumption to execute C cycles is $\frac{C}{f} \cdot P(f)$. There is a *critical frequency* f_{crit} between 50 MHz and 1000 MHz at which the energy consumption per cycle ($\frac{P(f)}{f}$) is minimized. What is the critical frequency f_{crit} of this processor?

(c) When the processor is idle at frequency $f_{\min}$ for t seconds, the consumed energy is $P(f_{\min}) \cdot t$. The *break-even time* is defined as the minimum idle interval, for which it is worthwhile for the processor to go into sleep mode. What is the break-even time of the processor?

(d) A *workload-conserving* schedule is defined as a schedule that is always executing a job when the ready queue is not empty. For the three jobs above, provide the workload-conserving schedule that minimizes the energy consumption without violating the timing constraints. For this subquestion, all tasks are executed at *critical frequency* f_{crit}. What is the energy consumption of this schedule?

(e) Is there another workload-conserving schedule without timing constraints violations for the three jobs that has a lower energy consumption than the schedule in (d)? There are no restrictions at what frequency tasks have to be executed. If so, provide the schedule, otherwise prove the optimality of the schedule in (d).

(f) Does a schedule for the three jobs without timing constraints violations exist that is not workload-conserving but consumes less energy than the optimal workload-conserving schedule? If so, provide the schedule, otherwise prove the optimality of the workload-conserving schedules.

Task 2: Dynamic Voltage Scaling for Real-Time Tasks

Consider a set $\mathbf{J}$ of aperiodic jobs as illustrated in Table 1 below. The system is considered to have negligible threshold voltage and a power consumption of $P(f) = (\frac{f}{10^6 \text{Hz}})^3$ W. The processor can operate at any frequency in the range of $[10^5, 10^7]$ Hz.

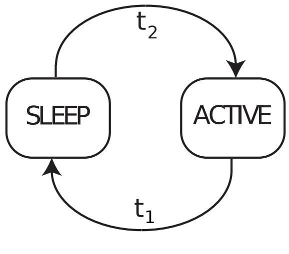
Job ID	1	2	3	4	5	6
arrival time (ms)	0	2	6	8	10	11
absolute deadline (ms)	8	12	10	20	25	15
cycles ($\times 10^3$)	1	6	8	2	5	3

Table 1: Job set $\mathbf{J}$

1. What is the optimal schedule to minimize the energy consumption without deadline misses for the set $\mathbf{J}$? What is the energy consumption of the resulting schedule? [**Hint**: Apply the YDS algorithm]
2. Suppose that we do not know a job before it arrives to the system. What is the schedule for the set $\mathbf{J}$ that the online YDS algorithm generates?

Task 3: Dynamic Power Management

Consider a micro-controller of type TI-MLP230, that consumes $P_{active} = 1.2$ mW in ACTIVE mode and P_{sleep} = 90.0 μW in SLEEP mode. Interrupts occur at times $t = iT$, $i \in \{0, 1, 2, ...\}$ to notify the processor of the arrival of a new task. Each task requires time t_{task} for processing. The transition from SLEEP mode to ACTIVE mode takes t_2; the transition from ACTIVE mode to SLEEP mode takes t_1 (see Figure below). We assume that the micro-controller cannot perform computations during these transitions. For simplicity, we assume that the power changes continuously and linearly during these transitions. The system is deployed with an energy source (battery) with $E_{bat} = 25.0$ kJ.

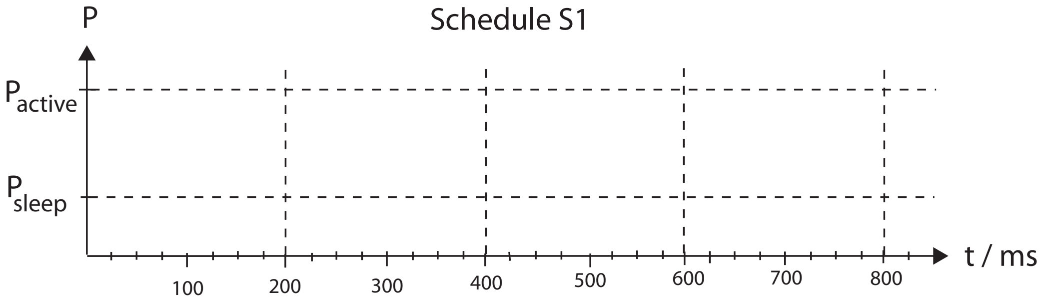


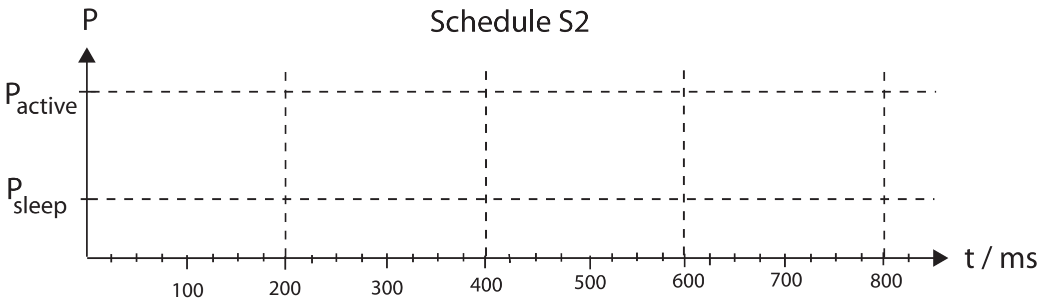
1. Initially, we neglect the transition times ($t_1 = t_2 = 0$). Assume that after each interrupt, the processor changes to ACTIVE mode and executes a task for $t_{task} = 2$ s. After the execution of the task, the processor returns to SLEEP mode. What is the maximal number of task executions N_{max} that can be

supported so the deployed system has a life time of 5 years? What condition does the period T have to satisfy to enable this deployment?

2. Next, assume that $t_1 = 25$ ms, $t_2 = 75$ ms, $t_{task} = 100$ ms, and $T = t_1 + t_2 + t_{task}$. At time $t = 0$ the processor is in SLEEP mode. Please sketch the power consumption function $P(t)$ of the processor in the given diagrams for the following two cases:

   - **Schedule S1:** Transition to the ACTIVE mode follows directly after an interrupt. After the task execution, i.e., after t_{task}, the processor immediately returns to the SLEEP mode.
   - **Schedule S2:** If the processor is in SLEEP mode when an interrupt occurs, then the transition to ACTIVE mode happens immediately. After the task execution, the processor decides whether to return to SLEEP mode or to remain in ACTIVE mode in order to execute the next task without delay when the next interrupt occurs. The processor makes this decision aiming at minimizing the energy consumption.





3. Compute the energy difference ΔE, which can be saved on average per period T (200 ms) when Schedule S2 is used instead of Schedule S1.

4. Consider a **Schedule S-OPT**, for which the following condition must hold: the task denoted by the $i-$th interrupt must have finished by the time the $(i + 1)-$th interrupt occurs. S-OPT must serve the arriving tasks with the minimum possible energy under the above condition. Draw the function $P(t)$ for Schedule S-OPT in the given diagram below. In addition, compute the energy difference $\Delta E'$, which can be saved on average per period T when Schedule S-OPT is used instead of Schedule S1.