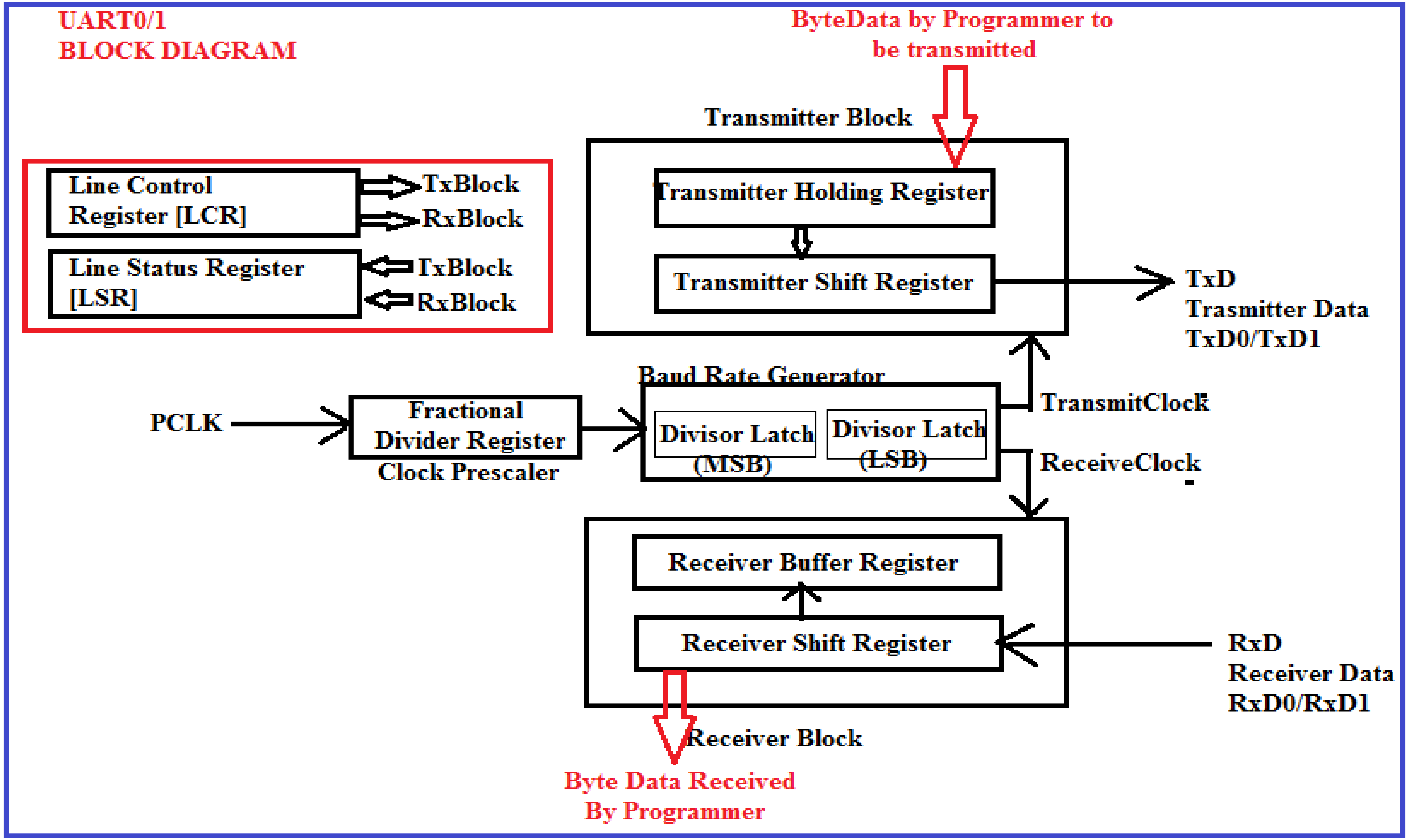
RV College of Engineering®

Go, change the world

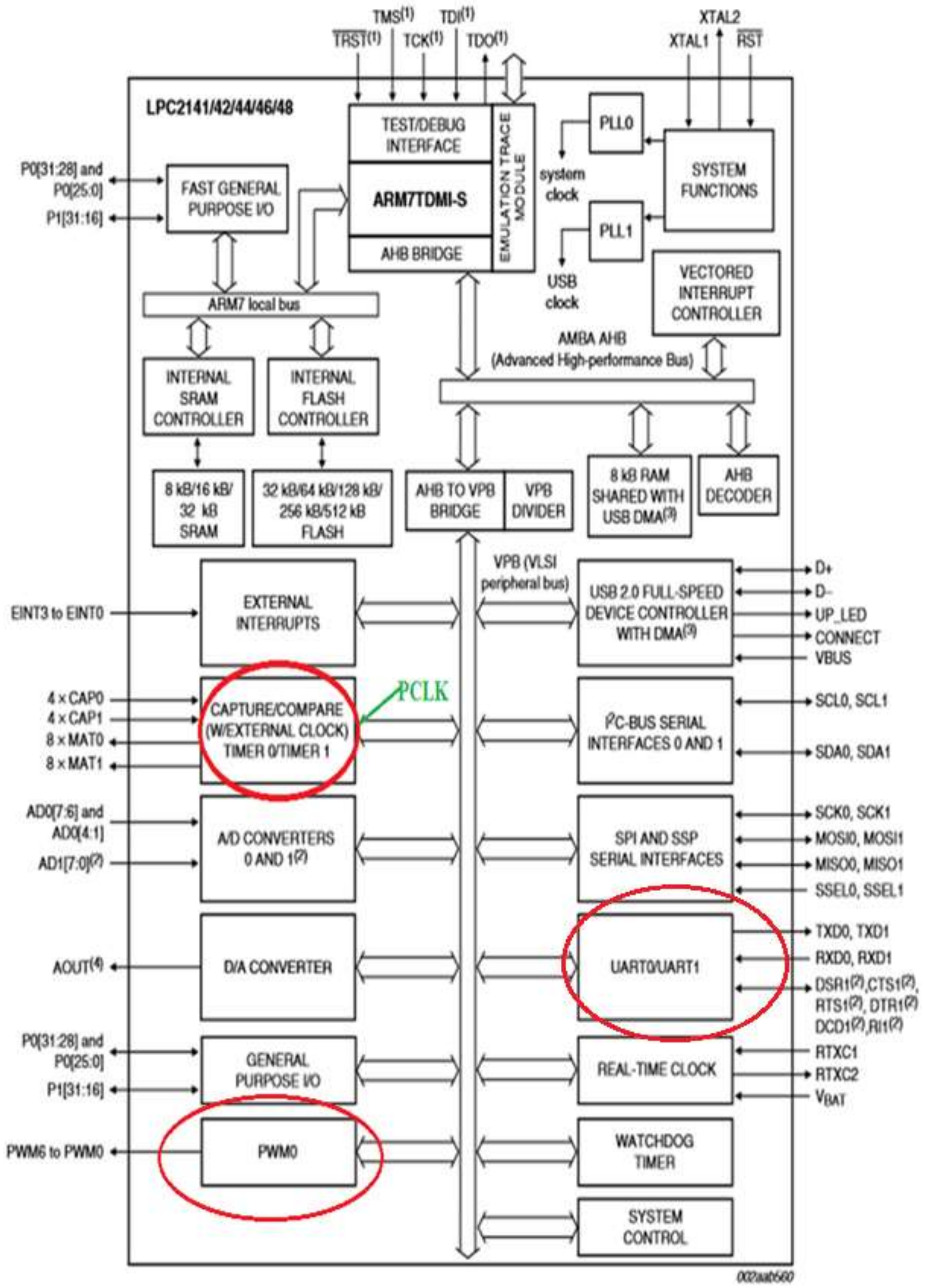
Course Code:CS344AI

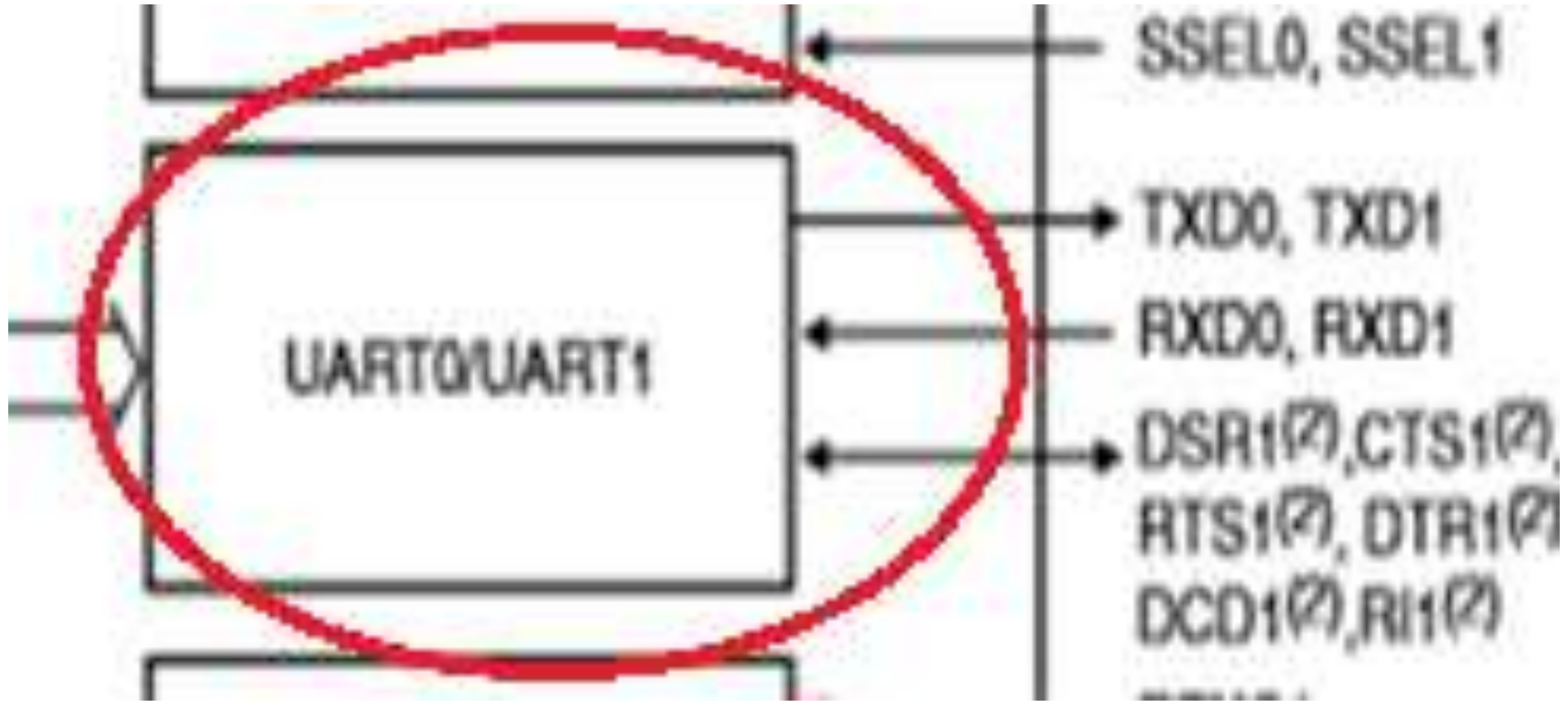
IOT & Embedded Computing

Programming LPC 2148 UART[Serial Port] Programming

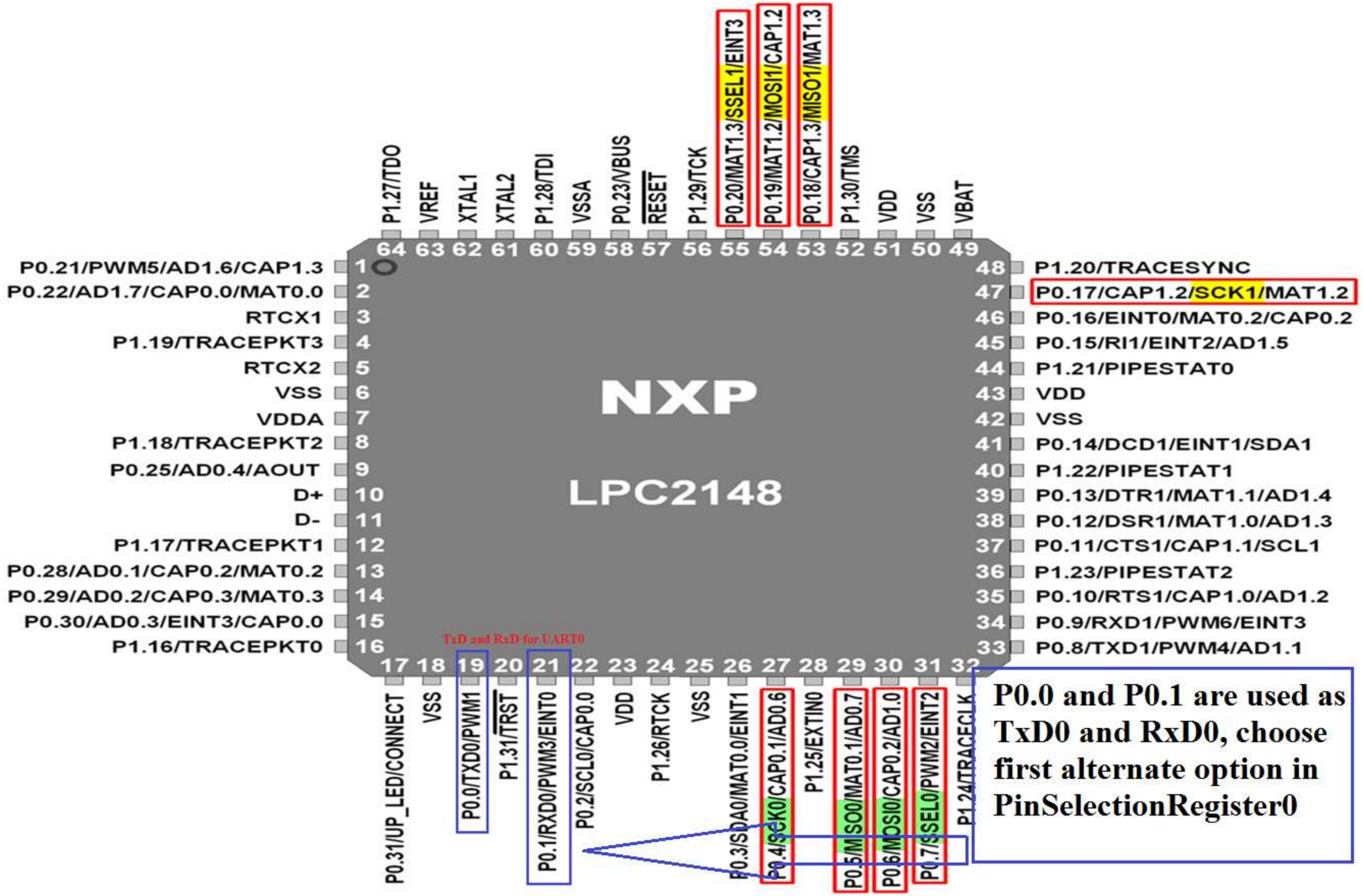

UART0/1
BLOCK DIAGRAM
ByteData by Programmer to be transmitted
Transmitter Block
Line Control Register [LCR]
TxBlock
RxBlock
Line Status Register [LSR]
TxBlock
RxBlock
Transmitter Holding Register
Transmitter Shift Register
TxD
Trasmitter Data
TxD0/TxD1
Baud Rate Generator
PCLK
Fractional Divider Register
Clock Prescaler
Divisor Latch (MSB)
Divisor Latch (LSB)
TransmitClock
ReceiveClock
Receiver Buffer Register
Receiver Shift Register
RxD
Receiver Data
RxD0/RxD1
Receiver Block
Byte Data Received By Programmer

LPC 2148 UART0 & UART1 Unit...





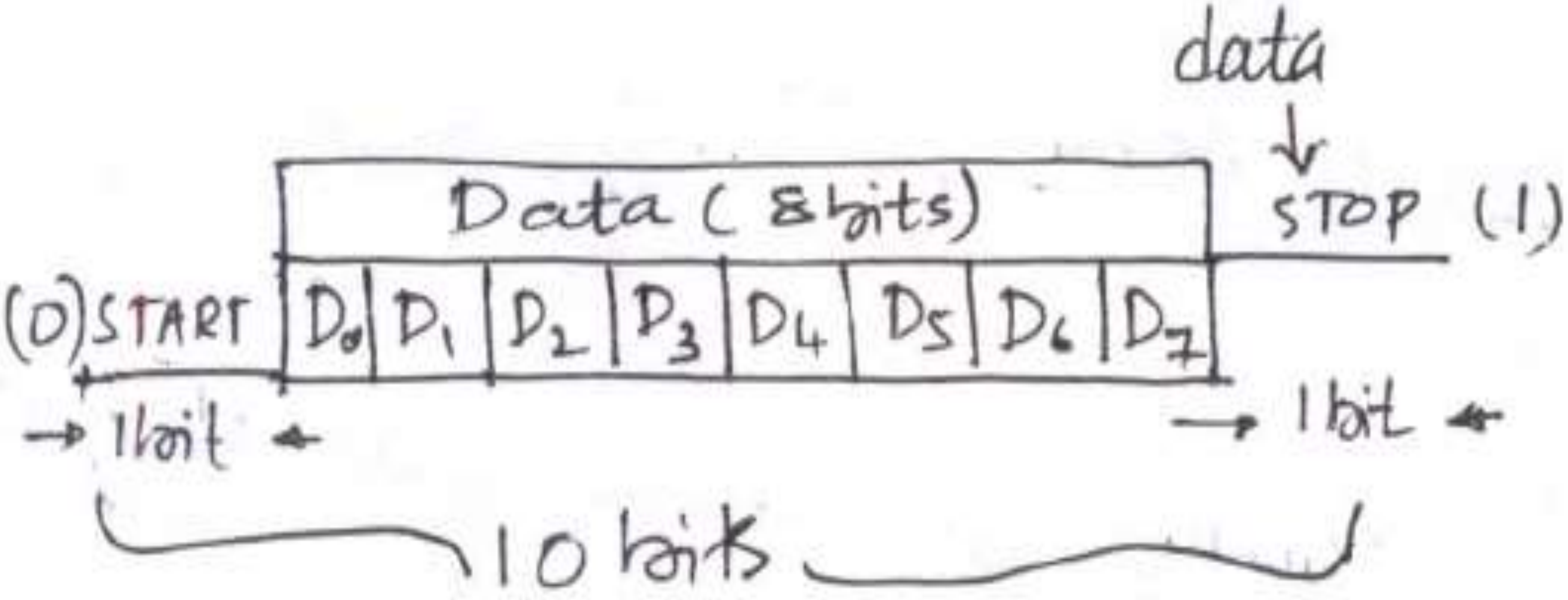
LPC 2148 UART0 & UART1 Unit...



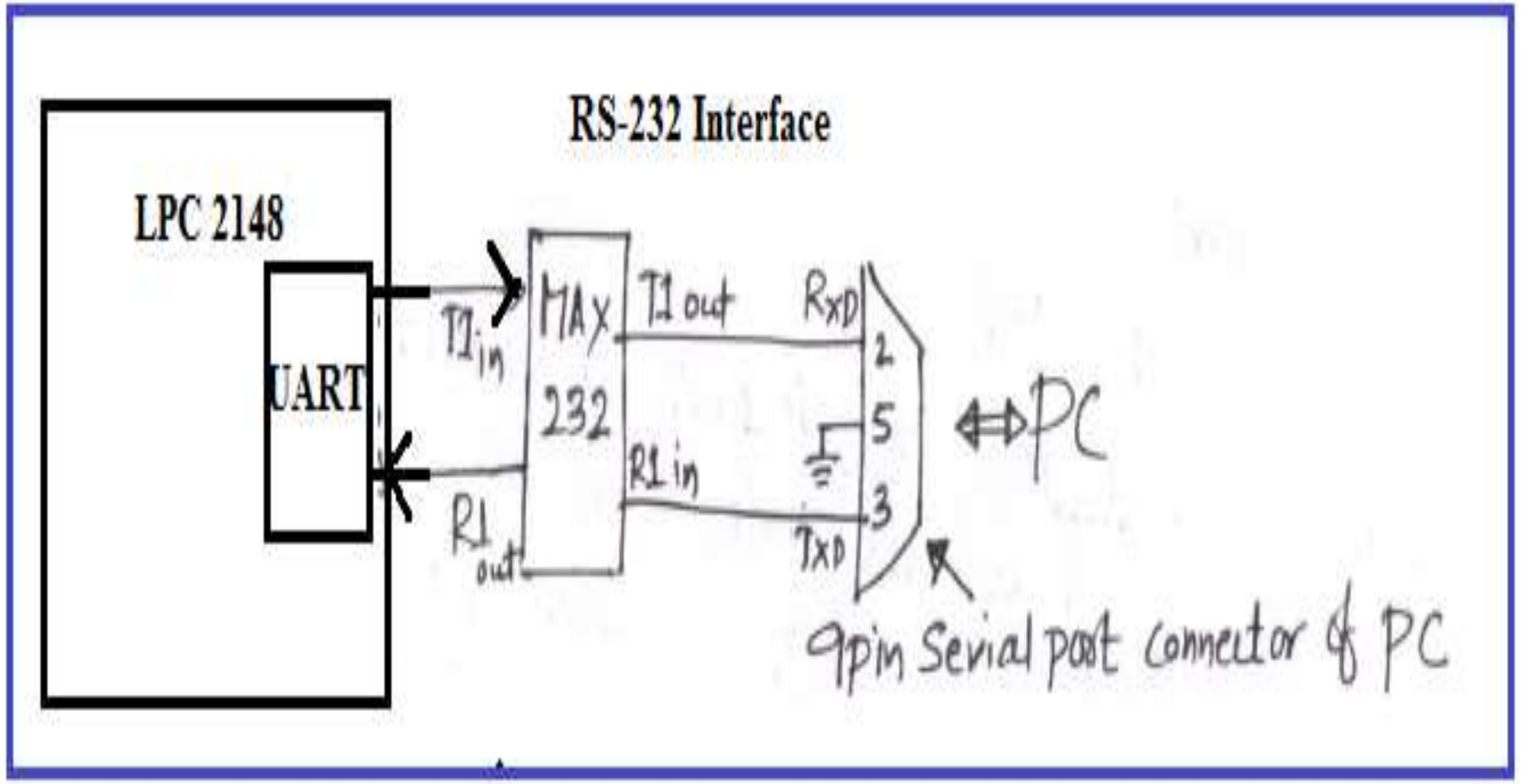
TxD and RxD for UART0

P0.0 and P0.1 are used as TxD0 and RxD0, choose first alternate option in PinSelectionRegister0

Asynchronous Serial Data Format...

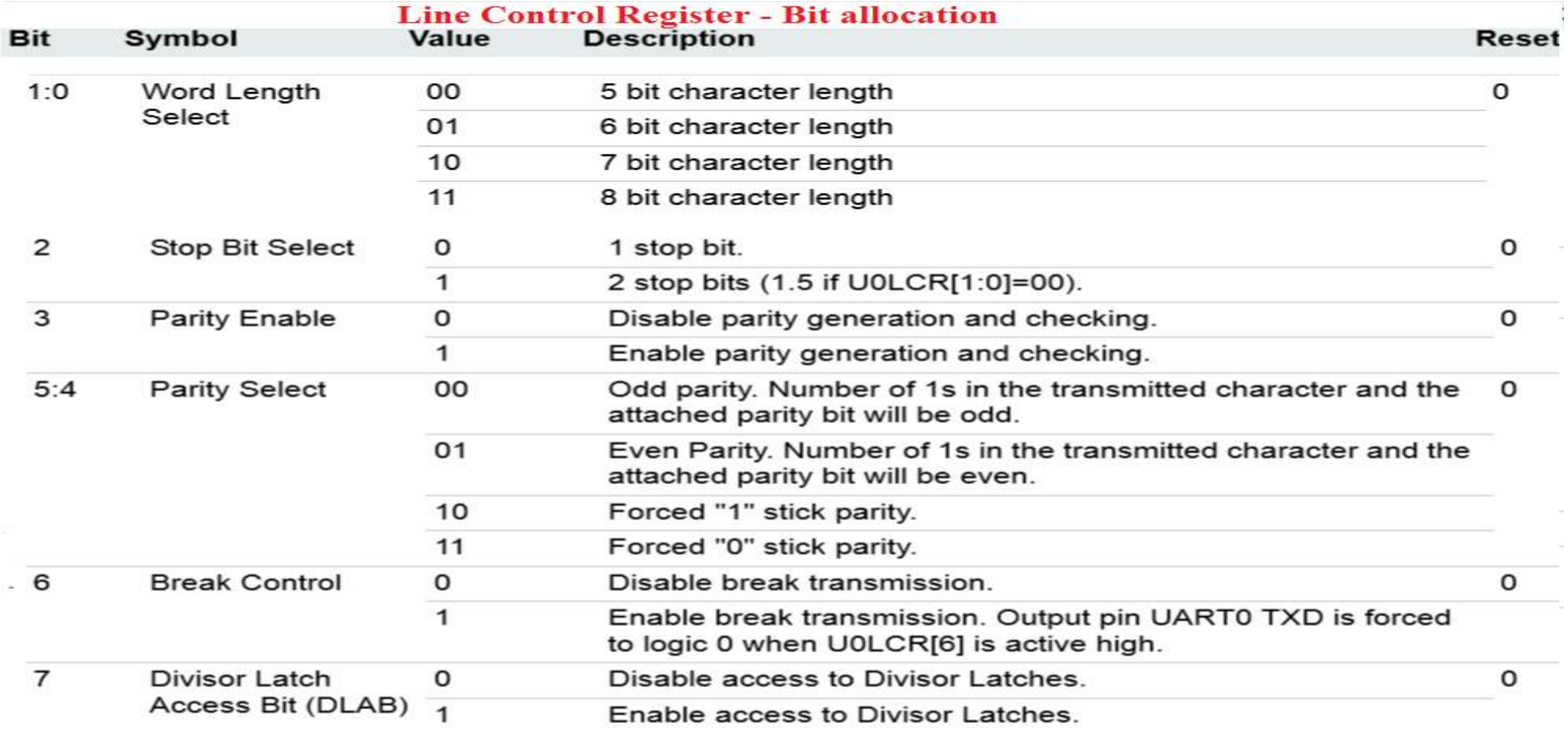


RS 232 Interface



UART Registers..

Register	Description
UxRBR	Contains the recently received Data
UxTHR	Contains the data to be transmitted
UxFCR	FIFO Control Register
UxLCR	Controls the UART frame formatting(Number of Data Bits, Stop bits)
UxDLL	Least Significant Byte of the UART baud rate generator value.
UxDLM	Most Significant Byte of the UART baud rate generator value.
UxLSR	Line status register, used to check transmitter is free / Receiver has data



Line Control Register - Bit allocation

Bit	Symbol	Value	Description	Reset
1:0	Word Length Select	00	5 bit character length	0
		01	6 bit character length	
		10	7 bit character length	
		11	8 bit character length	
2	Stop Bit Select	0	1 stop bit.	0
		1	2 stop bits (1.5 if U0LCR[1:0]=00).	
3	Parity Enable	0	Disable parity generation and checking.	0
		1	Enable parity generation and checking.	
5:4	Parity Select	00	Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.	0
		01	Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.	
		10	Forced "1" stick parity.	
		11	Forced "0" stick parity.	
6	Break Control	0	Disable break transmission.	0
		1	Enable break transmission. Output pin UART0 TXD is forced to logic 0 when U0LCR[6] is active high.	
7	Divisor Latch Access Bit (DLAB)	0	Disable access to Divisor Latches.	0
		1	Enable access to Divisor Latches.	

Example: U0LCR = (0x03<<0) | (1<<7); 8bit data, 1Stop bit, No parity

D7	D6	D5	D4	D3	D2	D1	D0
		THRE					RDR

RDR - Receiver Data Ready

THRE - Transmitter Holding Register Empty

Line Status Register

BaudRate Calculation..

$$\text{BaudRate} = \frac{\text{PCLK}}{(16 * (\text{ DLM:DLL}) * (1+ \text{DivAddVal/MulVal}))}$$

By disabling fraction divider, (taking default values on reset, DivVal=0,MulVal=1)

$$\text{BaudRate} = \frac{\text{PCLK}}{16 * (\text{ DLM:DLL})}$$

$$\text{DLM:DLL} = \frac{\text{PCLK}}{16 * \text{ BaudRate}} = \text{Result}$$

DLL = Lower 8 bits of Result

DLM = Upper 8 bits of Result

Ex: DLL = Result & 0xFF
DLM = (Result >> 8) & 0xFF

Configure UART for 115200 baud rate, PCLK = 15MHz

Calculation:
DLM:DLL = 15000000 / 16 * (115200) = result
Hence, DLL = result % 256 = 8 (lower 8 bits of result)
DLM = result / 256 (upper 8 bits of result)
(or , use calculator, convert answer in decimal to HEX, take lower two digits put into DLL, next two digits to DLM)

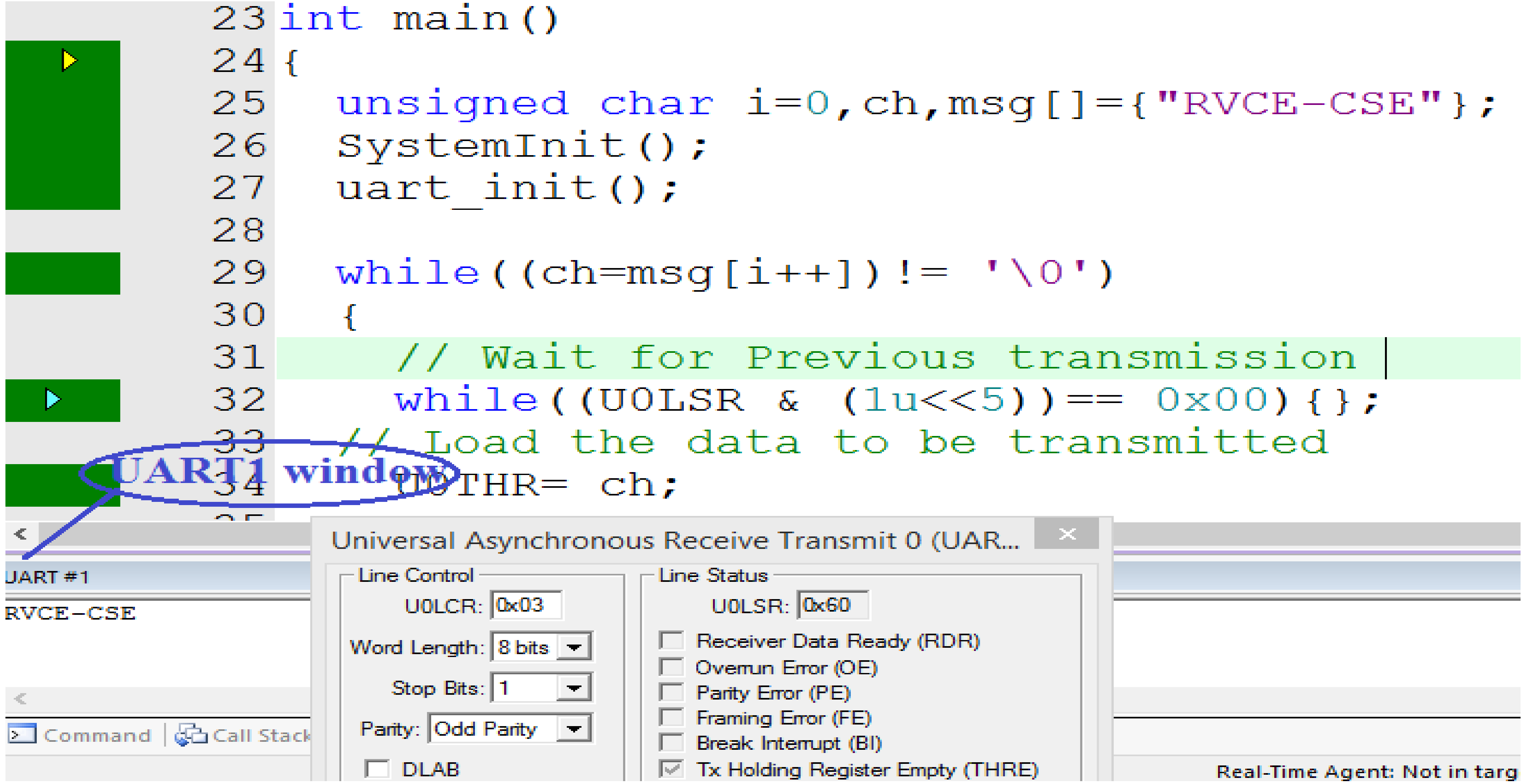
```
void uart_init(void)
{
 //configurations to use serial port, UART0
 PINSEL0 |= 0x00000005;  // P0.0 & P0.1 ARE CONFIGURED AS TXD0 & RXD0
//programming the baud rate
 U0LCR = 0x83;   /* 8 bits, no Parity, 1 Stop bit    & DLAB = 1 */
 U0DLM = 0; U0DLL = 8; // 115200 baud rate, PCLK = 15MHz
 U0LCR = 0x03; /* 8 bits, no Parity, 1 Stop bit    & DLAB = 0 */
}
```

Write a Program to transmit a string from LPC 2148 to PC using Serial port / UART0

```
int main()
{
    unsigned char i=0,ch,msg[]={"RVCE-CSE"};
    SystemInit();  // set PCLK = 15MHz
    uart_init();  // refer previous explanation

    while((ch=msg[i++])!= '\0')
    {
       // Wait for Previous transmission to complete
     while((U0LSR & (1u<<5))== 0x00){};
      // Load the data to be transmitted
     U0THR= ch;
    }
}
```

Write a Program to transmit a string from LPC 2148 to PC using Serial port / UART0



```
int main()
{
  unsigned char i=0,ch,msg[]={"RVCE-CSE"};
  SystemInit();
  uart_init();

  while((ch=msg[i++])!= '\0')
  {
    // Wait for Previous transmission
    while((U0LSR & (1u<<5))== 0x00){};
  // Load the data to be transmitted
  U0THR= ch;
```

UART #1

```
RVCE-CSE
```

Universal Asynchronous Receive Transmit 0 (UAR...

Line Control

U0LCR: 0x03

Word Length: 8 bits

Stop Bits: 1

Parity: Odd Parity

□ DLAB

Line Status

U0LSR: 0x60

□ Receiver Data Ready (RDR)

□ Overrun Error (OE)

□ Parity Error (PE)

□ Framing Error (FE)

□ Break Interrupt (BI)

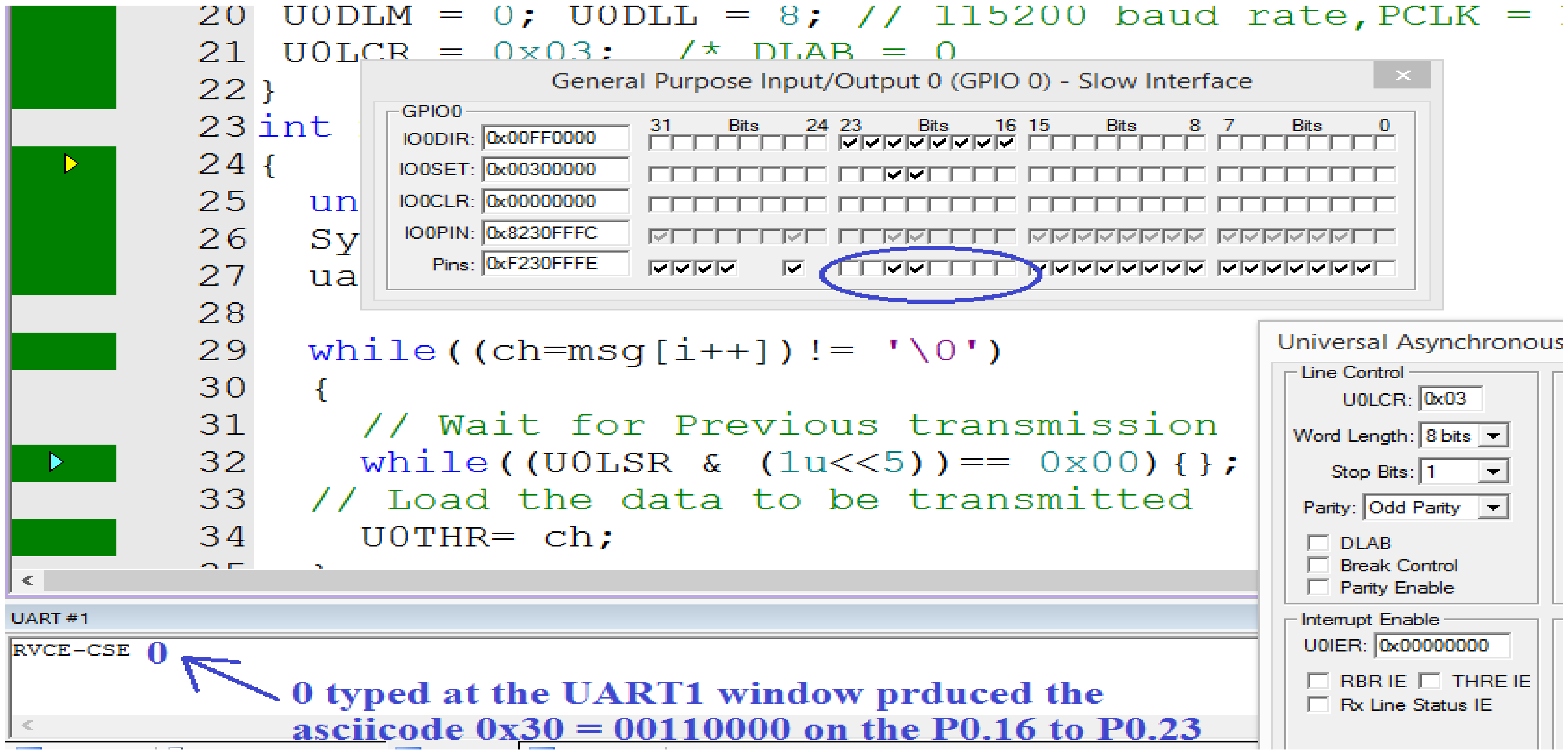
☑ Tx Holding Register Empty (THRE)

Write a Program to receive a character from PC and output the 8 bit data on P0.16 to P0.23.

```
int main()
{
    unsigned char i=0;
    SystemInit();
    uart_init();

    IO0DIR = 0XFF << 16;  // configure P0.16 to P0.23 as output pins
    do
    {
       while((U0LSR & (0x01<<0))== 0x00){};  // wait till, a character (8bit) is received from PC
       i = UORBR;  // read from the  UART0
       IO0CLR = 0XFF << 16;IO0SET = U0RBR << i; // output the  i(8bit code) to P0.16 – P0.23
    }
   while(1);
}
```

Write a Program to receive a character from PC and output the 8 bit data on P0.16 to P0.23.



```
U0DLM = 0; U0DLL = 8; // 115200 baud rate,PCLK =
U0LCR = 0x03;   /* DLAB = 0
}
int
{
  un
  Sy
  ua

  while((ch=msg[i++])!= '\0')
  {
    // Wait for Previous transmission
    while((U0LSR & (1u<<5))== 0x00){};
  // Load the data to be transmitted
    U0THR= ch;
```

General Purpose Input/Output 0 (GPIO 0) - Slow Interface

GPIO0

- IO0DIR: 0x00FF0000
- IO0SET: 0x00300000
- IO0CLR: 0x00000000
- IO0PIN: 0x8230FFFC
- Pins: 0xF230FFFE

Universal Asynchronous

Line Control

- U0LCR: 0x03
- Word Length: 8 bits
- Stop Bits: 1
- Parity: Odd Parity
- □ DLAB
- □ Break Control
- □ Parity Enable

Interrupt Enable

- U0IER: 0x00000000
- □ RBR IE □ THRE IE
- □ Rx Line Status IE

UART #1

```
RVCE-CSE 0
```

0 typed at the UART1 window prduced the asciicode 0x30 = 00110000 on the P0.16 to P0.23