Networked Embedded Systems Lab
Prof. Marco Zimmerling

Introduction to Embedded Systems – WS 2022/23

Sample Solution to 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.

Solution to Task 1:

(a) The constant term in the power consumption represents the minimal power the processor consumes while on. A non-negligible threshold voltage V_t and junction leakage are significant contributors to the minimal power. The former impacts the transistor input voltage required to turn on the transistor. Gate-oxide leakage also contributes to the static power but it is orders of magnitude smaller than the other two components.

(b) The power consumption monotonically decreases with decreasing frequency ($P(f)$ scales with f^3). However, when considering the energy consumption per cycle there are two opposing trends. Firstly, the energy consumption corresponding to the first term in the $P(f)$ divided by f decreases with decreasing frequency. Secondly, the energy consumption corresponding to the second term in $P(f)$ divided by f increases with decreasing frequency. The latter is a result of the increasing execution time of a cycle when decreasing the frequency, the processor has to be on for a longer time.

To compute the critical frequency, we have to minimize $\frac{P(f)}{f}$. Let f_{norm} be f normalized to 100 MHz. The first derivative of $\frac{P(f_{norm})}{f_{norm}}$ is $20f_{norm} - \frac{20}{f_{norm}^2}$, which is 0 when $f_{norm} = 1$. Therefore, the critical frequency is $f_{crit} = f_{norm} \cdot 100\,\text{MHz} = 100\,\text{MHz}$.

(c) Going into sleep mode must provide sufficient energy saving to compensate for the additional energy consumption (overhead) of 30 µJ + 0 J incurred by changing modes.

$$\text{Energy\_idle\_min\_frequency} \geq \text{Energy\_Overhead} + \text{Energy\_sleep}$$

The processor does not consume any power when in sleep mode (Energy_sleep $= 0$). The Energy_Overhead is $30\,\mu\text{J} + 0\,\text{J} = 30\,\mu\text{J}$. And the processor consumes Energy_idle_min_frequency $= P(f_{min}) \cdot t$ energy when idling for t seconds at the lowest frequency.

Thus, the above inequality is

$$\begin{aligned} P(f_{min}) \cdot t &\geq 30\,\mu\text{J} \\ t &\geq \frac{30\,\mu\text{J}}{(10 \cdot 0.5^3 + 20)\text{mW}} \\ t &\geq 1.412\,\text{ms} \end{aligned}$$

The break-even time $t_{\text{break-even}}$ is the minimal time for which the above inequality holds and therefore $t_{\text{break-even}} = 1.412\,\text{ms}$.

(d) The schedule is depicted in Figure 1. The energy consumption of the schedule is

$$\begin{aligned} E &= \frac{C_{\tau_1}}{f_{crit}} \cdot P(f_{crit}) + t_{idle,1} \cdot P(f_{min}) + \frac{C_{\tau_2}}{f_{crit}} \cdot P(f_{crit}) + E_{sleep} + E_{\text{mode change}} + \frac{C_{\tau_3}}{f_{crit}} \cdot P(f_{crit}) \\ &\quad + t_{idle,2} \cdot P(f_{min}) \\ &= 0.001\,\text{s} \cdot 30\,\text{mW} + 0.001\,\text{s} \cdot 21.25\,\text{mW} + 0.001\,\text{s} \cdot 30\,\text{mW} + 0\,\text{J} + 30\,\mu\text{J} + 0.0008\,\text{s} \cdot 30\,\text{mW} \\ &\quad + 0.0002\,\text{s} \cdot 21.25\,\text{mW} \\ &= 0.1395\,\text{mJ} \end{aligned}$$

Note that in the time interval $[1, 2]$ we do not go into sleep mode as the interval is smaller than the break-even time. During the idle time intervals $[1, 2]$ and $[6.8, 7]$, we select the minimum frequency f_{min}. Since no task is executed then, this choice minimizes power consumption.

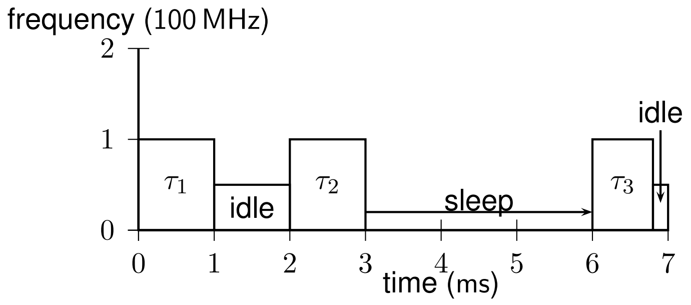


Figure 1: Solution to Question 1(d)

(e) Yes, the idea is to use the convex nature of the power consumption to slow down execution of tasks τ_1 and τ_3 to avoid idle times after their executions. Thus, even though we execute tasks below the critical frequency, the overall energy consumption is lower. The frequency at which τ_1, respectively τ_3, are executed is $f_{\tau_1} = \frac{C_{\tau_1}}{t_{\tau_1}}$, respectively $f_{\tau_3} = \frac{C_{\tau_3}}{t_{\tau_3}}$. The schedule is depicted in Figure 2. The energy consumption of the schedule is

$$\begin{aligned} E &= t_{\tau_1} \cdot P(f_{\tau_1}) + \frac{C_{\tau_2}}{f_{crit}} \cdot P(f_{crit}) + E_{sleep} + E_{\text{mode change}} + t_{\tau_3} \cdot P(f_{\tau_3}) \\ &= 0.002\,\text{s} \cdot 21.25\,\text{mW} + 0.001\,\text{s} \cdot 30\,\text{mW} + 0\,\text{J} + 30\,\mu\text{J} + 0.001\,\text{s} \cdot 25.12\,\text{mW} \\ &= 0.127\,62\,\text{mJ} \end{aligned}$$

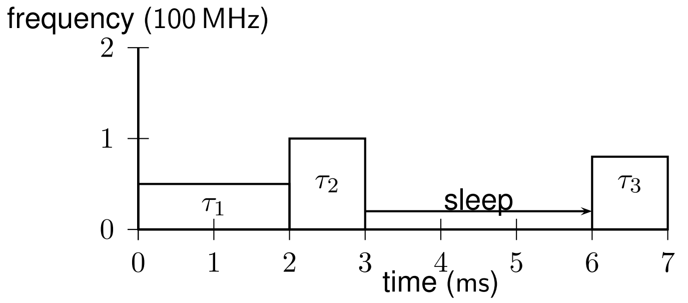


Figure 2: Solution to Question 1(e)

(f) Yes, the idea is to *batch* the inactive time into one block that the processor will spend in sleep mode, and execute task τ_1 with critical frequency. This illustrates that to conserve energy, workload conserving strategies are not necessarily the best. The schedule is depicted in Figure 3. The energy consumption of the schedule is

$$\begin{aligned} E &= \frac{C_{\tau_1}}{f_{crit}} \cdot P(f_{crit}) + E_{sleep} + E_{\text{mode change}} + \frac{C_{\tau_2}}{f_{crit}} \cdot P(f_{crit}) + t_{\tau_3} \cdot P(f_{\tau_3}) \\ &= 0.001\,\text{s} \cdot 30\,\text{mW} + 0\,\text{J} + 30\,\mu\text{J} + 0.001\,\text{s} \cdot 30\,\text{mW} + 0.001\,\text{s} \cdot 25.12\,\text{mW} \\ &= 0.115\,120\,\text{mJ} \end{aligned}$$

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.

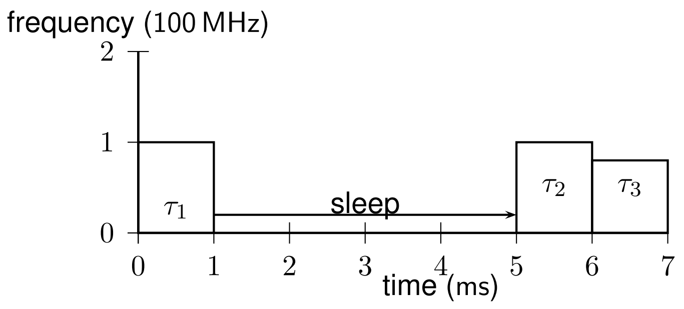


Figure 3: Solution to Question 1(f)

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 $\mathbf{J}$

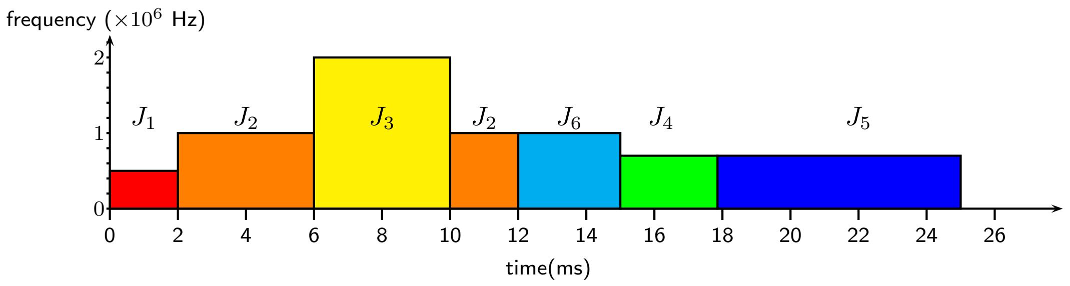
1. What is the optimal schedule to minimize the energy consumption without deadline misses for the set $\mathbf{J}$? What is the energy consumption of the resulting schedule? [**Hint**: Apply the YDS algorithm]
2. Suppose that we do not know a job before it arrives to the system. What is the schedule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

Solution to Task 2:

1. The critical intervals with the highest intensity at consecutive steps of the YDS algorithm are summarized in the following table. Note that the time intervals are defined with respect to the revised inputs (arrival times, deadlines of jobs) for each step. Additionally, if multiple intervals have the same (highest) intensity, by convention we select the one that enables scheduling the largest number of jobs.

Step #	Critical Interval (ms)	Intensity ($\times 10^6$)	Scheduled Tasks
1	[6,10]	2.0	{3}
2	[2,11]	1.0	{2,6}
3	[2,12]	0.7	{4,5}
4	[0,2]	0.5	{1}

The derived schedule is shown below and has an energy consumption of 44.68 mJ.

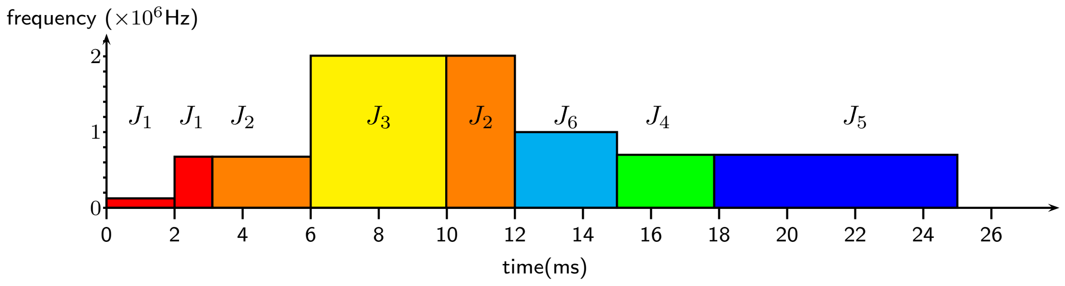


Job ID	1	2	3	4	5	6
speed ($\times 10^6$ Hz)	0.5	1.0	2.0	0.7	0.7	1.0

2. In the online version of YDS, the schedule is continuously updated. The new critical interval and intensity have to be determined whenever a job arrives and at the end of a critical interval. If the arrival of a new job does not change the critical interval, the jobs to be executed and the intensity do not change. The following table shows the time points at which the schedule is re-evaluated and the corresponding new critical interval and intensity if they changed. The table also shows all jobs that are considered (jobs that the system is aware of but has not completed yet) and the jobs that are scheduled to execute.

Time (ms)	Critical Interval (ms)	Intensity ($\times 10^6$)	Considered Jobs	Scheduled Jobs
0	[0,8]	$1/8 = 0.125$	{1}	{1}
2	[2,12]	$(0.75+6)/10 = 0.675$	{1,2}	{1,2}
6	[6,12]	$(8+4.05)/6 = 2.0083$	{2,3}	{2,3}
8	critical interval unchanged		{2,3,4}	
10	critical interval unchanged		{2,4,5}	
11	critical interval unchanged		{2,4,5,6}	
12	[12,15]	$3/3 = 1$	{4,5,6}	{6}
15	[15,25]	$(2+5)/10 = 0.7$	{4,5}	{4,5}

The final schedule is as follows:



Job ID	1		2		3	4	5	6
speed ($\times 10^6$ Hz)	0.125	0.675	0.675	2.0083	2.0083	0.7	0.7	1
time (ms)	0-2	2-3.111	3.111-6	9.983-12	6-9.983	15-17.857	17.857-25	12-15

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.

$$\begin{tikzcd} \boxed{\text{SLEEP}} \arrow[r, bend left, "t_2"] & \boxed{\text{ACTIVE}} \arrow[l, bend left, "t_1"] \end{tikzcd}$$

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 $t_{task} = 2\,\text{s}$. 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 $t_1 = 25\,\text{ms}$, $t_2 = 75\,\text{ms}$, $t_{task} = 100\,\text{ms}$, 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.

Solution to Task 3:

1. First we express the life time L in terms of the maximal number of task executions N_{max} and period T:

   $$L = N_{max} \cdot T$$

   Then we set up an equation for the total energy that is consumed during the lifetime L:

   $$\begin{aligned} E &= N_{max} \cdot (t_{task} \cdot P_{active} + (T - t_{task}) \cdot P_{sleep}) \\ &= N_{max} \cdot (t_{task} \cdot P_{active} + (\frac{L}{N_{max}} - t_{task}) \cdot P_{sleep}) \\ &= N_{max} \cdot (t_{task} \cdot P_{active} - t_{task} \cdot P_{sleep}) + L \cdot P_{sleep} \end{aligned}$$

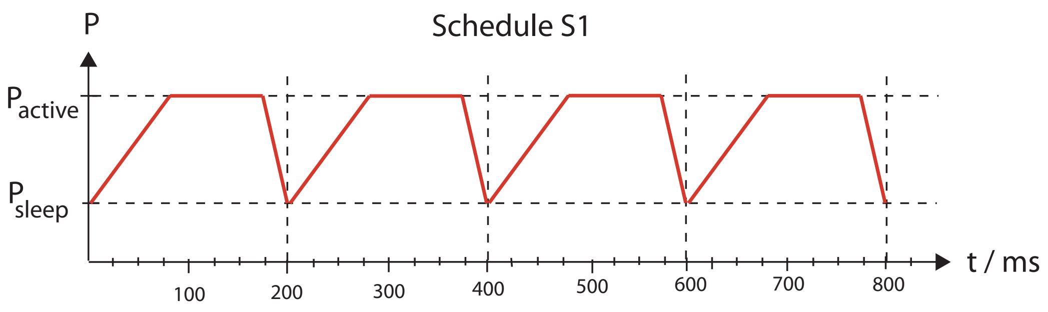
   Solving for N_{max} results in:

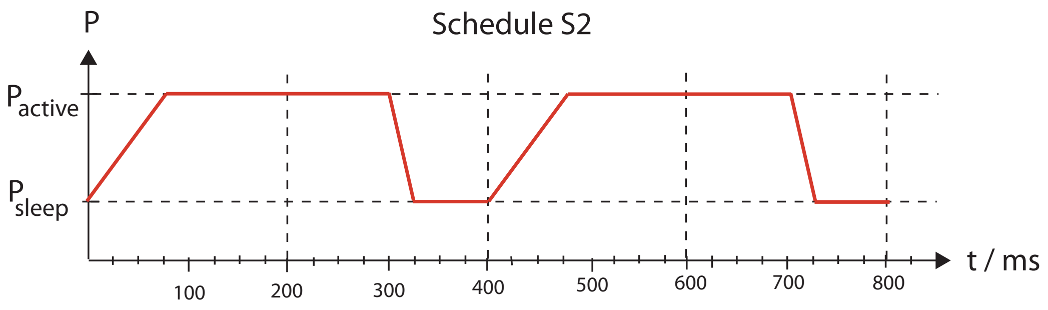
   $$N_{max} = \frac{E - L * P_{sleep}}{t_{task} * (P_{active} - P_{sleep})} = 4.87 * 10^6$$

   To support up to N_{max} executions, we can deduce that for T the following must hold:

   $$T \geq \frac{L}{N_{max}} = 32.39\,\text{s}$$

2. Function $P(t)$ for schedules S1 and S2, for $t \in [0, 800]$ is shown in the following diagram:





3. The energy consumption of the Schedule S2, has a periodicity of $2 \cdot T$. Therefore we compute the energy difference for the first two periods and then average those values to get the average energy difference per period T. In the 1st period:

$$\Delta E_1 = E_{S1} - E_{S2} = t_1 \cdot \frac{P_{sleep} + P_{active}}{2} - t_1 \cdot P_{active}$$

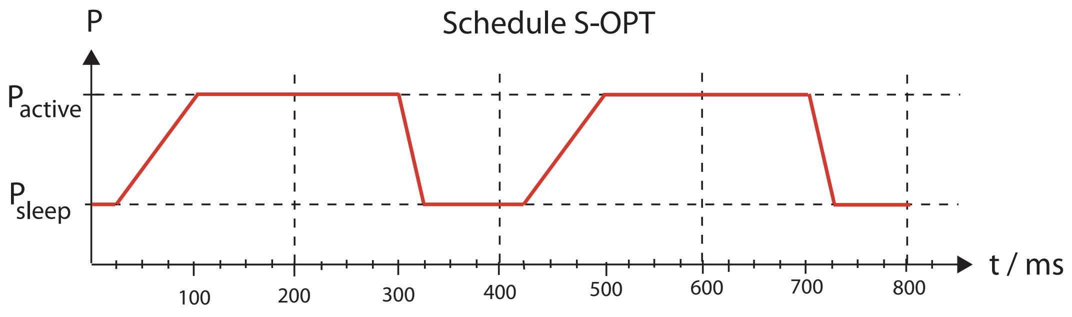
In the 2nd period:

$$\Delta E_2 = E_{S1} - E_{S2} = t_2 \cdot \frac{P_{active} + P_{sleep}}{2} - t_2 \cdot P_{sleep}$$

On average, the energy difference per period between S1 and S2 is:

$$\Delta E = \frac{\Delta E_1 + \Delta E_2}{2} = \frac{(t_2 - t_1)(P_{active} - P_{sleep})}{4} \approx 13.875\,\mu\text{J}$$

4. Function $P(t)$ for schedule S-OPT, for $t \in [0, 800]$ is shown in the following diagram:



The energy consumption of the Schedule S-OPT, has a periodicity of $2 \cdot T$. Therefore we compute the energy difference for the first two periods and then average those values to get the average energy difference per period T. In the 1st period:

$$\Delta E_1' = t_1 \cdot \frac{P_{active} + P_{sleep}}{2} - t_1 \cdot P_{sleep}$$

In the 2nd period:

$$\Delta E_2' = t_2 \cdot \frac{P_{active} + P_{sleep}}{2} - t_2 \cdot P_{sleep}$$

On average, the energy difference per period between S1 and S-OPT is:

$$\Delta E' = \frac{\Delta E_1 + \Delta E_2}{2} = \frac{(t_2 + t_1)(P_{active} - P_{sleep})}{4} \approx 27.75\,\mu\text{J}$$