APB-UART

Hardware design specification

Contents

List of tables

List of figures

1. System description

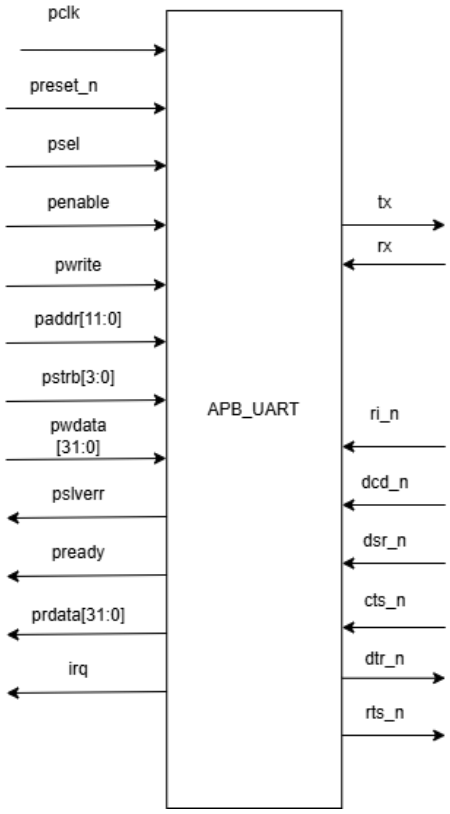

Figure 1.Interface of system

1.1Signals description

Table 1. apb- uart parameter description

Parameter	Value	Description
SYSTEM_FREQUENCY	100000000	System clock frequency

Table 2.apb_uart signals description

Singal name	Width	I/O	Description
Global signals			
clk	1	input	System clock signal
reset_n	1	input	System asynchronous reset, active LOW

APB signals			
pclk	1	Input	APB clock signal
preset_n	1	Input	APB reset signal, active LOW, connected directly to the system reset
psel	1	Input	Select, indicates that the APB completer is selected and that a data transfer is required
penalbe	1	input	Enable, indicates the second and subsequent cycles of an APB transfer
pwrite	1	Input	Direction, indicates an APB write access when HIGH and an APB read access when LOW
paddr	12	Input	Write strobe, indicates which byte lanes to update during write transfer
pstrb	1	Input	APB address bus
pwdata	32	Input	APB write data
pready	1	output	Ready, is used to extend an APB transfer by the completer
pslverr	1	output	Transfer error
prdata	32	output	APB read data
UART signals			
tx	1	output	Serial data receive
rx	1	output	Clear-to-send handshaking signal
cts_n	1	Input	Serial data transmit
rts_n	1	output	Request-to-send handshaking signal

1.2.Register block

Table 3.apb_uart register block

Register name	acronnym	Register description	Address	Acesss	Field name	Field reset value	position	Description
Tx_data_reg	TDR	TX data register	0x0	RW	tx_data	0	[7:0]	Parallel data from the host which is converted to serial data sent to peripheral device
					rfu	0	31:8	
rx_data_reg	RDR	RX data register	0x4	RO	rx_data	0	[7:0]	Parallel data archived after serial-to-parallel conversion on

								data received on peripheral device
					rfu	0	[31:8]	
line_cfg_reg	LCR	UART configuration register	0x8	RW	data_bit_num	0	[1:0]	2'b00: 5 bits 2'b01: 6 bits 2'b10: 7 bits 2'b11: 8 bits
					stop_bit_num	0	[2]	2'b0: 1 bit 2'b1: 2 bits
					parity_en	0	[3]	1'b1: enable 1'b0: disable
					parity_type	0	[4]	1'b1: Even 1'b0: Odd
					baudrate	0	[7:5]	3'b000: 4800 3'b001: 9600 3'b010: 14400 3'b011: 19200 3'b100: 38400 3'b101: 57600 3'b110: 115200 3'b111: 230400
					rfu	0	[31:8]	
o_ctrl_reg	OCR	Operation control register	0xC	RW	tx_en	0	[0]	1'b1: Transmitter is enabled 1'b0: Transmitter is disabled
					start_tx	0	[1]	1'b1 start transmit
					rx_en	0	[2]	1'b0: Receiver is disabled 1'b1: Receiver is enabled and begin searching for a start bit
					rfu	0	[31:3]	
line_stt_reg	LSR	Line status register	0x10	RO	tx_done	1	[0]	This bit is cleared as soon as the LSR is read
					rx_data_ready	0	[1]	0 - All the data in RBR or FIFO is read 1 - Complete incoming character has been received and transferred into the RBR of FIFO
					parity_error	0	[2]	This bit is cleared as soon as the LSR is read

					frame_error	0	[3]	It indicates that the received character did not have a valid stop bit. This bit is cleared as soon as the LSR is read
					TDRE	0	[4]	Transmitter Data Register Empty 0 - TDR or Transmitter FIFO has data to transmit 1 - TDR is empty, in FIFO mode, FIFO is empty
					FIFO DATA ERROR	0	[5]	If the FIFO is active, this bit will set as soon as any data character in the Receiver's FIFO has parity or framing error. The bit is cleared when the microprocessor read the LSR
					overrun error	0	[6]	This bit will be set when the next character is transferred into RDR before the RDR data is read by the CPU. The bit is cleared when the microprocessor read the LSR
					rfu	0	[31:7]	
fifo_ctr_reg	FCR	FIFO control register	0x14	RW	FIFO_en	0	[0]	When set '1' this bit enables both the transmitter and receiver FIFOs, changing it bit automatically resets both FIFOs.
					RxFIFO_reset	0	[1]	Setting '1' to this bit resets t Rx FIFO

					TxFIFO_reset	0	[2]	Setting '1' to this bit resets t Tx FIFO
					FIFO_trigger_level	0	[4:3]	00: 1 01: 4 10: 8 11: 14
					rfu	0	[31:5]	
intr_en_reg	IER	interrupt enable register	0x18	RW	data_ready_intr	0	[0]	0 - Disables received data interrupt 1 - Enables received data interrupt
					TDR empty intr	0	[1]	This bit enables the TDR Empty interrupt
					receiver_line_stt_intr	0	[2]	This bit enables the Receiver Line Status interrupt
					rfu	0	[31:3]	
intr_iden_reg	IIR	Interrupt identification register	0x1C	RO	intr_pending	0	[0]	0 - Interrupt is pending 1 - No Interrupt is pending
					intr_id	0	[2:1]	2'b11: LSR error flags (OE/FE...) 2'b10: LSR receiver data ready flag - RDR 2'b01: TDR empty intr
					rfu	0	[31:3]	
Hardwareflow_control_reg	HCR	Hardware flow control register	0x20	RW	hf_en		[0]	0: Disables hf 1: Enables hf
					force rts_n	0	[1]	0: 0 1: force rts_n to 0
					rfu	0	[31:2]	

2. IP Block Description

2.1 UART transmitter

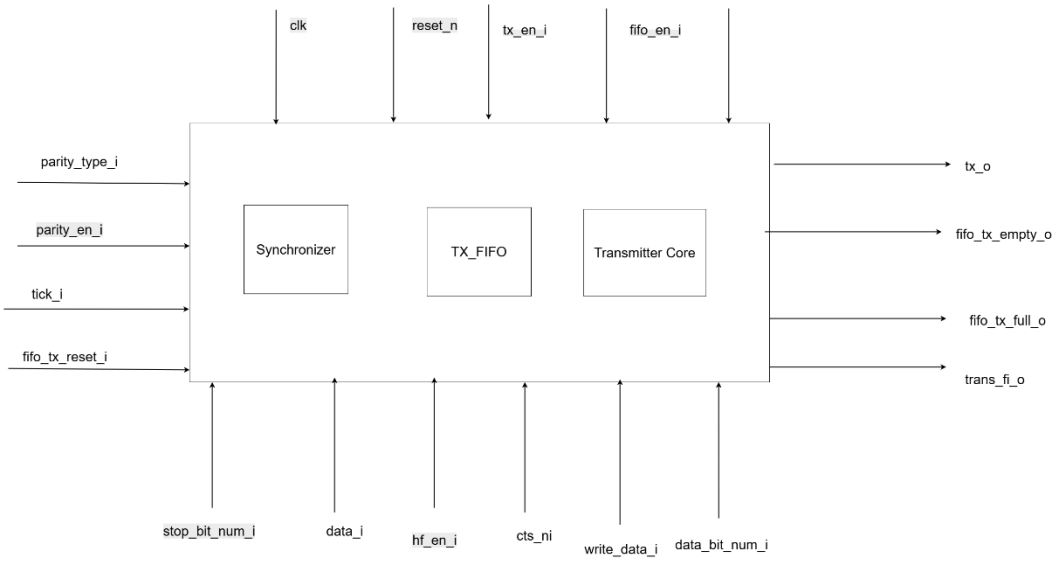

Figure 2.Uart transmitter block

2.1.1 Signals description

Table 4.Uart transmitter top signals description

Signals name	Width	I/O	Description
Clk	1	input	Clock system signal
reset_n	1	input	System asynchronous reset, active LOW
tx_en_i	1	input	Enable the transmitter
fifo_en_i	1	input	Enable the FIFO buffer
parity_type_i	1	input	Choose the type of parity
parity_en_i	1	input	Enable the parity bit
tick_i	1	input	Tick signal from Baudgenerator
fifo_tx_reset_i	1	input	Reset transmitter FIFO signal
stop_bit_num_i	1	input	Number of stop bit
start_tx_i	1	input	Starts UART transmission
data_i	8	input	Parallel data input
hf_en_i	1	input	Handshaking by requet to send and clear to send enable
cts_ni	1	input	Clear to send
write_data_i	1	input	Enable the pushing “data_i” to FIFO buffer
data_bit_num_i	2	input	Number of data bit in frame
tx_o	1	output	Transmitter data out
fifo_tx_empty_o	1	output	Notify that the FIFO buffer are empty
fifo_tx_full_o	1	output	Notify that the FIFO buffer are full
trans_fi_o	1	output	Notify Transmission finished

2.1.2 FSM for UART transmitter core

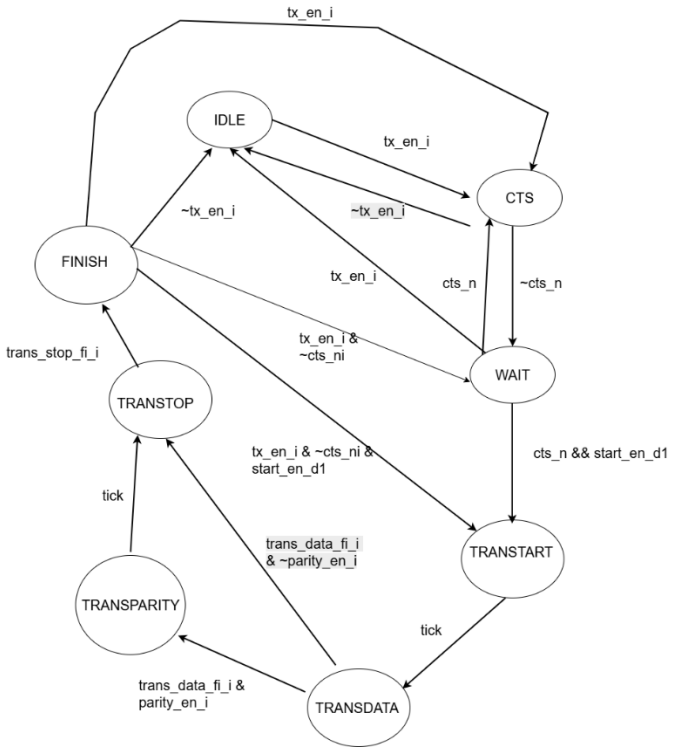

Figure 3. FSM of UART transmitter core

2.1.3: Transmitter FIFO

Table 5. Parameter of transmitter fifo

Parameter	Value	Description
DEPTH	16	Depth of receiver fifo

Table 6. Transmitter FIFO signals description

Signals name	Width	I/O	Description
clk	1	input	Clock system signal
reset_n	1	input	System asynchronous reset, active LOW
fifo_tx_i	8	input	Data input to FIFO
fifo_tx_reset_i	1	input	Reset FIFO buffer
fifo_rx_pop_i	1	input	Pop data from FIFO
fifo_rx_empty_o	1	output	Notify that the FIFO buffer are empty
fifo_tx_push_i	1	input	Push data into FIFO (write request)
fifo_tx_full_o	1	output	Notify that receiver FIFO was full
fifo_tx_o	8	output	Parallel data out from FIFO

2.2 UART receiver

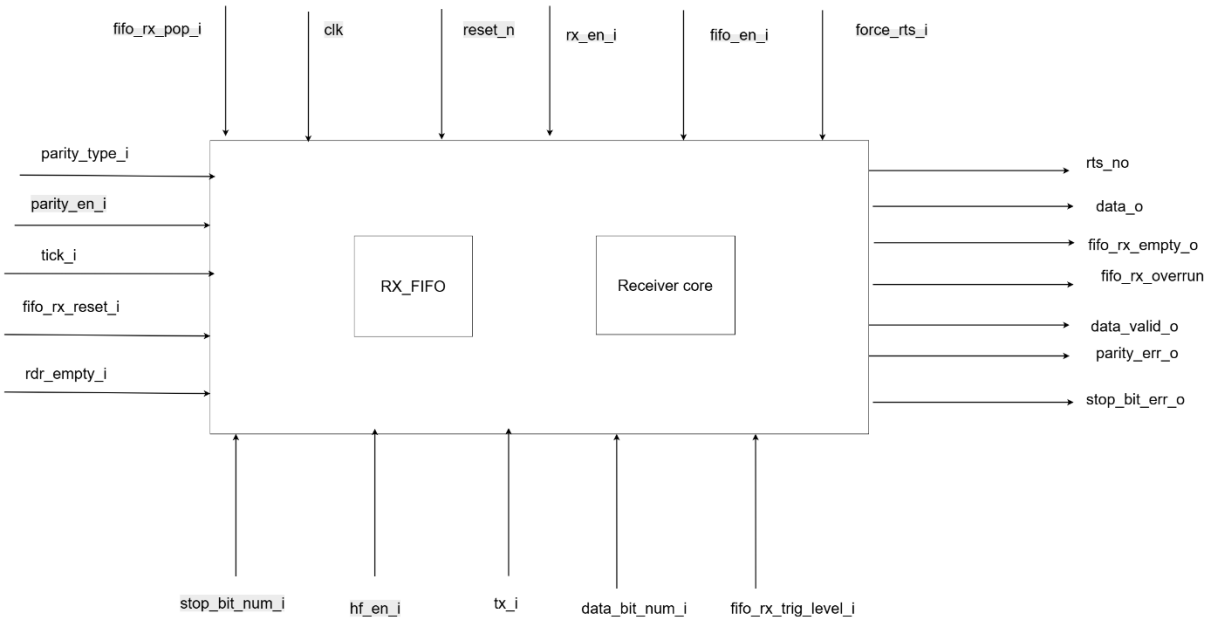

Figure 4. Uart recceiver block

2.2.1 Signals description

Table 7. Uart receiver top signals descirption

Signals name	Width	I/O	Description
clk	1	input	Clock system signal
reset_n	1	input	System asynchronous reset, active LOW
rx_en_i	1	input	Enable the receiver
fifo_en_i	1	input	Enable the FIFO buffer
parity_type_i	1	input	Choose the type of parity
parity_en_i	1	input	Enable the parity bit
tick_i	1	input	Tick signal from Baudrate generator (16x sampling for receiver)
fifo_rx_reset_i	1	input	Reset receiver FIFO signal
stop_bit_num_i	1	input	Number of stop bit
fifo_rx_trig_level_i	2	input	Configurable trigger level for RX FIFO (e.g., 1, 4, 8, 14 bytes)
hf_en_i	1	input	Handshaking by requet to send and clear to send enable
rts_no	1	output	Request to send, active LOW
force_rts_i	1	input	Force rts_no to 0, bypassing automatic control
fifo_rx_pop_i	1	input	Read data out from Receiver fifo
rdr_empty_i	1	input	Notify that the RDR register is empty
data_bit_num_i	2	input	Number of data bit in frame
tx_i	1	input	Serial Data input
fifo_rx_empty_o	1	output	Indicates that the RX FIFO is empty
fifo_rx_overrun	1	output	Notify that the FIFO buffer are overrun
data_o_valid	1	output	Confirm that the data is correct
parity_err_o	1	output	Notify that the parity bit of data frame not correct

stop_bit_err_o	1	output	Notify that the stop bit of data frame not correct
data_o	8	output	Received data output

2.2.2: FSM of UART receiver core

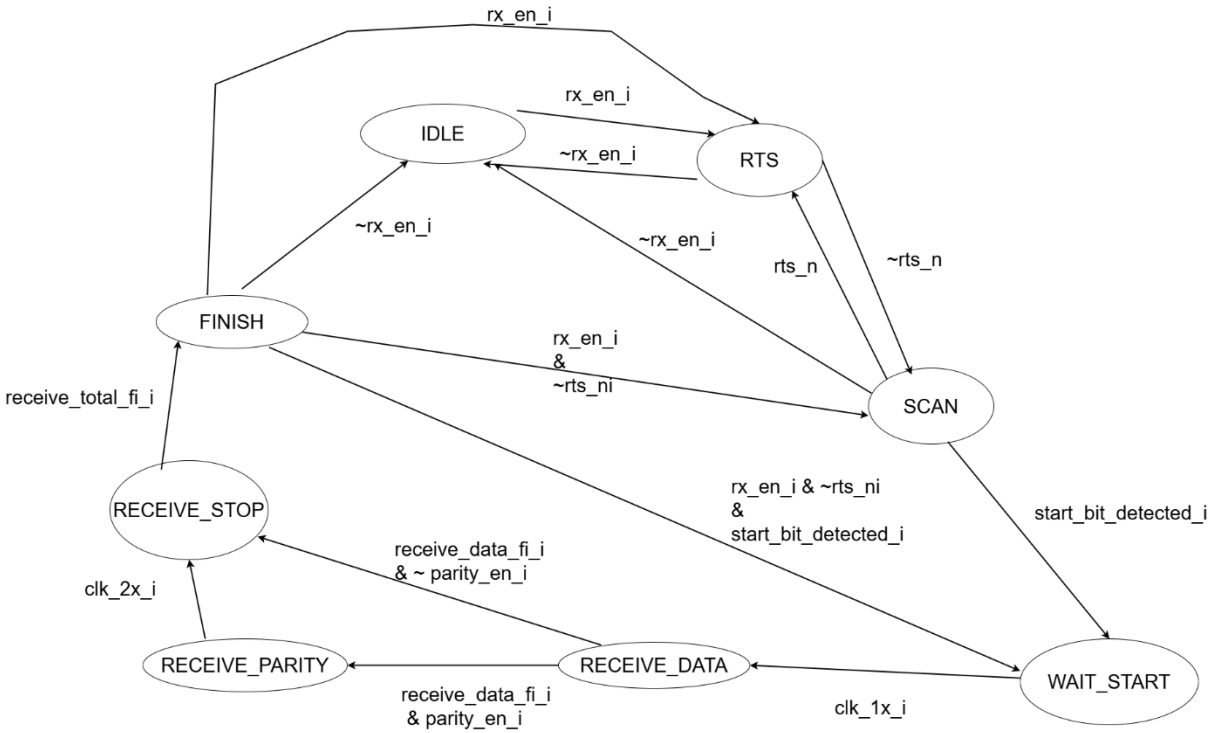

Figure 5. FSM of Uart recceiver core

2.2.3: Receiver FIFO

Table 8. Parameter of recceiver fifo

Parameter	Value	Description
DEPTH	16	Depth of receiver fifo
int TRIG_LEVELS[0:3]	'{1, 4, 8, 14}	These levels determine when the fifo_rx_triggered flag is asserted

Table 9. Uart recceiver fifo signals description

Signals name	Width	I/O	Description
clk	1	input	Clock system signal
reset_n	1	input	System asynchronous reset, active LOW
fifo_rx_i	8	input	Data input
fifo_rx_reset_i	1	input	Reset receiver FIFO signal
fifo_rx_trig_level_i	2	input	Trigger level of Receiver FIFO
fifo_rx_pop_i	1	input	Allow read data out from Receiver FIFO

fifo_rx_empty_o	1	output	Notify that the FIFO buffer are empty
fifo_rx_overrun	1	output	Notify that the FIFO buffer are full
fifo_rx_push_i	1	input	Push data to FIFO
fifo_rx_full_o	1	output	Notify that receiver FIFO was full
fifo_rx_o	8	output	Parallel data out from FIFO
fifo_rx_triggered_o	1	output	FIFO has reached or passed the trigger level

2.3 APB block

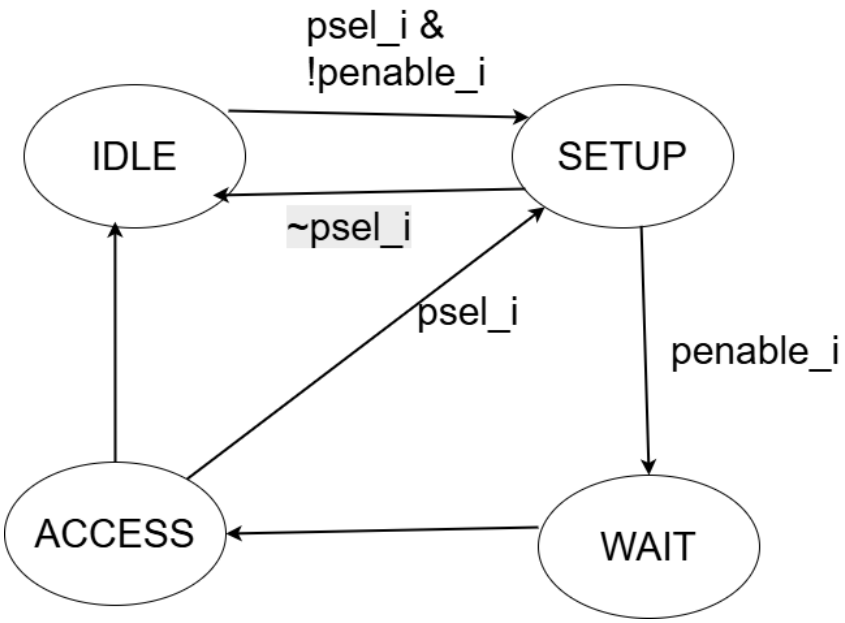

Figure 6. FSM of APB

3.Functional Description

APB_UART operates based on its register block, which provides information about the data to be processed, UART frame configuration, UART control signals, FIFO control signals, Interrupt controls and status signals.

The host also can use the FIFO buffer to store incoming and outgoing data temporarily, reducing the need for constant CPU intervention and enabling smoother data flow.

In addition, APB_UART provides an interrupt mechanism that enables efficient and responsive communication between the UART module and the host processor.

After reset, all control and status registers are set to their default values, including UART enable bits, FIFO enable, and interrupt masks. Both transmit and receive FIFOs are cleared, and the module remains idle until configured by the host. To enable, the host must:

- Configure the UART frame format (data bits, stop bits, parity) and baud rate via the LCR register
- Optionally enable FIFOs and Interrupt via the FCR and IER register
- Enable the transmitter and/or receiver through the OCR register

*Notes: In this system, the system clock signal and APB clock signal would use the same frequency.