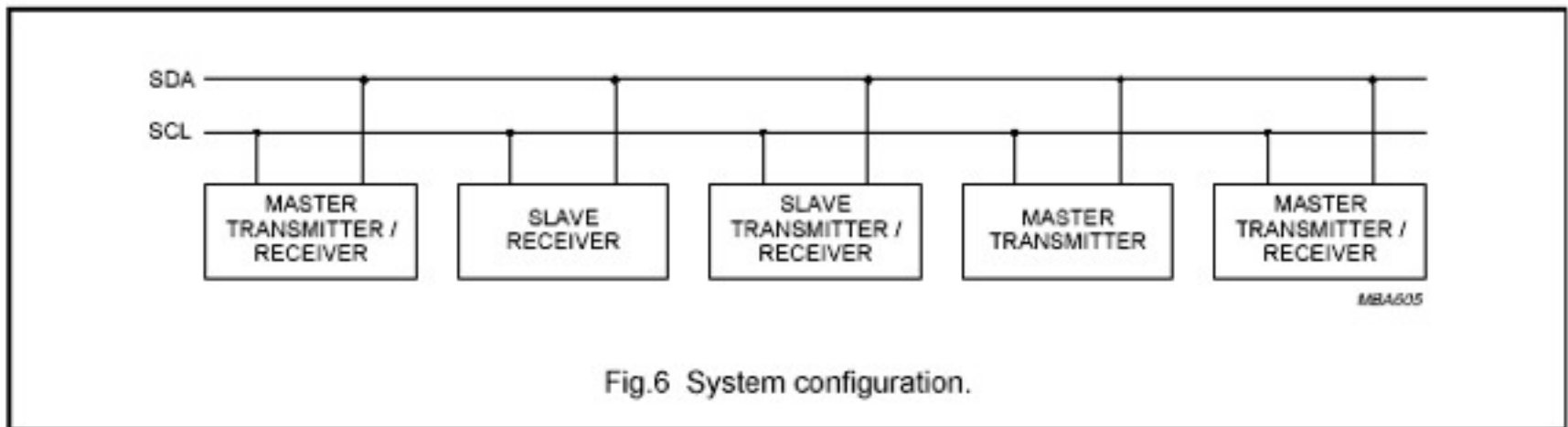


# I2C bus

## Inter IC Communication

2-draadsverbinding voor datatransport tussen IC's



# Geschiedenis

- Ontstaan  $\pm 1980$
- Doel: printoppervlak besparen voor communicatie tussen ic's
- Van 8 bits parallel naar 2 bits serieel
- 1 datalijn en 1 kloklijn dus synchrone communicatie

# Eigenschappen

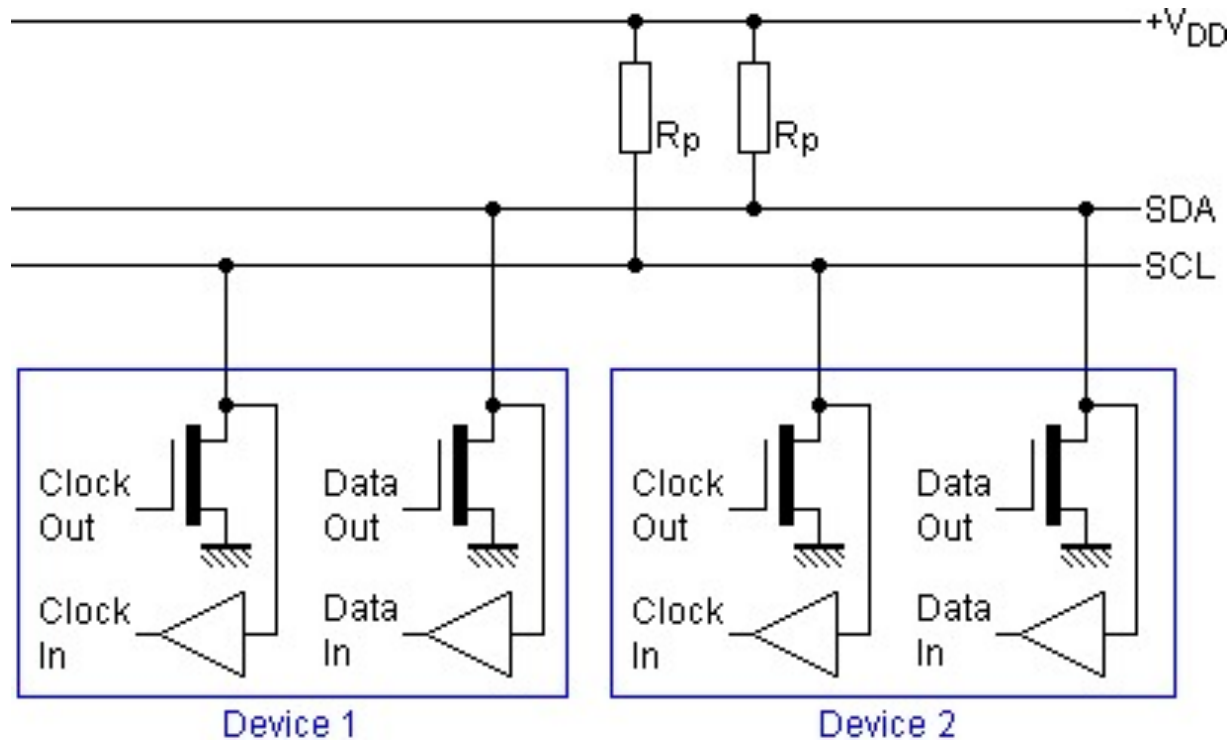
- Elk device op de **I2C** bus heeft zijn eigen unieke adres
- Communicatie vindt plaats in *master-slave mode*, waarbij meerdere masters op één bus zijn toegestaan
- Collision detection mechanismen zijn aanwezig om te voorkomen dat meer dan één apparaat op een gegeven moment de bus gebruikt
- Communicatie tussen master en slave is *bi-directioneel*
- Meerdere snelheden zijn mogelijk. Huidige implementaties staan snelheden van **100**, **400** en **3400** kbps toe.

# Fysieke eigenschappen

$R_p$ : pull-up weerstanden

$V_{DD}$  2-5 Volt

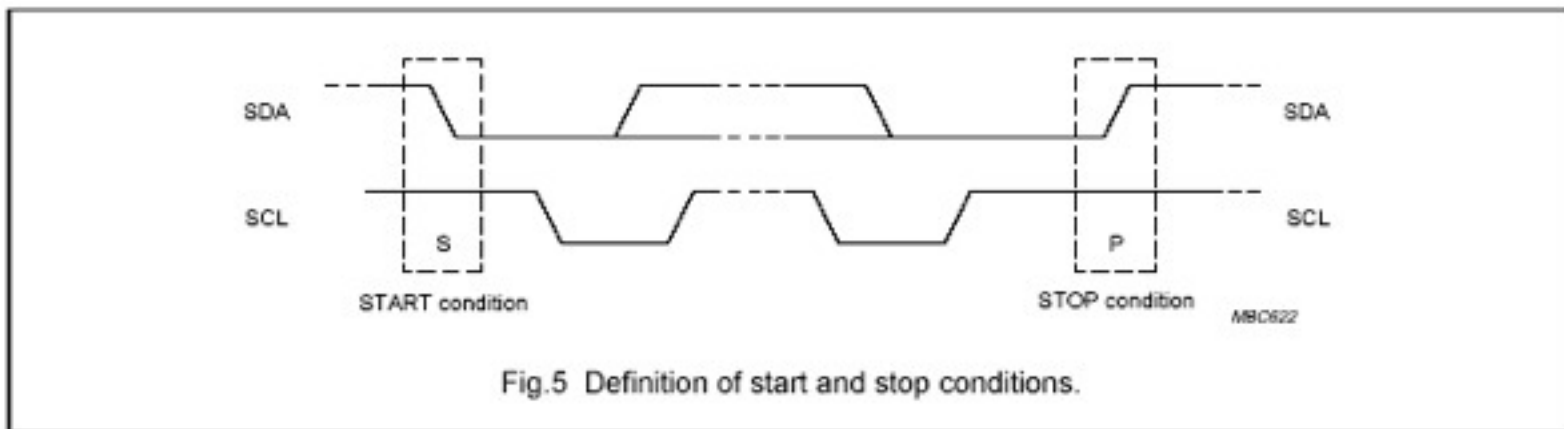
Aantal aansluitingen begrensd 400 pF



Serial Data Line  
Serial Clock Line

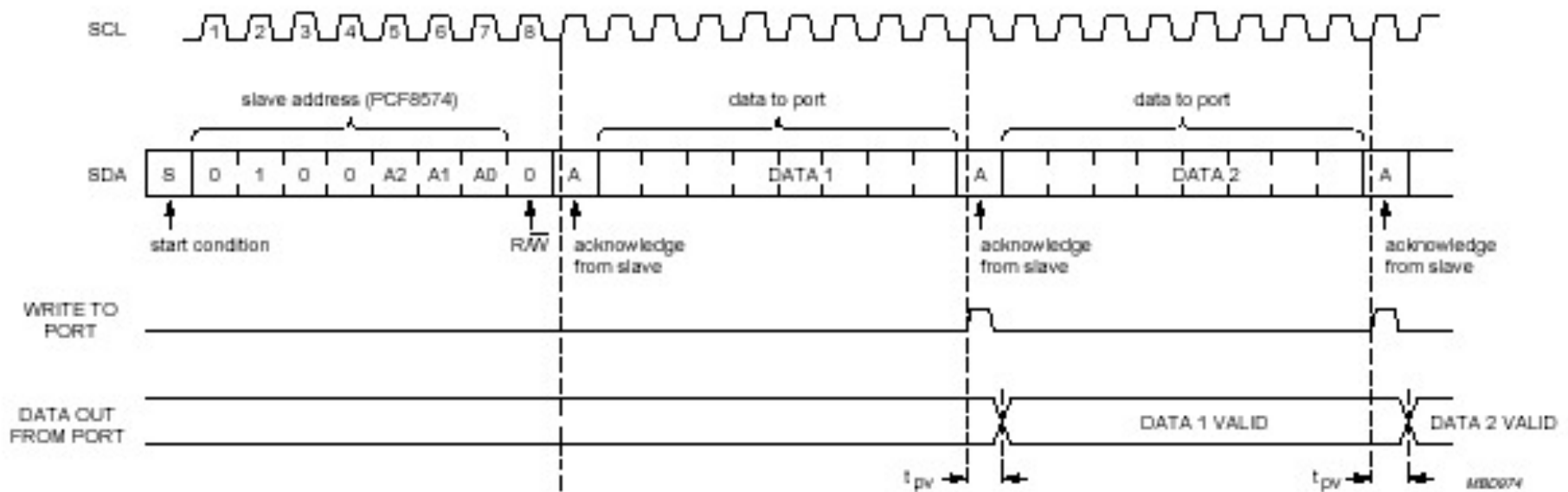
# Start en stop condities op de I2C bus

- Masters initieren datatransfer
- Kan alleen wanneer de bus 'idle' is (SDA en SCL lijn zijn hoog)
- Start: Master maakt SDA-lijn laag en SCL-lijn hoog
- Tijdens dataoverdracht van een bit is SCL-lijn laag
- Stop: **SDA** datalijn hoog terwijl de kloklijn **SCL** ook hoog is



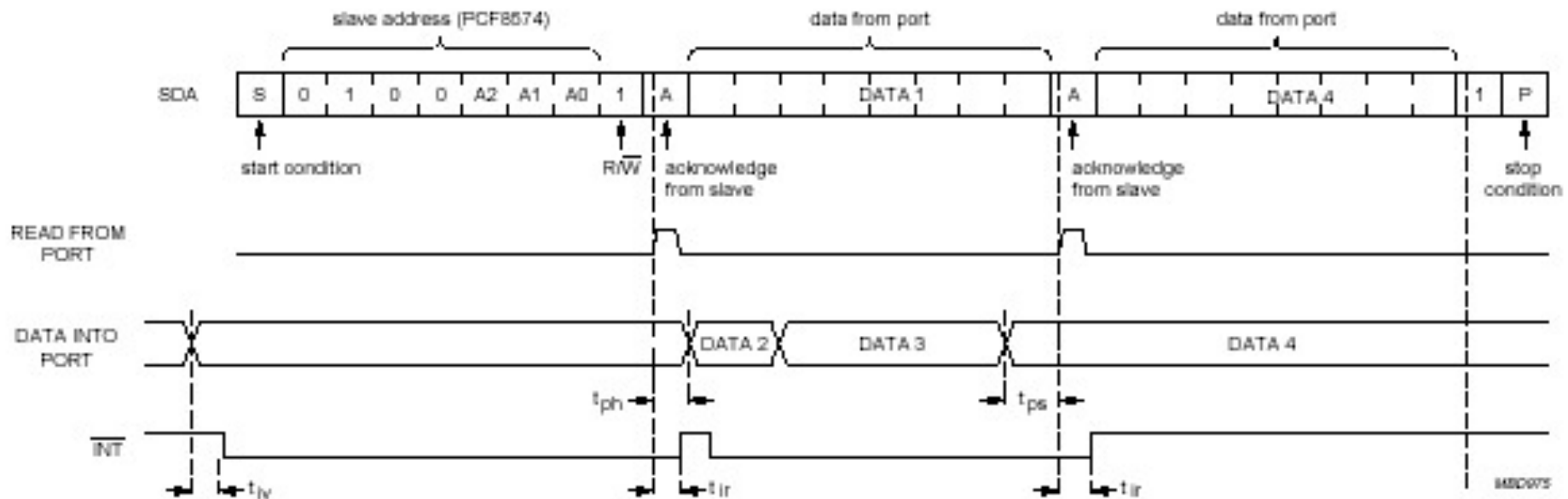
# Data overdracht van Master naar Slave

- Eerst adresbyte daarna databyte
- $0100A_2A_1A_0$



# Data overdracht van Slave naar Master

- Eerst adresbyte daarna databyte
- $0100A_2A_1A_0$ <sup>1</sup>



# PIDAC-module

- De PIDAC-module bevatten twee ic's type PCF8574A en de benodigde 1 k $\Omega$  'pull up' weerstanden.
- Adressen I/O expander:

Als pin AD1 = '0' (verbonden met ground = GND):

- A-bits is adres 0; B-bits = adres 1. (0100000x / 0100001x)

Als pin AD1 = '1' (verbonden met 5 V):

- A-bits is adres 2; B-bits = adres 3. (0100010x / 0100011x)

↑  
AD1



# I<sup>2</sup>C en de CCS compiler

```
#use I2C (Master, SDA=PIN_C4, SCL=PIN_C3, SLOW)
//Voorbeeld: Byte van Master → Slave met adres 1
i2c_start();
i2c_write( 0x42 );//write address (01000010)
i2c_write( 55 ); //write byte
i2c_stop();
```

```
//Voorbeeld: Byte van Slave met adres 0 → Master
i2c_start();
i2c_write( 0x41 );//write address (01000001)
Getal = i2c_read(0); // acknowledge
i2c_stop();
```

# Successive Approximation ADC

