**Features**

* byte oriented and four mode operation – master transmitter mode, master receiver mode, slave transmitter mode and slave receiver mode
* Same as studied in ARM processor
* there are I2C0, I2C1, I2C2
* I2C0 supports entire I2C specification with Fast-Mode

**Master Transmitter mode**

* Before entering the master transmitter mode, the **I2CnCONSET**must be as follows.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7** | **6 - I2EN** | **5 – STA** | **4 – STO** | **3 – SI** | **2 – AA** | **1:0** |
| **-** | **1** | **0** | **0** | **0** | **0** | **-** |

* SI Bit is cleared by wiring 1 to SIC bit in **I2CCONCLR**.
* **see the datasheet for operation – same as AVR**

**Register Description**

**I2C Control Set Register (I2C[0/1]CONSET: 0xE001 C000, 0xE005 0000)**

* controls setting of bits in I2CON register controlling the operation of I2C.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7-Reserved** | **6-I2EC** | **5-STA** | **4-STO** | **3-SI** | **2-AA** | **1:0-Reserved** |

**I2EN – I2C Interface Enable**

* **1 –** enabling the I2C
* Cleared by writing 1 to I2ENC bit in I2CONCLR register.

**STA – I2C START Flag**

* **1 –** Enter master mode and transmit a START condition or repeated START condition if already in master mode.
* Cleared by writing 1 to STAC bit in I2CONCLR register.

**STA – I2C STOP Flag**

* **1 -** Transmit a STOP condition if in master mode.
* cleared automatically by hardware

**SI – I2C Interrupt Flag**

* Is set when I2C state changes
* Causes the low period of SCL line to be stretched and serial transfer is suspended.
* SI must be reset by software by writing to SIC bit in I2CONCLR register.

**AA – Assert Acknowledge Flag**

* **1 –** causes to send an acknowledgment during acknowledge clock pulse.
* can be cleared by writing 1 to AAC bit in I2CONCLR register.

**I2C Control Clear Register (I2C [0/1] CONCLR: 0xE001 C018, 0xE005 0018)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7-Reserved** | **6-I2ENC** | **5-STAC** | **4 - Reserved** | **3-SIC** | **2-AAC** | **1:0-Reserved** |

**I2C Status Register (I2C[0/1]STAT - 0xE001 C004, 0xE005 C004)**

* gives the status of the I2C interface
* only 7:3 bits are used

**I2C Data Register (I2C[0/1]DAT - 0xE001 C008, 0xE005 C008)**

* 8-bit register containing the data transmitted and received

**I2C Slave Address Register (I2C[0/1]ADR - 0xE001 C00C,0xE005 C00C)**

* only when I2C is used in slave mode.
* LSB bit is general call mode
* 7:1 bit is I2C slave address

**I2C SCL High Duty Cycle Register (I2C[0/1]SCLH - 0xE001 C010,**

**0xE005 C010)**

* **16 bit register**
* names as SCL High time period selection
* I2CSCLH – number of PCLK cycles for SCL high time
* must be greater than or equal to 4

**I2C SCL Low Duty Cycle Register (I2C[0/1]SCLL - 0xE001 C014,**

**0xE005 C014)**

* **16 bit register**
* names as SCL Low time period selection
* I2CSCLL – number of PCLK cycles for SCL low time
* must be greater than or equal to 4

**I2C Date rate and duty cycle**

* Software must set the values of I2CSCLH and I2CSCLL.
* Both I2CSCLH and I2CSCLL should not have same value.
* Formula:



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **I2SCCL+**  **I2SCCH** | **PCLK = 1M** | **PCLK = 5M** | **PCLK = 10M** | **PCLK = 16M** | **PCLK = 20M** | **PCLK = 40M** | **PCLK = 60M** |
| **I2C Frequency(kHz)** | | | | | | |
| **8** | **125** |  |  |  |  |  |  |
| **10** | **100** |  |  |  |  |  |  |
| **25** | **40** | **200** | **400** |  |  |  |  |
| **50** | **20** | **100** | **200** | **320** | **400** |  |  |
| **100** | **10** | **50** | **100** | **160** | **200** | **400** |  |
| **160** | **6.25** | **31.25** | **62.5** | **100** | **125** | **250** | **375** |
| **200** | **5** | **25** | **50** | **80** | **100** | **200** | **300** |
| **400** | **2.5** | **12.5** | **25** | **40** | **50** | **100** | **150** |
| **800** | **1.25** | **6.25** | **12.5** | **20** | **25** | **50** | **75** |

**Master Transmitter Mode**

1. **I2CONSET** must be initialized as

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7** | **6 - I2EN** | **5 – STA** | **4 – STO** | **3 – SI** | **2 – AA** | **1:0** |
| **-** | **1** | **0** | **0** | **0** | **x** | **-** |

1. **I2C** rate must be configured in I2CSLL and I2CSLH registers
2. **Master Transmitter** mode is entered by setting STA bit to transmit the START condition
   * When start condition is transmitted, the ***SI*** bit is set and status code would be 0x08.
3. Next, the SLA+W is written into I2DAT register.
   * This is sent by resetting the ***SI*** bit by writing 1 in ***SIC*** bit.
   * Now, an acknowledgement bit is received and ***SI*** bit is set by hardware.
   * Different status codes are available in I2STAT – 0x18, 0x20, and 0x38.
4. Next, depending on the I2STAT from previous step, we can send data by writing data into I2DAT register.
   * This is sent by resetting the ***SI*** bit by writing 1 in ***SIC*** bit.
   * Now, an acknowledgement bit is received and ***SI*** bit is set by hardware.
   * Different status codes are available in I2STAT for dat transmission completion.

**Master Receiver Mode**

1. **I2CON** must be initialized as

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7** | **6 - I2EN** | **5 – STA** | **4 – STO** | **3 – SI** | **2 – AA** | **1:0** |
| **-** | **1** | **0** | **0** | **0** | **x** | **-** |

1. **I2C** rate must be configured in I2CSLL and I2CSLH registers
2. **Master Receiver** mode is entered by setting STA bit to transmit the START condition
   * When start condition is transmitted, the ***SI*** bit is set and status code would be 0x08.
3. Next, the SLA+R is written into I2DAT register.
   * This is sent by resetting the ***SI*** bit by writing 1 in ***SIC*** bit.
   * Now, an acknowledgement bit is received and ***SI*** bit is set by hardware.
   * Different status codes are available in I2STAT – 0x40, 048, and 0x38.
4. Next, depending on the I2STAT from previous step, we can send data by writing data into I2DAT register.
   * This is sent by resetting the ***SI*** bit by writing 1 in ***SIC*** bit.
   * Now, an acknowledgement bit is received and ***SI*** bit is set by hardware.
   * Different status codes are available in I2STAT for dat transmission completion.
   * we can send acknowledgement by setting AA bit.

**Slave Receiver Mode**

1. **I2CADR** is set with address.
2. **I2C0CONSET** is as follows

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7** | **6 - I2EN** | **5 – STA** | **4 – STO** | **3 – SI** | **2 – AA** | **1:0** |
| **-** | **1** | **0** | **0** | **0** | **1** | **-** |

1. **I2C** rate need not be set.
2. After addressed by master by its address, SI bit is set and valid code is available in I2CSTAT – 0x60 is received.
3. Now, data can be received and ACK will be sent to get status code of 0x80.