Serial Baud Rates, Bit Timing and Error Tolerance

Introduction

Asynchronous serial transmission is a mechanism to pass data from one device to another. It is termed asynchronous because the transmission timing conforms to a predefined timing specification as opposed to a synchronous mechanism where an additional clocking signal will indicate when a new data bit is being transmitted.

Byte data is transmitted as a series of eight bits with a preceding start bit to indicate when transmission is beginning and with a stop bit which indicates when all bits have been sent and to allow the next start bit to be detected; there needs to be a transition in the signal line to detect the start bit and the stop bit guarantees this. A minimum of 10 bits will therefore be transmitted to send an 8-bit data value.

Asynchronous serial is transmitted at a baud rate and, for a digital signal, this equates to the maximum number of bits that can be sent per second. The time each bit is present for (the bit time) is the reciprocal of the baud rate -

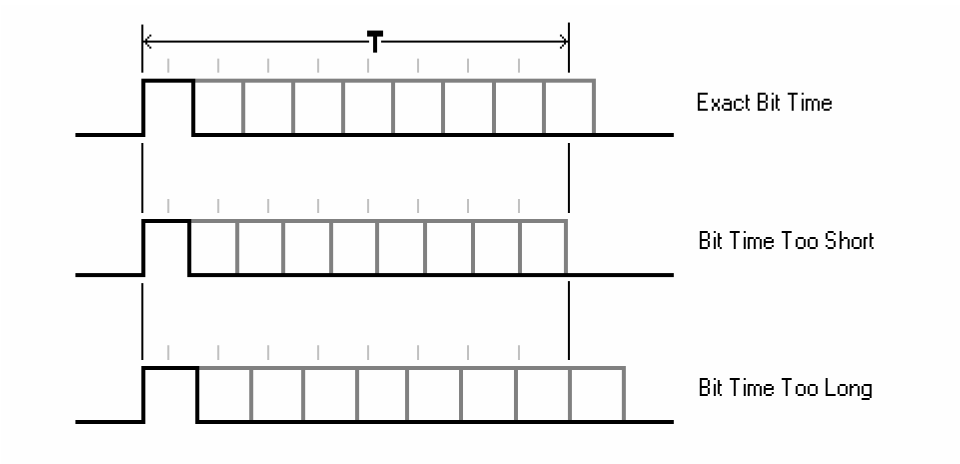
baud rate = 1 / bit time

bit time = 1 / baud rate

Asynchronous Serial Timing

A transmitting device should send its data at a specific baud rate with the correct bit time but that the bit timing actually used may sometimes be too short or too long.

The receiving device will expect the bit timing to be correct for the baud rate specified and will use that bit timing to determine the data received. What data is received will depend on the bit timing actually used by the transmitting device.



The top waveform shows 8-bit serial being received which has the correct bit time for a specific baud rate. The 8-bit data is preceded by a start bit which has the same bit time as each subsequent data bit.

To determine the 8-bit value sent the data stream is sampled in the middle of each data bit. The levels at those points will determine the data value. Note that the 8-bit data value is sent lsb first and msb last.

The internal bit timing synchronises to the leading edge of the start bit then one and a half bit times later a sample is taken in the middle of the first data bit. After a further bit time delay a sample is taken in the middle of the second data bit and so on until a sample has been taken in the middle of the eighth data bit.

The time taken from synchronising to the leading edge of the start bit to sampling in the middle of the eighth data bit (T) is equal to 8.5 times the bit time ($Tbit_{exact}$) -

$$T = 8.5 \times Tbit_{exact}$$

The middle waveform shows a transmission when the bit time is too short ($Tbit_{short}$).

When it comes to sampling the middle of the eighth data bit that bit has just passed; the sampling renders an inaccurate sample, a corrupt data byte.

Sampling fails when -

$$9 \times Tbit_{short} < T$$

The bottom waveform shows a transmission when the bit time is too long ($Tbit_{long}$).

When it comes to sampling the middle of the eighth data bit that bit has not yet started; the sampling renders an inaccurate sample, a corrupt data byte.

Sampling fails when -

$$8 \times Tbit_{long} > T$$

When the bit time is too short

Sampling fails when -

$$9 \times Tbit_{short} < T$$

$$9 \times Tbit_{short} < 8.5 \times Tbit_{exact}$$

$$Tbit_{short} < 8.5/9 \times Tbit_{exact}$$

Correspondingly, sampling succeeds when -

$$Tbit_{short} >= 8.5/9 \times Tbit_{exact}$$

When the bit time is too long

Sampling fails when -

$8 \text{ x } Tbit_{long} > T$

$8 \text{ x } Tbit_{long} > 8.5 \text{ x } Tbit_{exact}$

$Tbit_{long} > 8.5/8 \text{ x } Tbit_{exact}$

Correspondingly, sampling succeeds when -

$Tbit_{long} <= 8.5/8 \text{ x } Tbit_{exact}$

Putting it all together

We have seen that sampling succeeds when -

$Tbit_{short} >= 8.5/9 \text{ x } Tbit_{exact}$

and

$Tbitlong <= 8.5/8 \text{ x } Tbit_{exact}$

We can therefore say a valid bit time (Tbit) can range from $Tbit_{short}$ to $Tbit_{long}$ and when sampled using a $Tbit_{exact}$ timing the data will be sampled correctly and return the correct data value result -

$Tbit = Tbit_{short}$ to $Tbit_{long}$

$Tbit = (8.5/9 \text{ x } Tbit_{exact})$ to $(8.5/8 \text{ x } Tbit_{exact})$

Expressed in terms of percentage -

$Tbit = (94.44\%$ of $Tbit_{exact})$ to $(106.25\%$ of $Tbit_{exact})$

$Tbit = Tbit_{exact} -5.56\% / +6.25\%$

When we apply this to some common baud rates we can see the valid range of bit timings (in approximate microseconds) allowed for that baud rate -

Baud Rate	$Tbit_{exact}$	$Tbit_{short}$	$Tbit_{long}$
600	1667	1574	1771
1200	833	787	885
2400	417	394	443
4800	208	196	221

9600	104	98	110

Baud rate tolerance

When shortest and longest allowed bit times are converted to baud rates we can see the range of valid baud rates which can be sampled correctly using the nominal baud rate sampling time -

Baud Rate	Minimum	Maximum
600	565	635
1200	1130	1271
2400	2257	2538
4800	4525	5102
9600	9091	10204

This equates to a tolerance in baud rate errors of approximately +/- 6%

Note, that because baud rate and bit times are reciprocals of each other, the acceptable error percentages in bit time are not the same as the acceptable error percentages for baud rate.