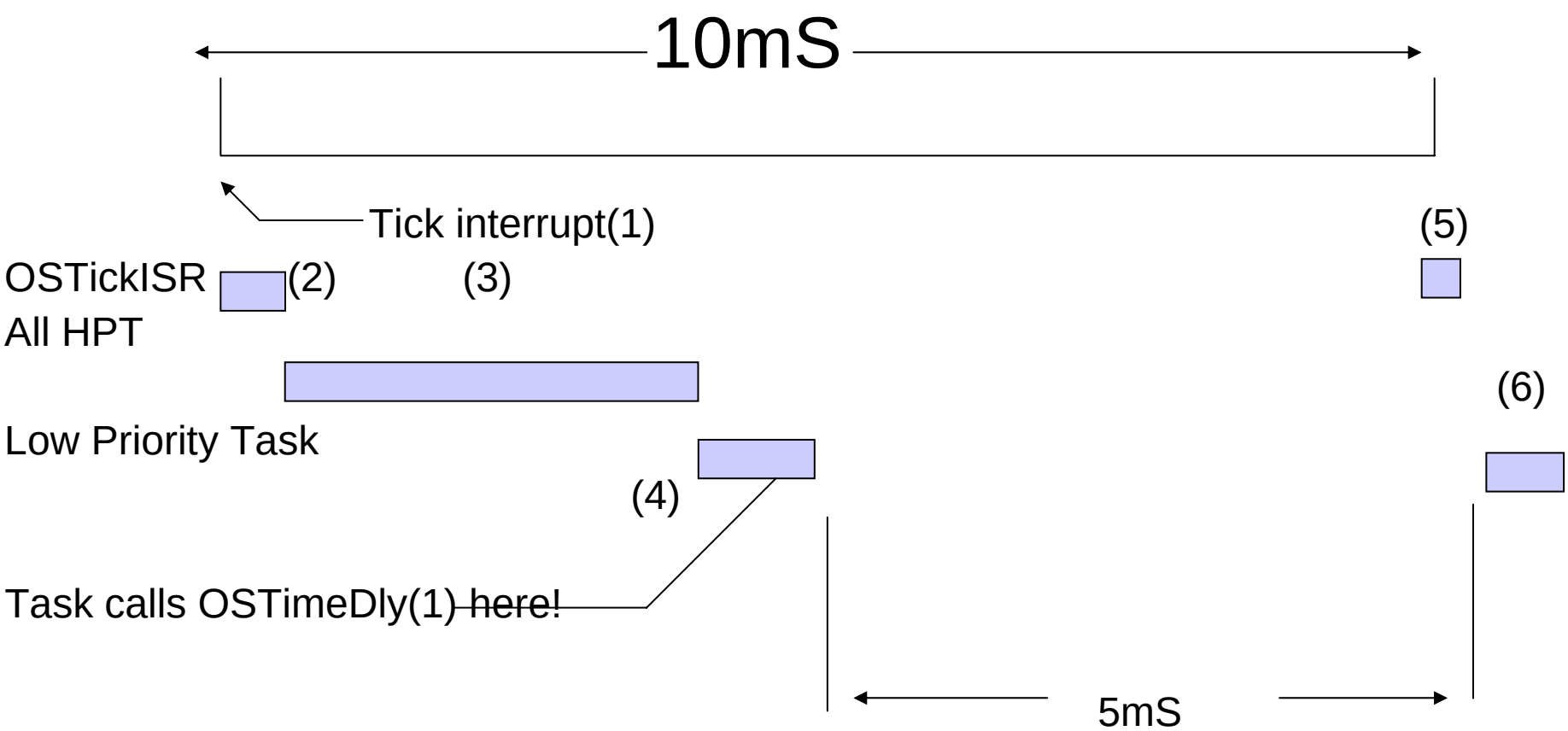
chapter 5

Time Management

- OSTimeDly()
- OSTimeDlyHMSM()
- OSTimeDlyResume()
- OSTimeGet()
- OSTimeSet()

Delay Resolution



OSTimeDly()

- DELAY TASK 'n' TICKS (n from 0 to 65535)
- Description:

  This function is called to delay execution of the currently running task until the specified number of system ticks expires.

  No delay will result If the specified delay is 0. If the specified delay is greater than 0 then, a context switch will result.

OSTimeDly()

```
void  OSTimeDly (INT16U ticks)
{
#if OS_CRITICAL_METHOD
   == 3 OS_CPU_SR  cpu_sr;
#endif
    if (ticks > 0) {OS_ENTER_CRITICAL();
        if ((OSRdyTbl[OSTCBCur->OSTCBY] &= ~OSTCBCur->OSTCBBitX)
            == 0)
        {
          OSRdyGrp &= ~OSTCBCur->OSTCBBitY;
        }
        OSTCBCur->OSTCBDly = ticks;
   OS_EXIT_CRITICAL();
        OS_Sched();
         }
}
```

OSTimeDlyHMSM()

- Description:

  This call allows you to specify the delay time in HOURS, MINUTES, SECONDS and MILLISECONDS instead of ticks.

OSTimeDlyHMSM()

```
#if OS_TIME_DLY_HMSM_EN > 0
INT8U  OSTimeDlyHMSM (INT8U hours, INT8U minutes, INT8U seconds,
   INT16U milli)
{
    INT32U ticks;
    INT16U loops;


    if (hours > 0 || minutes > 0 || seconds > 0 || milli > 0) {
        if (minutes > 59) {
            return (OS_TIME_INVALID_MINUTES); }
        if (seconds > 59) {
            return (OS_TIME_INVALID_SECONDS);
        }
        if (milli > 999) {
            return (OS_TIME_INVALID_MILLI);
        }
```

OSTimeDlyHMSM()

```
ticks = ((INT32U)hours * 3600L + (INT32U)minutes * 60L +
   (INT32U)seconds) * OS_TICKS_PER_SEC
               + OS_TICKS_PER_SEC * ((INT32U)milli + 500L /
   OS_TICKS_PER_SEC) / 1000L;
        loops = (INT16U)(ticks / 65536L); ticks = ticks % 65536L; */
        OSTimeDly((INT16U)ticks);
        while (loops > 0) {
            OSTimeDly(32768);
            OSTimeDly(32768);
            loops--;
        }
        return (OS_NO_ERR);
    }
    return (OS_TIME_ZERO_DLY);
}
#endif
```

OSTimeDlyResume()

- Description:

  This function is used resume a task that has been delayed through a call to either OSTimeDly() or OSTimeDlyHMSM().

OSTimeGet()

Description:

This function is used by your application to obtain the current value of the 32-bit counter which keeps track of the number of clock ticks.

OSTimeSet()

Description:

This function sets the 32-bit counter which keeps track of

the number of clock ticks.

OSTimeGet()

```
#if OS_TIME_GET_SET_EN > 0
INT32U  OSTimeGet (void)
{
#if OS_CRITICAL_METHOD == 3
     OS_CPU_SR  cpu_sr;
#endif
    INT32U     ticks;


    OS_ENTER_CRITICAL();
    ticks = OSTime;
    OS_EXIT_CRITICAL();
    return (ticks);
}
#endif
```

OSTimeSet()

```
#if OS_TIME_GET_SET_EN > 0
void  OSTimeSet (INT32U ticks)
{
#if OS_CRITICAL_METHOD == 3 OS_CPU_SR  cpu_sr;
#endif
    OS_ENTER_CRITICAL();
    OSTime = ticks;
    OS_EXIT_CRITICAL();
}
#endif
```