uart6551

© 2019 Robert Finch

# Overview

A UART component (Universal Asynchronous Transmitter / Receiver) is used for the asynchronous transmission and reception of data. Asynchronous referring to the lack of a clock signal during transmission or reception.

uart6551 is a WDC6551 register compatible uart. The uart is a 32-bit peripheral device. It may be used as an eight-bit peripheral by connecting the high order 24-bit data input lines to ground, and grounding select lines one to three.

Baud rate is controlled by clock divider which assumes a 200MHz baud reference clock input. If a different clock frequency is used, then the divider table will need to be updated. The baud rate may also be controlled via a clock divider register. This register is 24 bits so gives a minimum frequency of 11.92 Hz assuming a 200MHz clock. (200MHz / 2^24).

## Special Features

* WDC6551 register compatibility

# Register Description

There are only four registers in the design. The function of the low order eight bits of the registers matches the 6551 function. The controller honors byte lane selects so only the portion of the register selected is written.

|  |  |  |
| --- | --- | --- |
| Reg | Moniker | Description |
| 0 | UART\_TRB | Transmit and receive buffer. Data written is transmitted, on a read data available is read. Also reads / writes the clock multiplier if access to clock multiplier is enabled. |
| 1 | UART\_STAT | Status Register. Returns status bits on a read, a write of any value will cause a reset of some of the command register bits |
| 2 | UART\_CMD | Command register |
| 3 | UART\_CTRL | Control register |

## UART\_TRB

This register is 32-bits wide of which only the lower eight bits are used to transmit or receive data by the uart. Data written to the register is transmitted. A register read returns data received by the uart. When the fifo’s are enabled writing to this register writes to the transmit fifo. Reading this register reads the receive fifo. If clock divider access is enabled (via control register bit 31) then this register allows modifying or reading the clock divider value. Writing a clock divider value to this register automatically switches the function back to transmit / receive.

## UART\_STAT

Uart status register. Writing any value to the status register resets some of the uart’s command bits.

|  |  |  |
| --- | --- | --- |
| Bit | Status |  |
| 0 | Parity Error | 1 = parity error occurred, 0 = no error |
| 1 | Framing Error | 1 = framing error |
| 2 | Overrun | 1 = overrun |
| 3 | Rx Full | 1 = receiver data available |
| 4 | Tx Empty | 1 = open slot in transmit fifo |
| 5 | DCD | 0 = data carrier present |
| 6 | DSR | 0 = data set ready |
| 7 | IRQ | 1 = irq occurred |
|  | **Additional Line Status Byte** | |
| 8 | reserved |  |
| 9 | reserved |  |
| 10 | reserved |  |
| 11 | reserved |  |
| 12 | Break received | 1 if a break signal is received |
| 13 | Tx Full | 1 = transmit fifo full |
| 14 | reserved |  |
| 15 | G Rcv Err | 1 = global receiver error (set if any error status is set) |
|  | **Additional Modem Status Byte** | |
| 16 | CTS | 1 = CTS line changed state |
| 17 | DSR | 1 = DSR line changed state |
| 18 | RI | 1 = RI line changed state |
| 19 | DCD | 1 = DCD line changed state |
| 20 | CTS | CTS state |
| 21 | reserved |  |
| 22 | RI | RI state |
| 23 | reserved |  |
|  | **IRQ Status** | |
| 24,25 | zero | these two bits are zero |
| 26 to 28 | IRQENC | encoded irq value (0 to 7) |
| 29 to 30 | reserved |  |
| 31 | irq | IRQ is set |

## UART\_CMD

|  |  |  |
| --- | --- | --- |
| Bit |  |  |
| 0 | DTR | output 1 = low, 0 = high |
| 1 | RxIe | receiver interrupt enable 0 = enabled, 1 = disabled |
| 2,3 | RTS Control |  |
|  | 00 | output RTS high |
|  | 01 | output RTS low, enable transmit interrupt |
|  | 10 | output RTS low, |
|  | 11 | output RTS low, send a break signal |
| 4 | LLB | 1 = local loopback (receiver echo) |
| 5 to 7 | Parity Control |  |
|  | 000 | no parity |
|  | 001 | odd parity |
|  | 011 | even parity |
|  | 101 | transmit mark parity (parity error disabled) |
|  | 111 | transmit space parity (parity error disabled) |
| 8 | LSIe | line status change interrupt enable 1 = enabled |
| 9 | MSIe | modem status change interrupt enable 1 = enabled |
| 10 | RxToIe | receiver timeout interrupt enable 1 = enabled |
| 11 to 31 | reserved |  |

## UART\_CTRL

|  |  |  |
| --- | --- | --- |
| Bit |  |  |
| 0 to 3 | Baud Rate |  |
|  | |  |  | | --- | --- | | 0000 | Use 16x external clock | | 0001 | 50 | | 0010 | 75 | | 0011 | 109.92 | | 0100 | 134.58 | | 0101 | 150 | | 0110 | 300 | | 0111 | 600 | | 1000 | 1200 | | 1001 | 1800 | | 1010 | 2400 | | 1011 | 3600 | | 1100 | 4800 | | 1101 | 7200 | | 1110 | 9600 | | 1111 | 19200 | | This table is expanded using an extra control bit #27. |
| 4 | Rx clock source | 1 = external, 0 = baud rate generator |
| 5,6 | Word length   |  |  | | --- | --- | | 00 | 8 | | 01 | 7 | | 10 | 6 | | 11 | 5 | | code for word length in bits |
| 7 | Stop Bit   |  |  | | --- | --- | | 0 | 1 | | 1 | 1 if 8 bits and parity | | 1 | 1.5 if 5 bits and no parity | | 1 | 2 otherwise | |  |
| 8 to 15 | reserved | do not use |
| 16 | Fifo enable | 1 = fifo’s enabled |
| 17 | Rx Fifo Clear | 1 = clear receiver fifo |
| 18 | Tx Fifo Clear | 1 = clear transmit fifo |
| 19 | reserved |  |
| 20,21 | Transmit Threshold   |  |  | | --- | --- | | 0 | 1 byte | | 1 | ¼ full | | 2 | ½ full | | 3 | ¾ full | | Threshold for DMA signal activation  If the transit fifo count is less than the threshold then a DMA transfer is triggered. |
| 22, 23 | Receive Threshold   |  |  | | --- | --- | | 0 | 1 byte | | 1 | ¼ full | | 2 | ½ full | | 3 | ¾ full | | Threshold for DMA signal activation. If the receive fifo count is greater than the threshold then a DMA transfer is triggered. |
| 24 | hwfc | 1 = automatic hardware flow control |
| 25 | reserved |  |
| 26 | dmaEnable | 1 = dma enabled |
| 27 | Baud Rate bit 4   |  |  | | --- | --- | | 10000 | 38400 | | 10001 | 57600 | | 10010 | 115200 | | 10011 | 230600 | | 10100 | 460800 | | 10101 | 921600 | | 10110 | reserved | | 10111 | reserved | | 11xxx | reserved | | Extended baud rate selection bit, used in combination with bits 0 to 3. |
| 28,29 | reserved |  |
| 30 | selDV | 1 = use clock divider register, 0 = use baud table |
| 31 | accessDV | 1 = access clock divider via TRB register, 0 = normal TRB operation |

Selecting the clock divider register as the baud source allows any programmable baud rate.

# Ports

|  |  |  |  |
| --- | --- | --- | --- |
| Signal | I/O | Wid | Purpose |
| rst\_i | I | 1 | reset |
| clk\_i | I | 1 | bus clock input |
| cs\_i | I | 1 | circuit/core select |
| irq\_o | O | 1 | interrupt request |
|  | WISHBONE SIGNALS | | |
| cyc\_i | I | 1 | bus cycle valid |
| stb\_i | I | 1 | data transfer strobe |
| ack\_o | O | 1 | data transfer acknowledge |
| we\_i | I | 1 | write enable |
| sel\_i | I | 4 | byte lane selects (ground select bits 1 to 3 if using as an 8-bit peripheral) |
| adr\_i | I | 2 | address bits 2,3 (selects register) |
| dat\_i | I | 32 | data input bus (ground bits 8 to 31 if using as an 8-bit peripheral) |
| dat\_o | O | 32 | data output bus |
|  | Modem Controls | | |
| cts\_ni | I | 1 | clear to send input active low. |
| rts\_no | O | 1 | request to send output active low |
| dsr\_ni | I | 1 | data set ready active low |
| dcd\_ni | I | 1 | data carrier detect active low |
| dtr\_no | O | 1 | data terminal ready active low |
| ri\_ni | I | 1 | ring indicator active low |
|  |  |  |  |
| rxd\_i | I | 1 | serial data input (receive) |
| txd\_o | O | 1 | serial data output (transmit) |
| data\_present | O | 1 | data is present in the receiver |
| rxDRQ\_o | O | 1 | receiver DMA request |
| txDRQ\_o | O | 1 | transmitter DMA request |
| xclk\_i | I | 1 | external baud rate clock |
| RxC\_i | I | 1 | external receiver clock |