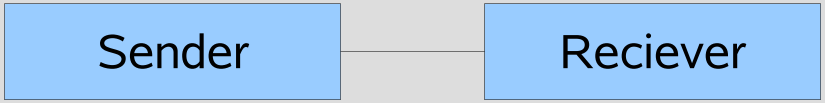
Data communication

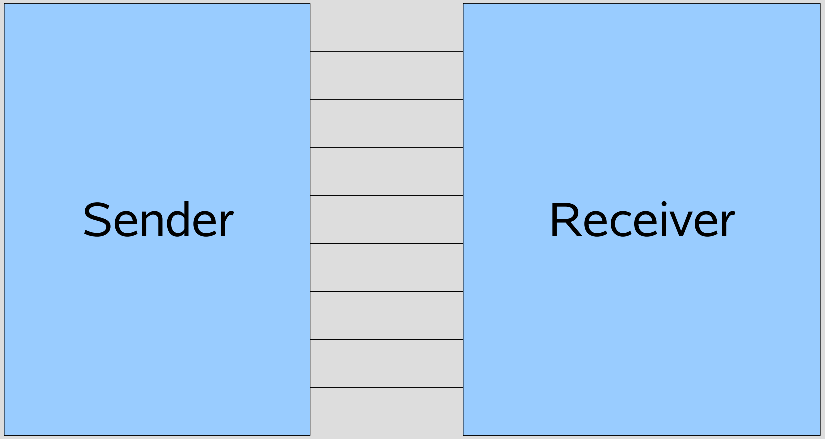
- Some Embedded systems cannot work in isolation.
- They may need to communicate with the other systems.
- Two approaches of the communication is possible.
- Serial communication & parallel communication.

Continued...


Sender

Reciever

- A bit is transmitted at a single time
- Slower [↓]
- Few wires [↑]


Sender

Receiver

- Byte or more at a time. [↑]
- Faster [↑]
- Large wire [↓]

Serial data transmission type

Simplex

DUPLEX

FULL DUPLEX

- Transmission only in one direction

- Data transmitted in one direction
- Direction of data can be changed

- Both direction & simultaneously

Serial com transmission mode

- Synchronous mode
- Asynchronous mode

Asynchronous mode

- Transmit one character at a time
- Data bits in a character ?
- Order of transmission LSB to MSB
- Start and stop bits tells receiver where one character stops and other starts
- Stop bits can be either one or two
- Parity bit for error detection
- Sending parity bit is optional

frame format

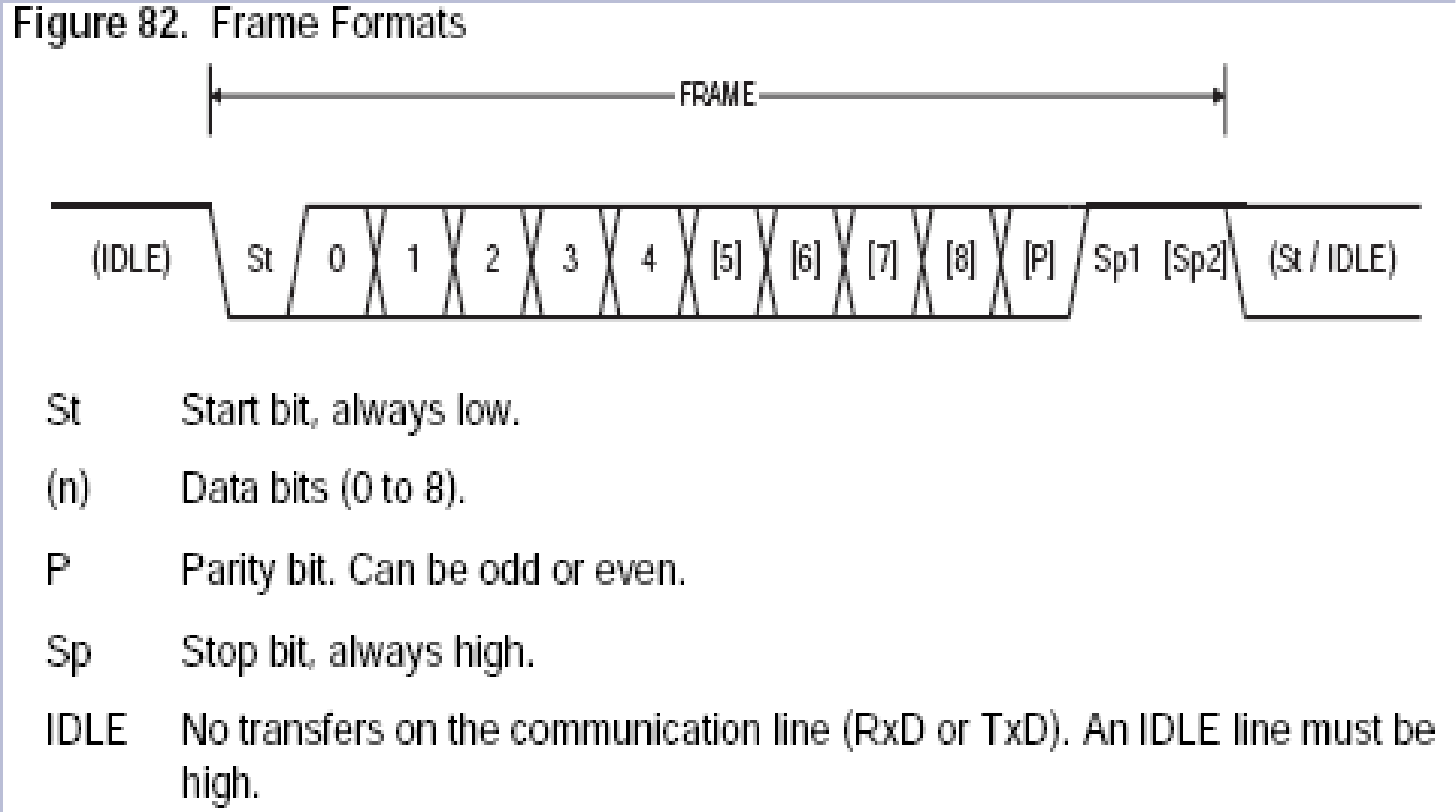


Figure 82. Frame Formats

St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high.

Async.. serial transfer

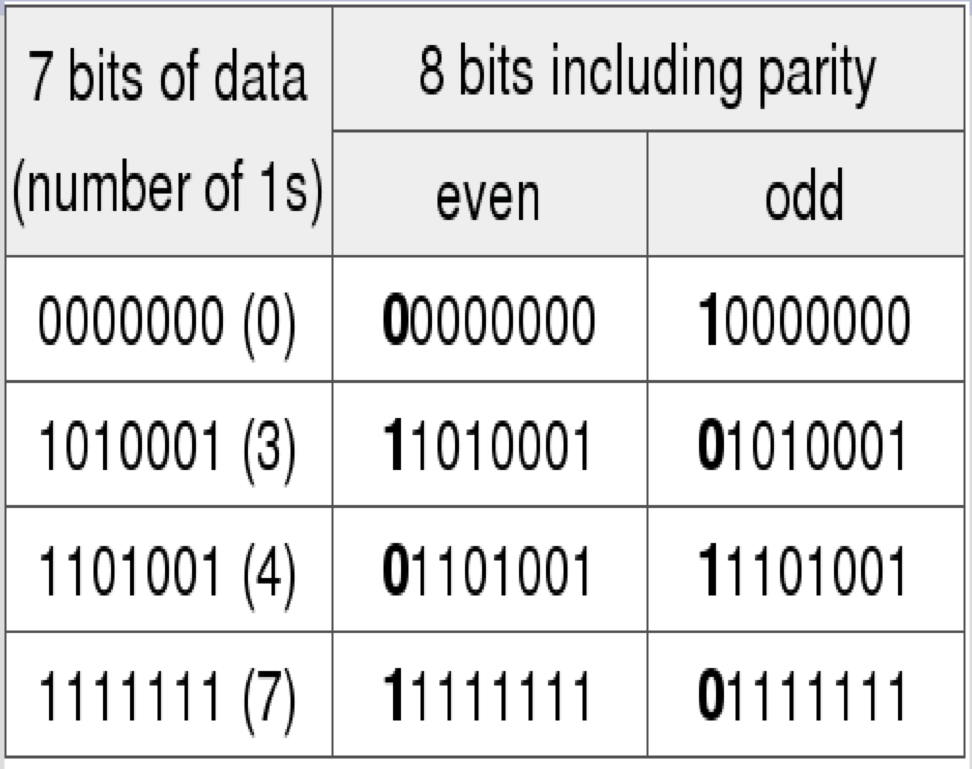
- NO data- signal is kept high
- Start bit – signal low for 1 bit time
- The data transmission rates are specified in terms of Baud rate
- Common values 2400,9600,19200,57600,115200
- Baud rate

Atmega8 USART facts

- Full duplex operation
  (independent receive and transmit register)
- High resolution baud rate register (UBRR)
- Serial frames with 5 -9 data bits
  (1-2 stop bits ,odd even parity generator,checking by H/W)
- Errors
  data overrun :data loss due to receive buff full frame error : error in the stops bits received   Parity error:

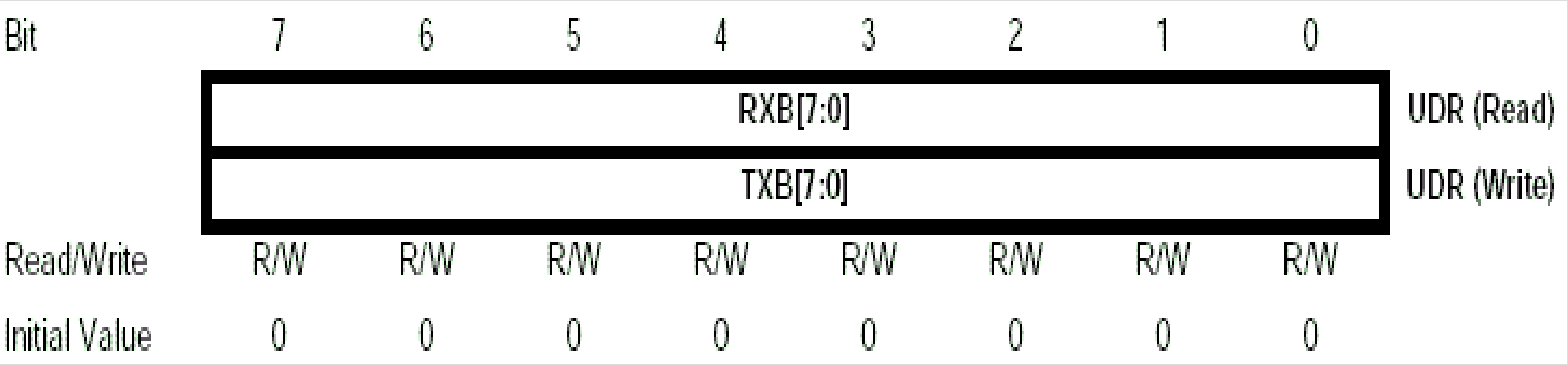
Parity Bit

- A parity bit is a bit that is added to ensure that the number of bits with value of one in a given set of bits is always even or odd.
- Parity bits are used as the simplest error detecting code.

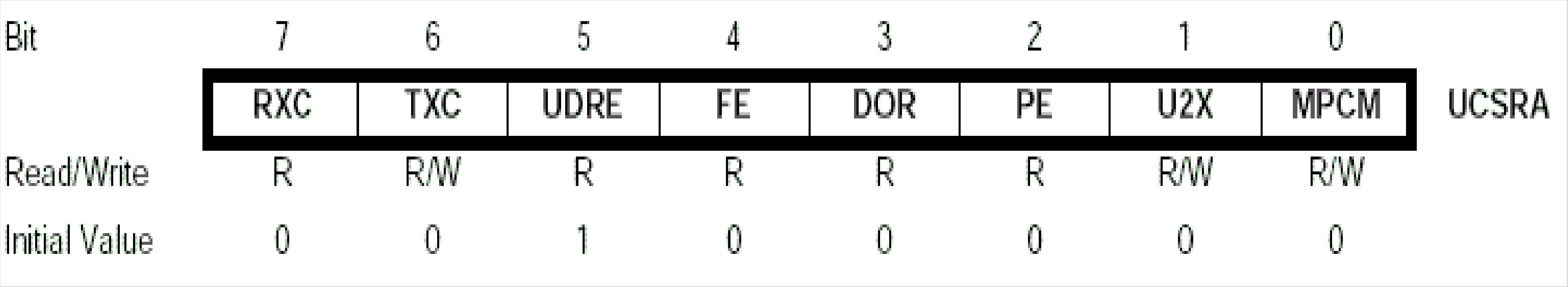


7 bits of data (number of 1s)	8 bits including parity	
	even	odd
0000000 (0)	**0**0000000	**1**0000000
1010001 (3)	**1**1010001	**0**1010001
1101001 (4)	**0**1101001	**1**1101001
1111111 (7)	**1**1111111	**0**1111111

Registers

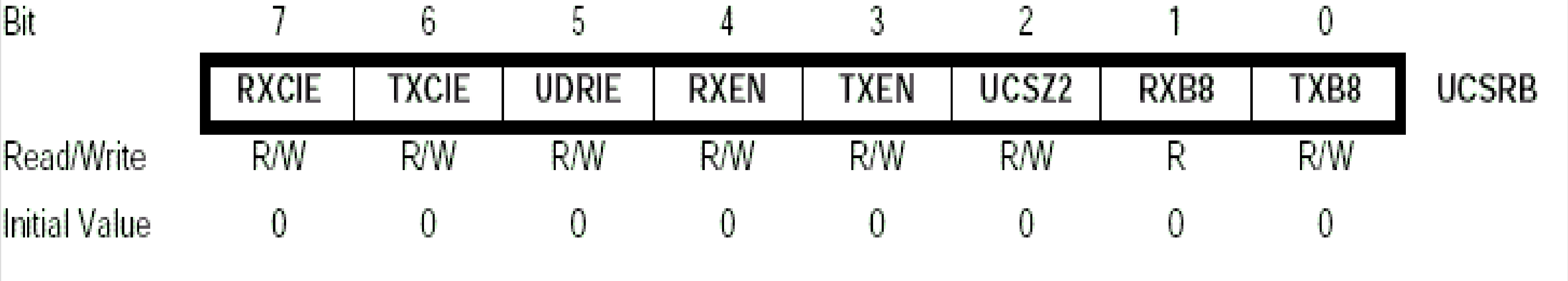


Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

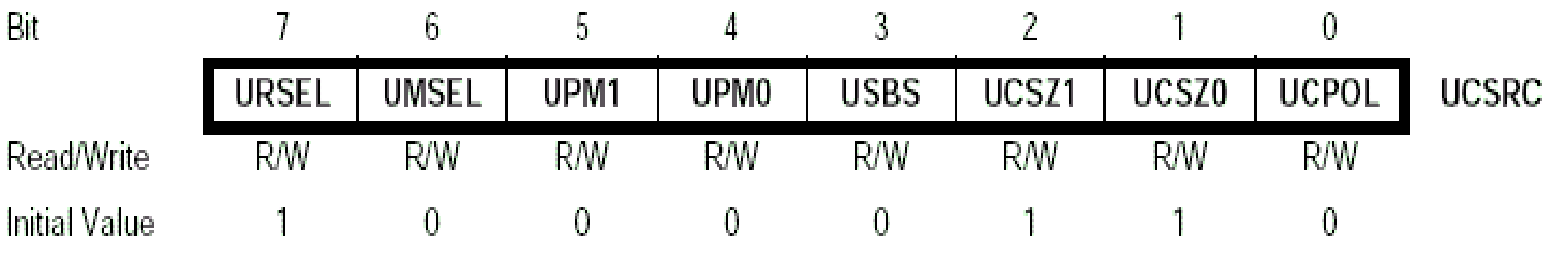


Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

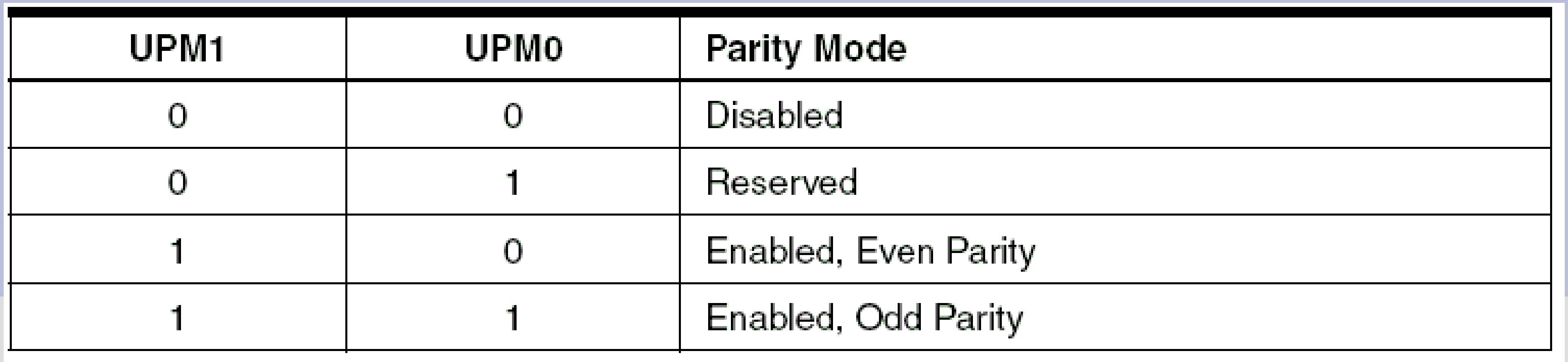
Registers



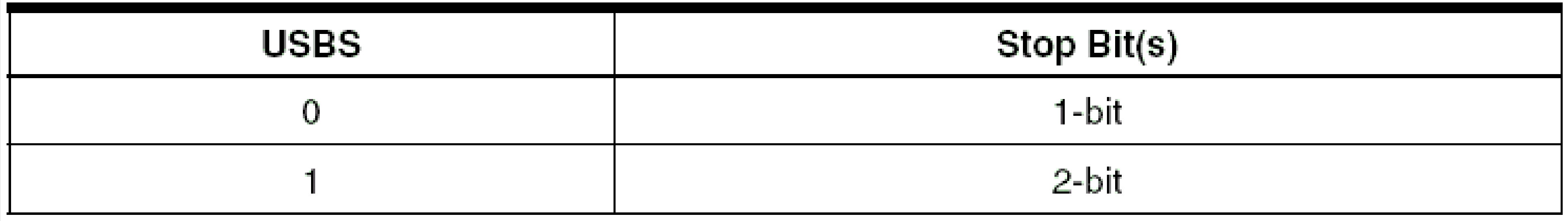
Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	



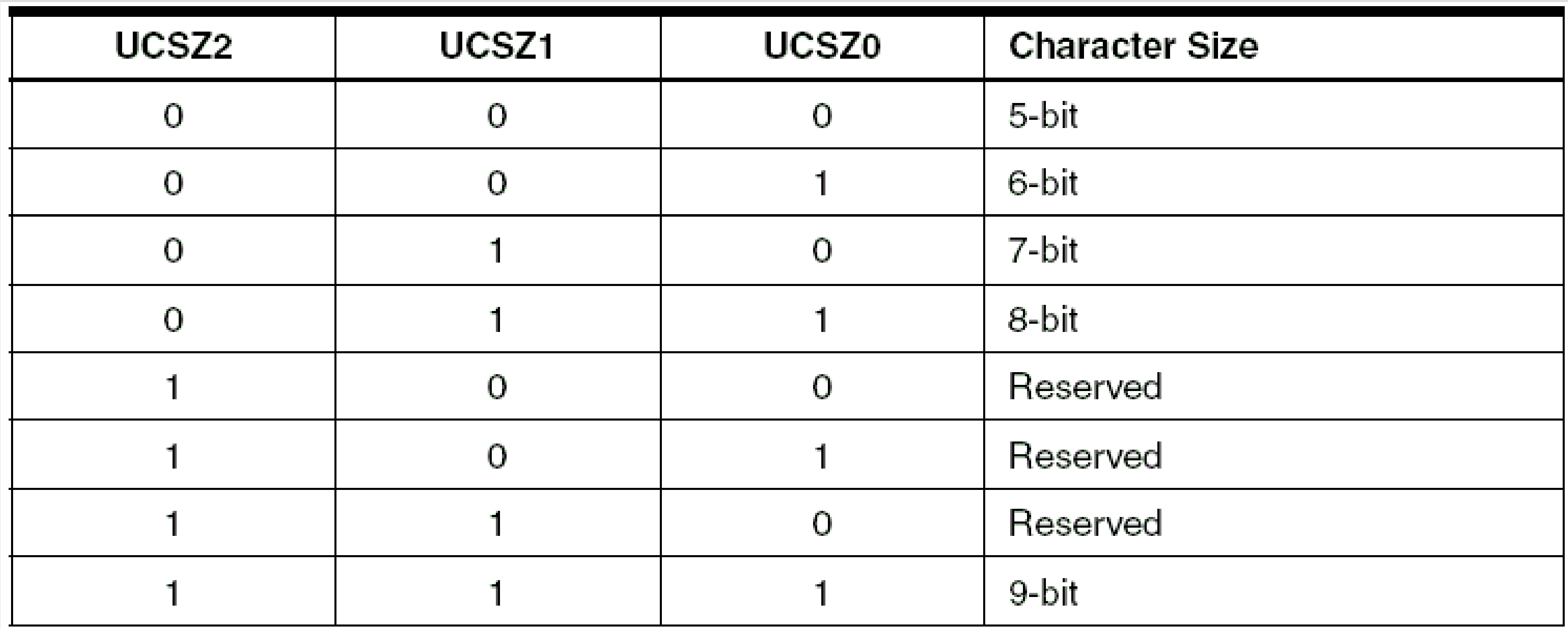
Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	



UPM1	UPM0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

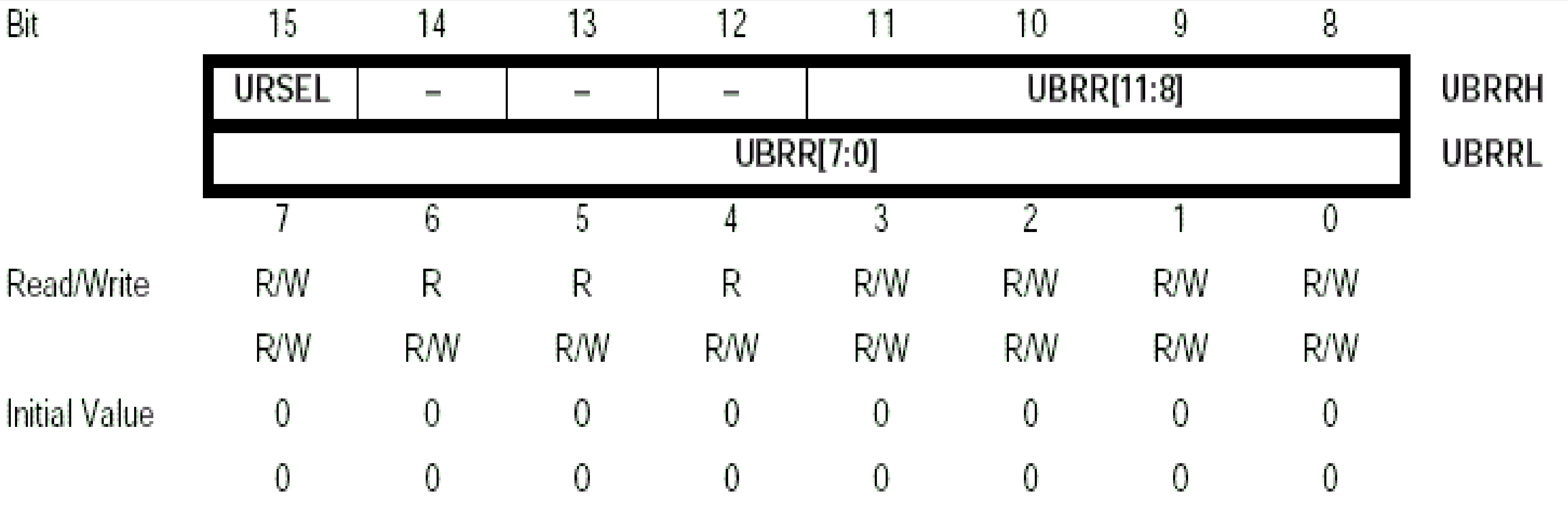


USBS	Stop Bit(s)
0	1-bit
1	2-bit



UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Baud Rate register



Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Equations

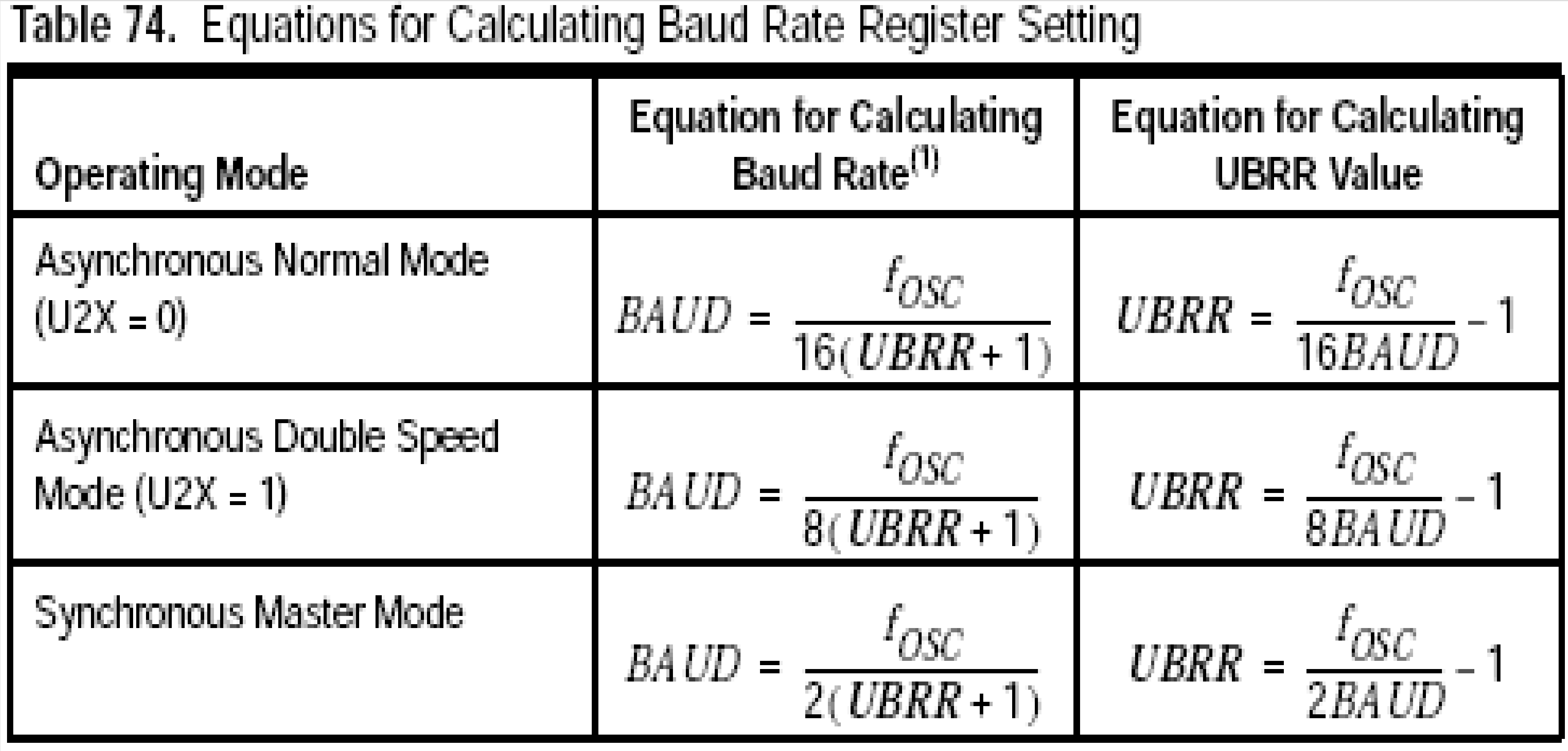
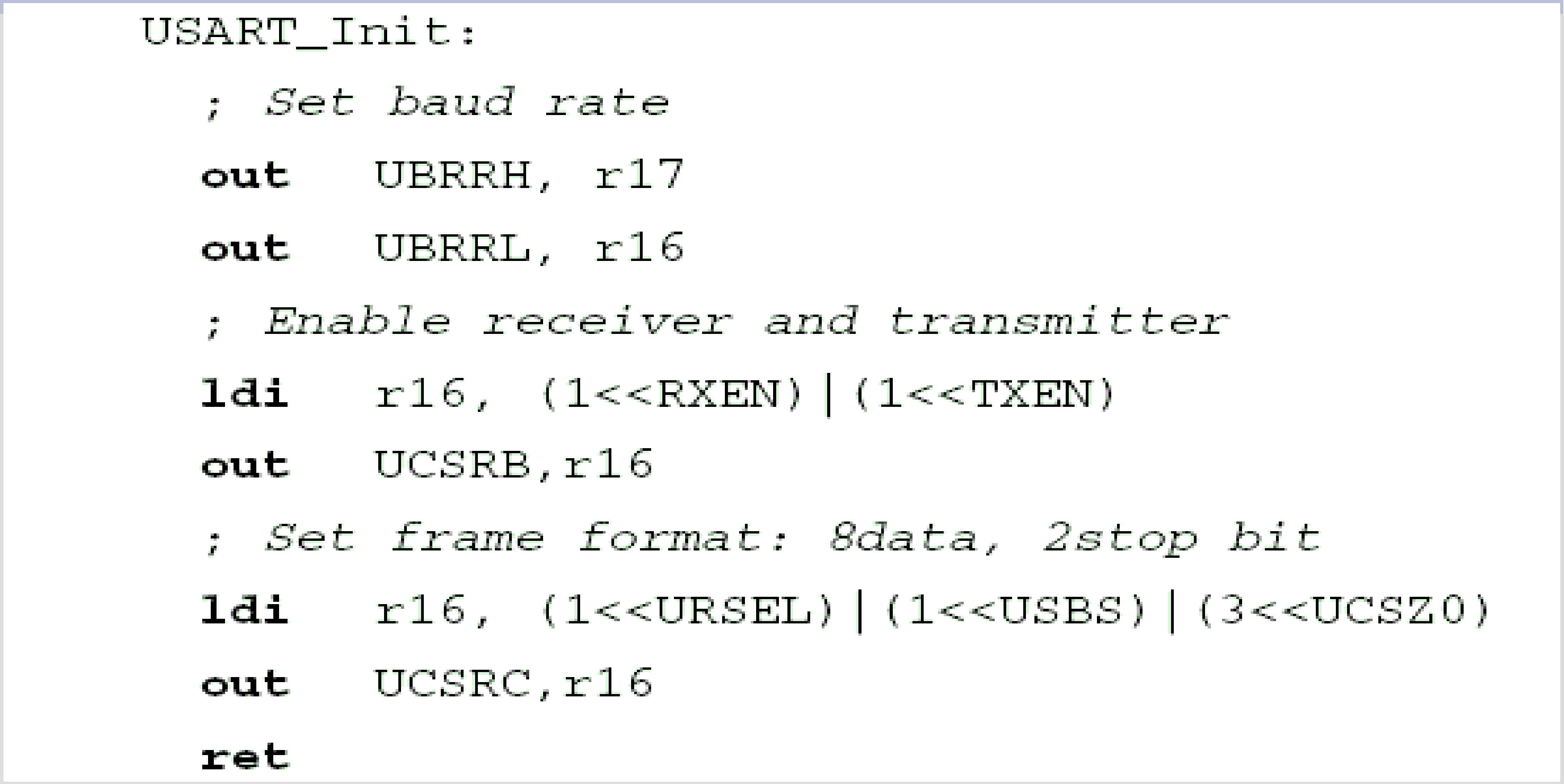


Table 74. Equations for Calculating Baud Rate Register Setting

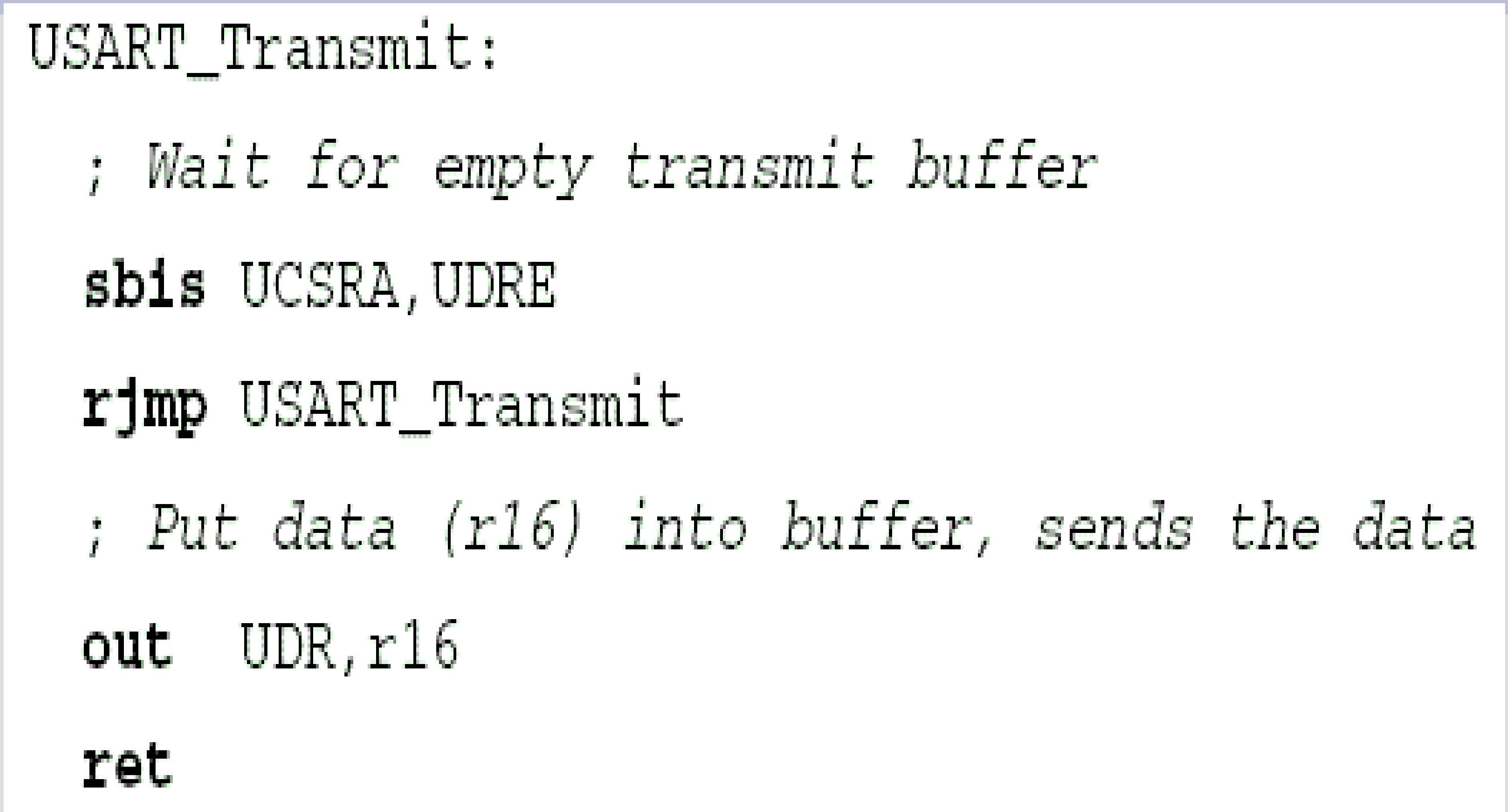
Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Uart Init



```
USART_Init:
  ; Set baud rate
  out   UBRRH, r17
  out   UBRRL, r16
  ; Enable receiver and transmitter
  ldi   r16, (1<<RXEN)|(1<<TXEN)
  out   UCSRB,r16
  ; Set frame format: 8data, 2stop bit
  ldi   r16, (1<<URSEL)|(1<<USBS)|(3<<UCSZ0)
  out   UCSRC,r16
  ret
```

Transmit function



```
USART_Transmit:
  ; Wait for empty transmit buffer
  sbis UCSRA,UDRE
  rjmp USART_Transmit
  ; Put data (r16) into buffer, sends the data
  out  UDR,r16
  ret
```

Refer>>>

page no (for usart registers) 153 – 159
page no 133 -159