

## Komunikasi antar IC dengan IIC

### 1. Latar Belakang dan Konsep IIC

Pada saat ini desain elektronik dituntut untuk semakin ringkas dan fleksibel, dimana ukuran fisik IC semakin diperkecil dan jumlah pin diminimalkan dengan tetap menjaga fleksibilitas dan komabilitas IC sehingga mudah untuk digunakan dalam berbagai keperluan desain yang berbeda, oleh karenanya banyak perusahaan semikonduktor yang berusaha mengembangkan cara baru komunikasi antar IC yang lebih akomodatif terhadap tuntutan diatas sebagai alternatif dari hubungan antar IC secara paralel (parallel bus) yang sudah kita kenal luas. Salah satu metode yang telah matang dan dipakai secara luas adalah IIC (sering ditulis juga I<sup>2</sup>C) singkatan dari *Inter Integrated Circuit bus* yang dikembangkan oleh Philips Semiconductor sejak tahun 1992, dengan konsep dasar komunikasi 2 arah antar IC dan/atau antar sistem secara serial menggunakan 2 kabel.

### 2. Fitur Utama IIC

Fitur utama I<sup>2</sup>C bus adalah sebagai berikut :

- Hanya melibatkan dua kabel yaitu *serial data line* (selanjutnya disebut SDA) dan *serial clock line* (selanjutnya disebut SCL).
- Setiap IC yang terhubung dalam I<sup>2</sup>C memiliki alamat yang unik yang dapat diakses secara *software* dengan *master/slave protocol* yang sederhana, dan mampu mengakomodasikan *multi master* (akan dijelaskan lebih detail pada bagian lain).
- I<sup>2</sup>C merupakan serial bus dengan orientasi data 8 bit (byte), komunikasi 2 arah, dengan kecepatan transfer data sampai 100Kbit/s pada mode standart dan 3,4 Mbit/s pada mode kecepatan tinggi.
- Jumlah IC yang dapat dihubungkan pada I<sup>2</sup>C bus hanya dibatasi oleh beban kapasitansi pada bus yaitu maksimum 400pF.

### 3. Keuntungan I<sup>2</sup>C

Keuntungan yang didapat dengan menggunakan I<sup>2</sup>C antara lain :

- Meminimalkan jalur hubungan antar IC (bandingkan dengan *parallel bus*!).
- Menghemat luasan PCB yang dibutuhkan.
- Membuat sistem yang didesain berorientasi software (mudah diekspan dan diupgrade).
- Membuat sistem yang didesain menjadi standart, sehingga dapat dihubungkan dengan sistem lain yang juga menggunakan I<sup>2</sup>C bus.

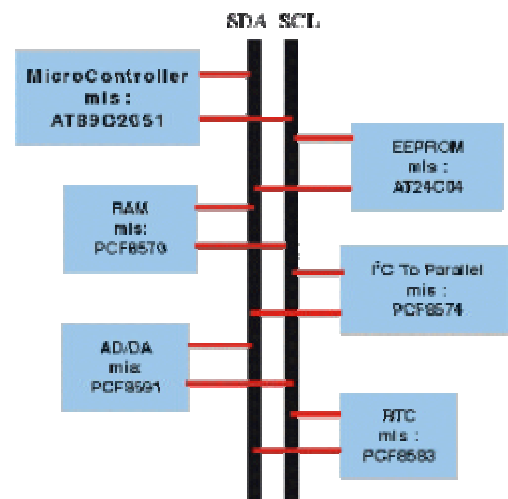
### 4. Cara kerja I<sup>2</sup>C Bus

Sebelum memahami cara kerjanya, ada beberapa terminologi, karakter dan kondisi penting dalam I<sup>2</sup>C yang harus dipahami terlebih dahulu, yaitu :

#### 4.1 Terminologi

- **Transmitter** yaitu *device* yang mengirim data ke bus.
- **Receiver** yaitu *device* yang menerima data dari bus.
- **Master** yaitu *device* yang memiliki inisiatif (memulai dan mengakhiri) transfer data dan yang membangkitkan sinyal clock.

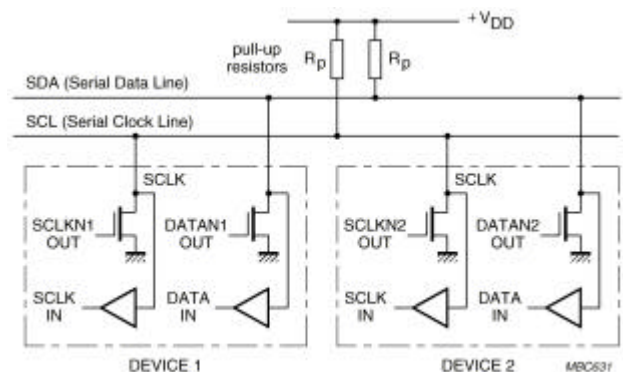
- **Slave** yaitu *device* yang dialamati (diakses berdasarkan alamatnya) oleh Master.
- **Multi-master** yaitu sistem yang memungkinkan lebih dari satu Master melakukan inisiatif transfer data dalam waktu yang bersamaan tanpa terjadi korupsi data.
- **Arbitration** yaitu prosedur yang memastikan bahwa jika ada lebih dari satu Master melakukan inisiatif transfer data secara bersamaan, hanya akan ada satu Master yang diperbolehkan dengan tanpa merusak data yang sedang ditransfer.
- **Synchronization** yaitu prosedur untuk menyelaraskan sinyal clock dari dua atau lebih *device*.



Gambar 1  
Contoh Sistem dengan IIC Bus

#### 4.2 Karakter perangkat keras

- Kedua pin pada I<sup>2</sup>C yaitu SDA dan SCL harus memiliki kemampuan input dan output serta bersifat *open drain* atau *open collector*.
- Kedua pin tersebut terhubung pada I<sup>2</sup>C bus yang telah di pull-up dengan resistor ke suplai positif dari sistem
- Semua *device* yang terhubung pada bus harus terhubung pada ground yang sama sebagai referensi.

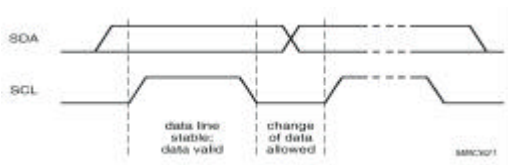


Gambar 2  
Koneksi IIC Bus

## Komunikasi antar IC dengan IIC

### 4.3 Karakter Transfer Data Bit

Data bit dikirim/diterima melalui SDA, sedangkan sinyal clock dikirim/diterima melalui SCL, dimana dalam setiap transfer data bit satu sinyal clock dihasilkan, transfer data bit dianggap valid jika data bit pada SDA tetap stabil selama sinyal clock *high*, data bit hanya boleh berubah jika sinyal clock dalam kondisi *low*, lihat gambar 3.



**Gambar 3**  
Transfer Data Bit Pada I<sup>2</sup>c Bus

### 4.4 Karakter Transfer Data Byte

I<sup>2</sup>C Bus berorientasi pada 8 bit data (byte), dengan *most significant bit* / MSB ditransfer terlebih dulu, serta 2 macam data byte yaitu **Address Byte** dan **Data Byte**.

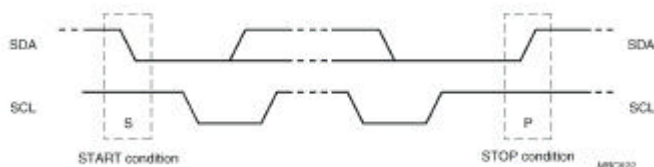
### 4.5 Kondisi

#### 4.5.1 START dan STOP

Apabila pada SDA terjadi transisi dari kondisi *high* ke kondisi *low* pada saat SCL berkondisi *high*, maka terjadilah kondisi START.

Apabila pada SDA terjadi transisi dari kondisi *low* ke kondisi *high* pada saat SCL berkondisi *high*, maka terjadilah kondisi STOP.

Kondisi START dan STOP selalu dibangkitkan oleh Master, dan bus dikatakan sibuk setelah START dan dikatakan bebas setelah STOP.

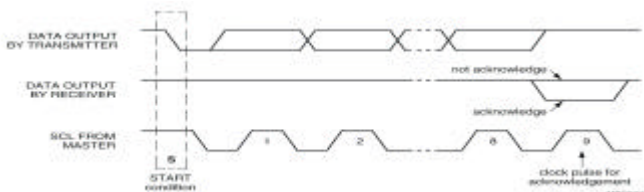


**Gambar 4**  
Kondisi Start dan Stop

#### 4.5.2 ACK dan NACK

Kondisi ACK terjadi apabila receiver “menarik” SDA pada kondisi *low* selama 1 sinyal clock.

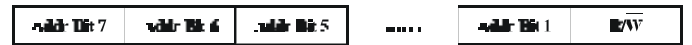
Kondisi NACK terjadi apabila receiver “membebaskan” SDA pada kondisi *high* selama 1 sinyal clock.



**Gambar 5**  
Kondisi Ack dan Nack

### 4.5 Cara kerja I<sup>2</sup>C Bus (Format 7 bit address)

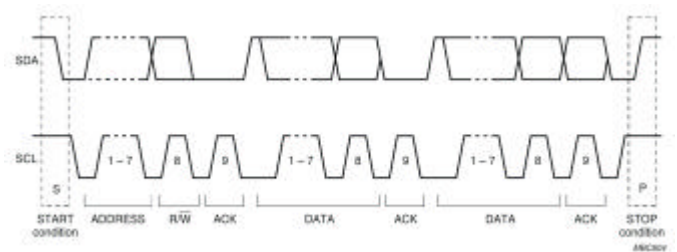
Cara kerja I<sup>2</sup>C bus dapat dibedakan menjadi format 7 bit addressing dan format 10 bit addressing, dalam artikel ini hanya akan dibahas format 7 bit addressing, untuk format 10 bit addressing dapat anda baca keterangan lengkapnya dalam “*The I<sup>2</sup>C-BUS Specification Version 2.1 January 2000*” yang dirilis oleh Philips Semiconductor. Inisiatif komunikasi/transfer data selalu oleh Master dengan mengirimkan kondisi START diikuti dengan address byte (7 bit address + 1 bit pengarah/data direction bit) seperti pada ilustrasi dibawah ini



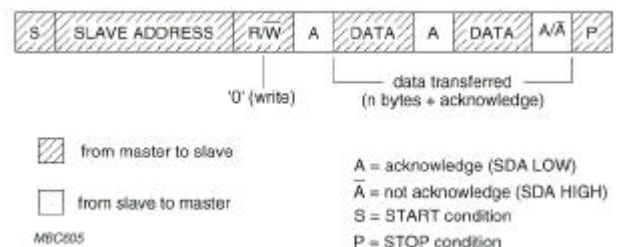
**Gambar 6**  
Format Address Byte

Ada dua jenis komunikasi dasar dalam I<sup>2</sup>C bus yaitu :

- Master-transmitter menulis data ke slave-receiver yang teralamat
- Master-receiver membaca data dari slave-transmitter yang teralamat

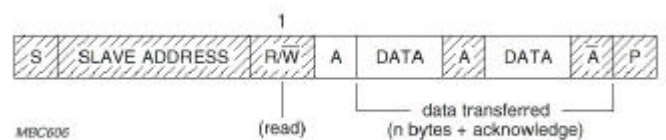


**Gambar 7**  
Transfer Data Lengkap I<sup>2</sup>c Bus



### Gambar 8

**Master-Transmitter Menulis Data Ke Slave-Receiver Yang Teralamat**



### Gambar 9

**Master-Receiver Membaca Data Dari Slave-Transmitter Yang Teralamat**

## Komunikasi antar IC dengan IIC



**Gambar 10**

### **Kombinasi Master-Transmitter dan Master Receiver**

Address byte terdiri dari bagian yang tetap dan bagian yang dapat diprogram, bagian yang tetap merupakan bawaan dari IC, sedangkan yang dapat diprogram biasanya berupa pin address pada IC yang bersangkutan, sebagai contoh IC PCF8591, memiliki address byte sbb : 1 0 0 1 A2 A1 A0, dimana 1001 adalah bagian yang tetap dan A2,A1,A0 adalah bagian yang dapat diprogram sesuai dengan kondisi logika pada pin IC PCF8591.

Sinyal Acknowledge (ACK) terjadi :

#### Dari Slave ke Master Transmitter :

- Sesudah address byte diterima dengan baik oleh slave
- Setiap kali slave selesai menerima data byte dengan baik

#### Dari Master Receiver ke Slave :

- Setiap kali Master selesai menerima data byte dengan baik

Sinyal Negative Acknowledge (NACK) terjadi :

#### Dari Slave ke Master Transmitter :

- Setelah slave gagal menerima address byte dengan baik
- Setiap kali slave gagal menerima data byte dengan baik
- Slave tidak terhubung pada bus

#### Dari Master Receiver ke slave :

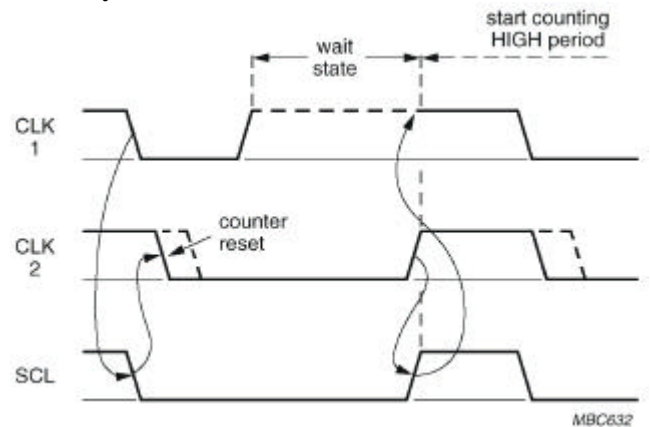
- Setelah Master menerima data byte yang terakhir dari slave

## **5. Multi Master**

Pada I<sup>2</sup>C bus bisa terjadi situasi dimana lebih dari 1 *device* mengambil inisiatif transfer data sebagai Master, dengan protocol Master/slave dan karakter hardware open drain/open collector yang bersifat wired AND, hal ini tidak menyebabkan terjadinya korupsi data, inilah yang disebut dengan Multi Master.

Untuk dapat melakukan Multi Master ada 2 hal yang penting yaitu Clock Synchronization dan Arbitration.

## **5.1 Clock Synchronization**



**Gambar 11**  
**Clock Synchronization**

Karena sifat wired AND dari I<sup>2</sup>C bus, dimana jika salah satu *device* menarik bus dalam kondisi *low* maka *device* lain tidak dapat membuat bus tersebut menjadi *high* (sifat dari logika AND), sehingga jika ada lebih dari satu *device* yang melakukan inisiatif transfer data sebagai Master dengan membangkitkan sinyal clock pada SCL pada saat yang bersamaan harus ada sinkronisasi clock yang dapat dijelaskan (seperti terlihat pada gambar 10 diatas) sebagai berikut :

- Jika Master1 (Clock 1) memulai periode *low* sinyal clock-nya, maka SCL menjadi *low*, Master2 mendeteksi kondisi tersebut dan harus juga memulai menghitung periode *low* sinyal clock-nya.
- Saat Master1 (Clock 1) akan memulai periode *high* sinyal clock-nya dan mendeteksi bahwa SCL masih dalam kondisi *low* (disebabkan periode *low* sinyal clock dari Master 2 (Clock 2) masih belum selesai) maka dia harus menunggu dan tidak menghitung periode *high* sinyal clock-nya terlebih dahulu.
- Saat Master2 (Clock 2) memulai periode *high* sinyal clock-nya, maka kondisi SCL menjadi *high*, Master1 (Clock1) yang mendeteksi kondisi tersebut juga harus memulai menghitung periode *high* sinyal clock-nya.
- Karena Master 1 (Clock1) terlebih dahulu menyelesaikan periode *high* sinyal clock-nya dan memulai periode *low* maka kondisi SCL menjadi *low*, maka Master 2 (Clock 2) yang mendeteksi kondisi tersebut juga harus memulai menghitung periode *low* sinyal clock-nya, demikian seterusnya sehingga terjadilah sinkronisasi sinyal clock antara Master1 dan Master2.

Secara singkat sinkronisasi clock dapat disimpulkan sbb:

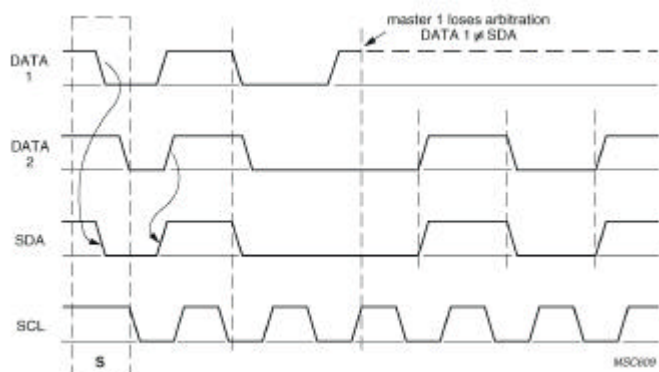
1. Jika kondisi SCL tetap *low* pada saat Master mencoba membuatnya *high*, maka Master tersebut harus memulai menghitung periode *low* sinyal clock-nya
2. Jika Master akan memulai periode *high* sinyal clock-nya, maka Master tersebut harus menunggu kondisi SCL menjadi *high* sebelum memulai menghitung periode *high* sinyal clock-nya.

Sehingga sinkronisasi clock yang terbentuk sbb:

## Komunikasi antar IC dengan IIC

- Periode *low* akan mengikuti periode *low* dari *device* yang membangkitkan sinyal clock dengan periode *low* yang terpanjang.
- Periode *high* akan mengikuti periode *high* dari *device* yang membangkitkan sinyal clock dengan periode *high* yang terpendek.

### 5.2 Arbitration



**Gambar 12**  
**Arbitration**

Dalam Multi Master, bisa terjadi kemungkinan lebih dari satu *device* melakukan inisiatif transfer data menjadi Master, walaupun transfer data hanya bisa dilakukan jika kondisi bus bebas, tetapi sangat memungkinkan lebih dari satu *device* mendeteksi kondisi bus sebagai bebas dan membangkitkan kondisi START sedikit berselisih waktu tetapi masih dalam batas-batas kondisi START yang valid. Untuk kondisi seperti dijelaskan diatas, maka arbitration diberlakukan bit demi bit hingga selesai, dimana sekali lagi sifat/karakter bus yang wired AND memungkinkan hal tersebut terjadi.

Untuk jelasnya perhatikan gambar 11 diatas yang dapat dijelaskan sebagai berikut :

- Ambil contoh Master 1 (Data 1) akan mentransfer data 101xxxxB sedangkan Master 2 (Data 2) akan mentransfer data 100101xxB.
- Kedua Master mendeteksi bus dalam keadaan bebas, dan membangkitkan sinyal START yang hampir bersamaan, Master 1 lebih dahulu membangkitkan START sehingga kondisi SDA mengikuti Master 1, baru kemudian Master 2 membangkitkan START, tetapi kondisi START pada SDA masih valid untuk Master2.
- Kedua Master mentransfer MSbit (sama-sama "1"), kemudian data bit berikutnya (sama-sama "0"), pada bit yang berikutnya Master1 berusaha untuk membuat SDA *high* sesuai dengan data bit-nya, sedangkan Master 2 berusaha untuk membuat SDA *low* (sesuai dengan data bit-nya), karena sifat wired AND dari SDA, maka kondisi SDA menjadi *low*, karena itu Master 1 dikatakan kehilangan arbitrase (dengan kata lain bisa disebut sebagai kehilangan kontrol) atas SDA.
- Bagi Master 1 yang kehilangan arbitrase bisa terus membangkitkan sinyal clock sampai transfer data selesai dan bus dalam kondisi bebas lagi, bagi Master 2 yang memenangkan arbitrase (mendapat kontrol) atas SDA dapat menyelesaikan transfer data-nya tanpa ada data yang terkorupsi sama sekali.

### 6. Kesimpulan

Komunikasi I<sup>2</sup>C dapat memenuhi target desain elektronika saat ini, dengan karakter hardware dan master/slave protocol yang sederhana tetapi tangguh.

Jika digunakan *device* yang telah dilengkapi dengan kemampuan I<sup>2</sup>C secara built-in, akan sangat mudah untuk melaksanakan komunikasi baik single Master maupun multi Master, tetapi bagi *device* yang tidak, masih dapat mengemulasikan I<sup>2</sup>C protocol pada pin-pin I/O-nya asalkan memenuhi syarat karakter hardware dengan cukup mudah untuk single Master tetapi sedikit rumit untuk multi Master.

Artikel ini tidak membahas secara lengkap seluruh kemampuan I<sup>2</sup>C seperti misalnya I<sup>2</sup>C mode kecepatan tinggi, tetapi cukup detail memberikan gambaran tentang I<sup>2</sup>C, untuk mengetahui secara lengkap bisa didapatkan dari referensi untuk artikel ini yang dicantumkan dibawah.

### 7. Referensi

1. THE I<sup>2</sup>C-BUS SPECIFICATION VERSION 2.1 JANUARY 2000, PHILIPS SEMICONDUCTOR
2. APPLICATION NOTE AN435A, PHILIPS SEMICONDUCTOR.
3. APPLICATION NOTES I<sup>2</sup>C SPECIFIC INFORMATION, PHILIPS SEMICONDUCTOR
4. DATA SHEET PCF8591 8-BIT A/D AND D/A CONVERTER, PHILIPS SEMICONDUCTOR

Artikel ini ditulis oleh FDBS dan team teknis Innovative Electronics.