## I2C

Master:

1. Initiate data transfer
2. Generate clock pulse to permit data transfer
3. Master Slave relationship is not permanent and depends on the direction of Data transfer.
4. Multi master capable
5. Master can be a transmitter or a receiver
6. Master at all time generates the clock pulse and terminates the operation
7. The clock signal generated by the master during data transfer can be altered when they are stretched by a slow slave device holding the clock down or by another master when arbitration occurs.
8. Clock can be a push pull driver design in case of single master application and no slave holding the clock low. (called no slave stretching the clock.)

Arbitration procedure:

Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the winning message is not corrupted. The procedure depends on the wired AND connection of all the I2C interfaces to the I2C bus.

IF two or more masters try to put information on the bus the first to produce ‘1’ when the other produces a zero loses the bus control. The clocks are synchronized.

Data Transfer

1. 5 speeds are possible from 100 kBits / sec in standard mode to 5 Mbits/sec in ultra fast mode.
2. Data must be stable when the clock is high. The data can only change when the clock signal is low.
3. One clock pulse is generated for each bit of data.
4. Logic levels are 0.3Vdd VIL and 0.7VDD VIH.
5. The number of devices connected to the bus are only limited by the capacitance of the bus.
6. Every byte is 8 bits long.
7. After every byte an ACK is received.
8. Data is transferred from MSB first to LSB.
9. If slave cannot receive or transmit data on request by the master at that instant it pulls the clock low till it is ready.

Start and Stop Conditions:

1. Generated by the master
2. SDA low when SCL high is start
3. SDA high when SCL is high is STOP condition.
4. If a repeated start occurs then the same master gains control of the bus in the next cycle. It can address another slave without giving up the bus.

ACK NACK

1. Receiver to transmitter
2. First acknowledge is always generated by the slave indication correct address has been reached.
3. When the master is in transmit mode all ACKS are generated by the slave.
4. When the Master is in receive mode subsequent acks are generated by the master.
5. Stop condition is generated by the master which sends NACK just before STOP.
6. NACK is also send when the transfer direction is changed, i.e. when master transmitter becomes master receiver then it sends a NACK before the start condition.
7. Signals that the byte was successfully received.
8. Acknowledge is sent in the 9th clock pulse generated by the master.
9. In this case the transmitter releases the SDA line during the 9th clock pulse and waits for the receiver to pull it low and hold it low during high of the 9th SCL pulse.

Reference: NXP, UM10204. "I2C-bus specification and user manual." (2007).

@article{semiconductor2007um10204,

title={UM10204: I2C-bus specification and user manual},

author={Semiconductor, NXP},

journal={URL: https://www. nxp. com/documents/user\% 20manual/UM10204. pdf},

year={2007}

}