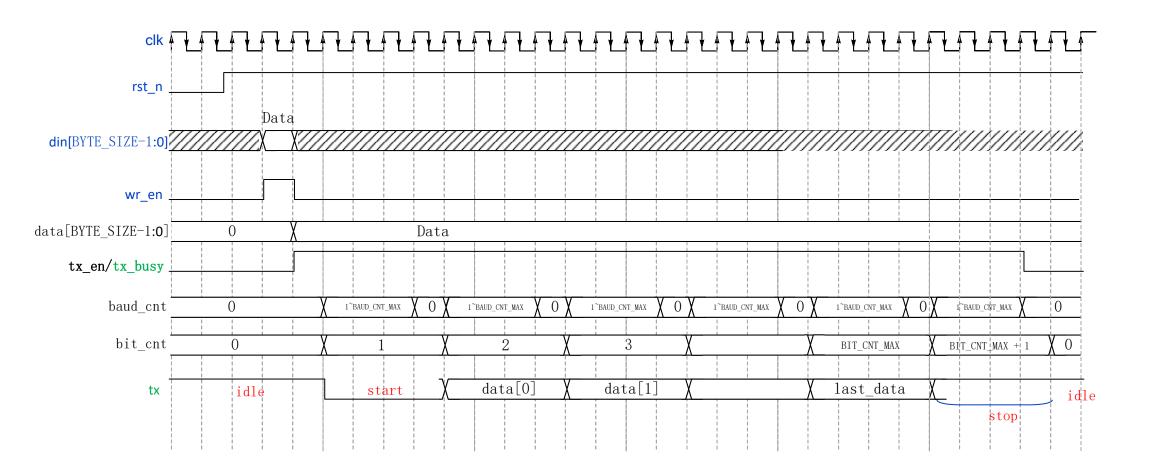


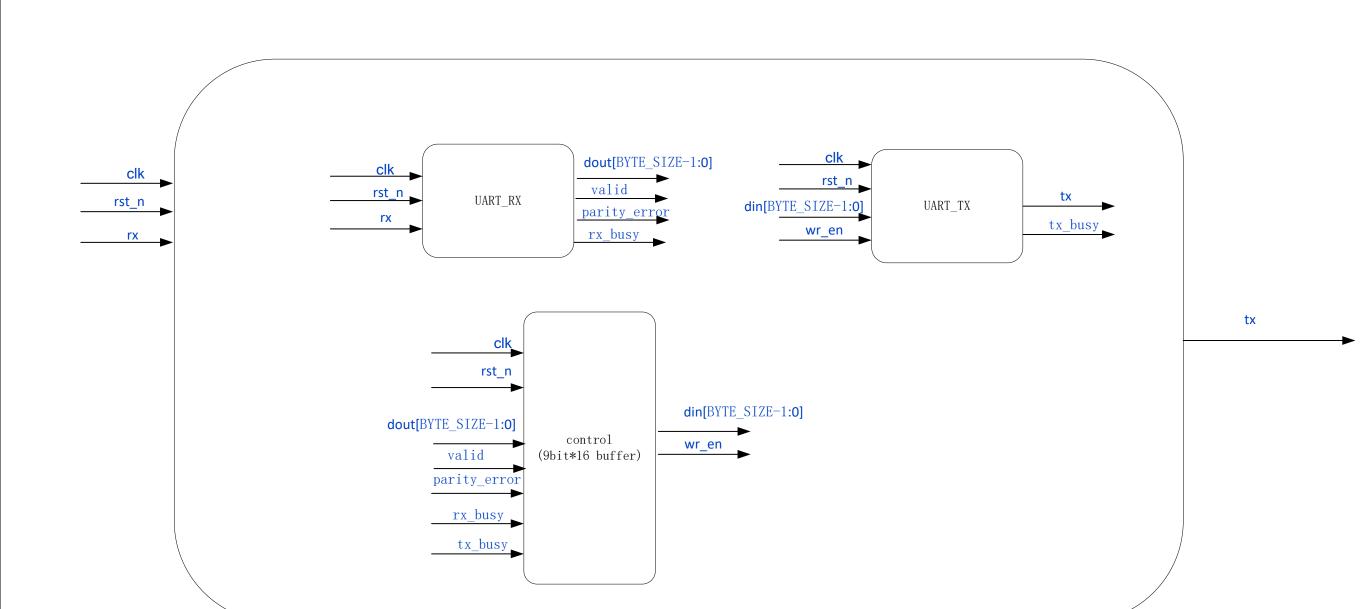
2

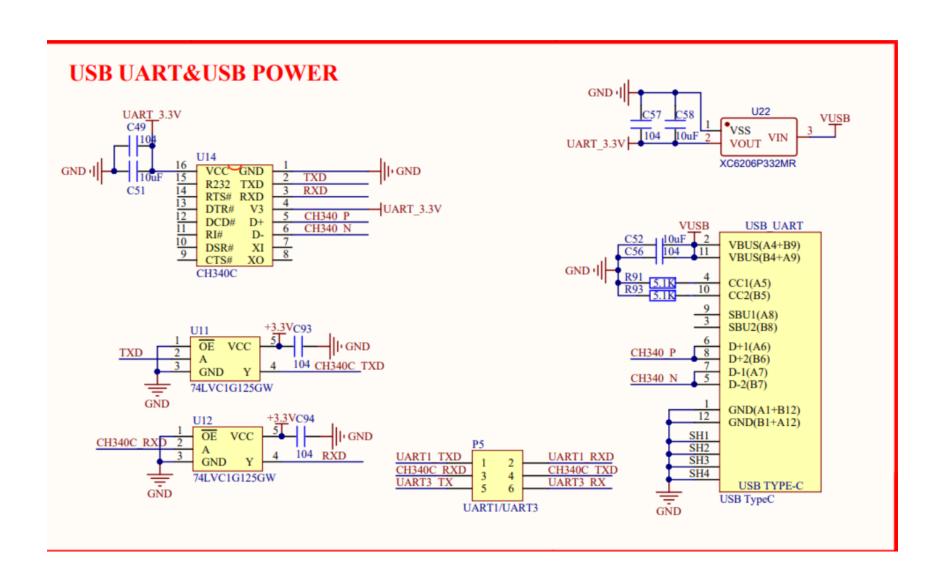
Global Parameters:
1. BAUD_RATE_BPS = 9600
2. CLK_MHZ = 50
3. PARITY = 0
4. STOP_SIZE = 1
5. BYTESIZE = 8

local parameters:
6. BAUD_CNT_MAX = CLK_MHZ * 10^6 / BAUD_RATE_BPS - 1
7. BIT_CNT_MAX = BYTESIZE + PARITY + STOP_SIZE

8. RX_BIT_CNT_MAX = BYTESIZE + PARITY

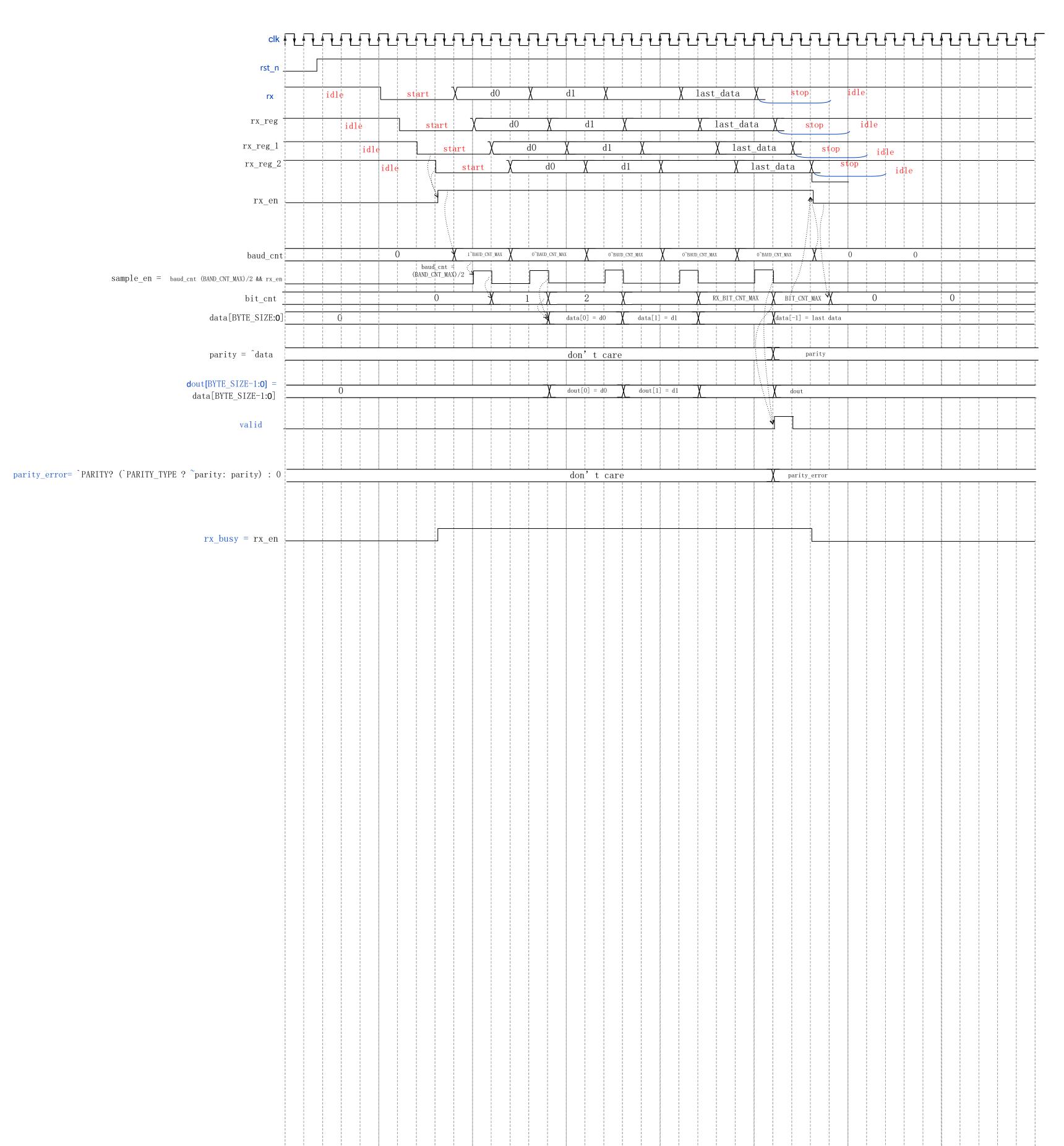


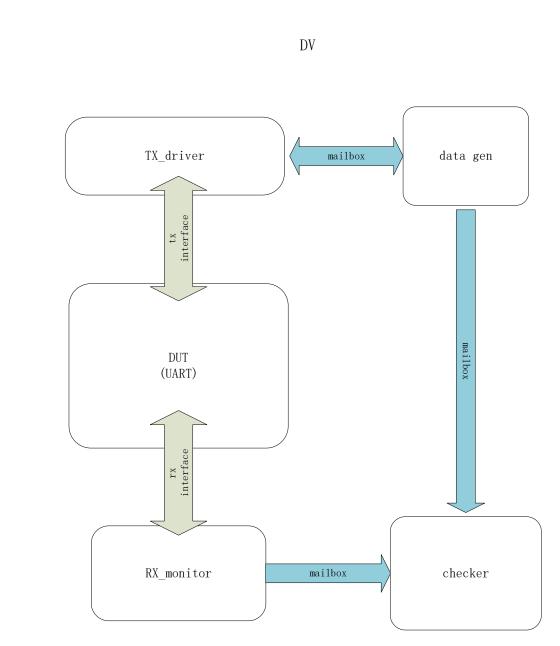




ATK MODULE	=		
uart_rx	input	T19	RXD端口 UART3_RX
uart_tx	output	J15	TXD端口







issue: 当内置FIFO的深度是16的时候,FPGA读写到大约4500个package的时候,会丢失16个package,也就是说在连续读了大约4500个package后,读的数据量超过了写的数据量16个package,FIFO会溢出,所以推断读的速度超过写的速度

通过simulation的结果和FPAG内置的逻辑分析仪,有两个方面的因素导致了读的速度超过写的速度,其中第一个因素占主导地位

```
[CMP], [4743], CMP=0, write=31, read=149
[CMP], [4744], CMP=0, write=195, read=74
 [CMP], [4745], CMP=0, write=7, read=218
[CMP], [4746], CMP=0, write=196, read=94
[CMP], [4747], CMP=0, write=\frac{102}{102}, read=\frac{243}{102}
[CMP], [4748], CMP=0, write=45, read=140
[CMP], [4749], CMP=0, write=64, read=184
[CMP], [4750], CMP=0, write=165, read=26
[CMP], [4751], CMP=0, write=187, read=92
[CMP], [4752], CMP=0, write=173, read=195
[CMP], [4753], CMP=0, write=65, read=13
[CMP], [4754], CMP=0, write=36, read=106
[CMP], [4755], CMP=0, write=105, read=223
[CMP], [4757], CMP=0, write=254, read=240
[CMP], [4758], CMP=0, write=152, read=144
[CMP], [4759], CMP=0, write=149, read=69
[CMP], [4760], CMP=0, write=74, read=245
[CMP], [4761], CMP=0, write=218, read=198
[CMP], [4762], CMP=0, write=94, read=56
[CMP], [4763], CMP=0, write=243, read=156
```

1. 因素一: 描述: 一个byte的读的绝对时间长度和写绝对时间长度的不一致。

原因:可能的原因是读的的波特率和写的波特率之间存在一定的误差,似乎是异步IO不可避免的。

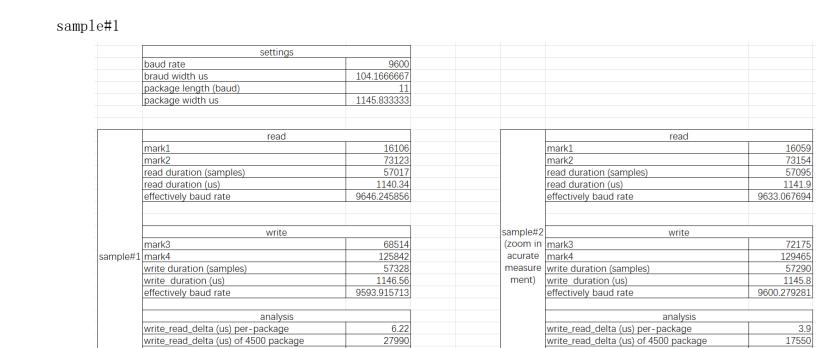
对于单向 1Mbps 及以上、或双向 500Kbps 及以上的应用,建议改用 CH343 启用硬件自动流控。

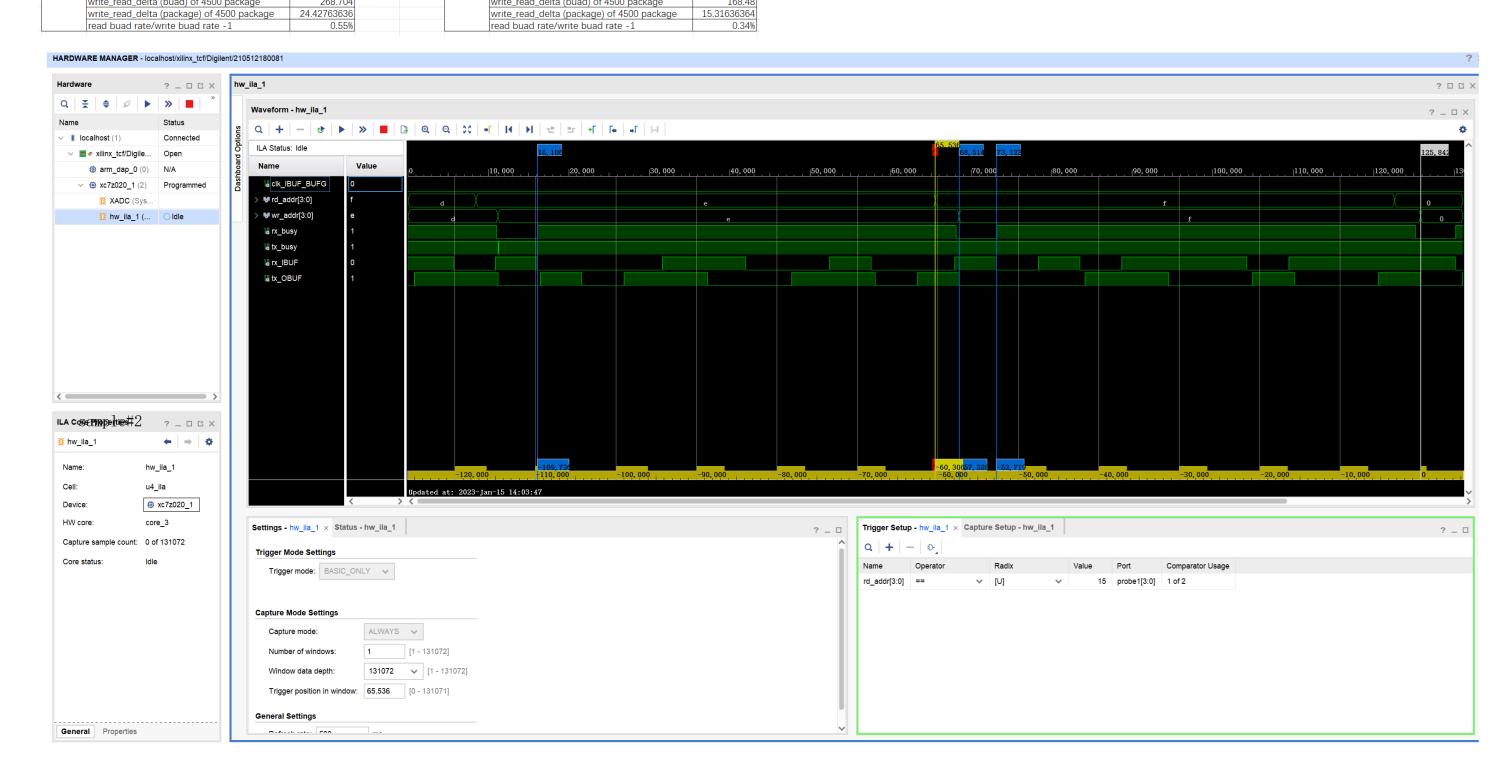
CH340 串口接收信号的允许波特率误差约 2%,CH340G/CH340T/CH340R 串口发送信号的波特率误差小于 0. 3%,CH340C/340N/340K/340E/340X/340B 串口发送信号的波特率误差小于 1. 2%。

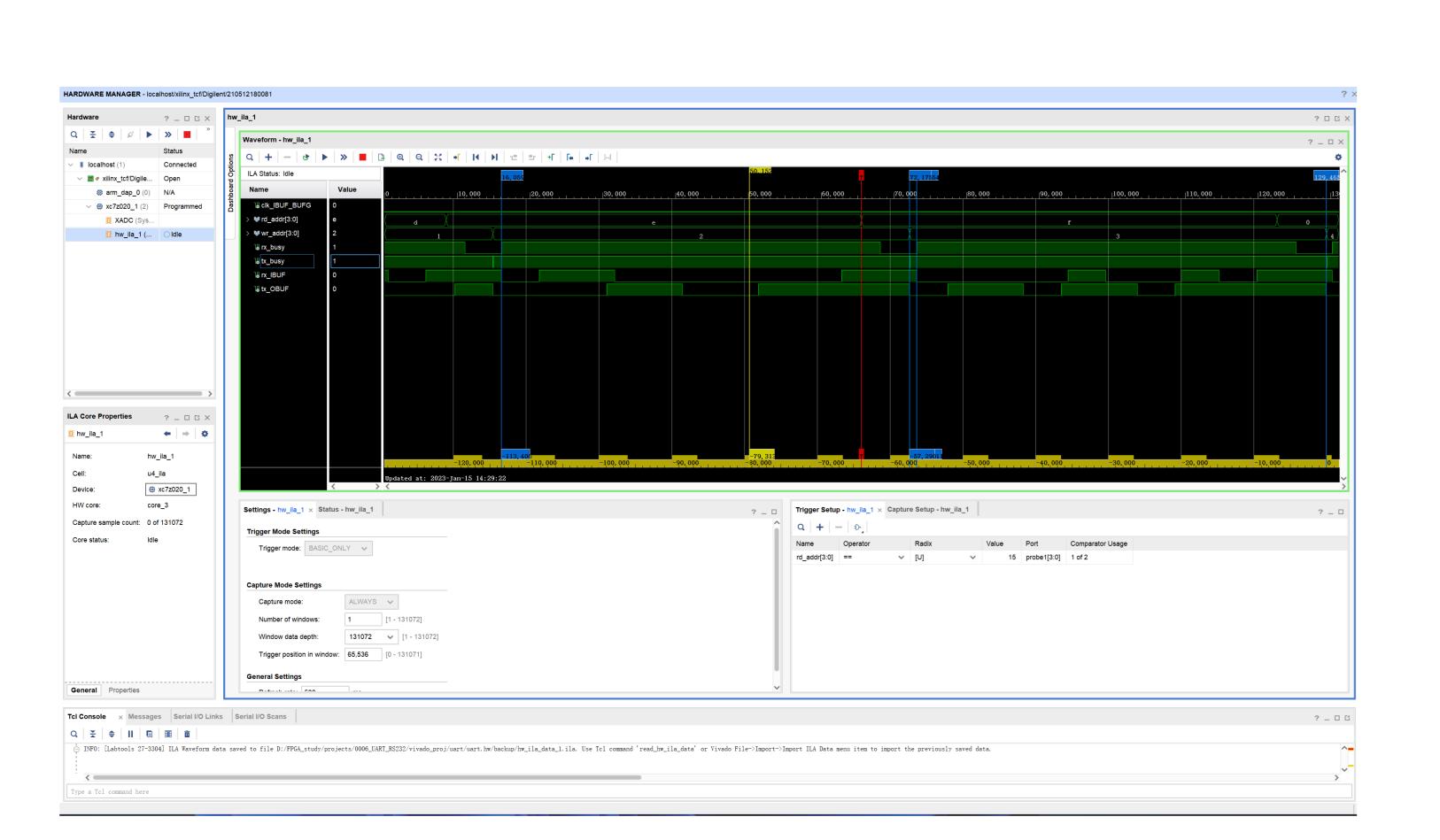
在计算机端的 Windows 操作系统下,CH340 的驱动程序能够仿真标准串口,所以绝大部分原串口

应用程序完全兼容,通常不需要作任何修改。 具体分析:从ila capture可以看到,读写之间有0.34%的波特率误差,而这仅仅0.34%的波特率误差累计 4500 cycle之后会产生15个package的delay,而fi'f 深度为16,所以和实际的上板测试结果很接近。 而且CH340C 芯片手册描述波特率的误差小于1.2%

为10,为646人别们工业队的超过水限及处。则且GIO100 亿分门,加强定伙们中间从土

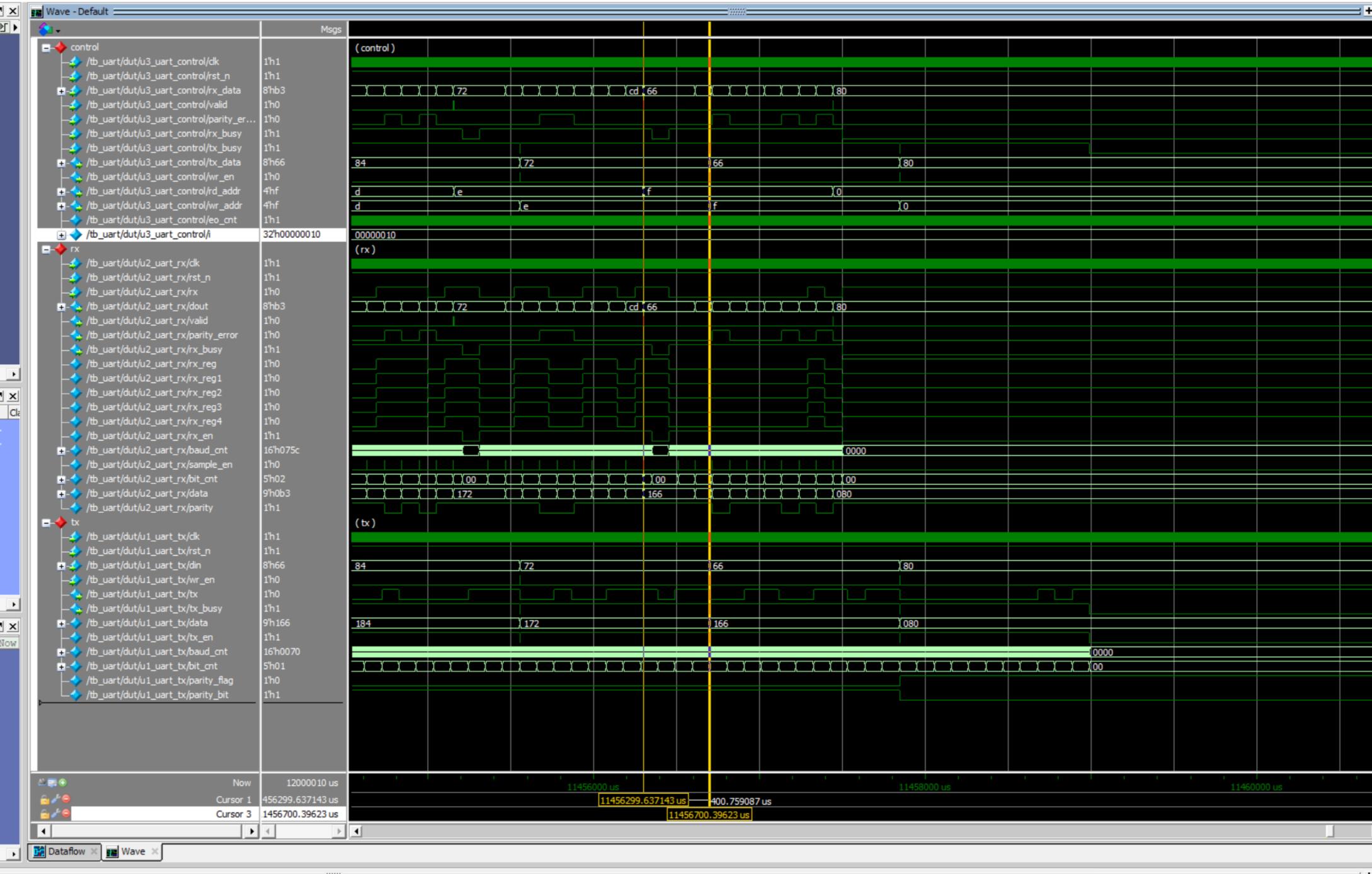






描述:从simulation 结果可以看到,写因为需要先判定 tx_busy 所以每个package就会比读增加2个clock的开销,但是2个clock的开销与9600的波特率比起来很小,不足以会在~4500package左右就会使得读写之间有16个 package的delay

具体分析:从simulation上可以看到,即使到了10000个package,读写之间由于整个timing造成的delay也不过400us(1个package长度是1145us),和理论计算结果相符;而且simuation在10000packa



```
9974, wr data = b8, wr parity = 1,rd_pkg =
@11430344730000 [WARNING] CMP : compare=1,wr pkg =
                                                                                                        9974, rd_data = b8, rd_parity = 1
@11431490530000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9975, wr_data = c4, wr_parity = 0,rd_pkg =
                                                                                                        9975, rd_data = c4, rd_parity = 0
 @11432636330000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9976, wr data = a5, wr parity = 1,rd_pkg =
                                                                                                        9976, rd_data = a5, rd_parity = 1
 @11433782130000 [WARNING] CMP : compare=1,wr pkg =
                                                            9977, wr data = 99, wr parity = 1,rd pkg =
                                                                                                        9977, rd data = 99, rd parity = '
 @11434927930000 [WARNING] CMP : compare=1,wr pkg =
                                                            9978, wr data = c0, wr parity = 1,rd pkg =
                                                                                                        9978, rd data = c0, rd parity = 1
 @11436073730000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9979, wr_data = ca, wr_parity = 1,rd_pkg =
                                                                                                        9979, rd data = ca, rd parity = 1
@11437219530000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9980, wr_data = 91, wr_parity = 0,rd_pkg =
                                                                                                        9980, rd_data = 91, rd_parity = 0
@11438365330000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9981, wr_data = d8, wr_parity = 1,rd_pkg =
                                                                                                         9981, rd_data = d8, rd_parity = 1
 @11439511130000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9982, wr data = 8c, wr parity = 0,rd pkg =
                                                                                                        9982, rd data = 8c, rd parity = 0
                                                                                                        9983, rd_data = b3, rd_parity = 0
 @11440656930000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9983, wr_data = b3, wr_parity = 0,rd_pkg =
 @11441802730000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9984, wr data = 63, wr parity = 1,rd pkg =
                                                                                                        9984, rd data = 63, rd parity = 1
@11442948530000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9985, wr data = 70, wr parity = 0,rd pkg =
                                                                                                        9985, rd data = 70, rd parity = 0
                                                           9986, wr data = a8, wr parity = 0,rd_pkg =
 @11444094330000 [WARNING] CMP : compare=1,wr_pkg =
                                                                                                        9986, rd data = a8, rd parity = 0
@11445240130000 [WARNING] CMP : compare=1,wr pkg =
                                                           9987, wr data = 1a, wr_parity = 0,rd_pkg =
                                                                                                        9987, rd data = 1a, rd parity = 0
@11446385930000 [WARNING] CMP : compare=1,wr pkg =
                                                            9988, wr data = 02, wr parity = 0,rd pkg =
                                                                                                        9988, rd_data = 02, rd_parity = 0
 @11447531730000 [WARNING] CMP : compare=1,wr_pkg =
                                                                                                        9989, rd_data = 86, rd_parity = 0
                                                            9989, wr data = 86, wr parity = 0,rd pkg =
@11448677530000 [WARNING] CMP : compare=1,wr_pkg =
                                                           9990, wr_data = cf, wr_parity = 1,rd_pkg =
                                                                                                        9990, rd_data = cf, rd_parity = 1
@11449823330000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9991, wr data = 91, wr parity = 0,rd pkg =
                                                                                                        9991, rd_data = 91, rd_parity = 0
 @11450969130000 [WARNING] CMP : compare=1,wr pkg =
                                                                                                        9992, rd data = e9, rd_parity = 0
                                                           9992, wr data = e9, wr parity = 0,rd pkg =
@11452114930000 [WARNING] CMP : compare=1,wr pkg =
                                                            9993, wr data = b0, wr parity = 0,rd pkg =
                                                                                                        9993, rd data = b0, rd parity = 0
@11453260730000 [WARNING] CMP : compare=1,wr pkg =
                                                           9994, wr data = d3, wr parity = 0,rd pkg =
                                                                                                        9994, rd_data = d3, rd_parity = 0
@11454406530000 [WARNING] CMP : compare=1,wr pkg =
                                                            9995, wr data = 2d, wr parity = 1,rd pkg =
                                                                                                        9995, rd_data = 2d, rd_parity = 1
@11455552330000 [WARNING] CMP : compare=1,wr_pkg =
                                                            9996, wr data = 84, wr parity = 1,rd pkg =
                                                                                                        9996, rd_data = 84, rd_parity =
                                                           9997, wr_data = 72, wr_parity = 1,rd_pkg =
@11456698130000 [WARNING] CMP : compare=1,wr_pkg =
                                                                                                        9997, rd data = 72, rd parity = ^{\circ}
@11457843930000 [WARNING] CMP : compare=1,wr pkg =
                                                           9998, wr data = 66, wr_parity = 1,rd_pkg =
                                                                                                        9998, rd data = 66, rd parity = '
@11458989730000 [WARNING] CMP : compare=1,wr pkg =
                                                           9999, wr data = 80, wr parity = 0,rd pkg =
                                                                                                        9999, rd data = 80, rd parity = 0
```

解决办法: 1. 软件端使用分段发送,避免超过1000个package的连续发送 2. 增加FIFO 深度(只能缓解)

it passed 800,000 package with insert 0.1s delay every 1000 package

```
f write_seq(self,size)
write_list = []
#write_list = list(range(size))
for i in range(size):
   write_list.append(random.randint(0,255))
write_data = bytes(write_list)
print(write_data)
 c_time = self.time_module.get_time_stamp()
self.write_info("[%.3f] %d bytes was writing\n" %(c_time, size))
 if size > 1000: #split each 1000 package
    for i in range(int((size-1)/1000 + 1)):
        print("sending sub_packag %d" %i)
        if (i*1000 + 1000) < size:
            write_data_sub = write_data[i*1000: i*1000 + 1000]
           write_data_sub = write_data[i*1000: size]
        written_bytes_nu = self.shandle.write(write_data_sub)
        c_time = self.time_module.get_time_stamp()
        self.write_info("[%.3f] %d bytes have been written\n" %(c_time,written_bytes_nu))
        time.sleep(0.1)
```

```
[CIVIT ],[13331 T],CIVIT - 1,VVITC-10,ICAG-10
[CMP],[799975],CMP=1,write=146,read=146
 [CMP],[799976],CMP=1,write=110,read=110
[CMP],[799977],CMP=1,write=212,read=212
[CMP],[799978],CMP=1,write=101,read=101
[CMP],[799979],CMP=1,write=24,read=24
[CMP],[799980],CMP=1,write=109,read=109
[CMP],[799981],CMP=1,write=153,read=153
[CMP],[799982],CMP=1,write=28,read=28
[CMP],[799983],CMP=1,write=138,read=138
[CMP],[799984],CMP=1,write=240,read=240
[CMP],[799985],CMP=1,write=133,read=133
[CMP],[799986],CMP=1,write=193,read=193
[CMP],[799987],CMP=1,write=58,read=58
[CMP],[799988],CMP=1,write=214,read=214
[CMP],[799989],CMP=1,write=230,read=230
[CMP],[799990],CMP=1,write=206,read=206
[CMP],[799991],CMP=1,write=104,read=104
[CMP],[799992],CMP=1,write=95,read=95
[CMP],[799993],CMP=1,write=134,read=134
[CMP],[799994],CMP=1,write=97,read=97
[CMP],[799995],CMP=1,write=147,read=147
[CMP],[799996],CMP=1,write=12,read=12
[CMP],[799997],CMP=1,write=62,read=62
[CMP],[799998],CMP=1,write=203,read=203
[CMP],[799999],CMP=1,write=124,read=124
[summary] pass_counts=800000, error_counts=0[buff_check],out_buffer=0, in_buff=0
```