# CSE 211: Introduction to Embedded Systems

Section 7

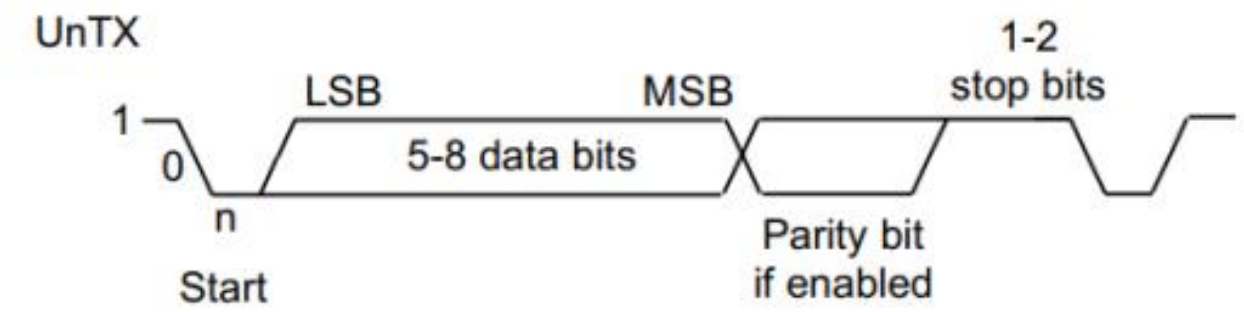
Frame

عبارة عن data و شوية headers قبل و بعد

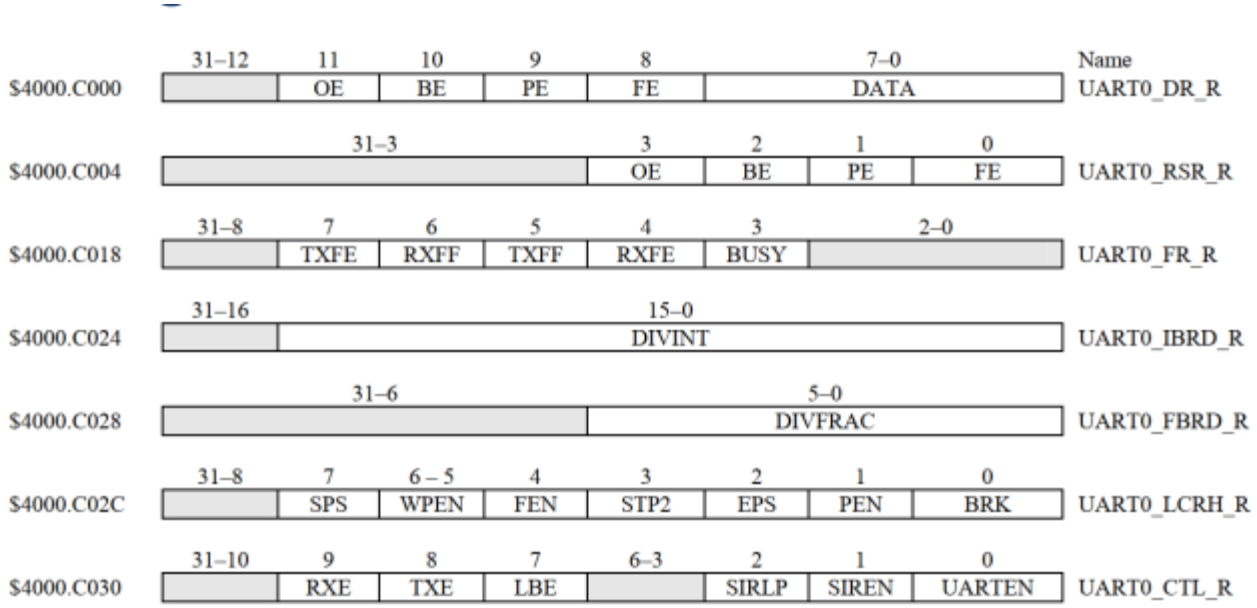
# UART Protocol

بيبعت serial bits على ال bus

ال format بتاعة ال frame



# UART Registers



| | 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 |

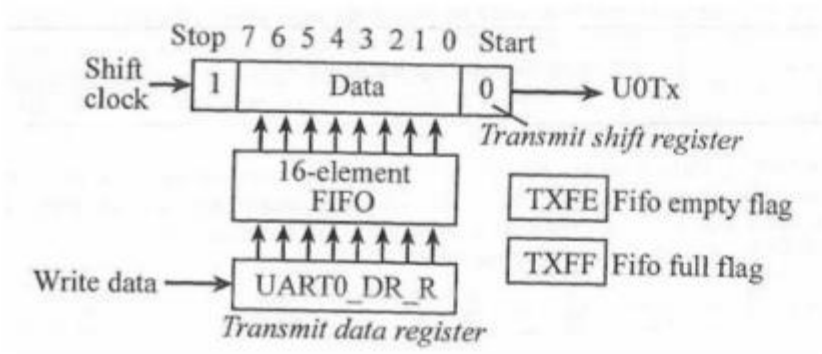
# UART Baud Rate Generation

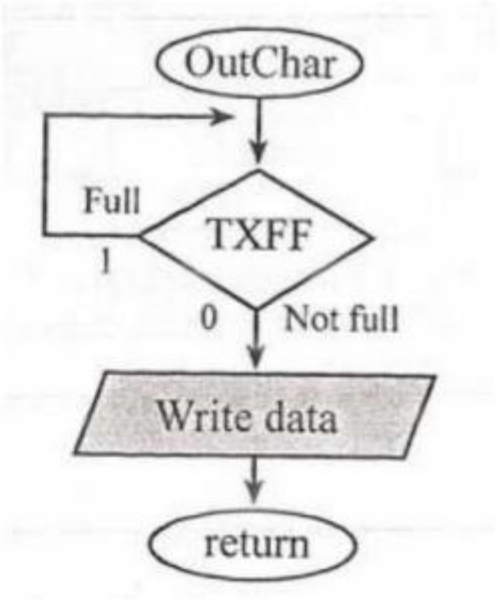
- BRD = IBRD + FBRD
- BRD = UARTSysClk / (ClkDiv * Baud Rate)
- DIVINT = IBRD = INT(BRD)
- DIVFRAC = INT(FBRD * 64 + 0.5)

# UART Setup

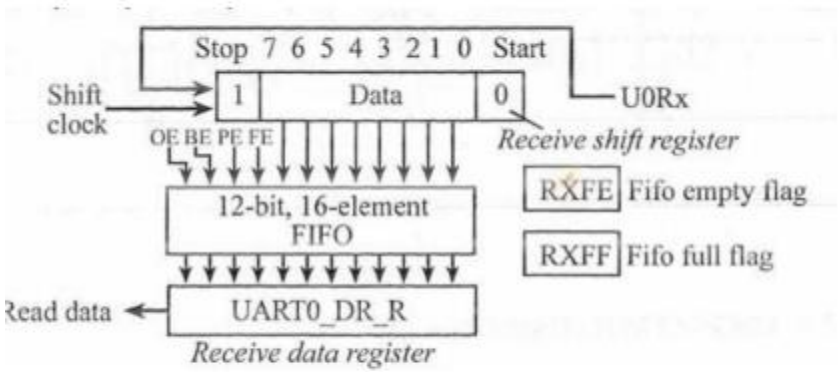
1. Enable clock using RCGCUART
2. Enable GPIO Clock (SYSCTL_RCGCGPIO_R)
3. Disable UART by clearing UARTEN in UARTx_CTL_R
4. Write the values for baud rate in UARTBRD and UARTFBRD
5. Write the desired parameters (word length, FIFO enable, number of stop bits, and parity enable and type) to UARTLCRH
6. Enable UART by setting UARTEN in UARTx_CTL_R
7. Set GPIO AFSEL, PCTL, and DEN

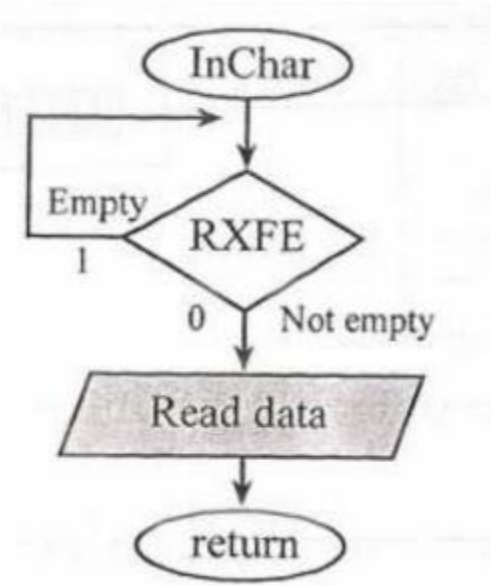
# UART Transmit



# UART Receive



# Sheet 6

- Assume System clock frequency=16MHz. Find the values for the divisor registers of UARTIBRD and UARTFBRD for the following standard baud rates:
  - (a) 4800 (b) 9600 (c) 57,600 (d) 115,200

# Answer

By default, 16 MHz System Clock is divided by 16 before it is fed to the UART.

Therefore, Divisor= 16MHz/(16*BaudRate) = 1MHz/BaudRate.

(a) 1MHz/4800 = 208.3333, UARTIBRD = 208 and UARTFBRD = (0.3333×64) + 0.5 = 21.8312 = 21

(b) 1MHz/9600 = 104.166666, UARTIBRD = 104 and UARTFBRD = (0.16666 × 64) +0.5=11

(c) 1MHz/57600 = 17.361, UARTIBRD = 17 and UARTFBRD = (0.361 × 64) + 0.5 =23

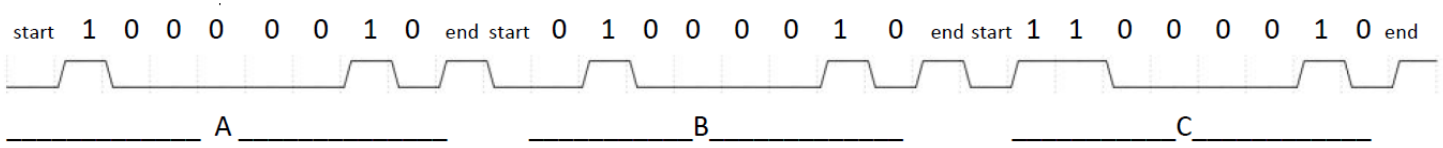
(d) 1MHz/115200 = 8.680, UARTIBRD = 8 and UARTFBRD = (0.680 × 64) +0.5=44

# Sheet 6

- 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?

# Answer

- A 65 -> 0100 0001
- B 66 -> 0100 0010
- C 67 -> 0100 0011




- Each char has 8 bits+ start bit + end bit = 10 bits.
- Time = (10*3) * bit-time = 30 / baud rate = 30 / 9600 = 3.125 mS.

# Sheet 6

- 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.

# Answer

```
void UART_Init(void){ // should be called only once
SYSCTL_RCGCUART_R |= 0x0001; // activate UART0
SYSCTL_RCGCGPIO_R |= 0x0001; // activate port A
UART0_CTL_R &= ~0x0001; // disable UART
UART0_IBRD_R = 520; // IBRD=int(80000000/(16*9600)) = int(520.8333)
UART0_FBRD_R = 53; // FBRD = int(0.8333 * 64 + 0.5)
UART0_LCRH_R = 0x0070; // 8-bit word length, enable FIFO 001110000
UART0_CTL_R = 0x0301; // enable RXE, TXE and UART 001100000001
GPIO_PORTA_AFSEL_R |= 0x03; // enable alt function PA0 ,PA1
GPIO_PORTA_PCTL_R = (GPIO_PORTA_PCTL_R&0xFFFFFF00)+0x00000011; /*
configure
UART 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
}
```

# Sheet 6

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

# Answer

**We need to check the empty flag of the receiver, if FIFO buffer is not empty, then there is data available to be received.

```
bool UART0_Available(void){
return (UART0_FR_R & 0x010 != 0) ? 0 : 1;
}
```

# Sheet 6

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

# Answer

```
char UART0_Read(void){
while(UART0_FR_R & 0x0010 != 0); //check if the buffer is empty
return (char)(UART0_DR_R & 0xFF); //return the first 8 bits (Data)
}
```

# Sheet 6

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

# Answer

```
void UART0_Write(char data){
while((UART0_FR_R & 0x0020) != 0); //check if the buffer is full
UART0_DR_R = data;
}
```

# Sheet 6

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

# Answer

```
Void read_writeToUpper(void){
char in;
char out;
while(1){
in = UART0_Read();
out = in - 0x20; // To upper case (ex. a = A + 32)
UART0_Write(out);
}}
```

# Thank You