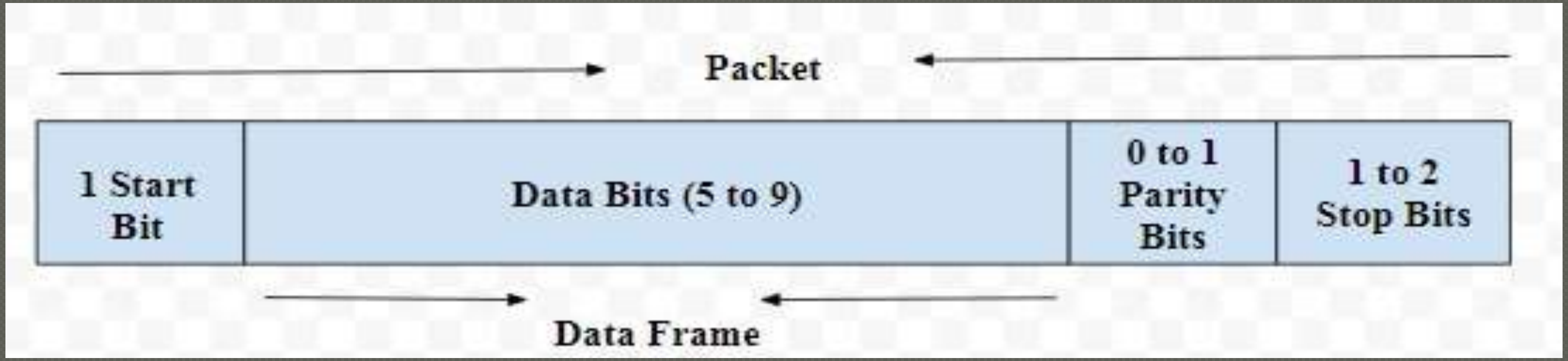
UART

Lakshmi R, Asst. Professor, Dept. of CSE

Universal Asynchronous Receiver Transmitter (UART)

- Asynchronous – no clock for data synchronization.
- For synchronization – predefined baud rate.
- Communication settings – Baud rate and UART data frame.



Packet

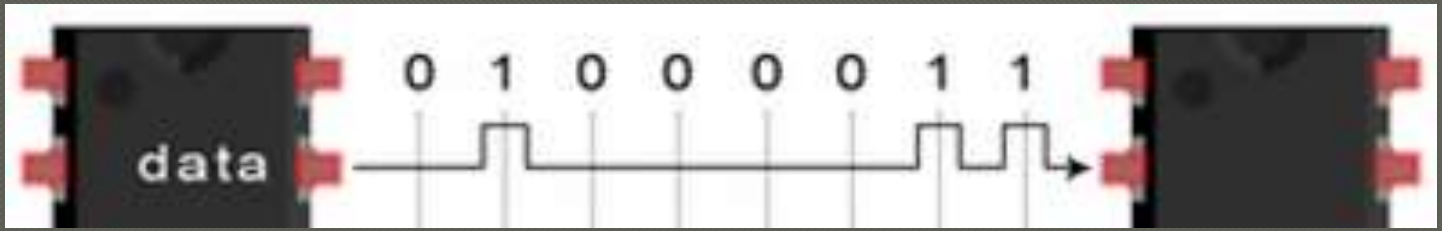
1 Start Bit

Data Bits (5 to 9)

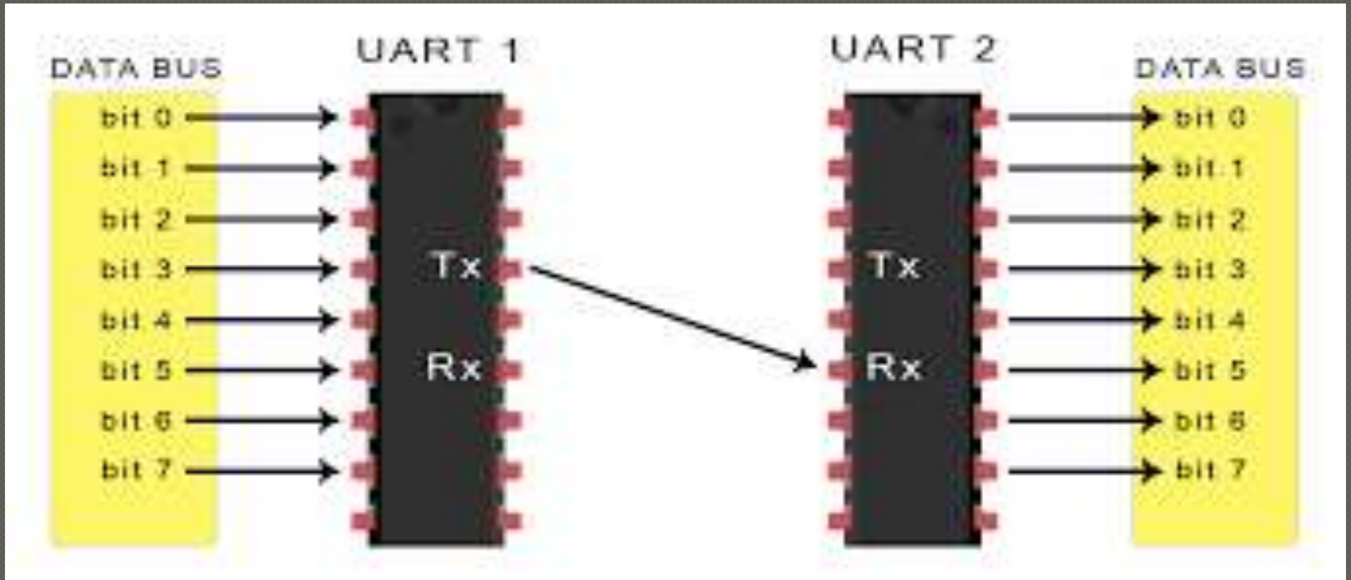
0 to 1 Parity Bits

1 to 2 Stop Bits

Data Frame



Working of UART



Start bit:

T_x – normally high (no transmission)

– Rx doesn't read

T_x – pulled low (begin transmission)

– Rx begins to read

UART Steps

1. Configure the functionality for pins
2. Set line control configuration
   a. Define frame formats (8 bits, no Parity, 1 Stop bit)
   b. Enable access to divisor latch to set baud rate
3. Set and load the baud rate
4. Disable access to divisor latch
5. Transmit ASCII bytes to display (HELLO WORLD)
6. Check the line status for data presence

   If data is present, then wait until it is written

   Else, write 8-bit (ASCII) data

1. Configure the functionality for pins

UART	T_x pin	R_x Pin
UART0	P0.0	P0.1
UART1	P0.8	P0.9

We are using UART0. ∴ pins P0.0 and P0.1 have to be configured. PINSEL0 to be used

P0.1 /	RXD0 /	PWM3 /	EINT0
00	01	10	11

P0.0 /	TXD0 /	PWM1
00	01	10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0.15		P0.14		P0.13		P0.12		P0.11		P0.10		P0.9		P0.8		P0.7		P0.6		P0.5		P0.4		P0.3		P0.2		P0.1		P0.0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

1. Configure the functionality for pins

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0.15		P0.14		P0.13		P0.12		P0.11		P0.10		P0.9		P0.8		P0.7		P0.6		P0.5		P0.4		P0.3		P0.2		P0.1		P0.0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

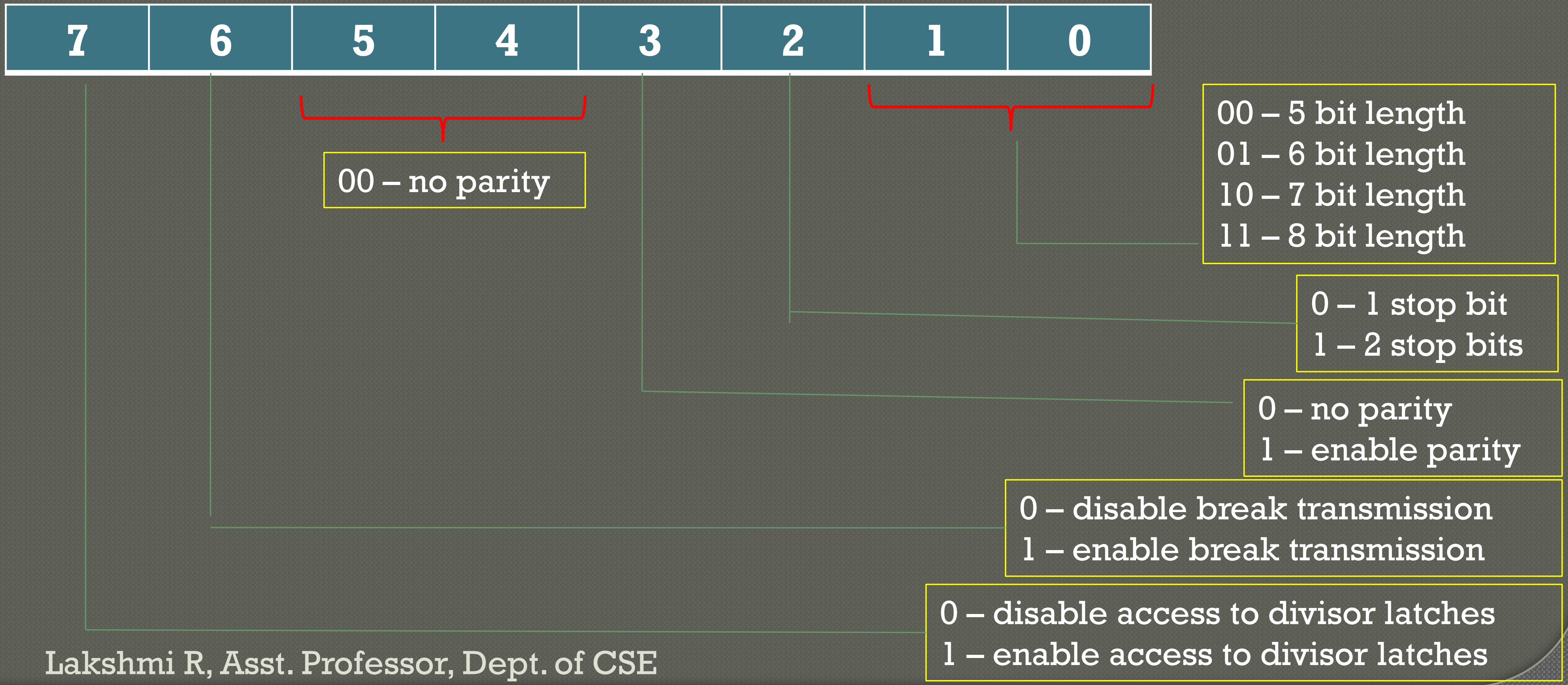
0 0 0 0 0 0 0 5

(Hex Value)

PINSEL0 = 0X00000005

2. Set line control configuration

UART line control register – U0LCR



7	6	5	4	3	2	1	0

00 – no parity

00 – 5 bit length
01 – 6 bit length
10 – 7 bit length
11 – 8 bit length

0 – 1 stop bit
1 – 2 stop bits

0 – no parity
1 – enable parity

0 – disable break transmission
1 – enable break transmission

0 – disable access to divisor latches
1 – enable access to divisor latches

2. Set line control configuration

UART line control register – U0LCR

8-bit ASCII value

1 stop bit

No parity

No break transmission

Enable to access divisor latch

7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	1

U0LCR = 0X83

3. Set and load baud rate

Baud rate → 9600 bits per second.

This rate to be set in Divisor Latch Register (DLR) →

U1DLM – most significant byte of DLR

U1DLL – least significant byte of DLR

DLM:DLL → together 16 bits

Formula:

$$\text{Baud Rate} = \frac{\text{PCLK}}{16 \text{ X } (\text{DLM:DLL})}$$

$$\text{DLM:DLL} = \frac{\text{PCLK}}{16 \text{ X Baud Rate}}$$

PCLK → Peripheral Clock cycle

= 1 /4 CPU clock ---- (12 MHz)

∴ PCLK = 3MHz

Baud rate = 9600 bps

16 →it takes 16 clocks to send 1 bit

$$\text{DLM:DLL} = \frac{3000000}{16 \text{ X } 9600} \approx 19 \text{ (only 8-bits)}$$

DLL = 19 and DLM = 0

U0DLM = 0

U0DLL = 0x13

4. Disable access to divisor latch

UART line control register – U0LCR

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

U0LCR = 0X03

5. Transmit ASCII bytes to display (HELLO WORLD)

- U0THR → Transmit Holding Register → 8-bit Write Data
- U0THR = ASCII byte

6. Check the line status for data presence

UART line status register – U0LSR

7	6	5	4	3	2	1	0
Error in RX FIFO (RXFE)	Transmitter Empty (TEMT)	Transmitter Holding Register Empty (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	Receiver Data Ready (RDR)

(THRE)

Bit 0 → U0THR contains valid data

Bit 1 → U0THR is empty

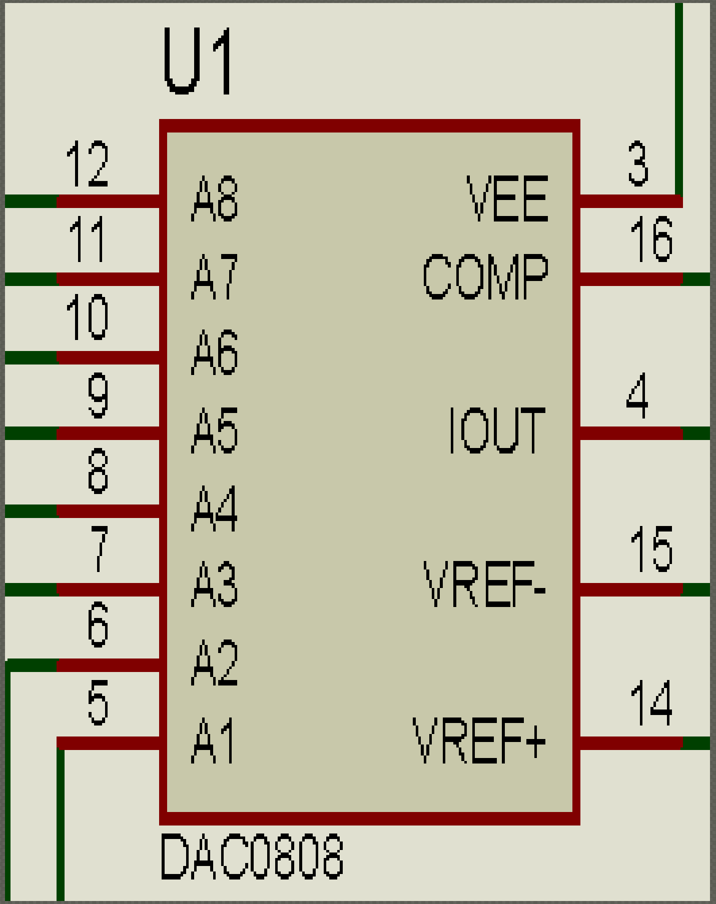
- If U0LSR = 0 0 0 0 0 0 0 0, then U1THR contains valid data
  (wait until it becomes empty to write next byte)
- If U0LSR = 0 0 1 0 0 0 0 0 (or 0x20) , then U0THR is empty
  (now write new byte to U0THR)

```
#include<lpc214x.h>

unsigned int delay;
unsigned char *ptr,arr[]=“HELLO WORLD\r";

int main()
{
        PINSEL0=0X0000005;              //select TXD0 and RXD0 lines   → 0101
        U0LCR = 0X00000083;            //enable baud rate divisor loading and
        U0DLM = 0X00;                  //select the data format
         U0DLL = 0x13;                 //select baud rate 9600 bps
         U0LCR = 0X00000003;
```

```
    while(1)
    {
            ptr = arr;
            while(*ptr!='\0')
            {
             U0THR=*ptr++;
              while(!(U0LSR & 0x20)== 0x20);
              for(delay=0;delay<=600;delay++);
            }
            for(delay=0;delay<=60000;delay++);
    }
}
```

DAC

External DAC0808



LPC2148 Port Pins Used : P0.16 –P0.23

External DAC0808

DAC0808 is an 8bit DAC.

Minimum value = 00000000 in binary

= 00 in decimal

= 0x00 in hexadecimal

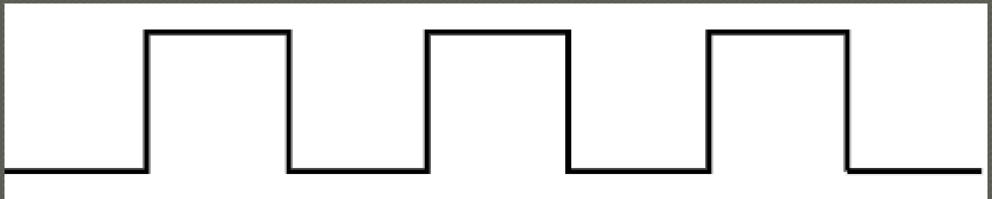
Maximum value = 11111111 in binary

= 255 in decimal

= 0xFF in hexadecimal

Program for Square Wave

```
//program to generate square wave with DAC interface
#include <lpc21xx.h>
unsigned int delay;
int main ()
{
        PINSEL1 = 0x00000000 ;   // Configure P0.16 to P0.31 as GPIO
        IO0DIR  = 0x00FF0000 ;
        while(1)
        {
                IO0PIN = 0x00000000;
                for(delay=0;delay<=950;delay++);
                IO0PIN = 0x00FF0000;
                for(delay=0;delay<=950;delay++);
         }
}
```



Program for Triangular Wave

```
#include <LPC21xx.h>
unsigned long int temp=0x00000000;

int main ()
{
          unsigned int i=0;
         IO0DIR=0x00FF0000;
         while(1)
          {
              // output 0 to FE
                  for(i=0;i!=0xFF;i++)
                       {
                                           temp=i;
                                           temp = temp << 16;
                                           IO0PIN=temp;
                         }
```

```
// output FF to 1
for(i=0xFF; i!=0;i--)
     {
             temp=i;
             temp = temp << 16;
             IO0PIN=temp;
     }
             }//End of while(1)
}//End of main()
```