uart6551sbi

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.

uart6551sbi is a WDC6551 register compatible uart. The uart is a 8-bit peripheral device. The SBI version of the uart uses a simple synchronous bus interface like the 6502 bus.

Baud rate is controlled by clock divider which assumes a 50MHz 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 2.98 Hz assuming a 50MHz clock. (50MHz / 2^24).

Note the use a reference clock separate from the bus clock. It is common in 65xx systems to create wait states by stretching the bus clock. This means the bus clock is not a suitable reference for other timing.

Special Features

- WDC6551 register compatibility
- Independent bus and baud clocks

Register Description

There are seventeen registers in the design. The function of the first four registers matches the 6551 function.

Reg	Moniker	Description
00	UART_TRB	Transmit and receive buffer. Data written is transmitted, on a read data available is read.
01	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
02	UART_CMD	Command register
03	UART_CTRL	Control register

UART_TRB (reg 0)

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.

UART_STAT (reg 1)

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

UART_LS (reg 11)

Uart line status register.

	Additional Line Status Byte	
0	reserved	
1	reserved	
2	reserved	
3	reserved	
4	Break received	1 if a break signal is received
5	Tx Full	1 = transmit fifo full
6	reserved	
7	G Rcv Err	1 = global receiver error (set if any error status is set)

UART_MS (reg 12)

Uart modem status

	Additional Modem Status Byte	
0	CTS	1 = CTS line changed state
1	DSR	1 = DSR line changed state
2	RI	1 = RI line changed state
3	DCD	1 = DCD line changed state
4	CTS	CTS state
5	reserved	
6	RI	RI state
7	reserved	

UART_IRQS (reg 13)

IRQ status register

	IRQ Status	
1,0	zero	these two bits are zero
2 to 4	IRQENC	encoded irq value (0 to 7)
5 to 6	reserved	
7	irq	IRQ is set

UART_CMD (reg 2)

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)

UART_CMD2 (reg 9)

0	LSIe	line status change interrupt enable 1 = enabled
1	MSIe	modem status change interrupt enable 1 = enabled
2	RxToIe	receiver timeout interrupt enable 1 = enabled
3 to 7	reserved	

UART_CTRL (reg 3)

<table>
<tr><td>Bit</td><td></td><td></td></tr>
<tr><td>0 to 3</td><td>Baud Rate</td><td></td></tr>
<tr><td></td><td><table>
<tr><td>0000</td><td>Use 16x external clock</td></tr>
<tr><td>0001</td><td>50</td></tr>
<tr><td>0010</td><td>75</td></tr>
<tr><td>0011</td><td>109.92</td></tr>
<tr><td>0100</td><td>134.58</td></tr>
<tr><td>0101</td><td>150</td></tr>
<tr><td>0110</td><td>300</td></tr>
<tr><td>0111</td><td>600</td></tr>
<tr><td>1000</td><td>1200</td></tr>
<tr><td>1001</td><td>1800</td></tr>
<tr><td>1010</td><td>2400</td></tr>
<tr><td>1011</td><td>3600</td></tr>
<tr><td>1100</td><td>4800</td></tr>
<tr><td>1101</td><td>7200</td></tr>
<tr><td>1110</td><td>9600</td></tr>
<tr><td>1111</td><td>19200</td></tr>
</table></td><td>This table is expanded using an extra control bit #27.</td></tr>
<tr><td>4</td><td>Rx clock source</td><td>1 = external, 0 = baud rate generator</td></tr>
<tr><td>5,6</td><td>Word length<table>
<tr><td>00</td><td>8</td></tr>
<tr><td>01</td><td>7</td></tr>
<tr><td>10</td><td>6</td></tr>
<tr><td>11</td><td>5</td></tr>
</table></td><td>code for word length in bits</td></tr>
<tr><td>7</td><td>Stop Bit<table>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1 if 8 bits and parity</td></tr>
<tr><td>1</td><td>1.5 if 5 bits and no parity</td></tr>
<tr><td>1</td><td>2 otherwise</td></tr>
</table></td><td></td></tr>
</table>

UART_CTRL2 (reg 15)

Controls the fifos

0	Fifo enable	1 = fifo's enabled
1	Rx Fifo Clear	1 = clear receiver fifo
2	Tx Fifo Clear	1 = clear transmit fifo
3	reserved	
4,5	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.
6, 7	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.

UART_CTRL3 (reg 16)

0	hwfc	1 = automatic hardware flow control
1	reserved	
2	dmaEnable	1 = dma enabled
3	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.
4,5	reserved	
6	selDV	1 = use clock divider register, 0 = use baud table
7	reserved	

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

UART_CLKD0 (reg 4)

Bits 0 to 7 of the divider value.

UART_CLKD1 (reg 5)

Bits 8 to 15 of the divider value.

UART_CLKD2 (reg 6)

Bits 16 to 23 of the divider value.

OTHERS (reg 8, 10, 14)

Reserved

Bus Interface

The synchronous bus interface transfers data according to the bus clock. Write data is latched on the negative edge of the bus clock. The address, control, and data outputs are latched on the positive edge of the bus clock. There is no bus acknowledge; transfers are assumed to complete within one clock cycle.

Ports

Signal	I/O	Wid	Purpose
rst_i	I	1	reset
refclk_i	I	1	Reference clock (50 MHz)
ph2_i	I	1	bus clock input
cs_i	I	1	circuit/core select
irq_o	O	1	interrupt request
	SYNCHRONOUS BUS SIGNALS		
rw_i	I	1	Read (1) / write (0) enable
adr_i	I	5	address bits 0 to 4 (selects register)
dat_i	I	7	data input bus
dat_o	O	7	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