

# Chapter 1

## IP - UART

### 1.1 Universal Asynchronous Receiver Transmitter - UART

History		
Target Specification; Current version: Rev. 1.0, 2010-03-31; Author: Andreas Siggelkow; Previous version: Rev. 0.9, 2009-10-19		
Paragraph in previous version	Paragraph in current version	Subjects
Current Chapter Version: C0.1, 2009-10-21		
5.2.1	5.2.1	History generated

#### 1.1.1 System Integration

- Supply domain: **VDD\_MAIN**
- Chip internal interfaces:
  - Clock domain: **clk\_i**
  - Bus domain: Wishbone
  - Interrupt sources: **tbd**
- DMA sources : none
- Other interfaces : none
- Chip external signals related to this book:

- tx\_o
- rx\_i
- Monitor Pins:
  - test\_o - baudrate clock

### 1.1.2 Features

This is a standard UART according the standard. The fixed baudrate is 9600bit/s. The transmission is limited to 8bit, one start bit, no parity and one stop bit - 8N1.

	Generic UART Feature Set	Details
yes	Separate kernel clock	100MHz
no	FIFO Data Buffering	<ul style="list-style-type: none"> <li>• TX-FIFO: 32 Bits 12 Stages</li> <li>• RX-FIFO: 32 Bits 12 Stages</li> </ul>
yes	Full Duplex Capability	UART Controller functioning: full-duplex
no	DMA support	-
no	FIFO Alignment	-
no	FIFO Flow Controller Modes	-

Table 1.1: UART Feature Set

### 1.1.3 Connections

Domain	Block Name	Chip Level Name	Remark
Supply		VDD_MAIN	-
Clock	clk_i	clk_i	100MHz system clock used as the kernel clock.
Bus	Wishbone	Wishbone	
Interrupt			
DMA			
Chip / Peripheral IO	tx_o rx_i	tx_o rx_i	UART (serial) transmit UART (serial) receive

## 1.1. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER - UARTS

Chip internal	ascii_o	ascii_rx_s	The received 8bit data. Register value; located in the BPI; synchronised to kernel/BPI clock domain.
	ascii_i	ascii_tx_s	The 8bit data to be transmitted. Register value; located in the BPI; synchronised to kernel/BPI clock domain.
	rx_ready_o	rx_ready_s	Status: The 8bit have been received. High, if no transmission is active, low during transmission. Register value; located in the BPI; synchronised to kernel/BPI clock domain.
	tx_ready_o	tx_ready_s	Status: The 8bit have been transmitted. High, if no transmission is active, low during transmission. Register value; located in the BPI; synchronised to kernel/BPI clock domain.
	tx_start_i	tx_start_s	Configuration: The transmission starts. Should be only one base clock (clk_i) period high. This could be achieved by an edge detection (synchronisation) circuit. Register value; located in the BPI; synchronised to kernel/BPI clock domain.
	rst_n_i	rst_n_s	The kernel reset, active low.
	(bpi_clk_i)	(bpi_clk_s)	The bus clock. For BPI-block only.
	kernel_clk_i	kernel_clk_s	The kernel clock.
	test_o	test_s	This test signals for the outside world can be chosen freely.

### 1.1.4 Registers of UART

Register Name in Chip	Register name in Module	Remarks
UART01_ID	REG_ID	located in BPI
UART01_CLC	REG_CLC	located in BPI
UART01_BPI_CFG	REG_BPI_CFG	located in BPI
UART01_KERN_CFG	REG_KERNEL_CFG	SFR of UART kernel; located in BPI
UART01_KERN_RX	REG_KERNEL_RX	SFR of UART kernel; located in BPI
UART01_KERN_TX	REG_KERNEL_TX	SFR of UART kernel; located in BPI

### 1.1.5 Functional Overview

The UART (RS232) is a pure receive/transmit serial interface. The fixed baudrate is 9600, the protocol is 8N1 (eight data bits, no parity, one stop-bit).

#### Math

The baudrate must be generated from a base clock. This could be done by means of a clock divider. The division factors are:

Base Clock	Baud Rate	Division Factor
50MHz	9600	5208
100 MHz	9600	10417

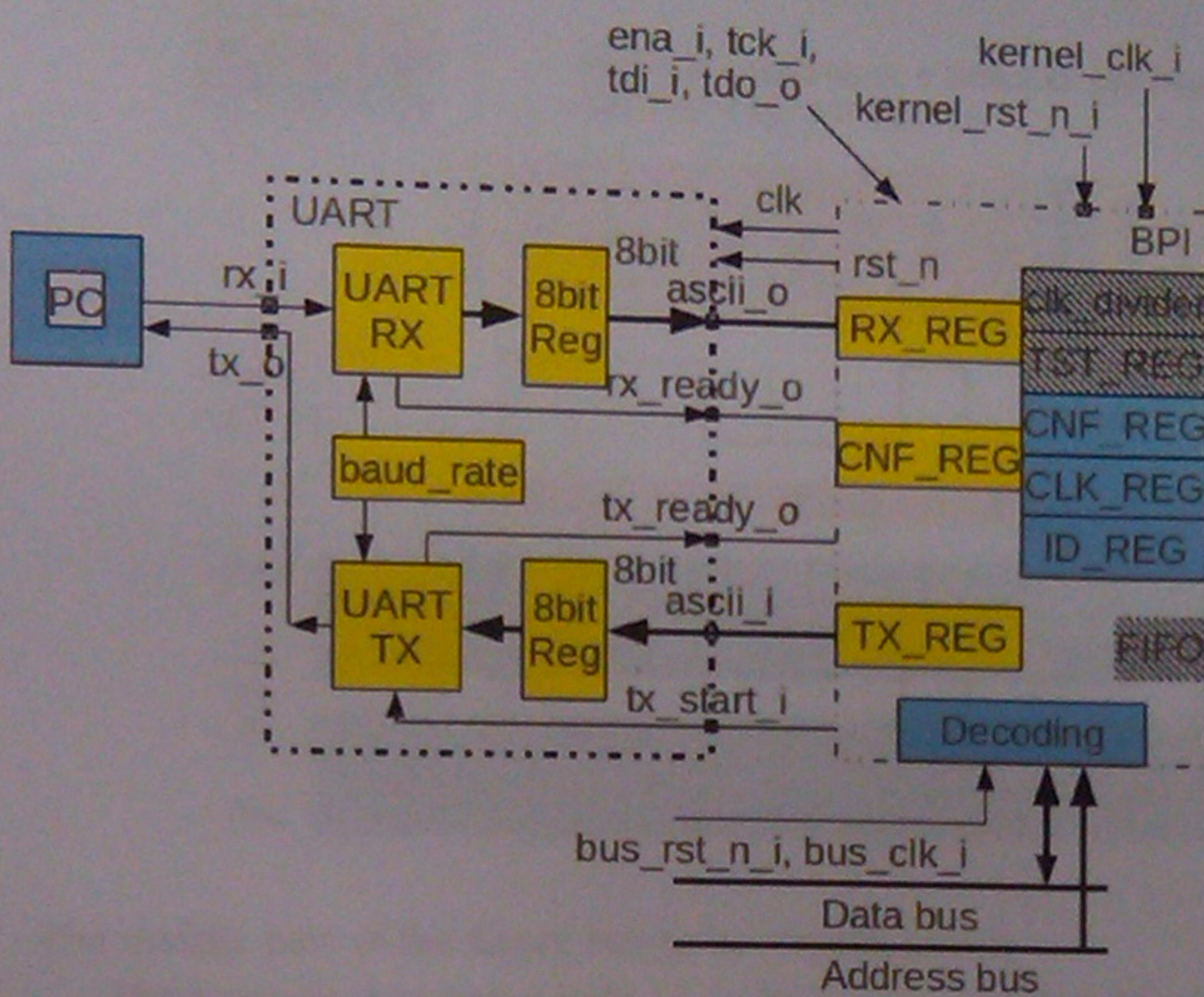
The formula is

$$\text{Division Factor} = \frac{\text{Base Clock}}{\text{Baudrate}}$$

### 1.1.6 Functional Description

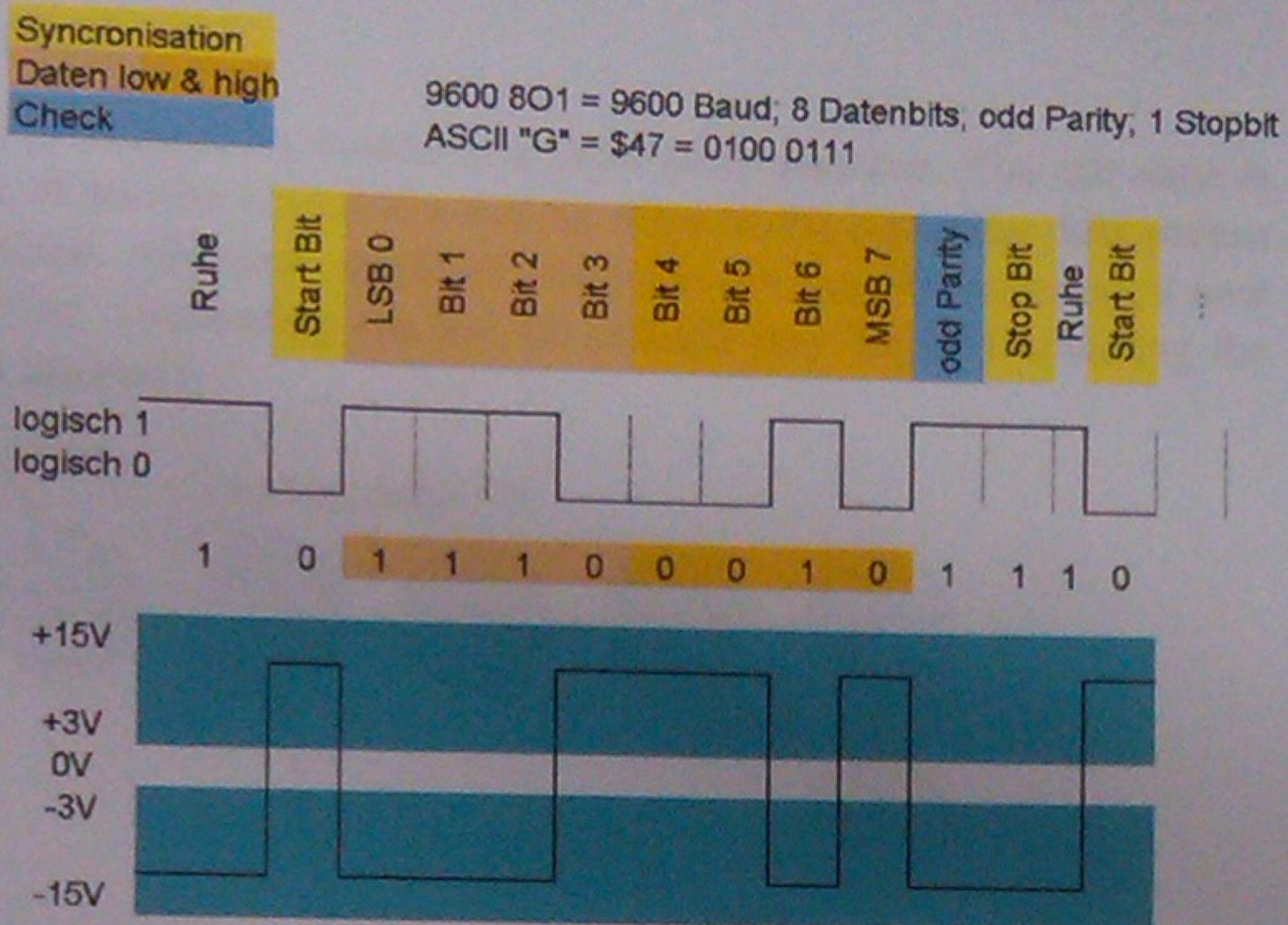
The base of the UART is a serial parallel conversion for the RX path and a parallel serial conversion for the TX path. Both sides interfaces a 8bit register chip internally. The RX path needs a ready signal to make visible, that the data has been received. The TX path needs a ready signal to make visible, that the data transmission is ready (next data could be transmitted).

## 1.1. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER - UART7



The UART kernel will be connected to a BPI (bus peripheral interface), which also contains the registers of the UART kernel. The data signals between the BPI and the UART must be synchronised to the respective clock domain (tbd).

The transmission signals will be adapted to TTL logic by a MAX232 chip. This chip inverts the polarity, so for the **HS-Weingarten Phase ASIC** point of view, the signal levels represent positive logic. This could be seen in the following figure (Wikipedia):



The middle part of the figure has to be implemented.

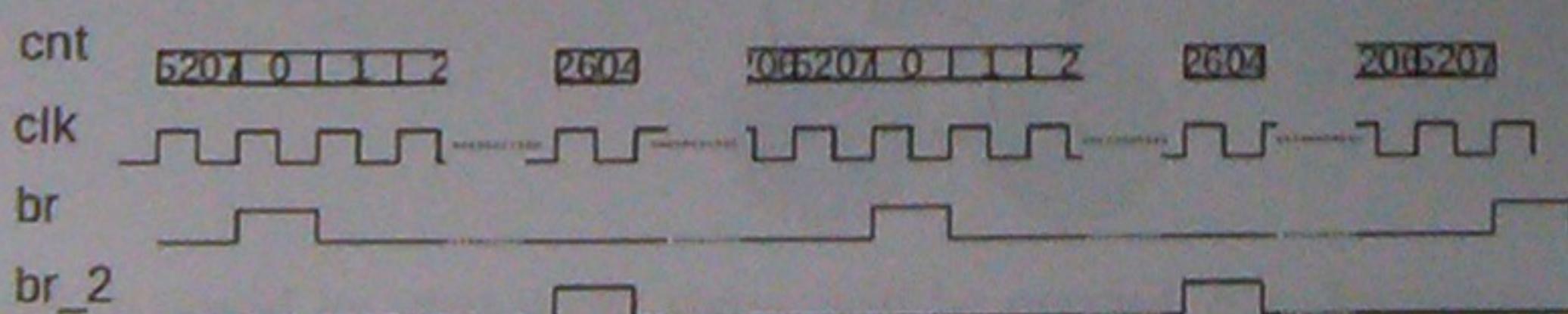
The transmission speed is fixed to 9600bit/s. The data block format is 8N1.

### Function of the “Start Transmission” Signal

The “starttx” signal should be only one base clock cycle high, else low. After releasing it, the transmission starts.

### Function of the Baud Rate Signals

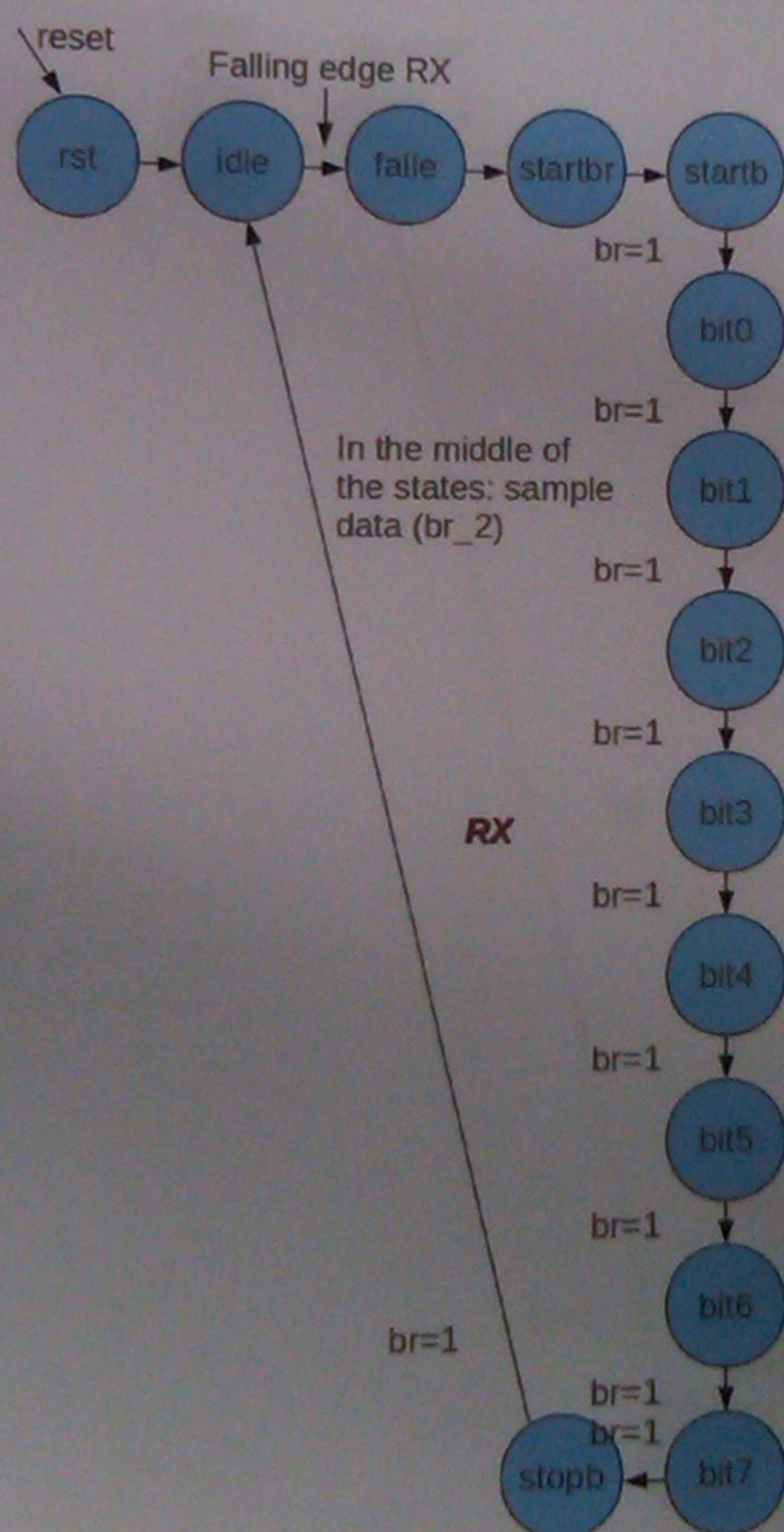
Two down-scaled signals (from base clock (e.g. 100MHz) down to 9600Hz) will be generated: one for the indication of the start of the bit interval and one for the indication of the middle of a bit interval. The down-scaled clock will have not a duty cycle of 50%, the high-level must have a duration of exactly one base clock period, the low-level a duration for the rest of the cycle time. This clock will not be used to clock a register, it will be used to enable a logic which is clocked with the base clock



## 1.1. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER - UART9

### FSM of the RX-Block

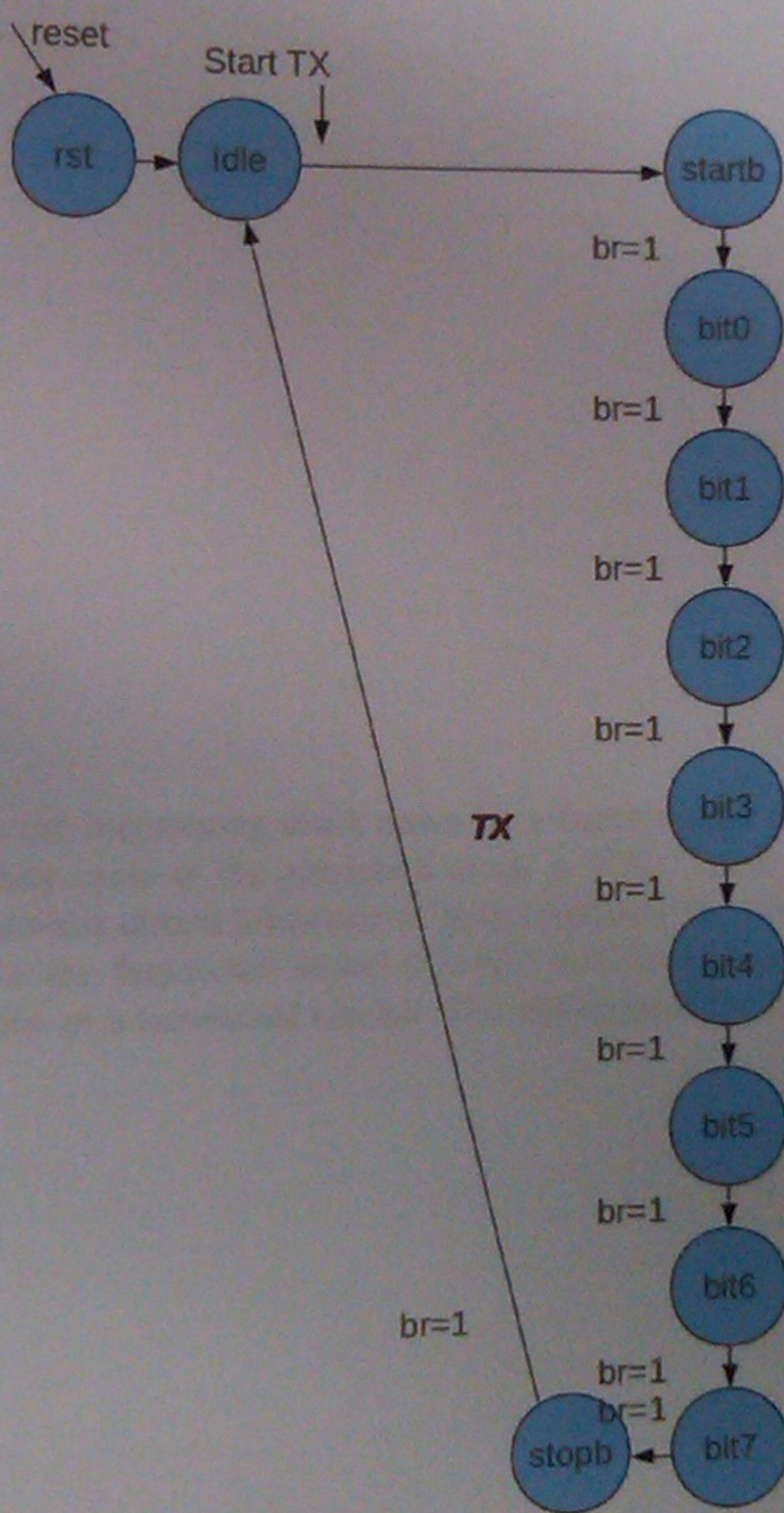
The RX function is mapped by means of a finite state machine. The idle state is the inactive state, it will be leaved if a falling edge of the incoming data stream (RX) will be detected. After this, the baud rate generator must be restarted, it now generates pulses "br" (indicating the full bit period) and "br\_2" (indicating the middle of one bit interval).



For a working FSM, it could be, that more states will be needed, e.g. for signal synchronisation.

### FSM of the TX-Block

The TX function is mapped by means of a finite state machine. The idle state is the inactive state, it will be leaved if a "start transmission" signal will be given. After this, the data to be send will be written to the serial line (TX) with every "br" pulse.



For a working FSM, it could be, that more states will be needed, e.g. for signal synchronisation.

## 1.1. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER - UART11

### Clock Divider (BPI)

The clock divider scales the incoming clock down by a factor defined by an incoming signal. The duty cycle of the generated clock is 50%. There are two incoming clocks, the divider circuit takes one of both (normally the two clocks are a dedicated clock for the functional kernel (UART) and the BPI bus clock which could also be taken as a functional clock). The multiplexor for the clock must be glitch-free.

The clock can be disabled, glich-free. If the clock is disabled, the output clock signal will be high.

### 1.1.7 Signal Description

Signal	Type	Description
clk_i	in	functional clock, e.g. 100MHz
reset_n_i	in	low active reset
rx_i	in	incomming serial message; if inactive, it should be '1'.
tx_o	out	serial message to be transmitted; if inactive, it should be '1'
ascii_o(7:0)	out	parallel message received; should be connected to the receive register (BPI-SFR)
ascii_i(7:0)	in	parallel message to be transmitted; should be connected to the receive register (BPI-SFR)
rx_ready_o	out	receive is ready; This signal indicates, that a serial data stream has been received. It is low, if the receive operation is running and it is high, if it is not running/finished. The rising edge could be taken to activate the storage into the register.
tx_ready_o	out	transmit is ready; This signal indicates, that a serial data stream has been transmitted. It is low, if the transmit operation is running and it is high, if it is not running/finished. The rising edge could be taken to activate the storage into the register.
starttx_i	in	starts the transmission; This signal duration (high time) should be shorter than the duration of the transmission, otherwise the data will be send again.

### 1.1.8 Register Description

- w: writable bit, by SW

## 1.1. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER - UART13

- r: readable bit, by SW
  - h: bit will be set by HW only

## Clock Control and ID Registers

**REG\_ID:** Identification register of this peripheral. A write will have no effect, a read will return the fixed ID. The ID will be defined on chip level.

REG CLC:

### Clock control register (in BPI)

[Reset value: 00H]

7	6	5	4	3	2	1	0
'_'	<b>clk_sel</b>	'_'	'_'	<b>kernel_clk_disable</b>	'_'	'_'	<b>bus_clk_disable</b>
'_'	<b>rw</b>	'_'	'_'	<b>rw</b>	'_'	'_'	<b>rw</b>

Field	Bits	Type	Description
reserved	7:5,2:1	rw	not implemented, a write has no effect, a read returns a '0'
clk_sel	6	rw	Selects the clock source for the UART kernel: <ul style="list-style-type: none"><li>• '1': kernel_clk_i (kernel clock) drives the functional part</li><li>• '0': bus_clk_i (bus clock) drives the functional part</li></ul>
kernel_clk_disable	3	rw	disables the kernel clock for the block.
bus_clk_disable	0	rw	disables the bus clock for the block.

## **REG BPI\_CFG:**

Configuration control register (in BPI)

[Reset value:  $00_H$ ]

## CHAPTER 1. IP - UART

Field	Bits	Type	Description
reserved	7:5	rw	not implemented, a write has no effect, a read returns a '0'
level	4	rw	active high or active low input and output. <b>TBD</b>
br_sel(3:0)	3:0	rw	Defines the baud rate. <b>TBD</b>

## Other Registers

**REG\_KERNEL\_CFG:** Control and status register of the UART, located as a SFR in the BPI.

Bit 0 is starttx,  
bit 1 is rx\_ready and  
bit 2 is tx\_ready

**REG\_KERNEL\_RX:** Receive register of the UART, located as a SFR in the BPI.

Receive register RX-path

[Reset value: 00<sub>H</sub>]

7	6	5	4	3	2	1	0
ascii(7:0)							
rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
ascii	7:0	h	Received data.

**REG\_KERNEL\_TX:** Transmit register of the UART, located as a SFR in the BPI.

Transmit register for TX-path

[Reset value: 30<sub>H</sub>]

7	6	5	4	3	2	1	0
ascii(7:0)							
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ascii	7:0	h	The data to be transmitted.

## 1.1. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER - UART15

**TEST\_REG:** The test register is accessible by the TAP controller and by the BPI (Bus Peripheral Interface). NOT IMPLEMENTED YET!!!

Test configuration register

[Reset value:  $00_H$ ]

7	6	5	4	3	2	1	0			
'-	'-	'-	'-	TST_MODE						
-	-	-	-	rw	rw	rw	rw			

Field	Bits	Type	Description
TST_MODE	3:0	rw	<p>Various test modi:</p> <ul style="list-style-type: none"> <li>• TEST_MODE <math>0h</math>: No test</li> <li>• TEST_MODE <math>1h</math>: [RX] The receive direction will be tested. The received value will be seen on the eight DIGILAND board LEDs.</li> <li>• TEST_MODE <math>2h</math>: [TX] The transmit direction will be tested. The four DIGILAND board switches will be used to implement the data to be transmitted (one switch for two bits). The start of the transmission will be given by an additional DIGILAND board switch.</li> <li>• TEST_MODE <math>3h</math>: [RX and TX] The received data will be send back unchanged.</li> <li>• TEST_MODE <math>4h</math>: [RX and TX] The received data will be send back changed somehow.</li> </ul>