10.1 Timer overrun

10.2 Utilization

A programmer can analyze a task to estimate whether a timer-overrun exception might occur on a microcontroller. For example, suppose a microcontroller executes a single synchSM task, the task's actions always require 450 ms to execute, and the task's period is 1000 ms. Then clearly timer overrun won't occur—for each task tick every 1000 ms, the microcontroller spends 450 ms executing actions, and 550 ms waiting for the next tick. The microcontroller is said to be ***idle*** during that waiting time. On the other hand, if the task's period was 300 ms, then clearly timer overrun will occur because the next tick will come along before the 450 ms needed to complete the previous tick's actions.

Utilization is a value that a programmer can compute to predict whether timer overrun will occur. For a microcontroller executing a single task, utilization can be defined as:

```
utilization = time-per-task-tick / task-period
```

A task requiring 450 ms per tick and with a 1000 ms period has utilization of 450 ms / 1000 ms or 0.45. Utilization is normally presented as a percentage, such as 45%. If that same task had a period of 300 ms, utilization would be 450 ms / 300 ms, or 150%. Utilization greater than 100% indicates timer overrun will occur.

Note that utilization as defined above is just for analysis, and does not measure an *actual* usage of the microcontroller; actual usage clearly cannot exceed 100%.

Participation Activity 10.2.1: Basic utilization.

A microcontroller executes a single synchSM task. Each task execution requires T ms. The task's period is P ms. Determine the utilization for the following T and P values.

#	Question	Your answer
1	T is 50, P is 500. ✔ 10 50 / 500 is 0.1 or 10%. Every 500 ms, the processor spends 50 ms executing actions, and 450 ms waiting for the next tick.	10 %
2	T is 4, P is 200. ✔ 2 4 / 200 is 0.02 or 2%. For some embedded systems with simple actions, utilizations of well below 1%, such as 0.001%, are common.	2 %
3	T is 240, P is 200. ✔ 120 240 / 200 is 120%. Timer overrun will occur. Compute intensive embedded systems may have high utilizations.	120 %

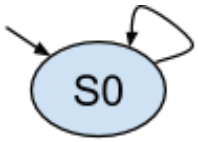
If utilization exceeds 100%, such as if a task requires 450 ms per tick but has a period of 300 ms (utilization is 150%), timer overrun will occur and some task ticks won't execute. The programmer then has several options, each either increasing the denominator or decreasing the numerator of the utilization calculation:

- Increase the task period. Increasing from 300 to 500 yields utilization of 450 / 500 or 90%. Of course, such increase has drawbacks like worse system latency, missing of some input events, etc.
- Reduce the time for actions per tick. This could be achieved by writing more efficient code, splitting a synchSM state into two states each doing half the actions, or removing some system functionality. Such reduction could also

be achieved by using a faster microcontroller or increasing a microcontroller's clock frequency.

Participation Activity 10.2.2: Decreasing utilization.

Consider the two synchSMs, each having a period of 100 ms and requiring 150 ms to execute the actions.




```
tmp1 = A * 12 * 12;
tmp2 = tmp1 * tmp1;
tmp3 = 15 * tmp2 + 15;
B = tmp3 + 400;
```


```
tmp1 = A * 12 * 12;
tmp2 = tmp1 * tmp1;
```

```
tmp3 = 15 * tmp2 + 15;
B = tmp3 + 400;
```

#	Question	Your answer
1	One way to decrease utilization is to increase the period from 100 to 200 microsec. ✔ Utilization will decrease from 150/100 or 150%, to 150/200 or 75%.	True False
2	One way to decrease utilization is to replace the first two statements with the one statement tmp2 = 20736 * A * A, assuming a compiler wouldn't already make that replacement. ✔ The replacement yields fewer instructions and thus faster execution, perhaps decreasing from 150 to 90, for example. However, note that modern compilers may make the replacement automatically.	True False
3	The change to the synchSM on the right does not impact utilization because the same actions must still be executed. ✔ *Each tick* executes fewer actions so utilization is reduced. If the time to execute S0's actions is reduced from 150 to 75 microsec, and S1 also requires 75 microsec, then the utilization is reduced from 150/100 or 150%, to 75/100 or 75%. Of course, system latency and the sampling rate are worsened.	True False
4	Increasing the microcontroller's clock frequency won't decrease utilization because the same actions must still executed.	True False

✔ Those actions will take less time. The task tick rate however was kept the same. So utilization will decrease.

10.3 WCET