SHEET 6

Q1. Assume System clock frequency=16MHz. Find the values for the divisor registers of UARTIBRD and UARTFBRD for the following standard baud rates:

* $BRD = \frac{SysClk}{ClkDiv * Baud\ Rate} = IBRD + FBRD$ (= 123.456) مثلاً

* $UARTIBRD = INT(BRD)$ * $UARTFBRD = INT(FBRD * 64 + 0.5)$

(a) 4800

$\therefore BRD = \frac{16 \times 10^6}{16 * 4800} = 208.3333$ (208 → IBRD, .3333 → FBRD)

$\therefore UARTIBRD = INT(208.3333) = 208$

$\therefore UARTFBRD = INT(0.3333 \times 64 + 0.5) = INT(21.8312) = 21$

(b) 9600

$\therefore BRD = \frac{16 \times 10^6}{16 * 9600} = 104.167$

$\therefore UARTIBRD = INT(104.167) = 104$

$\therefore UARTFBRD = INT(0.167 \times 64 + 0.5) = INT(11.188) = 11$

(c) 57,600

$\therefore BRD = \frac{16 \times 10^6}{16 * 57,600} = 17.361$

$\therefore UARTIBRD = INT(17.361) = 17$

$\therefore UARTFBRD = INT(0.361 \times 64 + 0.5) = INT(23.604) = 23$

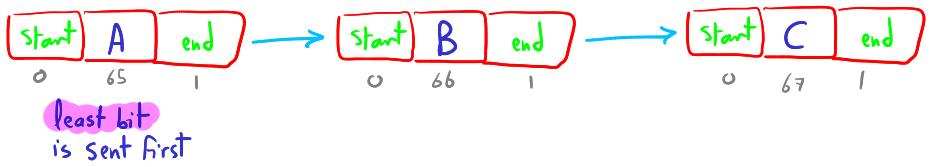
(d) 115,200

$$\therefore BRD = \frac{16 \times 10^6}{16 * 115{,}200} = 8.681$$

$$\therefore UARTIBRD = INT(8.681) = 8$$

$$\therefore UARTFBRD = INT(0.681 \times 64 + 0.5) = INT(44.084) = 44$$

Q2. Assume the baud rate is 9600 bits/sec. Show the serial port output versus time waveform that occurs when the ASCII characters "ABC" are transmitted one right after another. What is the total time to transmit the three characters?



least bit is sent first

i) Waveforms.

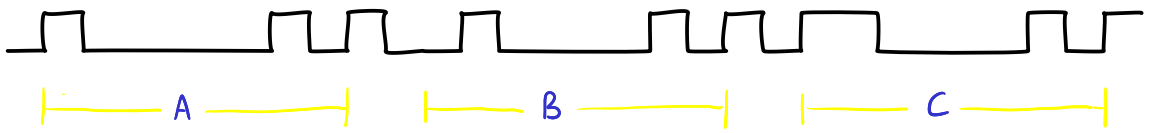
ascii (A) = 65 → 0100 0001

ascii (B) = 66 → 0100 0010

ascii (C) = 67 → 0100 0011

Each char has 8 bit + start bit + end bit → # total no. of bits = 3

by adding start & end bits & 0100000101000010001100001101

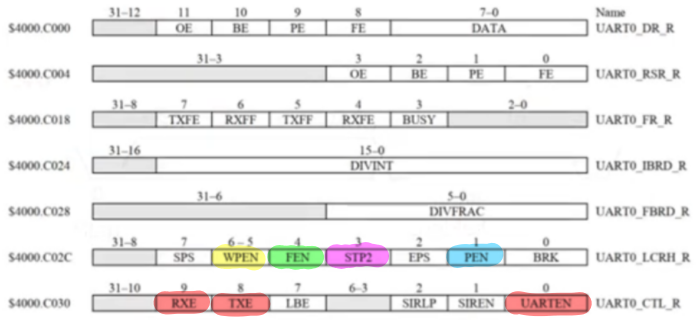


ii) total time:

$$\text{bit time} = \frac{1}{\text{baud rate}} = \frac{1}{9600} \text{ sec/bit}$$

$$\therefore \text{total time} = \text{no. of bits} * \text{bit time} = 30 * \frac{1}{9600} = 3.125 \text{ msec}$$

Q3. Write a C function to initialize UART0 with baud rate 9600 bits/s, 8 bits word length, no parity, one stop bit, and FIFO enabled.



	31-12	11	10	9	8	7-0	Name
$4000.C000		OE	BE	PE	FE	DATA	UART0_DR_R
$4000.C004	31-3				3 OE	2 BE	1 PE	0 FE	UART0_RSR_R
$4000.C018	31-8	7 TXFE	6 RXFF	5 TXFF	4 RXFE	3 BUSY	2-0	UART0_FR_R
$4000.C024	31-16	15-0 DIVINT						UART0_IBRD_R
$4000.C028	31-6	5-0 DIVFRAC						UART0_FBRD_R
$4000.C02C	31-8	7 SPS	6-5 WPEN	4 FEN	3 STP2	2 EPS	1 PEN	0 BRK	UART0_LCRH_R
$4000.C030	31-10	9 RXE	8 TXE	7 LBE	6-3	2 SIRLP	1 SIREN	0 UARTEN	UART0_CTL_R

enable fifo
disable parity
one stop
8 bit
= 0x0070

the tiva runs on 80MHz frequency

$$\rightarrow BRD = \frac{80 \times 16^{6}}{16 * 9600} = 520.833$$

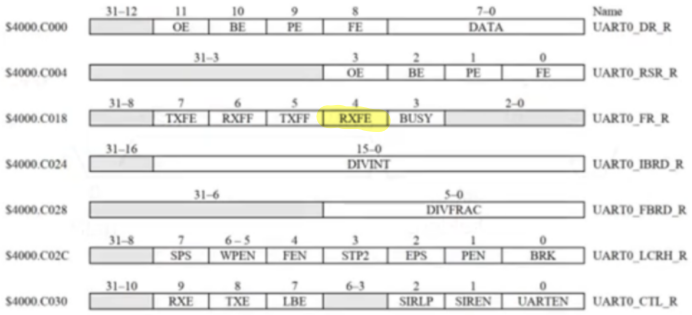
(16 = default value)

$$\therefore UARTIBRD = INT(520.833) = 520$$

$$\therefore UARTFBRD = INT(0.833 \times 64 + 0.5) = INT(53.812) = 53$$

```
void  UART_Init(void)
{

  SYSCTL_RCGCUART_R |= 0x0001;       // activate UART0
  SYSCTL_RCGCGPIO_R |= 0x0001;       // activate port A
  while ( (SYSCTL_PRGPIO_R&0x0001) == 0) {}
  UART0_CTL_R &= ~0x0001;            // disable UART

  UART0_IBRD_R = 520;
  UART0_FBRD_R = 53;

  UART0_LCRH_R = 0x0070;             //8-bit word length, enable FIFO

  UART0_CTL_R = 0x0301;              //enable RXE, TXE & UART

  GPIO_PORTA_AFSEL_R |= 0x03;        //enable alt function PA0(U0Rx) & PA1(U0Tx)

  GPIO_PORTA_PCTL_R = (GPIO_PORTA_PCTL_R&0xFFFFFF00)+0x00000011;      //Using UART Functionality for PA0 & PA1

  GPIO_PORTA_DEN_R |= 0x03;          //enable digital I/O on PA0 & PA1
  GPIO_PORTA_AMSEL_R  &= ~0x03;      //disable analog function on PA0 & PA1

}
```

Q4. Write a C function to check if there is data available to be received by UART0.



	31-12	11	10	9	8	7-0	Name
$4000.C000		OE	BE	PE	FE	DATA	UART0_DR_R

	31-3	3	2	1	0	
$4000.C004		OE	BE	PE	FE	UART0_RSR_R

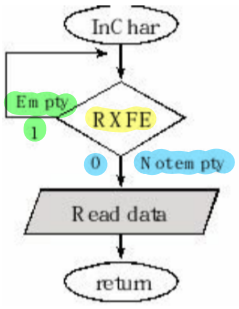
	31-8	7	6	5	4	3	2-0	
$4000.C018		TXFE	RXFF	TXFF	RXFE	BUSY		UART0_FR_R

	31-16	15-0	
$4000.C024		DIVINT	UART0_IBRD_R

	31-6	5-0	
$4000.C028		DIVFRAC	UART0_FBRD_R

	31-8	7	6-5	4	3	2	1	0	
$4000.C02C		SPS	WPEN	FEN	STP2	EPS	PEN	BRK	UART0_LCRH_R

	31-10	9	8	7	6-3	2	1	0	
$4000.C030		RXE	TXE	LBE		SIRLP	SIREN	UARTEN	UART0_CTL_R



→ to check if there is data available to be received or not

so you have to check if fifo buffer is empty or not

⇒ if fifo buffer is not empty, then there is data avaliable to be received

```
#include "tm4c123gh6pm.h"
#include "stdbool.h"

bool UART0_InChar(void)
{

  return ((UART0_FR_R & 0x10) != 0) ? 0:1;

}
```

return 0 if the Condition is true
return 1 if the Condition is false

Condition

if RxFE not equal 0
(equal 1)

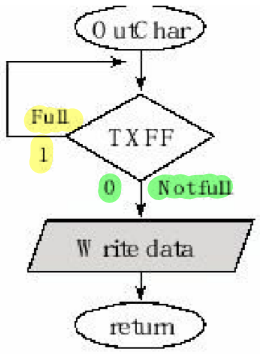
Q5. Write a C function to receive one byte using UART0.

Char

return

```
char UART0_Read(void)
{

  while ( UART0_InChar() == 0 ) {}; //first check whether there data to receive or not
    //another method: while ( (UART0_FR_R & 0x10) != 0 );

  return (char)(UART0_DR_R & 0xFF);      //return the first 8 bits (data)

}
```

Q6. Write a C function to transmit one byte using UART0.




```
bool UART0_OutChar (void)
{

  return ((UART0_FR_R & 0x20) != 0) ? 0:1;

}

void UART0_Write (char data)
{

  while ( UART0_OutChar() == 0 ) {};  //first check whether the buffer if full
    //another method: while ( (UART0_FR_R & 0x20) != 0 );
  UART0_DR_R = data;

}
```

Q7. Write a C program that receives from Device1 a lower-case character and transmits its upper-case to Device2.

// lowercase = uppercase + 32

```
int main()
{

  char letter;
  letter = UART0_Read();          //received char is lower-cased
  letter = letter - 32;           //uppercase = lowercase - 32
  UART0_Write(letter);            //transmitted char is upper-cased

}
```

Juba