COSE321 Computer Systems Design

Lecture 5. UART
- for serial communication

Prof. Taeweon Suh

Computer Science & Engineering

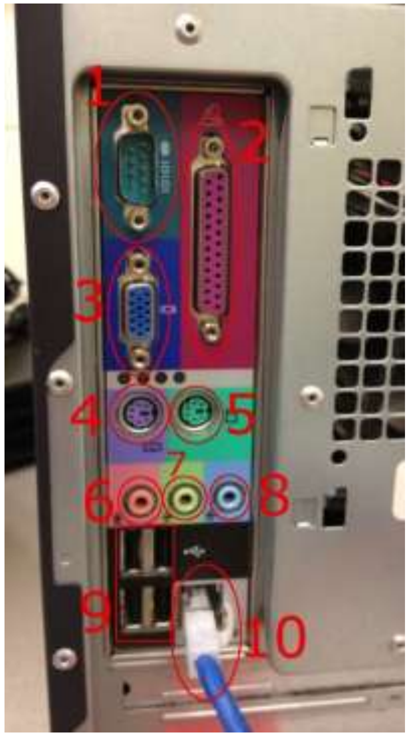
Korea University

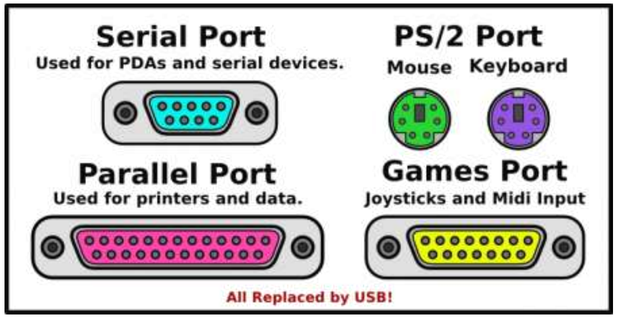
Handling I/O Devices in Software

- I/O devices typically provide 3 kinds of registers
  - Configuration (or control or mode) registers
  - Data registers
  - Status registers
- Software sets up I/O devices with configuration registers before normal operation
- Status registers indicate status of I/O devices

UART

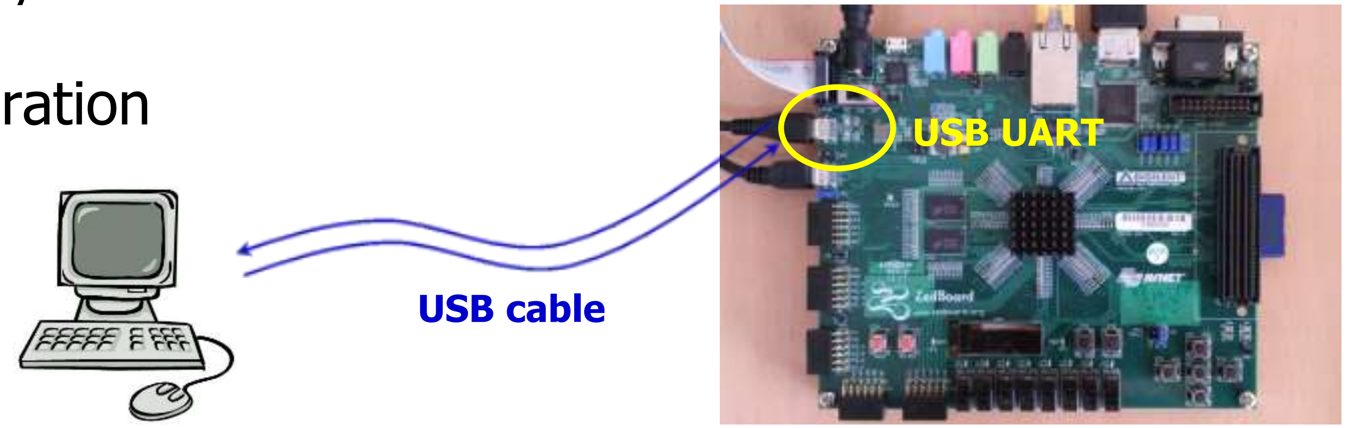
- Universal Asynchronous Receiver and Transmitter
  - Used for serial communication
  - Simply called serial port (or RS-232)
  - Has a long history (~1970)
  - Still widely used in embedded systems design for debugging purpose
  - Detected as a COM port in Windows
    - Its original shaped port has almost been disappeared in computers. Instead, the serial-to-USB is used whenever necessary





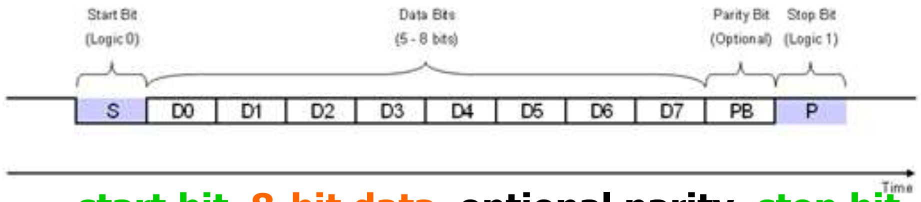
UART Controller in Zynq

- It is a full-duplex asynchronous receiver and transmitter
- Programmable baud rate (bit rate) generator
- 64-byte receive and transmit FIFOs
- Programmable protocol
  - 6, 7 or 8-bit data
  - #stop bits
  - Optional parity set
- Interrupt generation

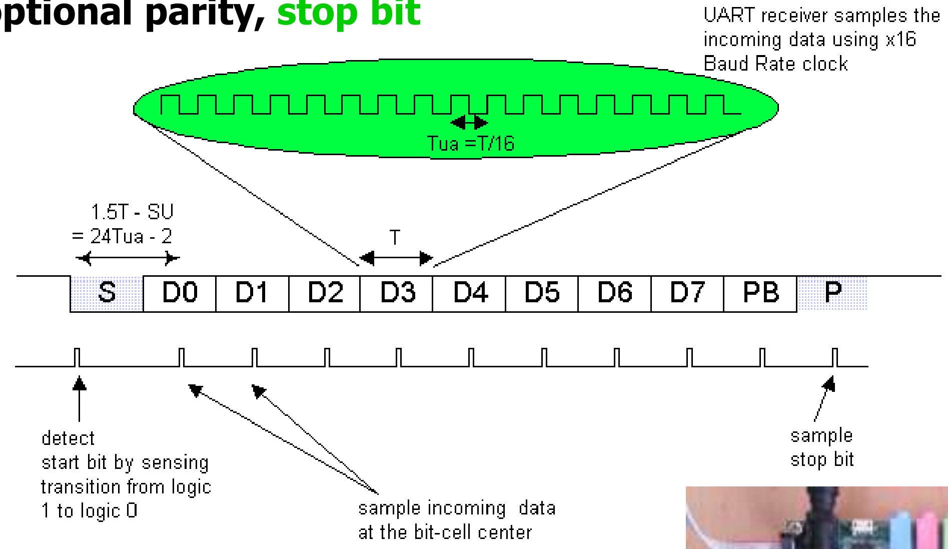


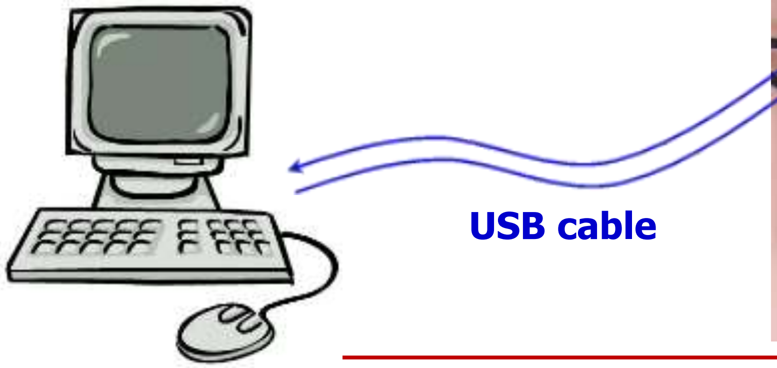
Korea Univ

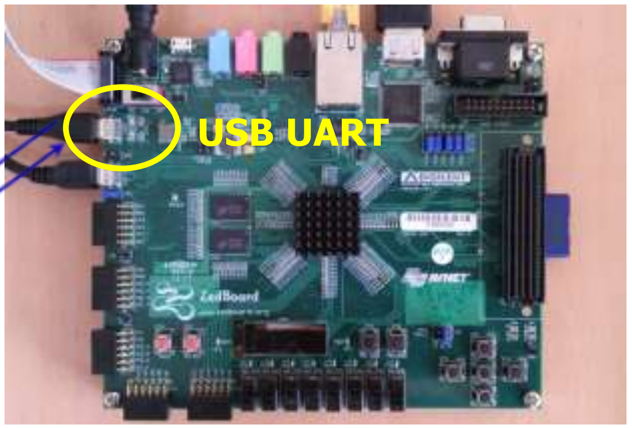
UART Packet format



start bit, 8-bit data, optional parity, stop bit


UART receiver samples the incoming data using x16 Baud Rate clock
Tua =T/16
1.5T - SU
= 24Tua - 2
T
S
D0
D1
D2
D3
D4
D5
D6
D7
PB
P
detect start bit by sensing transition from logic 1 to logic 0
sample incoming data at the bit-cell center
sample stop bit





http://pcsbyjohn.wordpress.com/
http://tutorial.cytron.com.my/2012/02/16/uart-universal-asynchronous-receiver-and-transmitter/

UART in a Zynq System

The system viewpoint diagram for the UART controllers is shown in Figure 19-1.

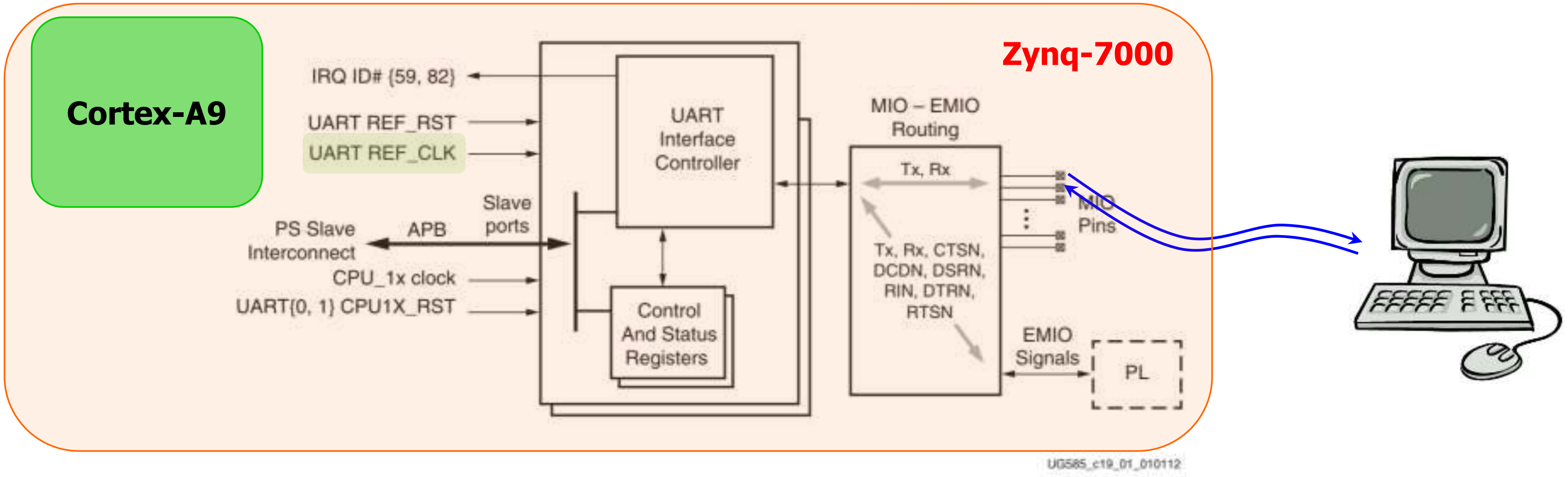


Figure 19-1: **UART System Viewpoint**

The slcr register set (refer to section 4.3 SLCR Registers) includes control bits for the UART clocks, resets and MIO-EMIO signal mapping. Software accesses the UART controller registers using the APB 32-bit slave interface attached to the PS AXI interconnect. The IRQ from each controller is connected to the PS interrupt controller and routed to the PL.

APB (Advanced Peripheral Bus), a part of AMBA standard, is used to connect peripherals (I/O devices) to the hardware system

UART in a Zynq System

The block diagram for the UART module is shown in Figure 19-2

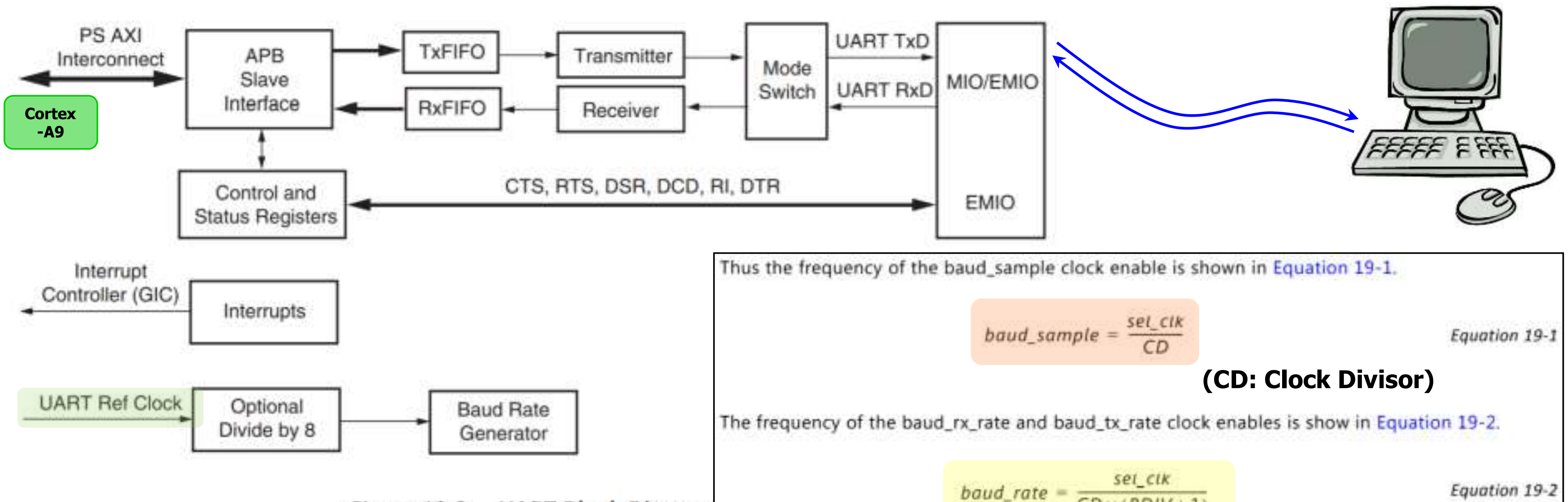


Figure 19-2: **UART Block Diagram**

Thus the frequency of the baud_sample clock enable is shown in Equation 19-1.

$$baud\_sample = \frac{sel\_clk}{CD} \quad \text{Equation 19-1}$$

(CD: Clock Divisor)

The frequency of the baud_rx_rate and baud_tx_rate clock enables is show in Equation 19-2.

$$baud\_rate = \frac{sel\_clk}{CD \times (BDIV+1)} \quad \text{Equation 19-2}$$

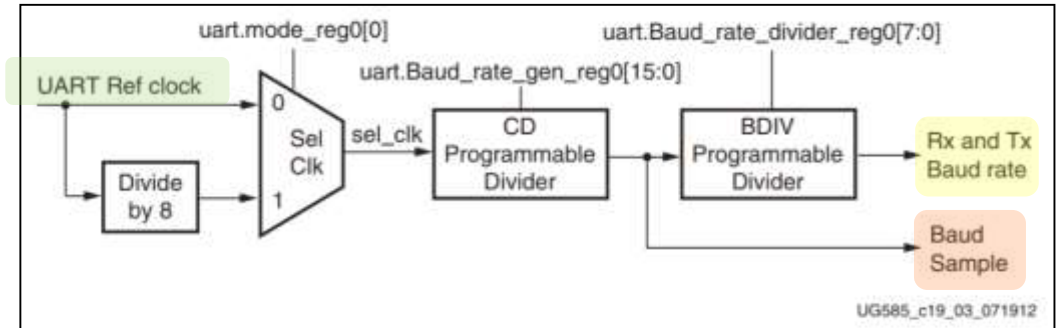


Figure 19-3: **UART Baud Rate Generator**

Transmitted Data & Received Data

Thus the frequency of the baud_sample clock enable is shown in Equation 19-1.

$$baud\_sample = \frac{sel\_clk}{CD}$$

The frequency of the baud_rx_rate and baud_tx_rate clock enables is show in Equati

$$baud\_rate = \frac{sel\_clk}{CD \times (BDIV + 1)}$$

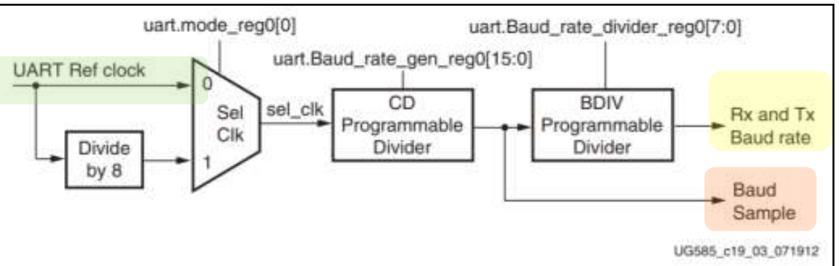


Figure 19-3: **UART Baud Rate Generator**

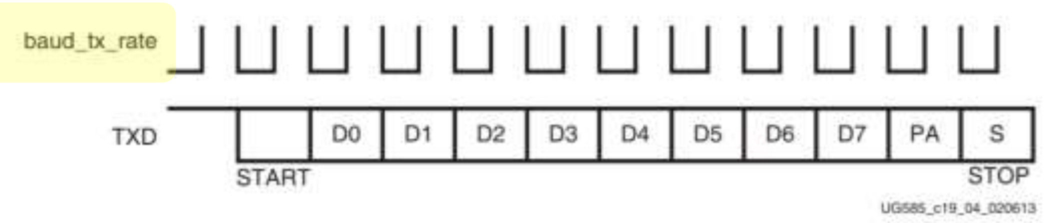


Figure 19-4: **Transmitted Data Stream**

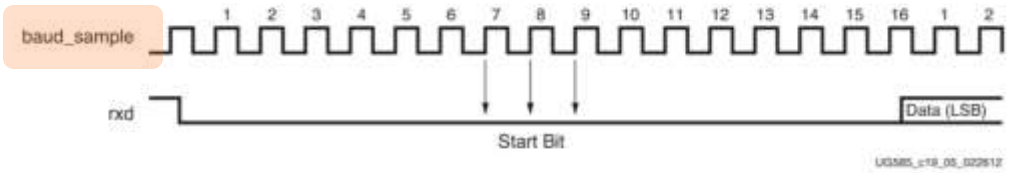


Figure 19-5: **Default BDIV Receiver Data Stream**

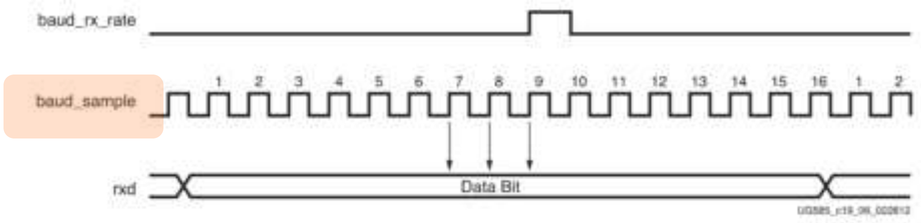
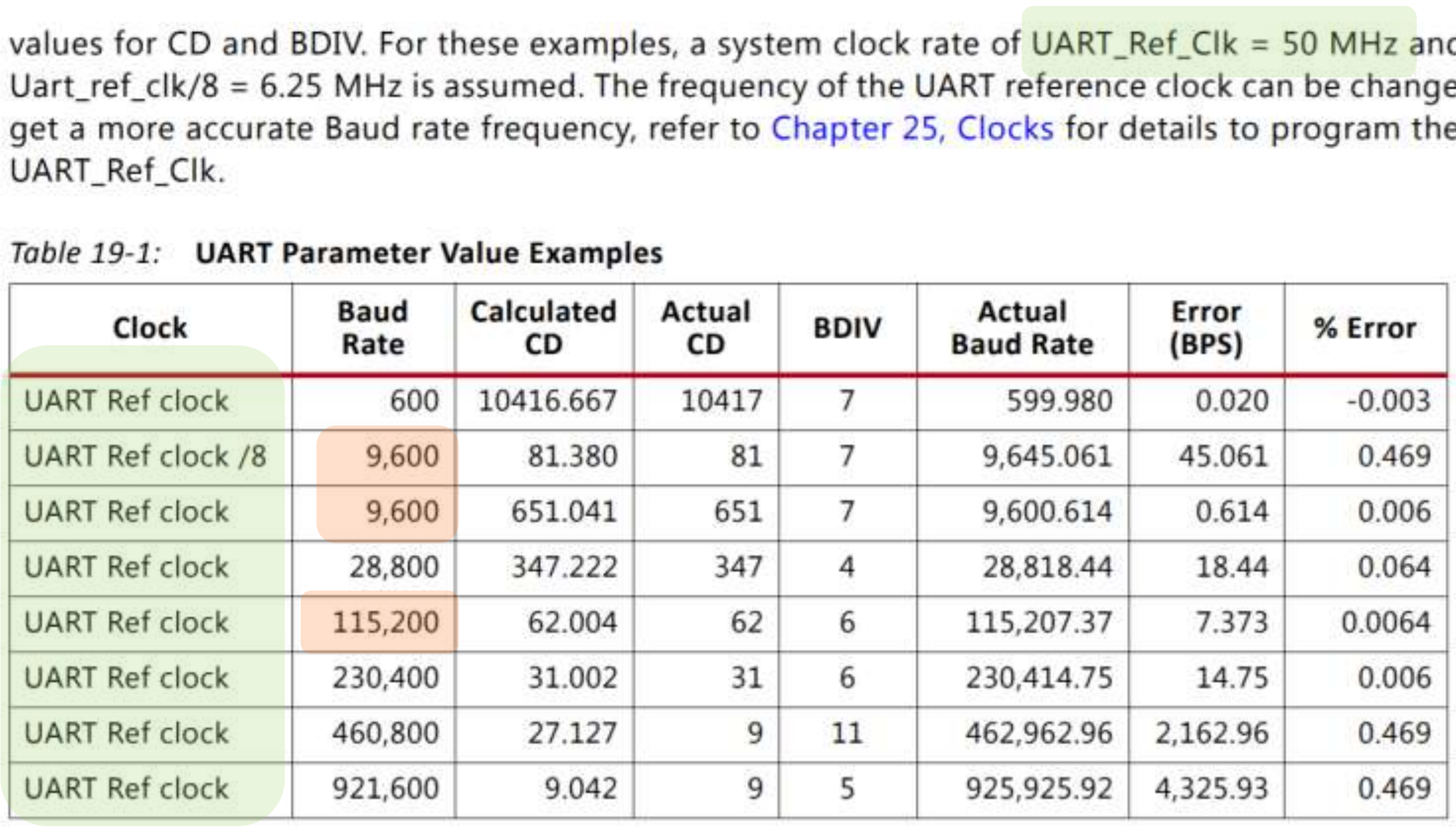


Figure 19-6: **Re-synchronized Receiver Data Stream**

Baud Rate Example



values for CD and BDIV. For these examples, a system clock rate of UART_Ref_Clk = 50 MHz and Uart_ref_clk/8 = 6.25 MHz is assumed. The frequency of the UART reference clock can be changed to get a more accurate Baud rate frequency, refer to Chapter 25, Clocks for details to program the UART_Ref_Clk.

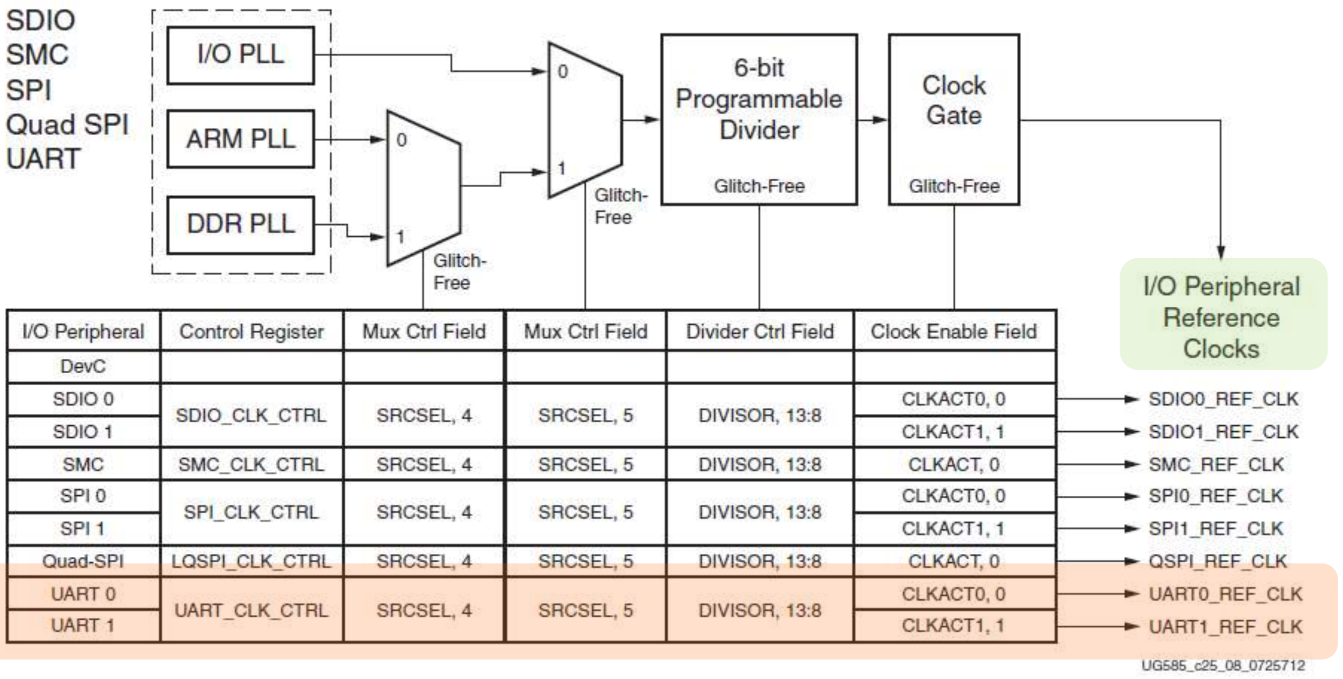
Table 19-1: **UART Parameter Value Examples**

Clock	Baud Rate	Calculated CD	Actual CD	BDIV	Actual Baud Rate	Error (BPS)	% Error
UART Ref clock	600	10416.667	10417	7	599.980	0.020	-0.003
UART Ref clock /8	9,600	81.380	81	7	9,645.061	45.061	0.469
UART Ref clock	9,600	651.041	651	7	9,600.614	0.614	0.006
UART Ref clock	28,800	347.222	347	4	28,818.44	18.44	0.064
UART Ref clock	115,200	62.004	62	6	115,207.37	7.373	0.0064
UART Ref clock	230,400	31.002	31	6	230,414.75	14.75	0.006
UART Ref clock	460,800	27.127	9	11	462,962.96	2,162.96	0.469
UART Ref clock	921,600	9.042	9	5	925,925.92	4,325.93	0.469

UART Clock Source in Zynq

25.6.3 SDIO, SMC, SPI, Quad-SPI and UART Clocks

The SDIO, SMC, Quad SPI, and UART peripheral clocks all have the same programming model (see Figure 25-8. The PLL source and divider values are shared for each I/O peripheral controller. The clocks for each SDIO, SPI and the UART controller can be individually enabled/disabled. There is a single clock, each, for the SMC and Quad SPI controllers.

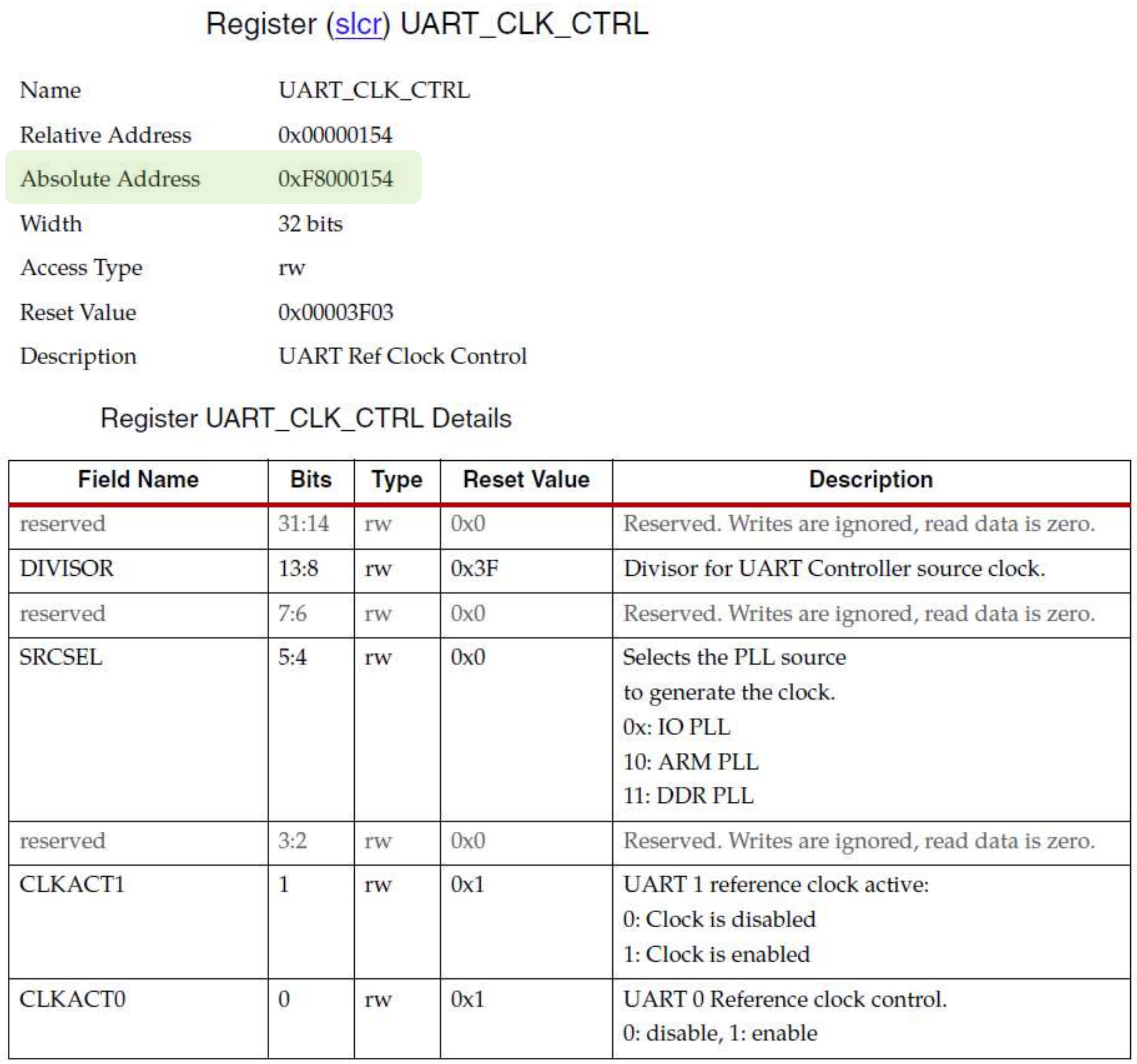


I/O Peripheral	Control Register	Mux Ctrl Field	Mux Ctrl Field	Divider Ctrl Field	Clock Enable Field	
DevC						
SDIO 0	SDIO_CLK_CTRL	SRCSEL, 4	SRCSEL, 5	DIVISOR, 13:8	CLKACT0, 0	SDIO0_REF_CLK
SDIO 1					CLKACT1, 1	SDIO1_REF_CLK
SMC	SMC_CLK_CTRL	SRCSEL, 4	SRCSEL, 5	DIVISOR, 13:8	CLKACT, 0	SMC_REF_CLK
SPI 0	SPI_CLK_CTRL	SRCSEL, 4	SRCSEL, 5	DIVISOR, 13:8	CLKACT0, 0	SPI0_REF_CLK
SPI 1					CLKACT1, 1	SPI1_REF_CLK
Quad-SPI	LQSPI_CLK_CTRL	SRCSEL, 4	SRCSEL, 5	DIVISOR, 13:8	CLKACT, 0	QSPI_REF_CLK
UART 0	UART_CLK_CTRL	SRCSEL, 4	SRCSEL, 5	DIVISOR, 13:8	CLKACT0, 0	UART0_REF_CLK
UART 1					CLKACT1, 1	UART1_REF_CLK

Figure 25-8: **SDIO, SMC, SPI, Quad SPI and UART Reference Clocks**

Check out the next slide for the register information

UART_CLK_CTRL Register



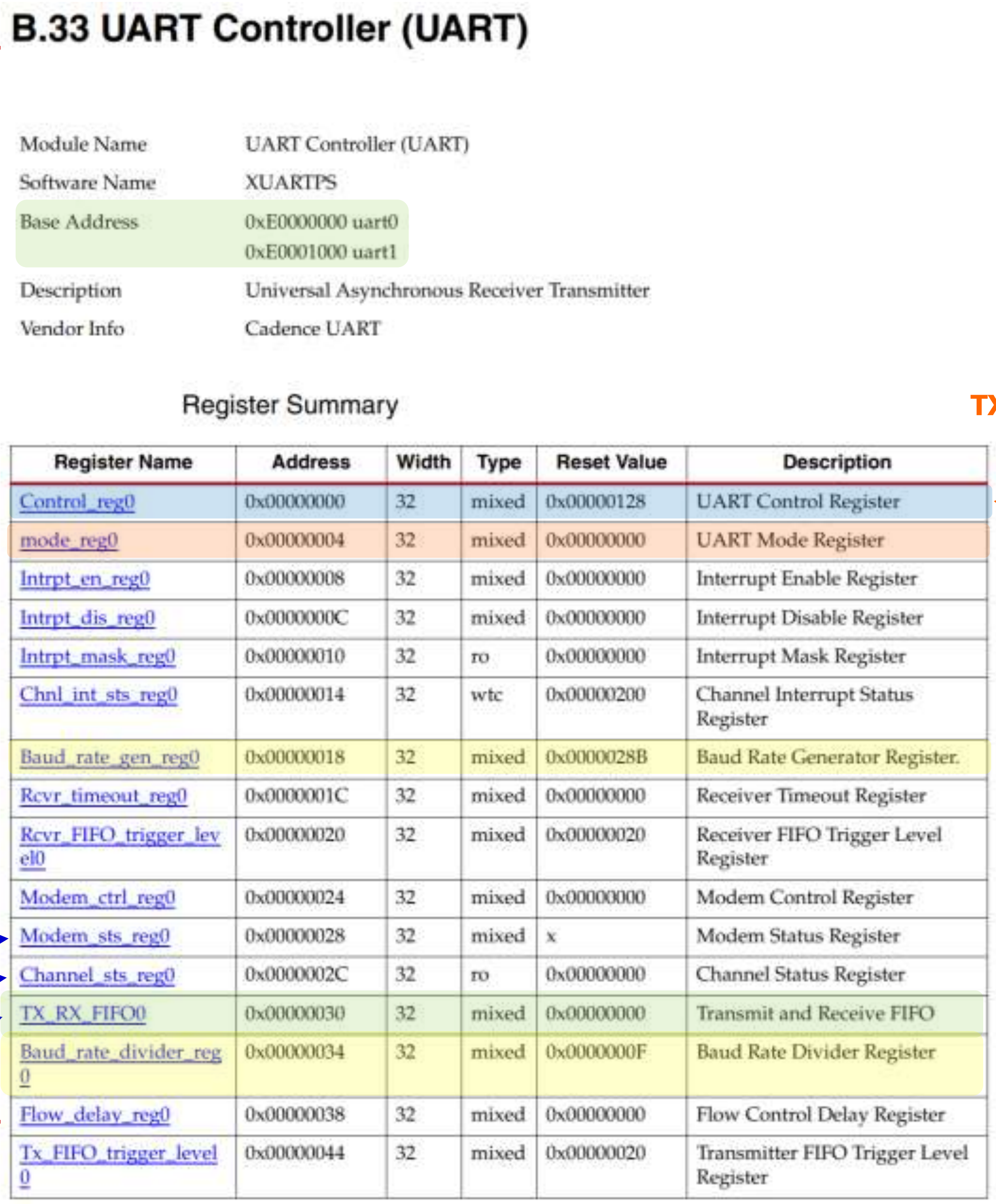
Register (slcr) UART_CLK_CTRL

Name	UART_CLK_CTRL
Relative Address	0x00000154
Absolute Address	0xF8000154
Width	32 bits
Access Type	rw
Reset Value	0x00003F03
Description	UART Ref Clock Control

Register UART_CLK_CTRL Details

Field Name	Bits	Type	Reset Value	Description
reserved	31:14	rw	0x0	Reserved. Writes are ignored, read data is zero.
DIVISOR	13:8	rw	0x3F	Divisor for UART Controller source clock.
reserved	7:6	rw	0x0	Reserved. Writes are ignored, read data is zero.
SRCSEL	5:4	rw	0x0	Selects the PLL source to generate the clock. 0x: IO PLL 10: ARM PLL 11: DDR PLL
reserved	3:2	rw	0x0	Reserved. Writes are ignored, read data is zero.
CLKACT1	1	rw	0x1	UART 1 reference clock active: 0: Clock is disabled 1: Clock is enabled
CLKACT0	0	rw	0x1	UART 0 Reference clock control. 0: disable, 1: enable

UART Registers



B.33 UART Controller (UART)

Module Name	UART Controller (UART)
Software Name	XUARTPS
Base Address	0xE0000000 uart0 0xE0001000 uart1
Description	Universal Asynchronous Receiver Transmitter
Vendor Info	Cadence UART

Register Summary

Register Name	Address	Width	Type	Reset Value	Description
Control_reg0	0x00000000	32	mixed	0x00000128	UART Control Register
mode_reg0	0x00000004	32	mixed	0x00000000	UART Mode Register
Intrpt_en_reg0	0x00000008	32	mixed	0x00000000	Interrupt Enable Register
Intrpt_dis_reg0	0x0000000C	32	mixed	0x00000000	Interrupt Disable Register
Intrpt_mask_reg0	0x00000010	32	ro	0x00000000	Interrupt Mask Register
Chnl_int_sts_reg0	0x00000014	32	wtc	0x00000200	Channel Interrupt Status Register
Baud_rate_gen_reg0	0x00000018	32	mixed	0x0000028B	Baud Rate Generator Register.
Rcvr_timeout_reg0	0x0000001C	32	mixed	0x00000000	Receiver Timeout Register
Rcvr_FIFO_trigger_level0	0x00000020	32	mixed	0x00000020	Receiver FIFO Trigger Level Register
Modem_ctrl_reg0	0x00000024	32	mixed	0x00000000	Modem Control Register
Modem_sts_reg0	0x00000028	32	mixed	x	Modem Status Register
Channel_sts_reg0	0x0000002C	32	ro	0x00000000	Channel Status Register
TX_RX_FIFO0	0x00000030	32	mixed	0x00000000	Transmit and Receive FIFO
Baud_rate_divider_reg0	0x00000034	32	mixed	0x0000000F	Baud Rate Divider Register
Flow_delay_reg0	0x00000038	32	mixed	0x00000000	Flow Control Delay Register
Tx_FIFO_trigger_level0	0x00000044	32	mixed	0x00000020	Transmitter FIFO Trigger Level Register

TX, RX enable/disable

Packet format

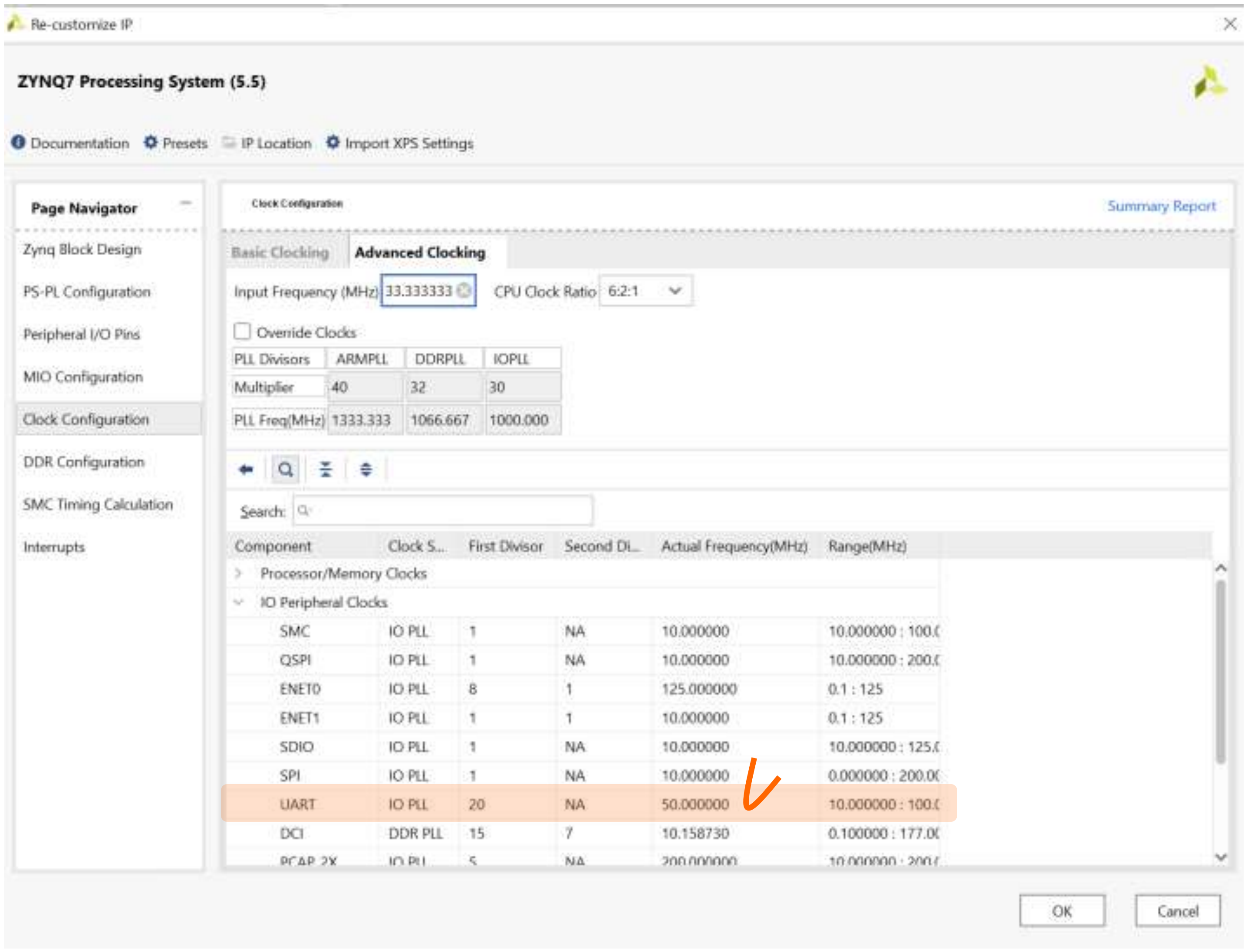
Baud rate setting

Status registers

Data register

Backup Slides

UART Clock Source



Re-customize IP

ZYNQ7 Processing System (5.5)

Documentation | Presets | IP Location | Import XPS Settings

Page Navigator

- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

Clock Configuration — Summary Report

Basic Clocking | **Advanced Clocking**

Input Frequency (MHz) 33.333333 CPU Clock Ratio 6:2:1

☐ Override Clocks

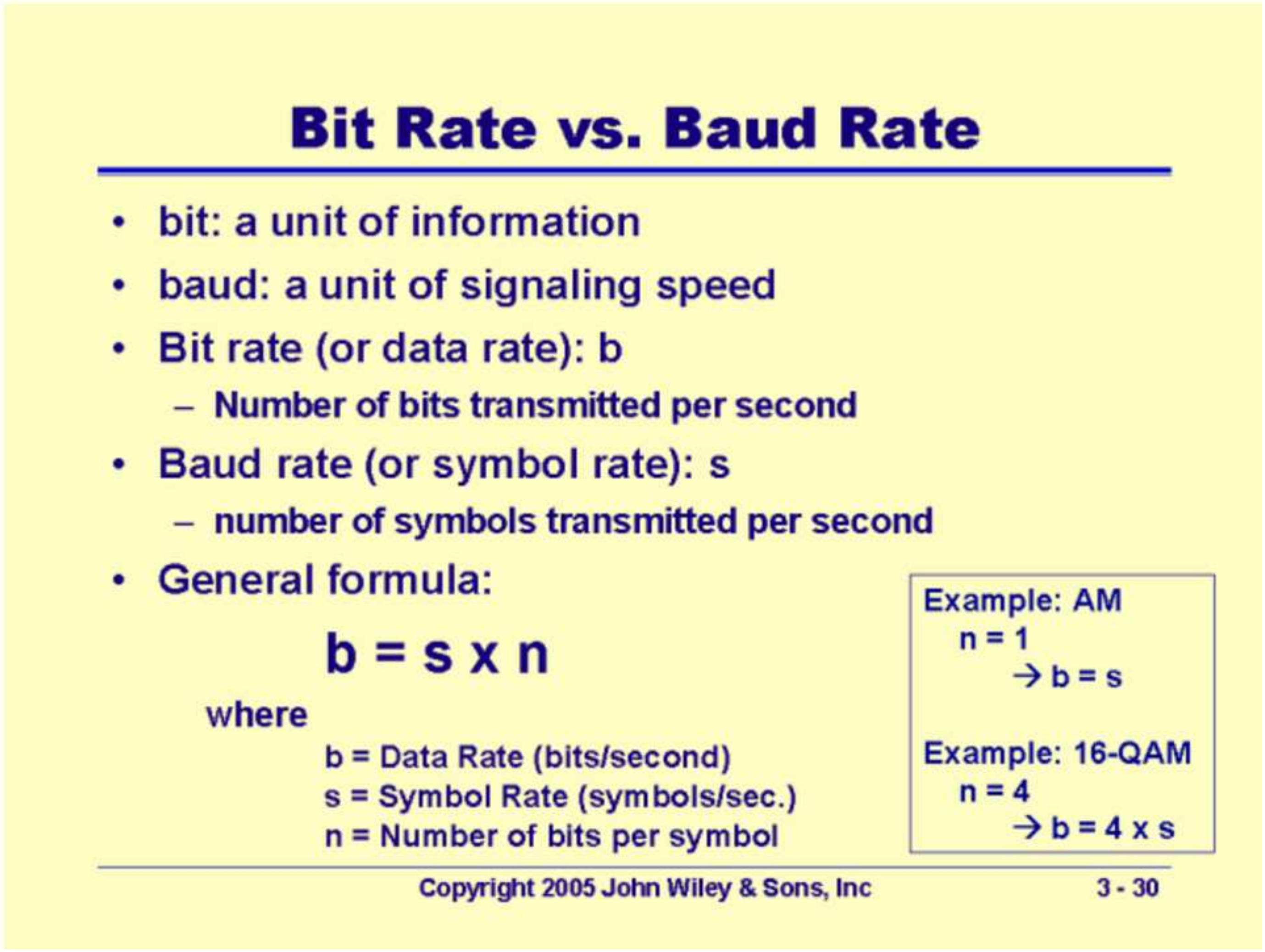
PLL Divisors	ARMPLL	DDRPLL	IOPLL
Multiplier	40	32	30
PLL Freq(MHz)	1333.333	1066.667	1000.000

Search:

Component	Clock S...	First Divisor	Second Di...	Actual Frequency(MHz)	Range(MHz)
› Processor/Memory Clocks					
⌄ IO Peripheral Clocks					
SMC	IO PLL	1	NA	10.000000	10.000000 : 100.0
QSPI	IO PLL	1	NA	10.000000	10.000000 : 200.0
ENET0	IO PLL	8	1	125.000000	0.1 : 125
ENET1	IO PLL	1	1	10.000000	0.1 : 125
SDIO	IO PLL	1	NA	10.000000	10.000000 : 125.0
SPI	IO PLL	1	NA	10.000000	0.000000 : 200.00
UART	IO PLL	20	NA	50.000000 ✓	10.000000 : 100.0
DCI	DDR PLL	15	7	10.158730	0.100000 : 177.00
PCAP_2X	IO PLL	5	NA	200.000000	10.000000 : 200.0

OK | Cancel

Bit Rate vs Baud Rate



Bit Rate vs. Baud Rate

- bit: a unit of information
- baud: a unit of signaling speed
- Bit rate (or data rate): b
  - Number of bits transmitted per second
- Baud rate (or symbol rate): s
  - number of symbols transmitted per second
- General formula:

$$b = s \times n$$

where

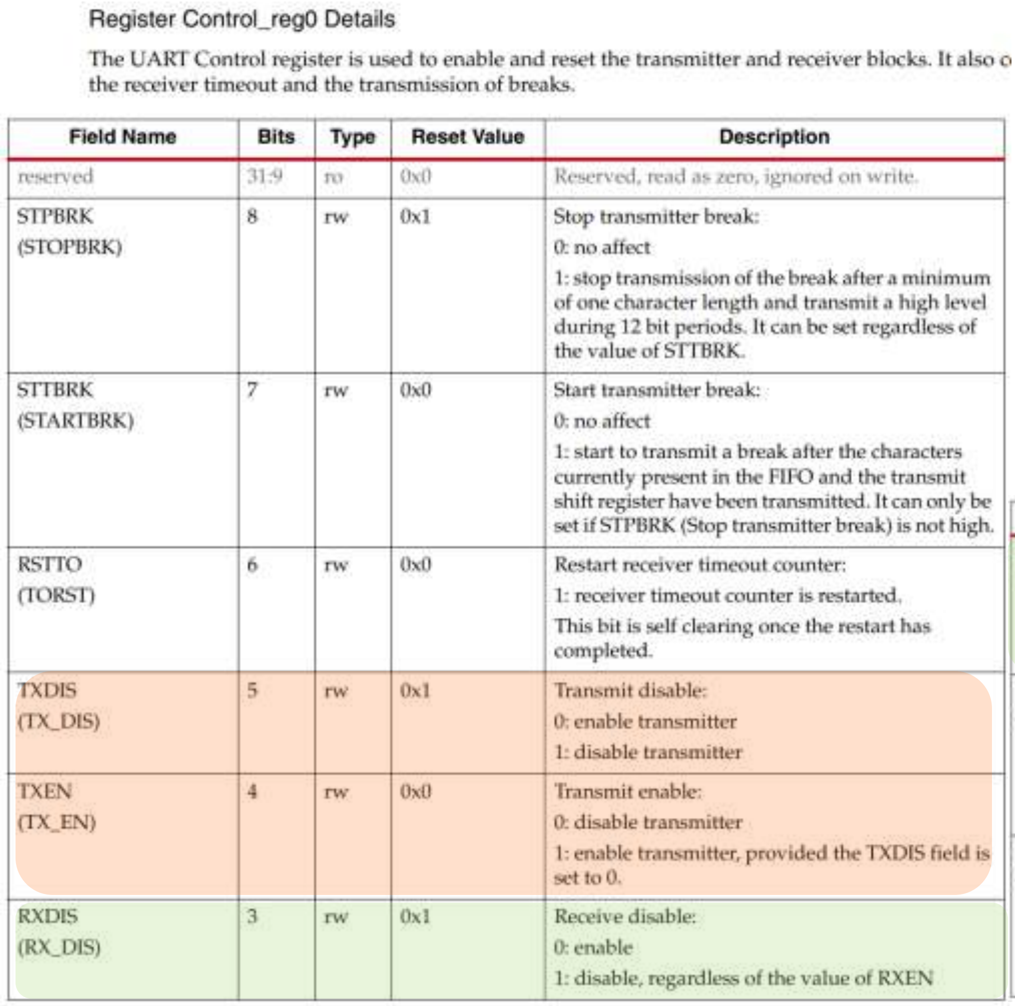
b = Data Rate (bits/second)
s = Symbol Rate (symbols/sec.)
n = Number of bits per symbol

Example: AM
$n = 1$
$\rightarrow b = s$

Example: 16-QAM
$n = 4$
$\rightarrow b = 4 \times s$

3 - 30

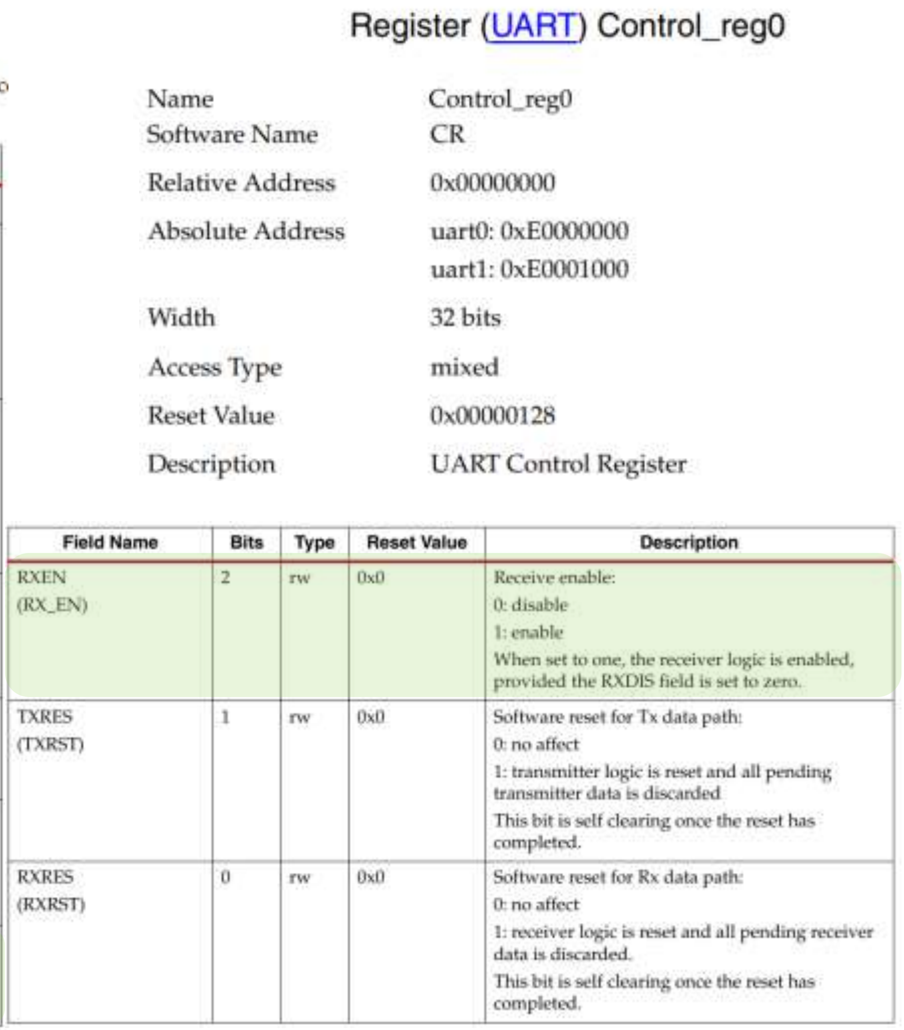
Control_reg0 register



Register Control_reg0 Details

The UART Control register is used to enable and reset the transmitter and receiver blocks. It also c
the receiver timeout and the transmission of breaks.

Field Name	Bits	Type	Reset Value	Description
reserved	31:9	ro	0x0	Reserved, read as zero, ignored on write.
STPBRK (STOPBRK)	8	rw	0x1	Stop transmitter break: 0: no affect 1: stop transmission of the break after a minimum of one character length and transmit a high level during 12 bit periods. It can be set regardless of the value of STTBRK.
STTBRK (STARTBRK)	7	rw	0x0	Start transmitter break: 0: no affect 1: start to transmit a break after the characters currently present in the FIFO and the transmit shift register have been transmitted. It can only be set if STPBRK (Stop transmitter break) is not high.
RSTTO (TORST)	6	rw	0x0	Restart receiver timeout counter: 1: receiver timeout counter is restarted. This bit is self clearing once the restart has completed.
TXDIS (TX_DIS)	5	rw	0x1	Transmit disable: 0: enable transmitter 1: disable transmitter
TXEN (TX_EN)	4	rw	0x0	Transmit enable: 0: disable transmitter 1: enable transmitter, provided the TXDIS field is set to 0.
RXDIS (RX_DIS)	3	rw	0x1	Receive disable: 0: enable 1: disable, regardless of the value of RXEN

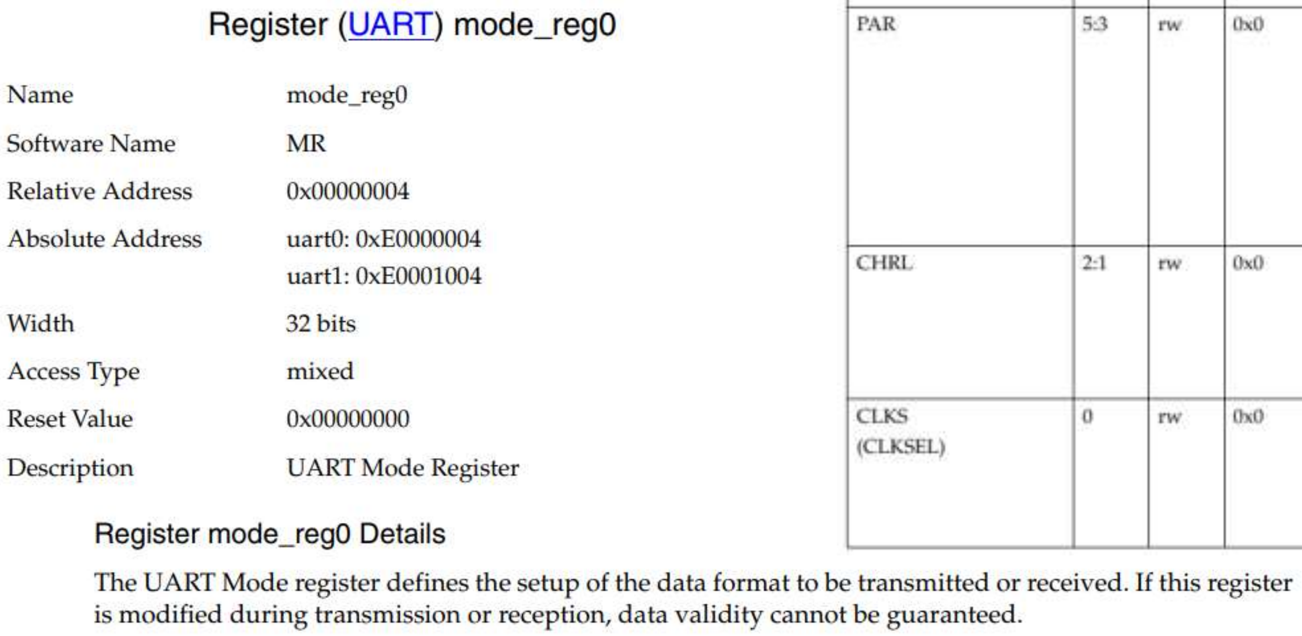


Register (UART) Control_reg0

Name	Control_reg0
Software Name	CR
Relative Address	0x00000000
Absolute Address	uart0: 0xE0000000 uart1: 0xE0001000
Width	32 bits
Access Type	mixed
Reset Value	0x00000128
Description	UART Control Register

Field Name	Bits	Type	Reset Value	Description
RXEN (RX_EN)	2	rw	0x0	Receive enable: 0: disable 1: enable When set to one, the receiver logic is enabled, provided the RXDIS field is set to zero.
TXRES (TXRST)	1	rw	0x0	Software reset for Tx data path: 0: no affect 1: transmitter logic is reset and all pending transmitter data is discarded This bit is self clearing once the reset has completed.
RXRES (RXRST)	0	rw	0x0	Software reset for Rx data path: 0: no affect 1: receiver logic is reset and all pending receiver data is discarded. This bit is self clearing once the reset has completed.

Mode_reg0 register

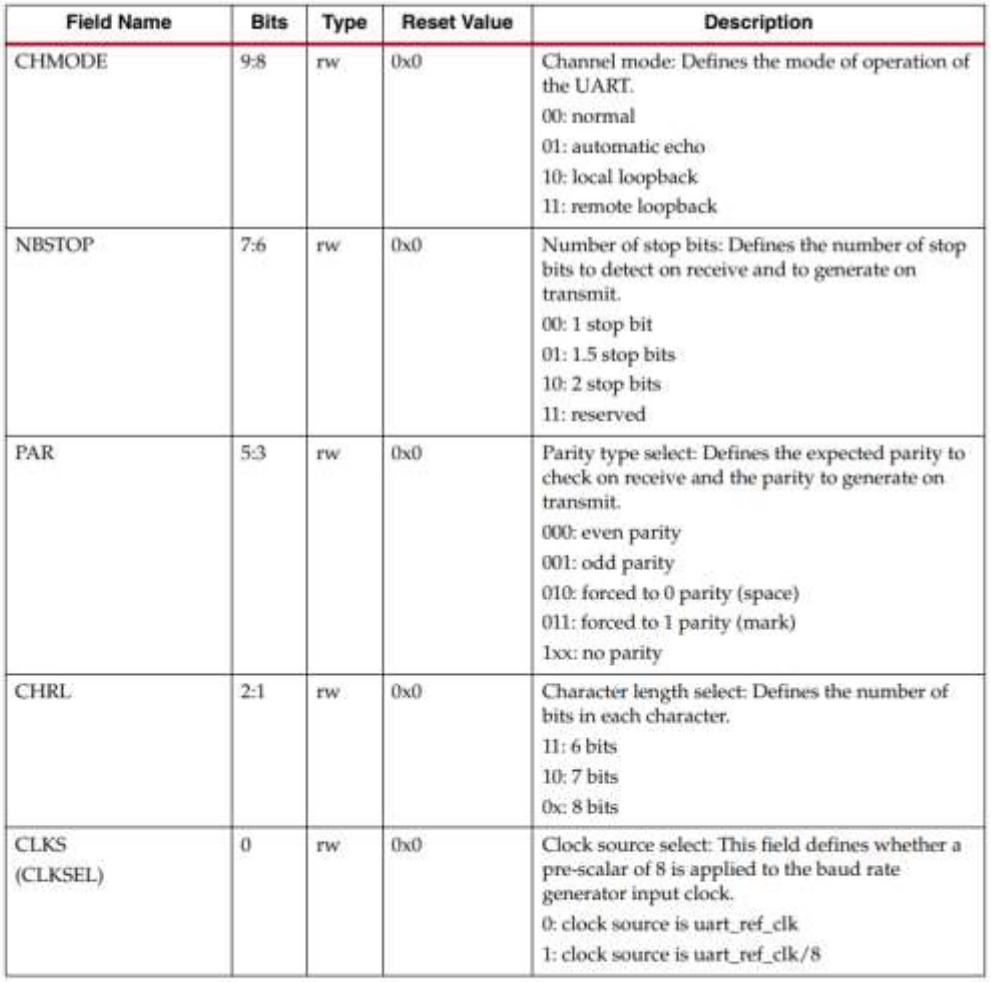


Register (UART) mode_reg0

Name	mode_reg0
Software Name	MR
Relative Address	0x00000004
Absolute Address	uart0: 0xE0000004 uart1: 0xE0001004
Width	32 bits
Access Type	mixed
Reset Value	0x00000000
Description	UART Mode Register

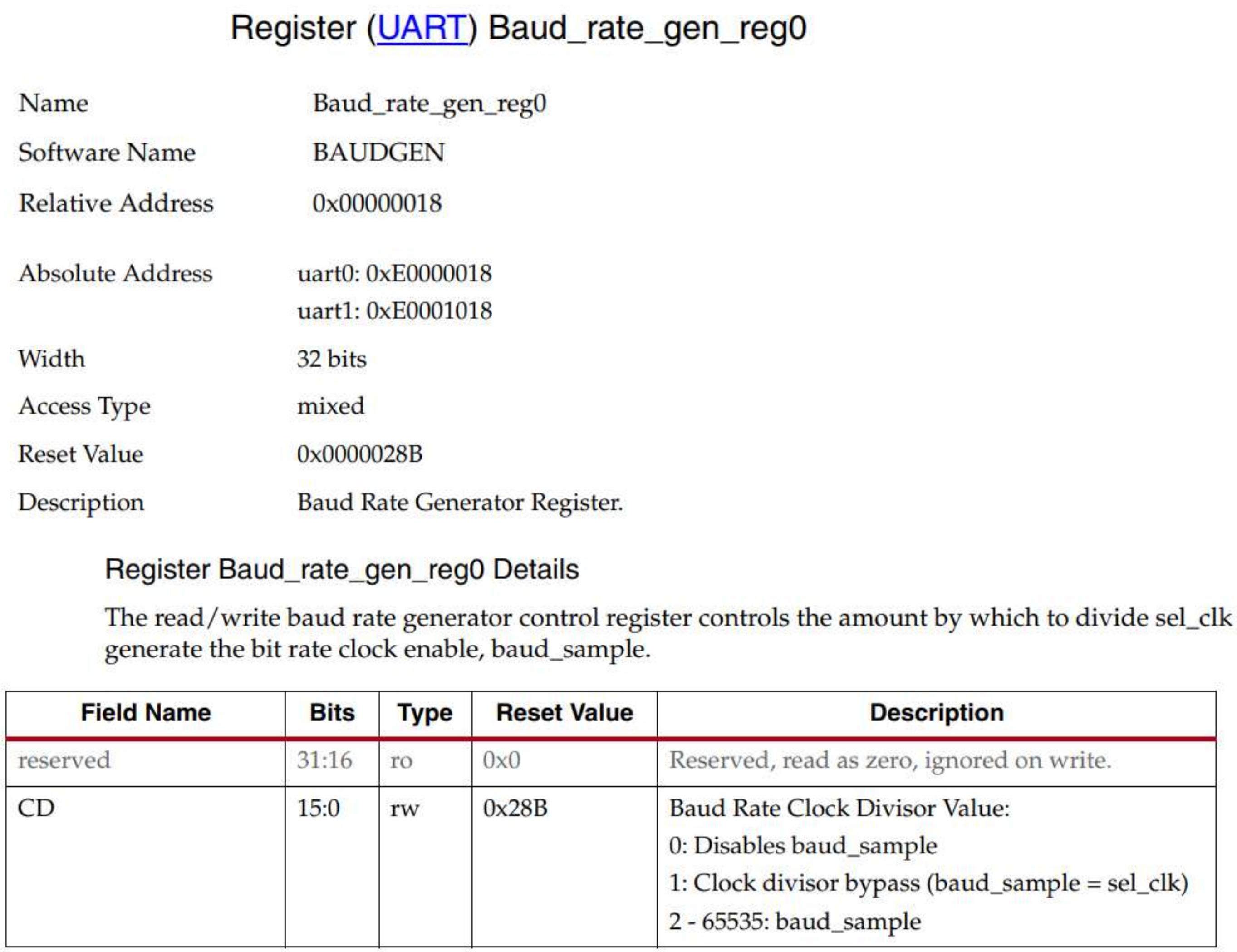
Register mode_reg0 Details

The UART Mode register defines the setup of the data format to be transmitted or received. If this register is modified during transmission or reception, data validity cannot be guaranteed.



Field Name	Bits	Type	Reset Value	Description
CHMODE	9:8	rw	0x0	Channel mode: Defines the mode of operation of the UART. 00: normal 01: automatic echo 10: local loopback 11: remote loopback
NBSTOP	7:6	rw	0x0	Number of stop bits: Defines the number of stop bits to detect on receive and to generate on transmit. 00: 1 stop bit 01: 1.5 stop bits 10: 2 stop bits 11: reserved
PAR	5:3	rw	0x0	Parity type select: Defines the expected parity to check on receive and the parity to generate on transmit. 000: even parity 001: odd parity 010: forced to 0 parity (space) 011: forced to 1 parity (mark) 1xx: no parity
CHRL	2:1	rw	0x0	Character length select: Defines the number of bits in each character. 11: 6 bits 10: 7 bits 0x: 8 bits
CLKS (CLKSEL)	0	rw	0x0	Clock source select: This field defines whether a pre-scalar of 8 is applied to the baud rate generator input clock. 0: clock source is uart_ref_clk 1: clock source is uart_ref_clk/8

Baud_rate_gen_reg0 register



Register (UART) Baud_rate_gen_reg0

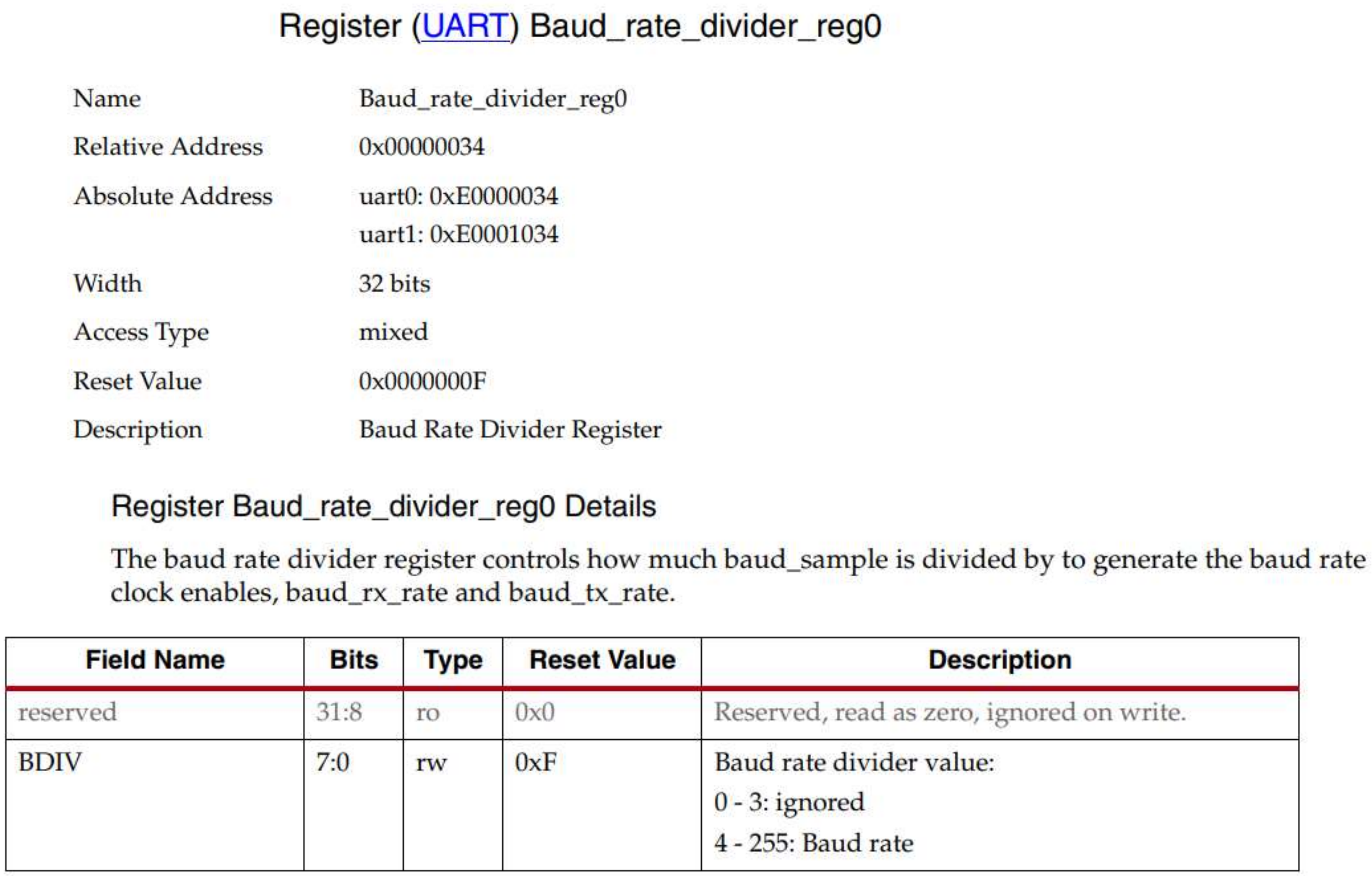
Name	Baud_rate_gen_reg0
Software Name	BAUDGEN
Relative Address	0x00000018
Absolute Address	uart0: 0xE0000018 uart1: 0xE0001018
Width	32 bits
Access Type	mixed
Reset Value	0x0000028B
Description	Baud Rate Generator Register.

Register Baud_rate_gen_reg0 Details

The read/write baud rate generator control register controls the amount by which to divide sel_clk to generate the bit rate clock enable, baud_sample.

Field Name	Bits	Type	Reset Value	Description
reserved	31:16	ro	0x0	Reserved, read as zero, ignored on write.
CD	15:0	rw	0x28B	Baud Rate Clock Divisor Value: 0: Disables baud_sample 1: Clock divisor bypass (baud_sample = sel_clk) 2 - 65535: baud_sample

Baud_rate_divider_reg0 register



Register (UART) Baud_rate_divider_reg0

Name	Baud_rate_divider_reg0
Relative Address	0x00000034
Absolute Address	uart0: 0xE0000034 uart1: 0xE0001034
Width	32 bits
Access Type	mixed
Reset Value	0x0000000F
Description	Baud Rate Divider Register

Register Baud_rate_divider_reg0 Details

The baud rate divider register controls how much baud_sample is divided by to generate the baud rate clock enables, baud_rx_rate and baud_tx_rate.

Field Name	Bits	Type	Reset Value	Description
reserved	31:8	ro	0x0	Reserved, read as zero, ignored on write.
BDIV	7:0	rw	0xF	Baud rate divider value: 0 - 3: ignored 4 - 255: Baud rate

TX_RX_FIFO register

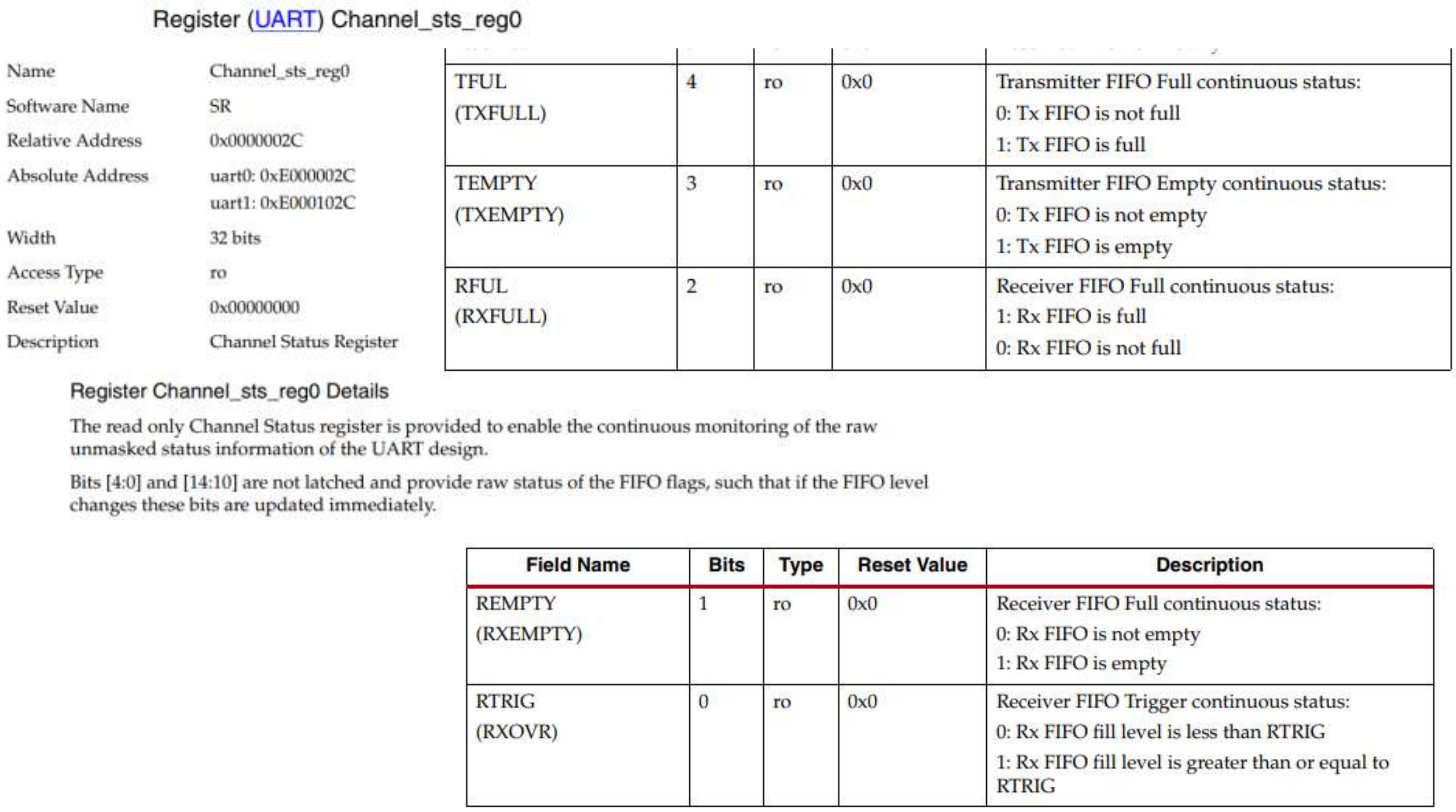
Register (UART) TX_RX_FIFO0

Name	TX_RX_FIFO0
Software Name	FIFO
Relative Address	0x00000030
Absolute Address	uart0: 0xE0000030 uart1: 0xE0001030
Width	32 bits
Access Type	mixed
Reset Value	0x00000000
Description	Transmit and Receive FIFO

Register TX_RX_FIFO0 Details

Field Name	Bits	Type	Reset Value	Description
reserved	31:8	ro	0x0	Reserved, read as zero, ignored on write.
FIFO	7:0	rw	0x0	Operates as Tx FIFO and Rx FIFO.

Channel_sts_reg0 register



Register (UART) Channel_sts_reg0

Name	Channel_sts_reg0
Software Name	SR
Relative Address	0x0000002C
Absolute Address	uart0: 0xE000002C uart1: 0xE000102C
Width	32 bits
Access Type	ro
Reset Value	0x00000000
Description	Channel Status Register

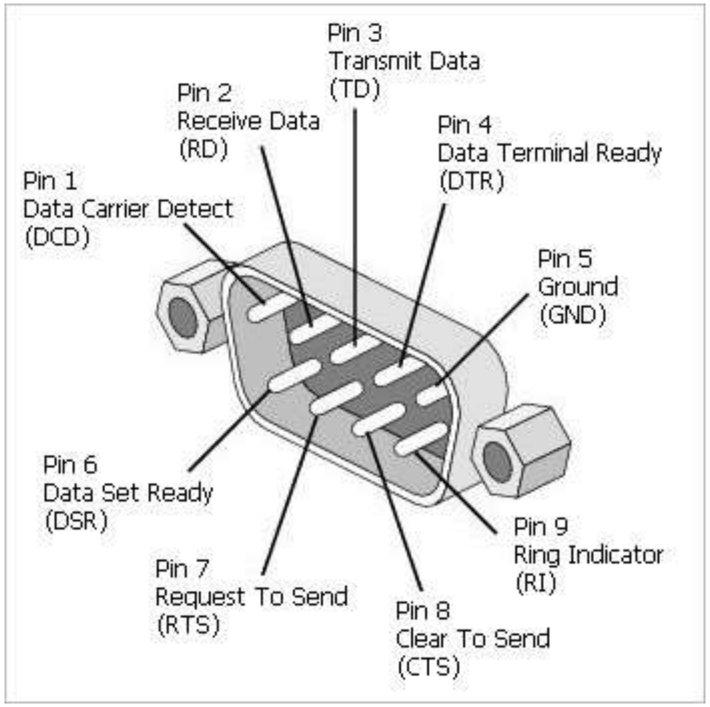
Register Channel_sts_reg0 Details

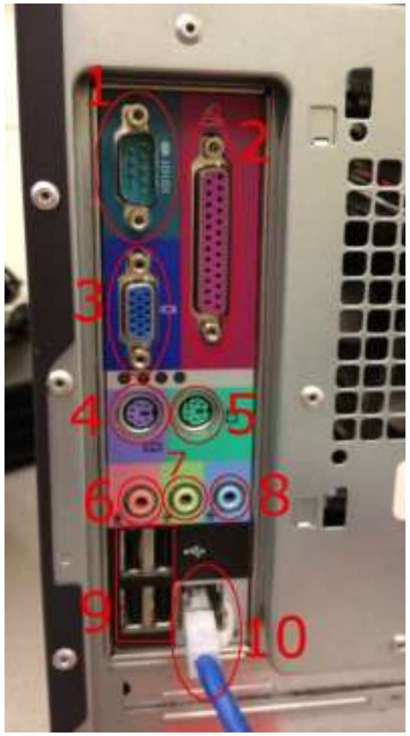
The read only Channel Status register is provided to enable the continuous monitoring of the raw unmasked status information of the UART design.

Bits [4:0] and [14:10] are not latched and provide raw status of the FIFO flags, such that if the FIFO level changes these bits are updated immediately.

Field Name	Bits	Type	Reset Value	Description
TFUL (TXFULL)	4	ro	0x0	Transmitter FIFO Full continuous status: 0: Tx FIFO is not full 1: Tx FIFO is full
TEMPTY (TXEMPTY)	3	ro	0x0	Transmitter FIFO Empty continuous status: 0: Tx FIFO is not empty 1: Tx FIFO is empty
RFUL (RXFULL)	2	ro	0x0	Receiver FIFO Full continuous status: 1: Rx FIFO is full 0: Rx FIFO is not full
REMPTY (RXEMPTY)	1	ro	0x0	Receiver FIFO Full continuous status: 0: Rx FIFO is not empty 1: Rx FIFO is empty
RTRIG (RXOVR)	0	ro	0x0	Receiver FIFO Trigger continuous status: 0: Rx FIFO fill level is less than RTRIG 1: Rx FIFO fill level is greater than or equal to RTRIG

UART 9-Pinouts





- RTS (Request to Send) / CTS (Clear to Send) for flow control
  - Flow control means the ability to slow down the flow of bytes in a wire
  - For serial ports, this means the ability to stop and then restart the flow without any loss of bytes

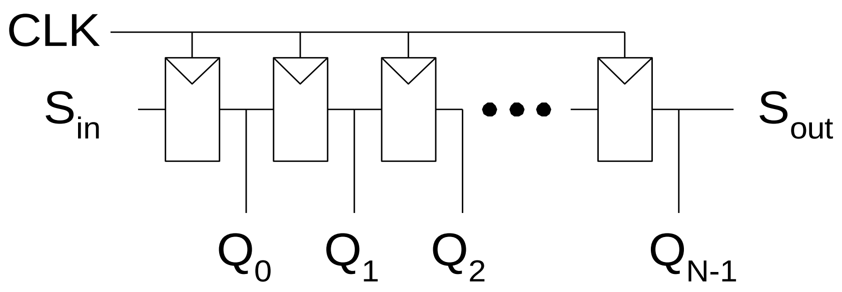
https://www.ingenuityworking.com/knowledge/w/knowledgebase/1016.aspx
http://www.tldp.org/HOWTO/Serial-HOWTO-4.html#flow_control

Line Break

- RS232-C also specifies a signal called a Break, which is caused by sending continuous Spacing values (no Start or Stop bits). When there is no electricity present on the data circuit, the line is considered to be sending Break.
- The Break signal must be of a duration longer than the time it takes to send a complete byte plus Start, Stop and Parity bits. Most UARTs can distinguish between a Framing Error and a Break, but if the UART cannot do this, the Framing Error detection can be used to identify Breaks.
- **In the days of teleprinters**, when numerous printers around the country were wired in series (such as news services), any unit could cause a Break by temporarily opening the entire circuit so that no current flowed. This was used to allow a location with **urgent news** to interrupt some other location that was currently sending information.
- **In modern systems** there are **two types of Break signals**. If the Break is longer than 1.6 seconds, it is considered a "Modem Break", and some modems can be programmed to terminate the conversation and go on-hook or enter the modems' command mode when the modem detects this signal. If the Break is smaller than 1.6 seconds, it signifies a Data Break and it is up to the remote computer to respond to this signal. Sometimes this form of Break is used as an Attention or Interrupt signal and sometimes is accepted as a substitute for the ASCII CONTROL-C character.

Shift Register

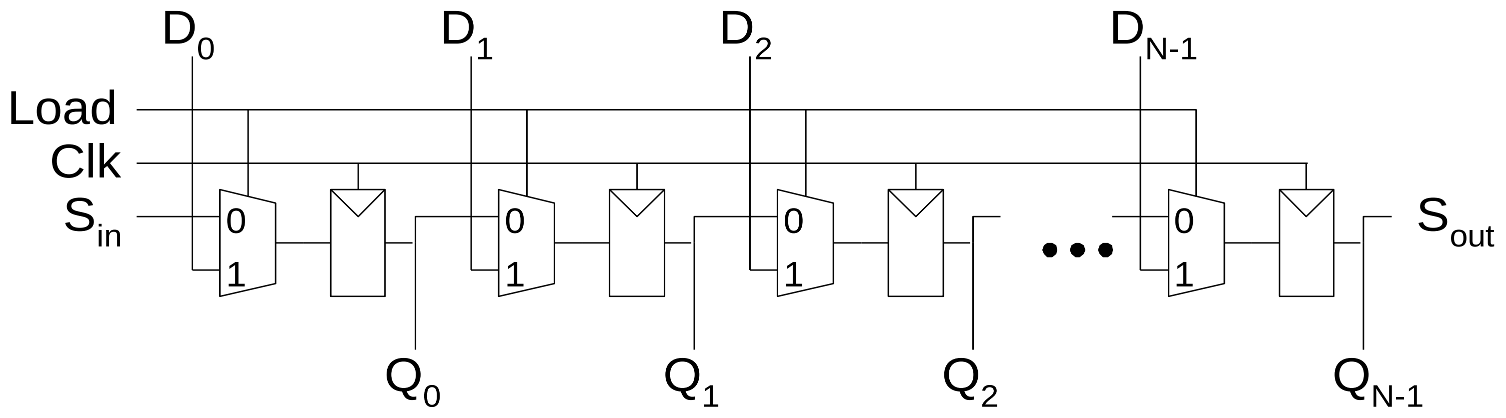
- Shift register has a clock, a serial input (S_{in}), a serial output (S_{out}), and N parallel outputs (Q[N-1:0])
- A new bit is shifted in from S_{in} on each clock edge
  - All the subsequent contents are shifted forward
  - The last bit in the shift register is available at S_{out}
- Used as serial-to-parallel converter
  - Converts serial input (S_{in}) to parallel output (Q[N-1:0])
- Don't be confused with shifters, which are combinational logic blocks that shift an input by a specified amount



Shift Register with Parallel Load

- Parallel-to-serial converter
  - When Load = 1, acts as a normal N-bit register
  - When Load = 0, acts as a shift register

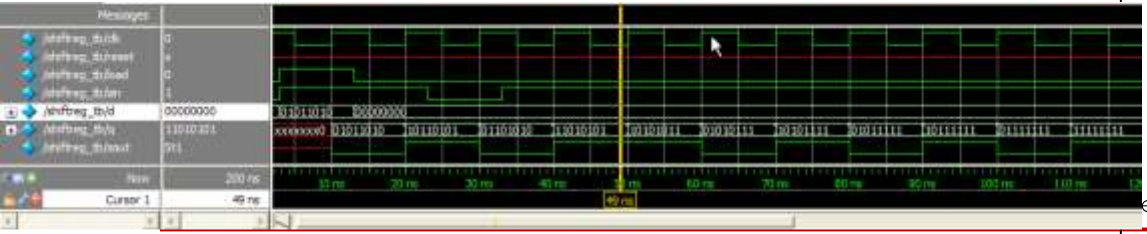
- Now a shift register can act as
  - Serial-to-parallel converter (S_{in} to Q[N-1:0])
  - Parallel-to-serial converter (D[N-1:0] to S_{out})



Verilog Representation

```
module shiftreg #(parameter N = 8)
      (input              clk,
       input              reset, load,
       input              sin,
       input       [N-1:0]  d,
       output reg  [N-1:0]  q,
       output             sout);

  always @(posedge clk or posedge reset)
  begin
    if (reset)  q <= 0;
    else
      if (load) q[N-1:0] <= d;
      else      q[N-1:0] <= {q[N-2:0], sin};
  end

  assign sout = q[N-1];

endmodule
```

```
`timescale 1ns / 1ns

module shiftreg_tb();
 reg  clk, reset, load, sin,
 reg  [7:0]  d;

 wire [7:0]  q;
 wire           sout;

 parameter  clk_period = 10;

 shiftreg shiftreg_uut
   (.clk (clk), .reset (reset), .load (load),
    .sin (sin), .d (d), .q (q), .sout
(sout));

 always
 begin
       clk = 1;
       forever #(clk_period/2) clk = ~clk;
 end

 initial
 begin
       load   = 1'b0; d = 8'h00; #3;
       load   = 1'b1; d = 8'h5A;
#(clk_period);
       load   = 1'b0; d = 8'h00;
 end

 initial
 begin
      sin    = 1'b0; #3;
      sin    = 1'b1; #(clk_period);
      sin    = 1'b1; #(clk_period);
      sin    = 1'b0; #(clk_period);
      sin    = 1'b1; #(clk_period);
      sin    = 1'b1; #(clk_period);
 end

endmodule
```



Messages
/shiftreg_tb/clk
/shiftreg_tb/reset
/shiftreg_tb/load
/shiftreg_tb/sin
/shiftreg_tb/d
/shiftreg_tb/q
/shiftreg_tb/sout
Now
200 ns
Cursor 1
49 ns
10 ns
20 ns
30 ns
40 ns