INTERFACING

I2C

AMIT

Internal Integrated Circuit:

Specs of IIC:

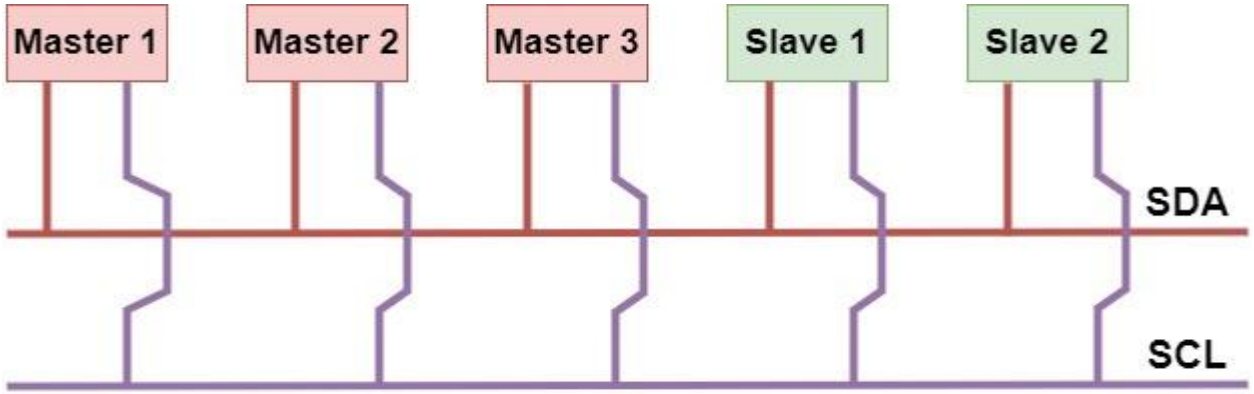
- The Two-wire Serial Interface (TWI) is ideally suited for typical microcontroller applications. TWI is same IIC with limited features.
  - Wired.
  - Serial.
  - Multi Master Multi Slave "**MMMS**".
  - Synchronies & Half Duplex.
  - Min Throughput = 44.44 %.
  - Open Drain circuit.

AMIT

Internal Integrated Circuit:

Hardware Connection of IIC:

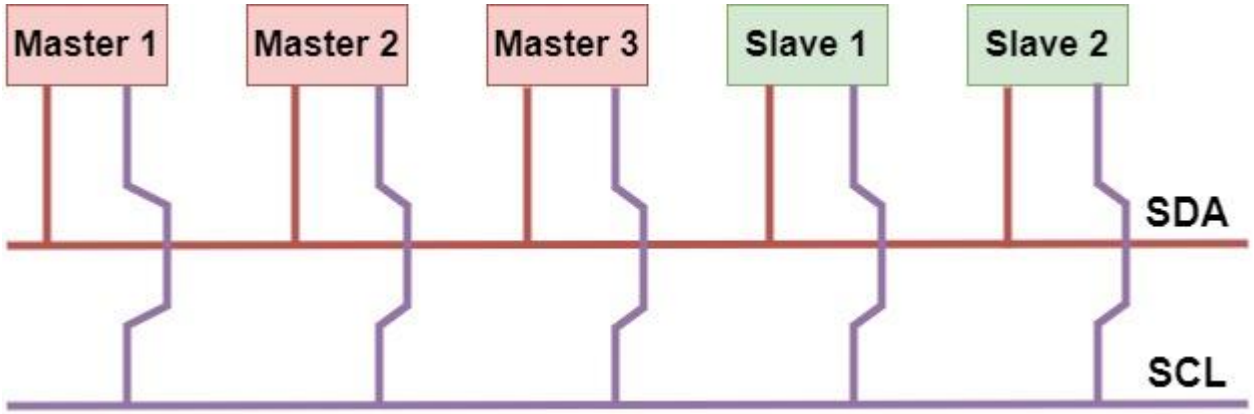
- The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA).
- Because it is a MMMS protocol, so, any master can start the communication at any time, so assume that Master 1 wants to send “0x38” to Slave 1, and Master 2 wants to send “0x23” to Master 3 and both of them have the same clock and they generate it at the same time, “master can send to other master it is valid, because when any master starts the protocol, the remaining node turns to slaves temporarily” and IIC sends the MSB first only, so, M1 will send “0” same with M2, then M1 will send “0” and same M2, and M1 will send “1” and same M2, but what will happen at sending 4th bit?



Internal Integrated Circuit:

Hardware Connection of IIC:

- When the fourth bit is written from Master 1 and Master 2, M1 will write “1” and M2 will write “0”, High and low signal will appears at the same time as the same bus so it will cause a short circuit.



- It is a mandatory to ignore the effect of one of them.

- To ignore the effect of one of these signals, it is a mandatory to use the open drain circuit.

AMIT

Internal Integrated Circuit:

Open-Drain Circuit:

- It looks like the open collector circuit, we mentioned it at Stepper motor, the only difference is that the open collector circuit uses **BJT** transistor, but the open-drain circuit uses **MOSFET** transistors.

- NOT Gate also is used to make the output more logic.

Truth table without NOT Gate

B	C
0	float
1	0

Truth table with NOT Gate

B	C
0	0
1	float

AMIT

Internal Integrated Circuit:

Open-Drain Circuit:

- the float is not a signal, so the two buses of IIC must be Pulled-Up to become one, so according to the previous example when M2 writes “0” and M1 writes “1”, the “0” signal only will be appeared because of the pull-up resistor.

B	C
0	0
1	1

- According to the previous, we will find that there is a bit can blur the effect of the second,
  so we have two concepts:
  - **Dominant Bit**:
    it is the bit ones it is writes to the bus, it clears and blurs the effect of other bit, ZERO is the Dominant in IIC.
  - **Recessive Bit**:
    it is the bit which is cleared and blurred if a Dominant bit is written, ONE is the recessive bit in IIC.

AMIT

Internal Integrated Circuit:

Data Frame of IIC:

- IIC is an address oriented to select the desired node because its networks has several nodes.
- The idle state of IIC is high because the bus is pulled up and the dominant bit is zero, so the start of frame must trigger the network by a bit influences the bus.
- The frame consists of:
  - Start Condition:
    it is called by condition not bit because it is time is very less than a time of bit to avoid the bus holding, its time about 50 ns to be bigger than the spikes to ensure that it is a start of frame not a noise.

Internal Integrated Circuit:

Data Frame of IIC:

- The frame consists of:
  - Slave Address:

    The next byte sent after the start condition is considers as slave address and the operation.

    The most significant seven bits is considered as the address of desired slave, so the network can contain about 128 slaves.

    The least significant bit indicates to the type of operation on the slave, "**0**" to write, "**1**" to read from the salve.

Internal Integrated Circuit:

Data Frame of IIC:

- The frame consists of:
  - Acknowledge:
    it is the ninth bit that the slave itself writes on the bus to indicates that it listened its address, also at this bit the slave can write zero on the clock bus to stretch it until acknowledge replying.
  - Data:
    It is the byte after SLA+R/W byte, master writes it if the operation is “write”, the salve write it if the operation is “read”, the acknowledge after it is the same before, but if the operation is “read” the master itself writes

Internal Integrated Circuit:

Data Frame of IIC:

- The frame consists of:
  - Stop Condition:

    it indicates to the end of frame, and when it is detected on the bus, any other master can start its own protocol.

    The frame does not carry only byte of data, any master can send any quantity of bytes into the same frame, and the master can change the slave address or the operation or both also in the same frame by sending a repeated start condition within the frame, finally after ending, the master can write the stop condition, so as we mentioned, the max throughput of IIC can not be calculated.

AMIT

Internal Integrated Circuit:

Arbitration in IIC:

- As we mentioned, it is possible that more than master starts the protocol any time, also we explained that only one master must complete the protocol and the remaining will wait until the stop condition, so arbitration circuit is a mandatory to use.

- Arbitration circuit works as the following: the first master write the dominant bit first, it wins and completes the protocol, the others will wait until the stop condition.

AMIT

Internal Integrated Circuit:

States of Arbitration in IIC:

- *Master 1 wants to communicate with "0x29" Slave, Master 2 wants to communicate with "0x18" Slave*:
  Master 2 will win the arbitration because it writes the first dominant bit, "0x18 → 0b00011000", but "0x29 → 0b00111001".

- *Master 1 & 2 want to communicate with the same Slave, but Master 1 wants to Write, Master 2 wants to read*:
  Master 1 will win the arbitration because it writes the first dominant bit, "writing operation means that the LSB is Zero".

AMIT

Internal Integrated Circuit:

States of Arbitration in IIC:

- *Master 1 & 2 want to communicate with the same Slave with same operation, but Master 1 wants to Write "0x41", Master 2 wants to write "0x42"*:
  Master 1 will win the arbitration because it writes the first dominant bit, "0x41 → 0b01000001", but "0x42 → 0b01000010".

- *Master 1 & 2 want to communicate with the same Slave, with same operation and same data, and both will stop the frame*:
  Both Masters tie the arbitration because they write or read the same data from the same slave without any arbitration lost.

AMIT

Internal Integrated Circuit:

States of Arbitration in IIC:

- *Master 1 & 2 want to communicate with the same Slave, with same operation and same data, but Master 1 will stop the frame, Master 2 will complete*:
  None will win or lose the arbitration because at first they write or read the same data from the same slave without any arbitration lost, then Master 1 withdraws without any arbitration lost, and Master 2 will complete without any issues.

- Finally, the arbitration circuit is used to avoid more than one master to communicate at the same time, it is a mandatory into any multi-master network.

- The arbitration sentences the master which writes the first dominant bit to win no matter what the place of arbitration “Slave address or operation or data or completing the frame”

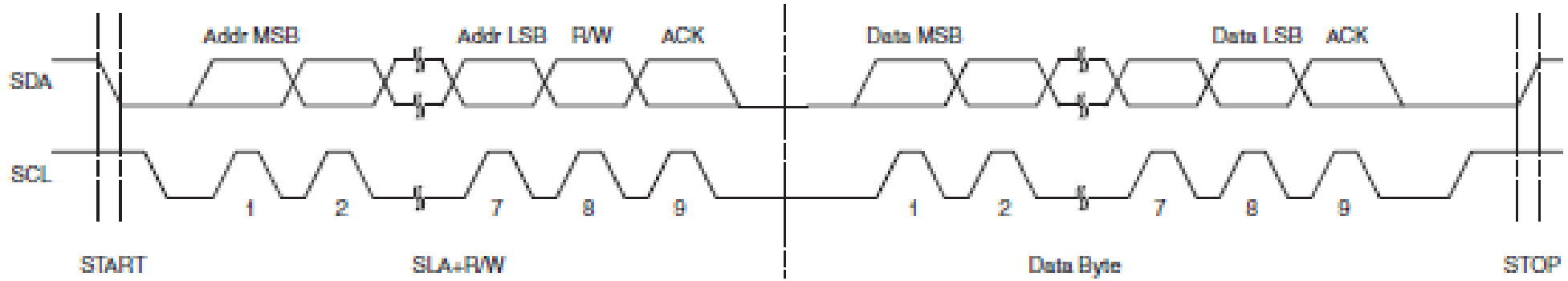
AMIT

Internal Integrated Circuit:

Typical Data Transmission in IIC:

- In the beginning of frame, the start bit is written when the clock is high, to differ from the writing of data, as the data is written when the clock is low, the same when the stop condition is written, it is written when the clock is high.

- The node knows if it loses the arbitration because, the master writes the bit when the clock is low, then it reads the bus when the clock is high to ensure that the written bit appears on the bus, if not, the master will realizes that the bus is busy and it loses the arbitration.



Internal Integrated Circuit:

Register Description in IIC:

- TWI Bit Rate Register "**TWBR**": this register is used to store a value that is used as parameter to calculate the frequency of Clock Bus of IIC

TWI Bit Rate Register – TWBR

Bit	7	6	5	4	3	2	1	0
	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0
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

- To calculate the frequency of the clock of IIC bus, which it is recommended not to be less than 100KHz and it must not exceed 400KHz at normal speed mode.
-

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot 4^{TWPS}}$$

- TWBR is calculated after assuming the IIC frequency as we mentioned, and the value of prescaler of IIC "**TWPS**".

AMIT

Internal Integrated Circuit:

Law of TWBR in IIC:

$$\text{TWBR value} = \frac{System\ Frequency\ - (16 \times IIC\ frequency)}{2 \times IIC\ frequency \times prescaler}$$

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 7 – TWINT: TWI Interrupt Flag:

TWI Control Register – TWCR

Bit	7	6	5	4	3	2	1	0
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W
Initial Value	0	0	0	0	0	0	0	0

This bit is set by hardware when the TWI has finished its current job and expects application software response, means it is set every operation which is occurred into the Two Wired Interface.

If the I-bit in SREG and TWIE in TWCR are set, the MCU will jump to the TWI Interrupt Vector. While the TWINT Flag is set, the SCL low period is stretched.

The TWINT Flag must be cleared by software by writing a logic one to it. Note that this flag is not automatically cleared by hardware

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 7 – TWINT: TWI Interrupt Flag:

The TWINT Flag must be cleared by software by writing a logic one to it.

TWI Control Register – TWCR

Bit	7	6	5	4	3	2	1	0
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W
Initial Value	0	0	0	0	0	0	0	0

Note that this flag is not automatically cleared by hardware when executing the interrupt routine.

Also note that clearing this flag starts the operation of the TWI, so all accesses to the TWI Address Register (TWAR), TWI Status Register (TWSR), and TWI Data Register (TWDR) must be complete before clearing this flag.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 6 – TWEA: TWI Enable Acknowledge Bit:

    The TWEA bit controls the generation of the acknowledge pulse. If the TWEA bit is written to one, the ACK pulse is generated on the TWI bus if the following conditions are met:

    1. The device's own slave address has been received.
    2. A general call has been received, while the TWGCE bit in the TWAR is set.
    3. A data byte has been received in Master Receiver or Slave Receiver mode.

    By writing the TWEA bit to zero, the device can be virtually disconnected from the Two-wire Serial Bus temporarily, so it must be set to be active into the network again.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 5 – TWSTA: TWI START Condition Bit:

    The application writes the TWSTA bit to one when it desires to become a master on the Two wire Serial Bus.

    The TWI hardware checks if the bus is available, and generates a START condition on the bus if it is free.

    However, if the bus is not free, the TWI waits until a STOP condition is detected, and then generates a new START condition to claim the bus Master status.

    TWSTA must be cleared by software when the START condition has been transmitted because it must be always set to detect if the bus is free or wait until it becomes free.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 4 – TWSTO: TWI STOP Condition Bit:

    Writing the TWSTO bit to one in Master mode will generate a STOP condition on the Two-wire Serial Bus.

    When the STOP condition is executed on the bus, the TWSTO bit is cleared automatically.

    In slave mode, setting the TWSTO bit can be used to recover from an error condition.

    This will not generate a STOP condition, but the TWI returns to a well-defined unaddressed slave mode and releases the SCL and SDA lines to a high impedance state.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 3 – TWWC: TWI Write Collision Flag:

    The TWWC bit is set when attempting to write to the TWI Data Register – TWDR when TWINT is low, because ,as we mentioned before the TWI will start the operation when TWINT is cleared, so writing any data while TWINT is low means that writing data while transmission operation, so it generates a write-collision error.

    This flag is cleared by writing the TWDR Register when TWINT is high.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 2 – TWEN: TWI Enable Bit:

    The TWEN bit enables TWI operation and activates the TWI interface. When TWEN is written to one, the TWI takes control over the I/O pins connected to the SCL and SDA pins, enabling the slew-rate limiters and spike filters.

    It is the only peripheral that DIO peripheral has no control on the pins, it must have an external-pulled-up registers on SDA and SCL, it does not like UART or SPI or any peripheral exists into atmega32 that the direction of their pins must be configured by using DIO.
    If this bit is written to zero, the TWI is switched off and all TWI transmissions are terminated, regardless of any ongoing operation.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Control Register "**TWCR**":
  - Bit 1 – Reserved Bit:
    This bit is a reserved bit and will always read as zero.

  - Bit 0 – TWIE: TWI Interrupt Enable:
    When this bit is written to one, and the I-bit in SREG is set, the TWI interrupt request will be activated for as long as the TWINT Flag is high.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Status Register "**TWSR**":
  - Bits [7:3] – TWS: TWI Status:

    These five bits reflect the status of the TWI logic and the Two-wire Serial Bus.

TWI Status Register – TWSR

Bit	7	6	5	4	3	2	1	0
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0
Read/Write	R	R	R	R	R	R	R/W	R/W
Initial Value	1	1	1	1	1	0	0	0

The different status codes are described later in this section. Note that the value read from TWSR contains both the 5-bit status value and the 2-bit prescaler value. The application designer should mask the prescaler bits to zero when checking the Status bits by ANDing it with "**0xF8**".

Every operation in TWI, these five bits will be updated according to the operation and the mode of Node.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Status Register "**TWSR**":
  - Bits [7:3] – TWS: TWI Status:

    There are four modes in TWI, there are four tables into the datasheet describe these modes:
    - Master Transmitter "Page 184, Table 74".
    - Master Receiver "Page 187, Table 75".
    - Slave Receiver "Page 191, Table 76".
    - Slave Transmitter "Page 194, Table 77".
    - Miscellaneous States "Page 195, Table 78".

    The most common status codes are used into TWI:
    - Start Condition has been transmitted:

      it equals "**0x08**" if it is a Master transmitter or Receiver.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Status Register "**TWSR**":
  - Bits [7:3] – TWS: TWI Status:

    The most common status codes are used into TWI:
    - Repeated Start Condition has been transmitted:

      it equals "**0x10**" if it is a Master transmitter or Receiver.

    - Slave address with operation with Acknowledge:

      - it equals "**0x18**" if it is a Master transmitter.
      - it equals "**0x40**" if it is a Master Receiver.
      - it equals "**0x60**" if it is a Slave Receiver.
      - it equals "**0xA8**" if it is a Slave transmitter.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Status Register "**TWSR**":
  - Bits [7:3] – TWS: TWI Status:

    The most common status codes are used into TWI:
    - Slave address with operation without Acknowledge:

      - it equals "**0x20**" if it is a Master transmitter.
      - it equals "**0x48**" if it is a Master Receiver.

    - Data has been transmitted with Acknowledge:

      - it equals "**0x28**" if it is a Master transmitter.
      - it equals "**0x50**" if it is a Master Receiver.
      - it equals "**0xB8**" if it is a Slave transmitter.
      - it equals "**0x80**" if it is a Slave Receiver.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Status Register "**TWSR**":
  - Bits [7:3] – TWS: TWI Status:

    The most common status codes are used into TWI:
    - Data has been transmitted without Acknowledge:

      - it equals "**0x30**" if it is a Master transmitter.
      - it equals "**0x58**" if it is a Master Receiver.
      - it equals "**0xC0**" if it is a Slave transmitter.
      - it equals "**0x88**" if it is a Slave Receiver.

    - Arbitration lost in SLA+R/W as a Master:

      - it equals "**0x38**" if it is a Master transmitter.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI Status Register "**TWSR**":
  - Bit 2 – Reserved Bit:
    This bit is reserved and will always read as zero.

  - Bits [1:0] – TWPS: TWI Prescaler Bits:
    These bits can be read and written, and control the bit rate prescaler, they equal the power of 4 that exist into the SCL frequency formula, according to their values, the parameter of 4^{TWPS} will equal the values exist into the following table, according to the value of these bits.

Table 73. TWI Bit Rate Prescaler

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

Internal Integrated Circuit:

Register Description in IIC:

- TWI Data Register "**TWDR**":
  - Bits [7:0] – TWI Data Register:

This register is used to send any data must be written on the data bus, like Slave address and the operation, or any data must be transmitted on the data bus if the node is a master,
and to receive any data which was written on the data bus if the node is a slave.

TWI Data Register – TWDR

Bit	7	6	5	4	3	2	1	0
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	1	1	1	1	1	1	1

In Transmit mode, TWDR contains the next byte to be transmitted.
In Receive mode, the TWDR contains the last byte received.

AMIT

Internal Integrated Circuit:

Register Description in IIC:

- TWI (Slave) Address Register "**TWAR**":
  - Bits [7:1] – TWA: TWI (Slave) Address Register:

    This register is used to save the own address of node itself, when any node exists into the network can call it by this address.

TWI (Slave) Address Register – TWAR

Bit	7	6	5	4	3	2	1	0
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	1	1	1	1	1	1	0

  - Bit 0 – TWGCE: TWI General Call Recognition Enable Bit:

    If set, this bit enables the recognition of a General Call given over the Two-wire Serial Bus.

    The general call can be generated by writing the slave address byte is "**0000 000**".

AMIT

Internal Integrated Circuit:

Sequence Code to Running IIC:

- Send the start condition:

```
TWCR = (1<<TWINT) | (1<<TWSTA) | (1<<TWEN);
```

- Wait for rising TWINT flag:

```
while ( ! ((TWCR >> TWINT) & 1) ) ;
```

- Check the Status Register:

```
if ( (TWSR & 0xF8) == 0x08 )  //complete the frame;
```

- Sending the slave address + operation:

```
TWDR = SLA + operation;
//clear the flag, clear the start condition;
TWCR |= (1<<TWINT);   TWCR &=~(1<<TWSTA);
```

AMIT

Internal Integrated Circuit:

Sequence Code to Running IIC:

- Wait for rising TWINT flag:

```
while ( ! ((TWCR >> TWINT) & 1) ) ;
```

- Check the Status Register:

```
if ( (TWSR & 0xF8) == 0x18 || 0x40 )  //complete the frame;
```

- Sending the data:

```
TWDR = Data;
//clear the flag;
TWCR |= (1<<TWINT);
```

- Wait for rising TWINT flag:

```
while ( ! ((TWCR >> TWINT) & 1) );
```

AMIT

Internal Integrated Circuit:

Sequence Code to Running IIC:

- Check the Status Register:

```
if ( (TWSR & 0xF8) == 0x28 || 0x50 )  //complete the frame;
```

- Send the stop condition if you do not want to send more:

```
TWCR = (1<<TWINT) | (1<<TWSTO) | (1<<TWEN);
```

AMIT

Internal Integrated Circuit:

Demerits of IIC:

- Bus Starvation:

  It occurs because:

  - Master sends infinite data.
  - Master sends lots of repeated start condition.
  - Master does not send a stop condition.

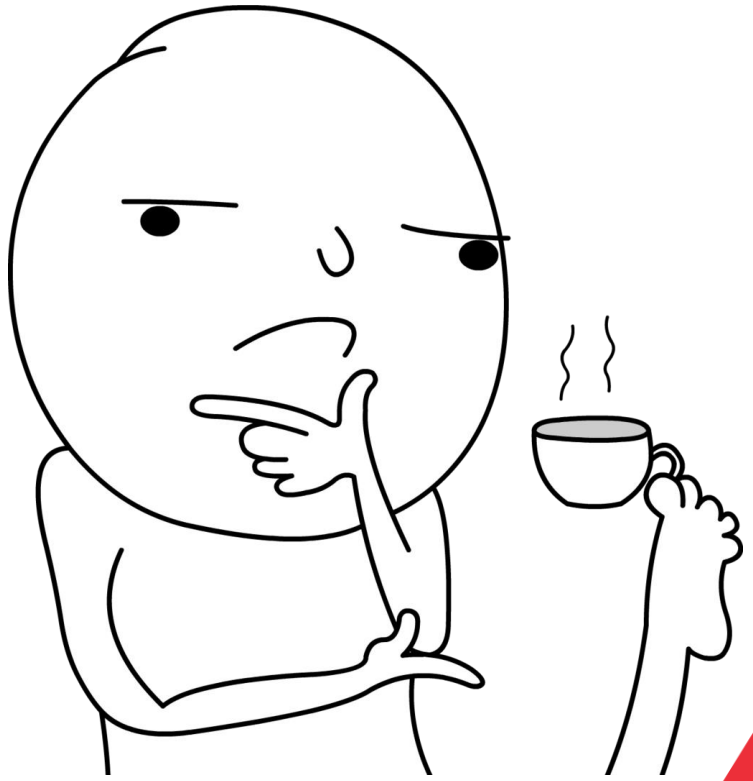
- Clock Stretching:

  It occurs at the ninth bit, because the slave controls the SCL bus at this bit to delay the master from sending more data before it processes the received data, so if the slave does not lay the SCL bus, the network of TWI will fail.

AMIT

IIC Driver:

Time To Code



AMIT

THANK YOU!