Embedded Systems
LA7 (R5-R6)

Name: Ahmed ElShaarany

Date: 7/5/2015

P1: (100 points)

Swi1 Priority	Swi2 Priority	Swi2End - Swi1Start (us)
3	1	458
3	3	458
3	5	366

How does your measurement compare against the measurement done as part of original Lab 6 (ledToggle benchmark, single Swi)? Explain.

The measurement done as part of the original Lab 6 was for a single Swi from Swi_enter to Swi_exit was sometimes 153 usec and other times 183 usec.

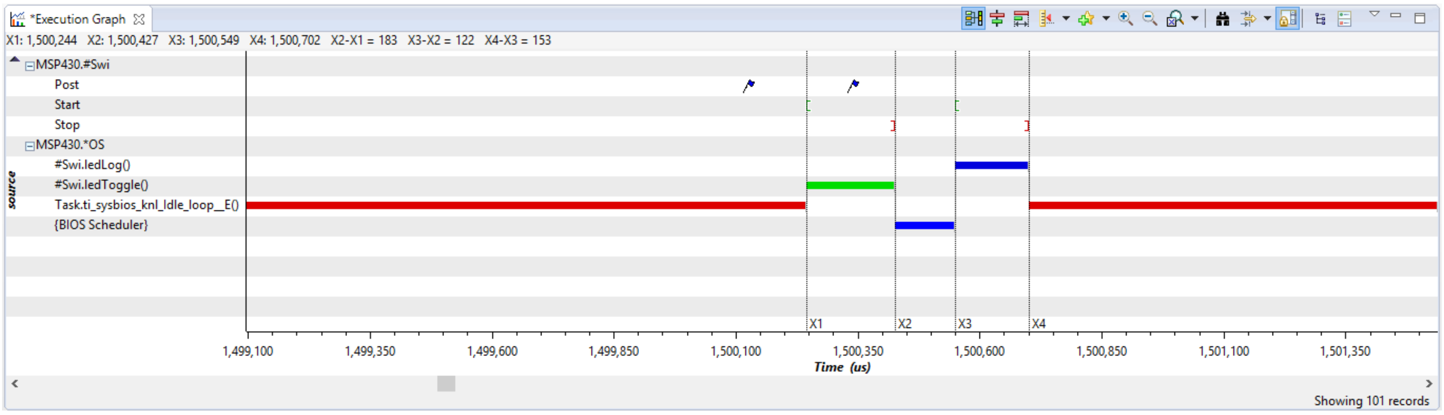
The original measurement is clearly much less than the measurement of the modified lab. The reason for this is that for the modified lab, we have split the Swi into two Swi's. This means that we will have more overhead caused by the scheduler. The scheduler now has to handle and switch between two Swi's instead of one and therefore, taking more time than the original lab.

How does the single Swi measurement you got compares against the value provided in the Lab manual? Explain.

The single Swi measurement I got was close to the one in the lab manual. The lab manual mentioned that the measurement is 183 usec cycles. The measurement I got was sometimes 183 usec and other times 153 usec. This is probably due to two scenarios. For the 153 usec measurement, the Hwi occurs during the idle state and hence, it doesn't interrupt the servicing of the Swi. This means that no extra overhead from the scheduler is incurred. On the other hand, for the 183 usec measurement, the Hwi probably occurs during the servicing of the Swi, and since the Hwi has a higher priority, the Swi is interrupted and the Hwi will get serviced. This causes the scheduler to spend time switching between the Hwi and Swi and therefore, this measurement is longer.

Explain similarities or differences in behavior (scheduling) and measurements in the above table.

For the first case, Swi1 will be posted first, therefore, Swi1 will run and post Swi2. Swi2 is now ready to run. Since Swi2 has lower priority, it will not interrupt Swi1's run. Therefore, Swi1 will run and finish, the scheduler will run, and then Swi2 will run and finish. The figure below depicts this explanation.

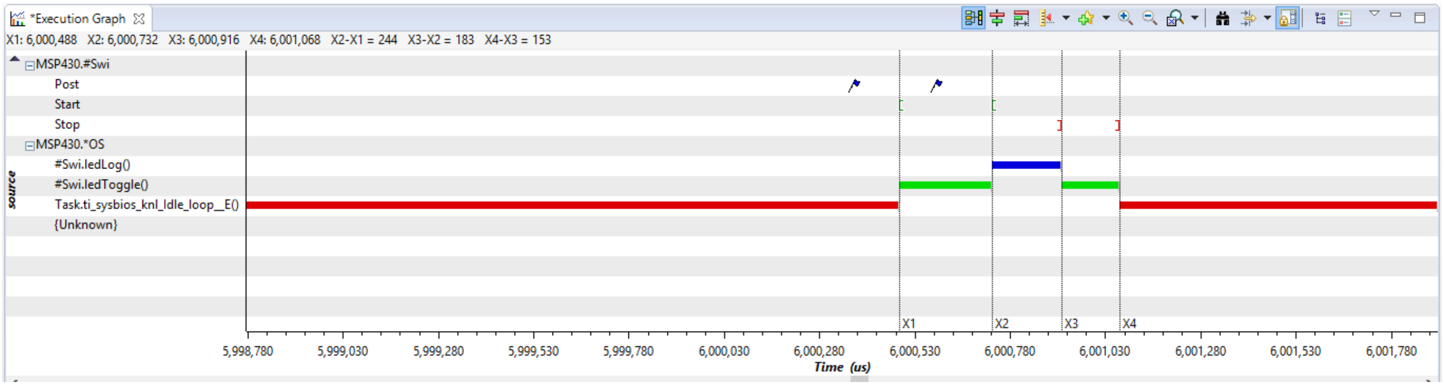


```
X1: 1,500,244   X2: 1,500,427   X3: 1,500,549   X4: 1,500,702   X2-X1 = 183   X3-X2 = 122   X4-X3 = 153
```

Showing 101 records

For the second case, both Swi1 and Swi2 have the same priority. Therefore, they will be served on a FIFO basis and the timing graph is similar to the first case. (i.e. Swi1 starts and finishes because it was posted first and then Swi2 will get served)

For the third case, Swi2 has higher priority than Swi1, so during the serving of Swi1, Swi2 is posted and since it has a higher priority than Swi1, the running thread (Swi1), gets interrupted and Swi2 runs until it finishes. After that, Swi1 gets to run again until it finishes. This is depicted by the figure below.



```
X1: 6,000,488   X2: 6,000,732   X3: 6,000,916   X4: 6,001,068   X2-X1 = 244   X3-X2 = 183   X4-X3 = 153
```