Cairo University

Faculty of Engineering year

Dept. of Electronics and Communications

ELC 3030

EECE – 1nd

Advanced Microprocessor Architecture

UART & USB Emulator project

Team: 36

Name	Section	B N
Ali Badry	3	4
Hassan sayed hassan	2	5
Doaa Mohamed shafiy	2	13

UART PART

data Bits	stop bits=1			stop bits=2		
	no Parity	even Parity	odd Parity	no Parity	even Parity	odd Parity
7 bits	22.22%	30%	30%	30%	36.36%	36.36%
8 bits	20%	27.27%	27.27%	27.27%	33.33%	33.33%

Table1.1_UART percentage Overhead results

From the result above , We found that if we increase the number of stop bits to two bits or adding parity bit for error detection or using 7 bits data rather than 8 bits (as many pure data decrease overhead),the Overhead on system increases ,and vise versa

data Bits	stop bits=1			stop bits=2		
	no Parity	even Parity	odd Parity	no Parity	even Parity	odd Parity
7 bits	77.78%	70%	70%	70%	63.64%	63.64%
8 bits	80%	72.73%	72.73%	72.73%	66.67	66.67%

Table1.2_UART Efficiency results

From the result above , we found that if we increase the number of stop bits to two bits or adding parity bit for error detection or using 7 bits data rather than 8 bits (as many pure data increase efficiency),the Efficiency on system decreases ,and vise versa, and also that make sense as Efficiency is opposite to Overhead

data Bits	stop bits=1			stop bits=2		
	no Parity	even Parity	odd Parity	no Parity	even Parity	odd Parity
7 bits	1.152sec	1.28sec	1.28sec	1.28sec	1.408sec	1.408sec
8 bits	1.28sec	1.408sec	1.408sec	1.408sec	1.536sec	1.536sec

Table1.3_UART Time Total results

From the result above ,we found that if we increase any bits over the lowest minimum number of required bits, total time of transmitting increases , regardless the bits type that's from make sense.

First two bytes in UART result , which have 1 start Bit , 8 bit Data , no Parity , 1 stop Bit
The First two character in input file is ***Th*** which has a binary data 0001010101 and 0000101101 Respectively

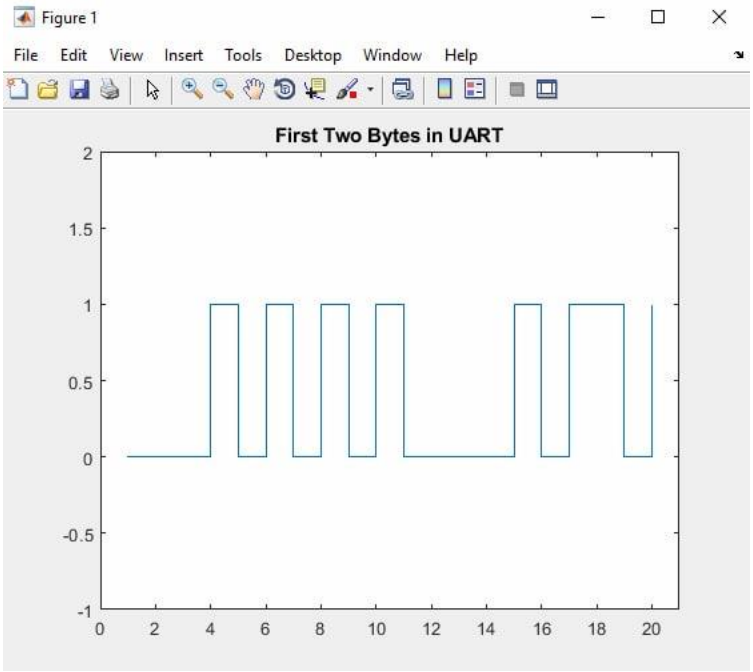

Figure1.1_UART First two bytes

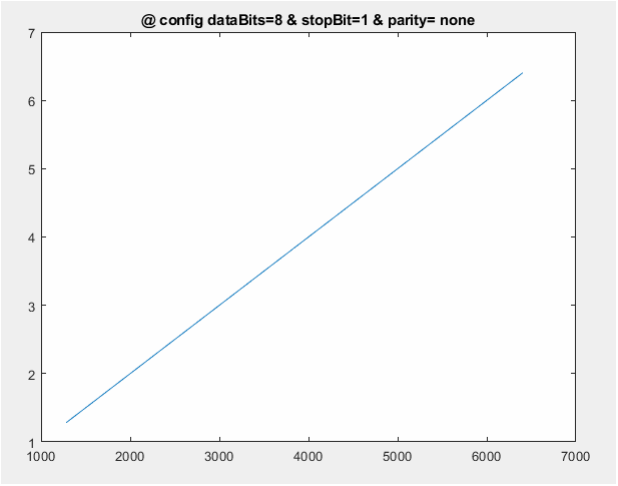

Figure1.2_UART Transmitting Time

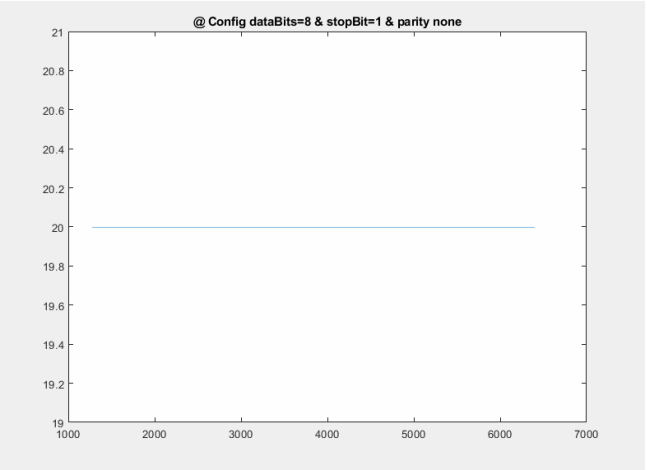

Figure1.3_UART Percentage Overhead

USB PART

Payload	Overhead		
	File size = 1280 bytes	File size = 2560 bytes	File size = 3840 bytes
128 bytes	2.754%	2.754%	2.754%
90 bytes	8.8563%	5.7133%	4.617%
200 bytes	10.1991%	3.291%	5.709%

Table2.1_USB percentage Overhead

When we made our the file size a multiple of the payload, the overhead remained constant as all the packets will be full. Thus, the minimum overload is (without the existence of bit stuffing) is 2.754%. When the file size is not a multiple of the payload, it changes as shown in table 2.1.

Payload	Efficiency		
	File size = 1280 bytes	File size = 2560 bytes	File size = 3840 bytes
128 bytes	97.246%	97.246%	97.246%
90 bytes	91.1437%	94.2867%	95.383%
200 bytes	89.8009%	96.709%	94.291%

Table2.2_USB percentage Efficiency

Payload	Transmission time at (0.0001 sec bit duration)		
	File size = 1280 bytes	File size = 2560 bytes	File size = 3840 bytes
128 bytes	1.053	2.106	3.159
90 bytes	1.1235	2.1721	3.2207
200 bytes	1.1403	2.1177	3.258

Table2.3_USB percentage Transmission time (sec)

The Transmission time increases by increasing the file size despite the payload difference. The transmission time is linear proportional with the bit duration. The relation is also nearly linear with the file size as shown in table 2.3.

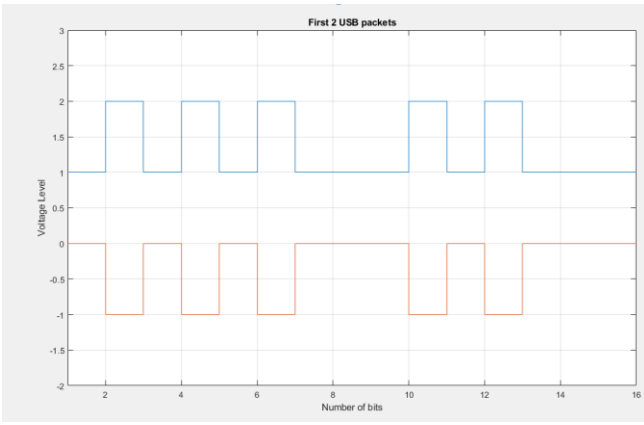

Figure2.1_USB First two bytes (First 15 bits)

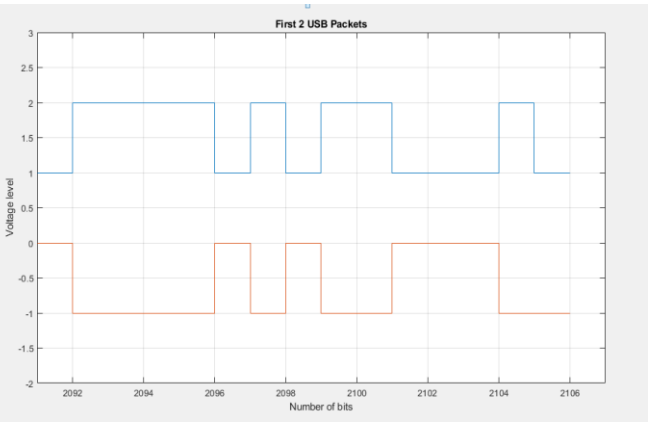

Figure2.2_USB First two bytes (Last 15 bits)

20	21	22	23	24	25	26	27	28	29	30
1	1	1	1	1	1	1	1	0	0	1

Figure2.3_USB Address before bit stuffing

20	21	22	23	24	25	26	27	28	29	30	31
1	1	1	1	1	1	0	1	1	0	0	1

Figure2.4_USB Address after bit stuffing

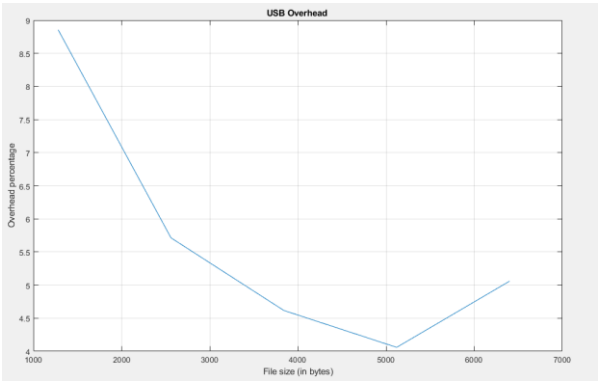

Figure2.5_USB overhead at 90 bytes payload

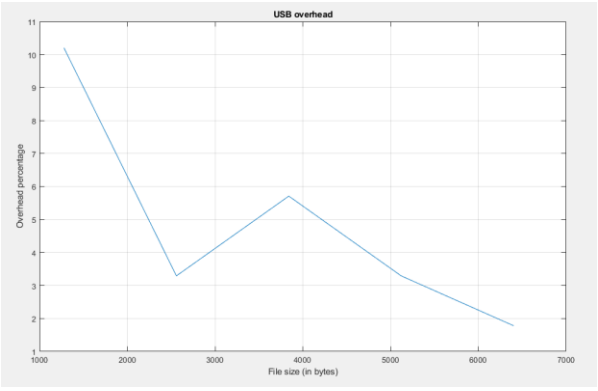

Figure2.6_USB overhead at 200 bytes payload

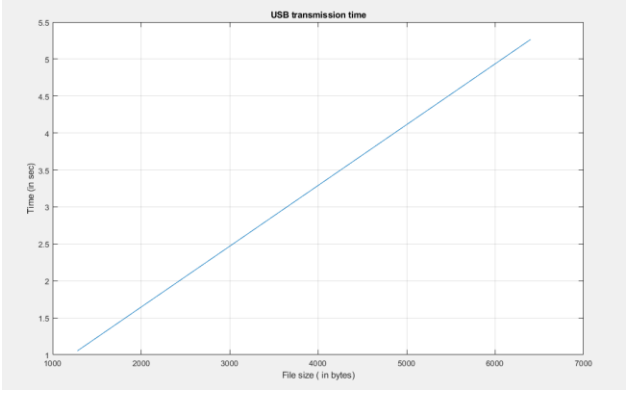

Figure2.7_USB Transmission time at 0.1 msec/ bit

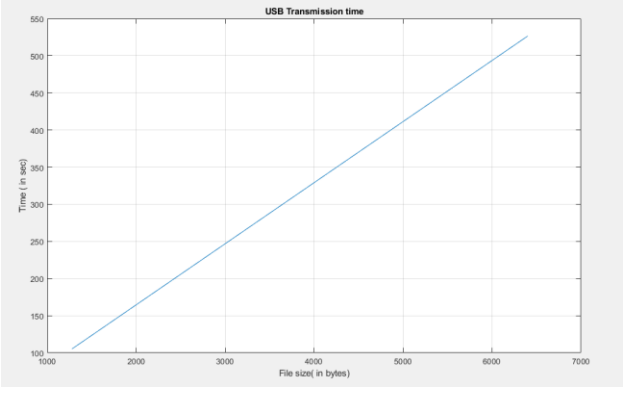

Figure2.8_USB Transmission time at 0.01 sec/ bit

OUTPUT PART


```
[{
   "Protocolname":"UART",
      "outputs":{
   "total_tx_time":1.28,
      "overhead":20,
      "efficiency":80
   }
   },
{
   "Protocolname":"USB",
      "outputs":{
   "total_tx_time":1.0530000000000002,
      "overhead":2.7540360873694207,
      "efficiency":97.245963912630586
   }]
```