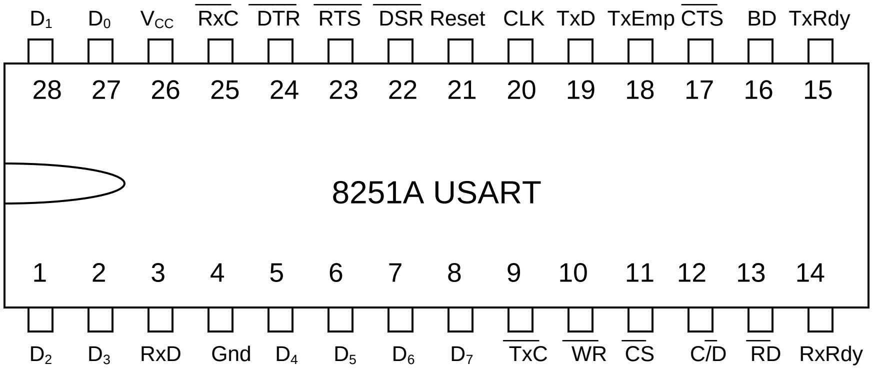
# ENBEDDED HARDWARE DESIGN

**Experiment 1 Asynchronous Serial Port 2009-10/I**

This experiment will address the operational features of the programmable serial interface 8251A, often referred to as a USART (Universal Synchronous/Asynchronous Receive/Transmit interface). This chip is used extensively in microprocessor systems for providing an asynchronous serial port, but not so much as a synchronous interface. In this experiment, therefore, we will concentrate only on the asynchronous port behaviour.



$D_1$ $D_0$ $V_{CC}$ $\overline{\text{RxC}}$ $\overline{\text{DTR}}$ $\overline{\text{RTS}}$ $\overline{\text{DSR}}$ Reset CLK TxD TxEmp $\overline{\text{CTS}}$ BD TxRdy
28 27 26 25 24 23 22 21 20 19 18 17 16 15
8251A USART
1 2 3 4 5 6 7 8 9 10 11 12 13 14
$D_2$ $D_3$ RxD Gnd $D_4$ $D_5$ $D_6$ $D_7$ $\overline{\text{TxC}}$ $\overline{\text{WR}}$ $\overline{\text{CS}}$ C/$\overline{\text{D}}$ $\overline{\text{RD}}$ RxRdy

## 1. Connections

Note the following connections made on the breadboard to the 8251A chip, which will remain unchanged throughout this experiment:

(a) The data bus of the 8251A is connected to a 8-bit Tri-state Buffer (TSB) as used for the ALU-Regster experiment, consisting of 8 Input Switches and 8 LED displays.

(b) The C/$\overline{\text{D}}$ input and the RxD input of the 8251A are connected to the two remaining Input Switches.

(c) The $\overline{\text{TxC}}$ and $\overline{\text{RxC}}$ inputs (giving the transmission and reception clocks) of the 8251A are connected together to the output of the manual PULSER.

(d) The TTL output of the Function Generator, with its frequency set at 1 MHz, is connected to the CLK input of the 8251A.

(e) The $\overline{\text{WR}}$ input is connected to the common control line of the TSB. The $\overline{\text{RD}}$ and $\overline{\text{WR}}$ pins are normally kept HIGH (and hence inactive) by connecting them to $V_{CC}$ through resistors, as in Experiment 10. Read/write pulses are applied manually, whenever necessary, by connecting the corresponding pin momentarily to Gnd by a wire.

## 2. Structures of the Mode, Command and Status bytes

(a) Mode Byte, written with C/$\overline{\text{D}}$ = 1 after RESET:

| Number of STOP bits | Parity | Parity Enable | Bits per Character | TxC Divider |
|---|---|---|---|---|
| 00 – Invalid, 01 – 1,<br>10 – 1.5, 11– 2 | 0 – Odd<br>1 – Even | 0 – Disabled<br>1 – Enabled | 00 – 5, 01 – 6,<br>10 – 7, 11 – 8 | 00 – Sync, 01 – 1x<br>10 – 16x, 11 – 64x |

(b) Command Byte, written with C/D = 1 after Mode has been configured:

| Enter Hunt (EH) | Internal Reset (IR) | Set RTS (RTS) | Error Reset (ER) | Send Break (SBRK) | Enable RxRdy (RxE) | Set DTR (DTR) | Enable Tx (TxEn) |
|---|---|---|---|---|---|---|---|

(c) Status Byte, read with C/D = 1:

| $\overline{DSR}$ | SYNDET/ Break | Framing Error (FE) | Overrun Error (OE) | Parity Error (PE) | TxEmpty | RxRdy | TxRdy |
|---|---|---|---|---|---|---|---|

## 3. Testing Serial Data Transmission by the 8251A

(a) Connect the $\overline{CTS}$ pin to $V_{CC}$ so as to disable transmission. Reset the 8251A by disconnecting the Reset pin from its normal position (Gnd), connecting it momentarily to $V_{CC}$ and restoring the original position of the connecting wire.

(b) Programme the 8251A for asynchronous data transmission with 2 STOP bits, even PARITY, 6 BITS/CHARACTER and 1x TxC Divider by applying a write pulse with the required mode byte (**11110101)** applied through the TSB Input Switches and C/D = 1.

(c) Apply a command to reset all previous error flags and to enable Tx only, by applying a write pulse with the required command byte (**00010001)** applied through the TSB Input Switches and C/D = 1.

(d) Read the Status byte by applying a read pulse manually with $C/\overline{D}$ = 1. Note down all the bits and interpret their observed values. Compare these values with the logic levels at the TxRdy and TxEmpty pins by testing the pins with the LED probe.

(e) Connect the $\overline{CTS}$ pin to Gnd and repeat step **3**(d).

(f) Write a suitable byte into the Tx buffer by setting the desired byte on the TSB Switches and applying a write pulse with C/D = 0. Note the change in the values of the TxRdy and TxEmpty bits, both by reading the Status byte and by testing the pins.

(g) With the LED probe connected to the TxD output, apply TxC pulses from the PULSER and note down the sequence of the TxD output. Interpret the observed 10-bit sequence in terms of the byte written into the Tx buffer in step (f) and the mode programming.

## 4. Testing Serial Data Reception by the 8251A

(a) Repeat steps (a) and (b) of part **3**, and apply a command to reset error flags and enable Rx only, by applying a write pulse with the required command byte (**00010100)** applied through the TSB Input Switches and C/D = 1.

(b) Note the logic level at the RxRdy pin by the LED probe and read the Status byte by applying a read pulse manually with C/D = 1. Note down and interpret the observed bits.

(c) Determine the sequence of 10 bits which would correspond to a suitable 6-bit data according to the mode programming done, and apply this sequence of 10 bits, LSB first, to the RxD pin by manually applying 10 RxC pulses from the PULSER.

(d) Repeat step (b) above and also read the received data by applying a read pulse with $C/\overline{D}$ = 0. Interpret the observed bits.

(e) Repeat the above steps with an erroneous bit sequence and observe the error flags.

## 5. Loopback Test

(a) Disconnect the Input Switch connection to the RxD pin. Connect TxD to RxD to make a loop-back arrangement, where the data written into the Tx register gets transmitted serially into the Rx register of the same chip and can be read back.

(a) Repeat steps (a) and (b) of part **3**, and apply a command to reset error flags and enable both Tx and Rx, by applying a write pulse with the required command byte (**00010101)** applied through the TSB Input Switches and C/D = 1.

(b) Repeat steps **3**(e), **3**(f) and **4**(d).