EE461 Microprocessor-based Digital Design

Synchronous Serial I/O

Solutions

1. There are two principal differences between asynchronous and synchronous serial I/O operations with the PIC16F874:

   - Asynchronous serial I/O does not require the use of a clock line. With two dedicated lines available it is possible to transmit data simultaneously in both directions. This is known as full-duplex operation. With synchronous serial I/O this is not possible: data can only be transmitted in one direction at a time. This is known as half-duplex operation (the line can be turned around for transmission in the opposite direction when desired.)

   - The asynchronous serial I/O protocol requires a start bit and a stop bit in addition to the data bits. This overhead is not necessary in the case of serial synchronous I/O. For an 8-bit data word, these two extra bits in every 10 constitute a 20% overhead not needed for serial synchronous I/O. (If a parity bit is in use, then that is an additional bit requirement, but this overhead applies whether we use asynchronous or synchronous communications.)

   In view of the difference in overhead, the synchronous serial I/O scheme is inherently more efficient. Alternatively, though, we can view it as requiring a lower BAUD rate.

   To achieve a data throughput of 1.5 kB/s in a system driven by a crystal oscillator with frequency f_{OSC} = 12 MHz, find choices for BRGH and SPBRG (in the asynchronous case without parity) and SPBRG alone (in the synchronous case without parity) that will achieve this throughput as accurately as possible.' Assume that data are transmitted in 8-bit words. Calculate the actual data throughput rate achieved and the relative error in the achieved throughput rate in each case.

   SOLUTION

   Because the synchronous case does not require any overhead bits, we will consider it first. Table 10-1 in the data sheet shows that for synchronous

'1 B = 1 byte = 8 bits. Data throughput is the amount of data (excluding overhead bits) transmitted through a channel in one unit of time.

serial transmission and reception, the BAUD rate is

$$f_{BAUD} = \frac{FOSC}{4(x + 1)}$$

where

$$x = \text{SPBRG} \in [0,255].$$

Solving for x we find

$$x = \frac{FOSC}{4F_{baud}} - 1$$

$$= \frac{12MHz}{4(1.5kB/s)\left(\frac{8\ bits}{B}\right)} - 1$$

$$= 249.$$

Since we did not need to round x off, the solution is exact. The achieved BAUD rate and the achieved data throughput rate both are 1.5 kB/s because every transmitted bit is a useful data bit. The relative error in the throughput rate is $p = 0\%$.

Now consider the asynchronous case. It requires two extra bits for each eight bits of data, for a total of 10 bits per byte. We therefore require a higher BAUD rate to achieve the same throughput rate. The required BAUD rate is

Fbaud = fthroughput X 10 transm bits / 8 data bits

= 1.5 kB/s X 8 bits/byte X 10 transmitted bits/ 8 data bits

= 15 kb/s

An expression for baud in this case is

Fbaud = fosc/ 16(4-3m)(x+1)

where

$$x = \text{SPBRG} \in [0,255] \text{ and}$$
$$m = \text{BRGH} \in \{0, 1\}.$$

Solving for $(4 - 3m)(x + 1)$ yields

Fosc/16 X fbaud = 12Mhz/ 16 X 15 kB/s

= 50

There are two solutions to this:

$m = 0$ $x = 11.5 = 11$ (rounded down to meet the required throughput.)

and

$m = 1$ $x = 4\ 9$

The first of these is inexact because rounding is required. The second is exact and achieves a BAUD rate of

$$f_{BAUD} = 15 \text{ kb/s}$$

and it corresponds to a throughput of

$$\begin{aligned} f_{throughput} &= f_{BAUD} \times 8 \text{ data bits}/10 \text{ transmitted bits} \\ &= 12\text{kb/s} \\ &= 1.5\text{kB/s} \end{aligned}$$

with a relative error of 0.

The two ideal solutions, then, are SPBRG = 249 for the synchronous case and SPBRG = 49 and BRGH = 1 for the asynchronous case. Both solutions have zero relative error and an achieved throughput rate of 1.5 kB/s.