**Features**

* 16 byte receive and transmit FIFO
* high baud rates
* use RXD0 and TXD0 pins

**Register Description**

**UART0 Receiver Buffer Register (U0RBR - 0xE000 C000)**

* top byte of UART0 Rx FIFO
* contains oldest character received
* Divisor Latch Access bit(DLAB) in U0LCR must be zero to access
* first read U0LSR and then U0RBR

**UART0 Transmit Holding Register (U0THR- 0xE000 C000)**

* top byte of UART0 TX FIFO
* newest character in the TX FIFO
* Divisor Latch Access bit(DLAB) in U0LCR must be zero to access
* Writing into it causes data to be stored in UART0 transmit FIFO

**UART0 Divisor Latch Registers (U0DLL - 0xE000 C000) and (U0DLM -**

**0xE000 C004)**

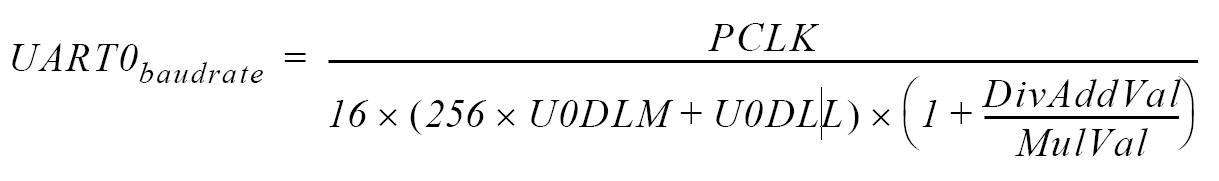
* **U0DLL** and **U0DLM** both form 16 bit divisor
* part of the UART0 Fractional Baud Rate generator
* holds the value used to dive the clock supplied by factional prescalar to produce baud rate clock
* Divisor Latch Access bit(DLAB) in U0LCR must be one to access

**UART0 Fractional Divider Register (U0FDR - 0xE000 C028)**

* contains clock pre-scalar for baud rate generation
* pre-scalar takes APB clock and generates an output clock

|  |  |  |
| --- | --- | --- |
| **Pins** | **Function** | **Description** |
| **3:0** | **DIVADDVAL** | Baud rate generation prescalar divisor value.   * If this is 0, then fractional baud rate generator will not impact the UART0 Baud rate |
| **7:4** | **MULVAL** | Baud rate generation prescalar multiplier value.   * value must be one or above |
| **31:8** | **-** | Reserved |

**Baud Rate Formula**



* **U0DLM** and **U0DLL** are standard UART0 baud rate divider register
* **DIVADDVAL** and **MULVAL** are fractional baud rate parameters
* **MULVAL**  should be greater than 0 but less than or equal to 15
* **DIVADDVAL** should be greater than or equal to 0 but less than or equal to 15
* If **DIVADDVAL** is 0, then fractional baud rate is disabled.

**UART0 Interrupt Enable Register (U0IER - 0xE000 C004)**

* enable UART0 interrupt sources

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **0** | **RBR Interrupt Enable** | Enables Receive data available interrupt for UART0 |
| **1** | **THRE Interrupt Enable** | THRE interrupt for UART0 |
| **2** | **RX Line Status Interrupt Enable** | Enables the UART0 RX line status interrupt |
| **7:4** | **-** | Reserved |
| **8** | **ABEOIntEn** | Enables the end of auto-baud interrupt |
| **9** | **ABTOIntEn** | Enables the auto-baud time-out interrupt |
| **31:10** | **-** | Reserved |

**UART0 Interrupt Identification Register (U0IIR - 0xE000 C008)**

* provides a status code denoting priority and source of pending interrupt
* interrupts occurring during U0IIR access are recorded for next U0IIR access

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Symbol** | **Description** | |
| **0** | **Interrupt Pending** | 0 represents interrupt is pending | |
| **3:1** | **Interrupt Identification** | Identifies an interrupt corresponding to UART0 RX FIFO | |
| 011 | 1 – Receive Line Status(RLS) |
| 010 | 2a – Receive Data Available(RDA) |
| 110 | 2b – Character Time-Out Indicator(CTI) |
| 001 | 3 – THRE Interrupt |
| **5:4** | **-** | Reserved | |
| **7:6** | **FIFO Enable** | FIFO enable indication | |
| **8** | **ABEOInt** | End of auto-baud interrupt. True if auto-baud has finished successfully and interrupt is enabled | |
| **9** | **ABTOInt** | Time-out of auto-baud interrupt. True if auto-baud has timed-out and interrupt is enabled | |
| **31:10** | **-** | Reserved | |

* The Interrupt Service routine determine the cause of interrupt and clear active interrupt by reading U0IIR’s U0IIR[3:0]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **U0IIR[3:0]** | **Priority** | **Interrupt type** | **Interrupt Source** | **Interrupt Cleared by** | **Description** |
| **xxx1** | **-** | **None** | **None** | **-** | Since UIIR[0] is 1, no pending interrupt |
| **0110** | **Highest** | **RX Line Status / Error** | **OE, PE,FE,BI** | **Reading U0LSR** | Set when Overrun error, Parity error, framing error, Break interrupt occurs.  Can be observed via U0LSR[4:1] |
| **0100** | **Second Level** | **RX Data Available** | **Rx data available or trigger level reached in FIFO** | **UART0 RX FIFO depth drops below trigger level**  **Or U0RBR read** | Activated when UART0 RX FIFO reaches trigger level defined in U0FCR[7:6] |
| **1100** | **Second** | **Character Time-out Indication** |  | **U0RBR read** |  |
| **0010** | **Third** | **THRE** | **THRE** |  | Activated when UART0 THR FIFO is empty |

**UART0 FIFO Control Register (U0FCR - 0xE000 C008)**

* Controls the operation of UART0 RX and TX FIFO

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **0** | **FIFO Enable** | Enable both UART0 TX and RX FIFO and U0FCR[7:1] access |
| **1** | **RX FIFO Reset** | Clear all bytes in UART0 RX FIFO and reset pointer logic |
| **2** | **TX FIFO Reset** | Clear all bytes in UART0 TX FIFO and reset pointer logic |
| **5:3** | **-** | Reserved |
| **7:6** | **RX Trigger Level** | Determine how many receiver UART0 FIFO character must be written before and interrupt is activated.  00 – Trigger Level 0(1 character)  01 – Trigger Level 1(4 character)  10 – Trigger Level 2(8 character)  11 – Trigger Level 3(14 character) |

**UART0 Line Control Register (U0LCR - 0xE000 C00C)**

* controls format of data characters in UART0

|  |  |  |  |
| --- | --- | --- | --- |
| **Bits** | **Symbols** | **Value** | **Description** |
| **1:0** | **Word Length select** | **00** | 5 bit character length |
| **01** | 6 bit character length |
| **10** | 7 bit character length |
| **11** | 8 bit character length |
| **2** | **Stop Bit Select** | **0** | 1 stop bit |
| **1** | 2 stop bit |
| **3** | **Parity Enable** | **0** | Disable Parity generation and checking |
| **1** | Enable Parity generation and checking |
| **5:4** | **Parity Select** | **00** | Odd Parity |
| **01** | Even Parity |
| **10** | Forced 1 stick parity |
| **11** | Forced 0 stick parity |
| **6** | **Bread Control** | **0** | Disable break transmission |
| **1** | Enable break transmission – UART0 TXD line is forced to 0 |
| **7** | **Divisor Latch Access Bit(DLAB)** | **0** | Disable access to Divisor Laches |
| **1** | Enable access to Divisor Laches |

**UART0 Transmit Enable Register (U0TER - 0xE000 C030)**

* TXEn = 1, transmits as long as data is availaibale

|  |  |
| --- | --- |
| **7 : TXEN** | **6:0 Reserved** |

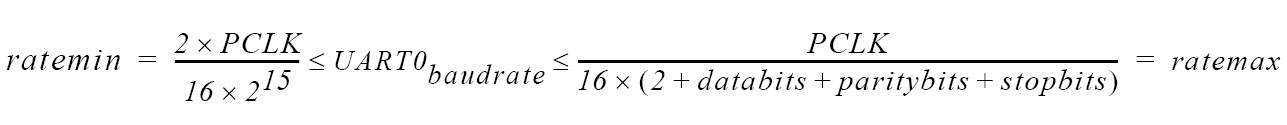
**UART0 Line Status Register (U0LSR - 0xE000 C014)**

* status information on UART0 TX and RX blocks

|  |  |  |
| --- | --- | --- |
| **Bits** | **Symbol** | **Description** |
| **0** | **Receiver Data Ready(RDR)** | Is set when U0RBR holds an unread character and is cleared when UART0 RBR FIFO is empty |
| **1** | **Overrun Error(OE)** |  |
| **2** | **Parity Error(PE)** |  |
| **3** | **Framing Error(Fe)** |  |
| **4** | **Break Interrupt(BI)** |  |
| **5** | **Transmitter Holding Register Empty(THRE)** | Empty UART0 THR |
| **6** | **Transmitter empty(TEMT)** | When both U0THR and U0TSR are empty |
| **7** | **Error in RX FIFO(RXFE)** |  |

**Auto-Baud**

* measure the incoming baud-rate based on AT protocol
* measures the bit time of receive data and sets the divisor latch register U0DLM and U0DLL
* started by setting on U0ACR Start bit and stopped by clearing U0ACR Start bit
* Also, the U0ACR Start bit is cleared once auto-baud has finished and reading the bit and return the status of auto-baud
* Two auto-baud modes by U0ACR Mode bit
  + In mode 0, the baud rate is measured on two subsequent falling edge of the UART0 RX pin (i.e. the falling edge of the start bit and the falling edge of LSB bit).
  + In mode 1, the baud rate is measured between falling edge and the subsequent rising edge on the UART0 RX pin (i.e. the falling edge of the start bit and the rising edge of start bit – which the length of the start bit).
* The U0ACR AutoRestart bit can be used automatically restart the baud-rate measurement if time-out occurs – when the rate measurement counter overflows.
  + Will restart at the next falling edge of UART0 RX pin
* auto-baud rate function generates two interrupts
  + U0IIR ABTOInt Interrupt if the auto-baud rate measurement counter overflows.
  + U0IIR ABEOInt Interrupt if the auto-baud had completed successfully.
  + Interrupts can be enabled by setting U0IER registers ABTOIntEn bit and ABEOIntEn bit.
  + Interrupts can be cleared by setting U0ACR registers ABTOIntClr bit and ABEOIntClr bit.
* The minimum and maximum baud rate supported depends on PCLK, no of data bits, stop-bits and parity bits.



**Steps – for UART0 – rx and tx without Interupts**

1. First, the corresponding pins for RXD and TXD and made to their particular function by selecting appropriate values in **PINSEL0** register.
2. The value of **U0DLM, U0DLL** and **U0FDR** are calculated using the formula.
3. Next, **DLAB** bit in **U0LCR** register is set to write into **U0DLM, U0DLL** and **U0FDR** registers.
4. The data size, stop bit, parity, etc. are configured in **U0LCR** register.
5. Finally, the **DLAB** bit in **U0LCR** register is cleared to lock the values of divider and multiplier values.

**Steps – AutoBaud:**

* **Not Know so far**