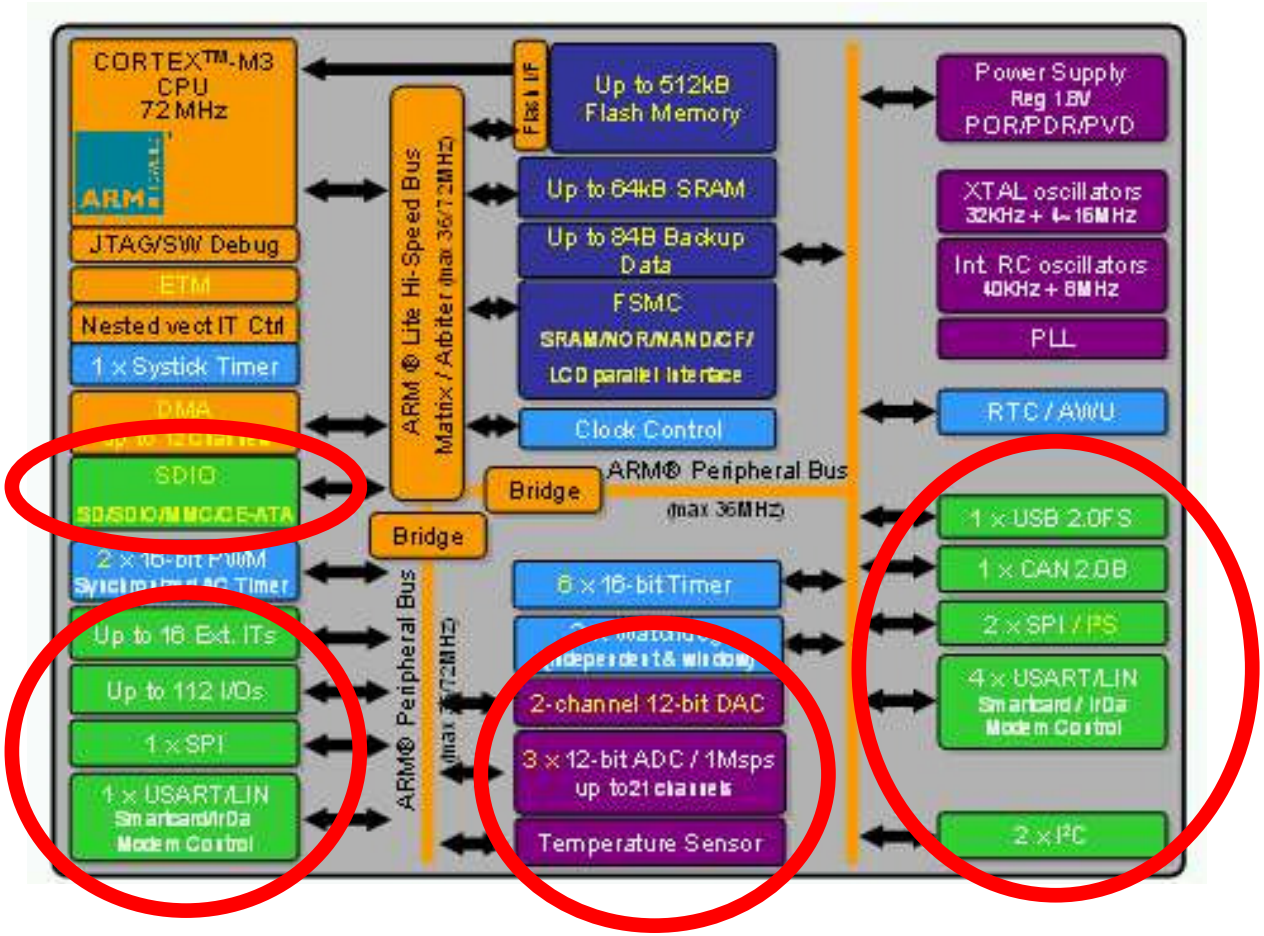
Microcontrollori – Periferiche Interne



Microcontrollori – Periferiche Interne

Peripherals registers

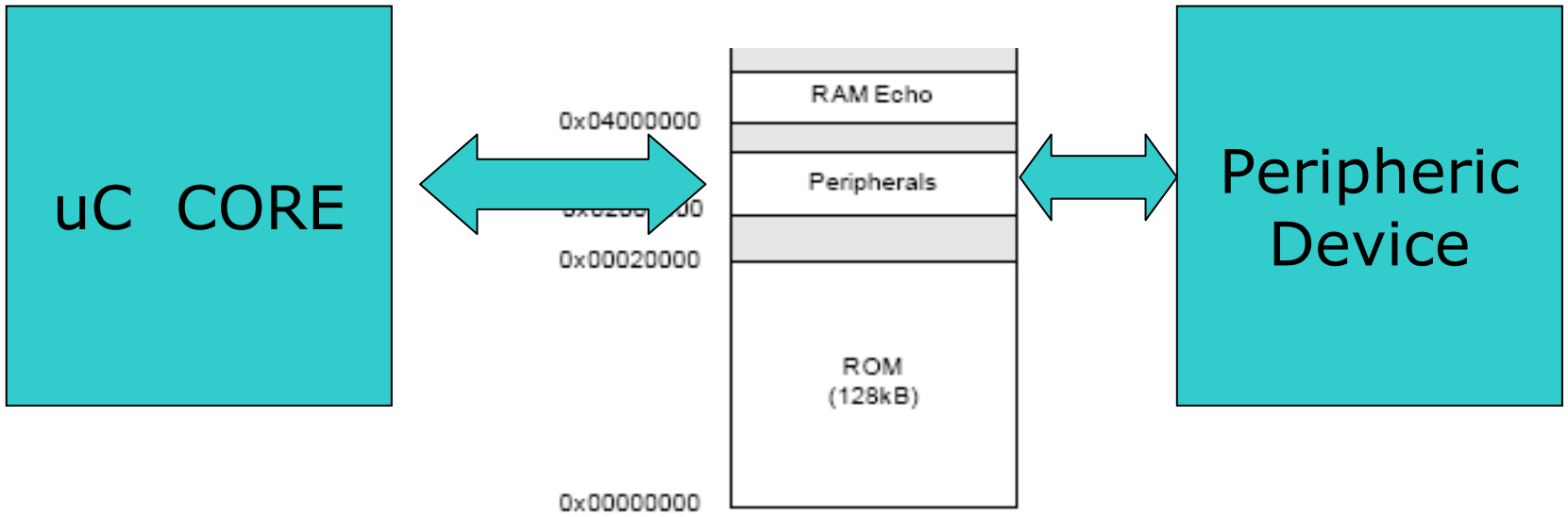
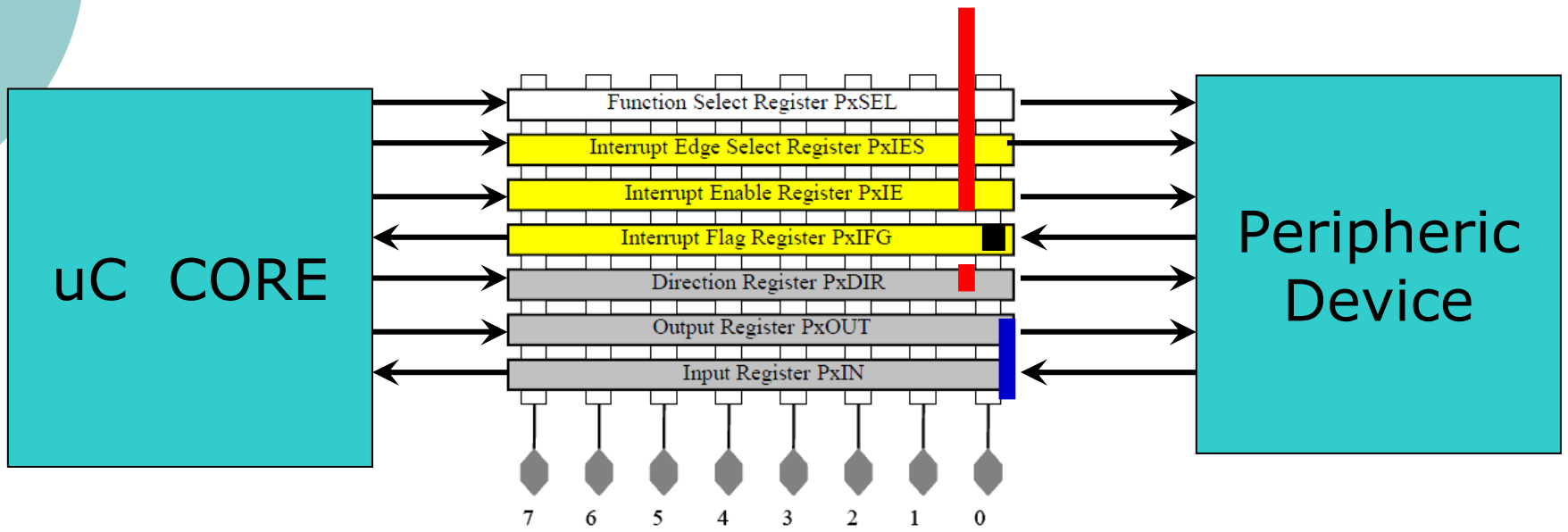


Figure 5: JN5148 Memory Map

Microcontrollori – Periferiche Interne

Peripherals registers



Microcontrollori – Periferiche Interne

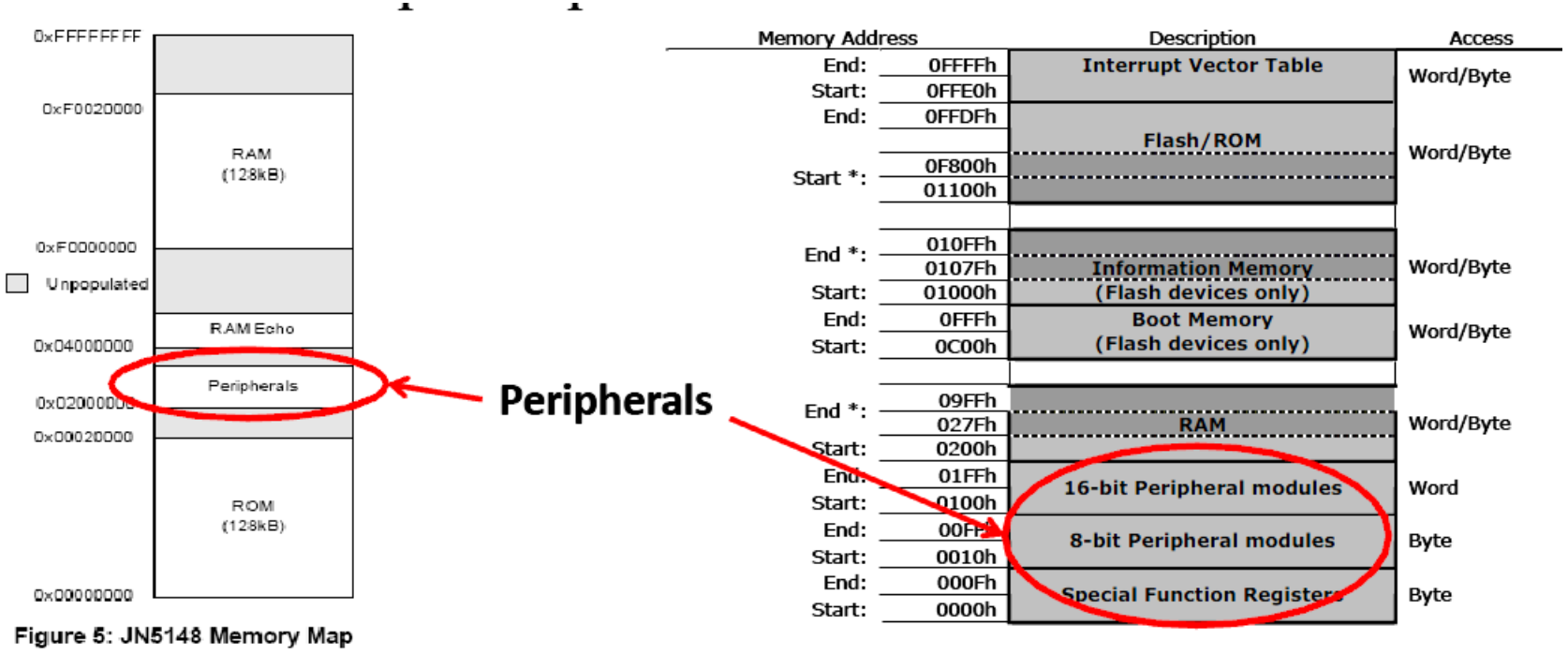
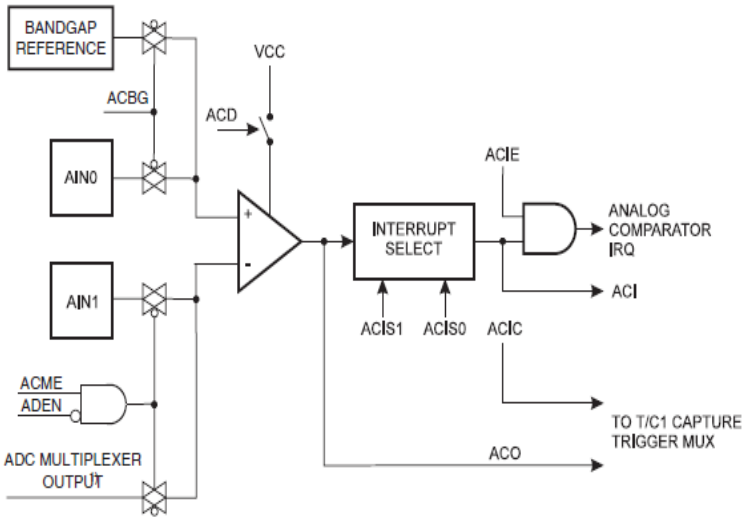


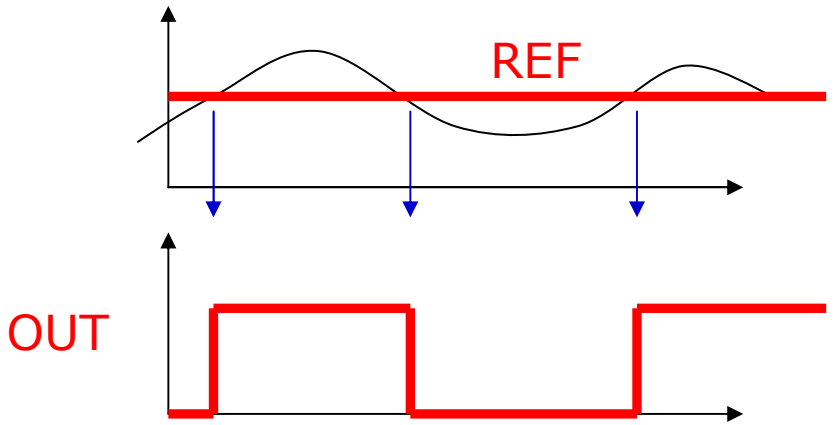
Figure 5: JN5148 Memory Map

Microcontrollori – Analogical Device

Comparator



- Monitoring of external analog signals.
- Internally reference voltage generator
- Externally reference voltage applicable



Microcontrollori – Analogical Device

ADC

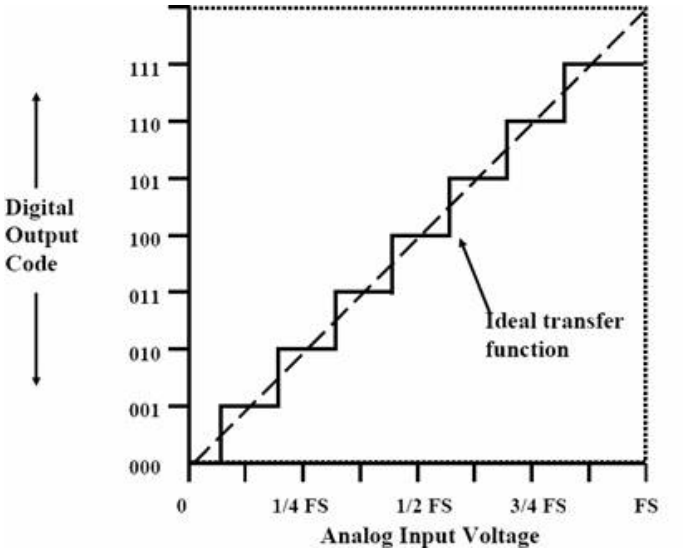
Risoluzione

Indica il numero di bit di uscita:

10, 12, 16, 24 Bit ADC

La variazione minima che si riesce

a discriminare è legata alla risoluzione



Accuratezza

Livello di conformità fra dato convertito e dato analogico

Velocità di conversione

Massimo data rate espresso in sample per second (sps)

$$V_{LSB} = \frac{V_{ref}}{2^n}$$

LSB= Least Significant Bit

Microcontrollori – Analogical Device

ADC

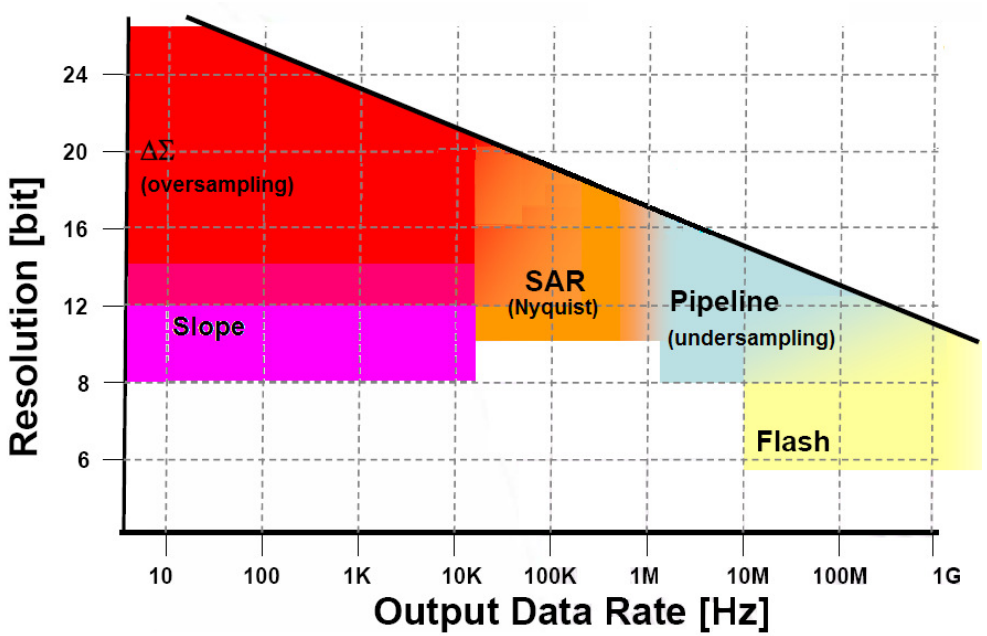
Successive Approximation (SAR)

Sigma Delta (SD or ΔΣ)

Slope or Dual Slope

Pipeline

Flash



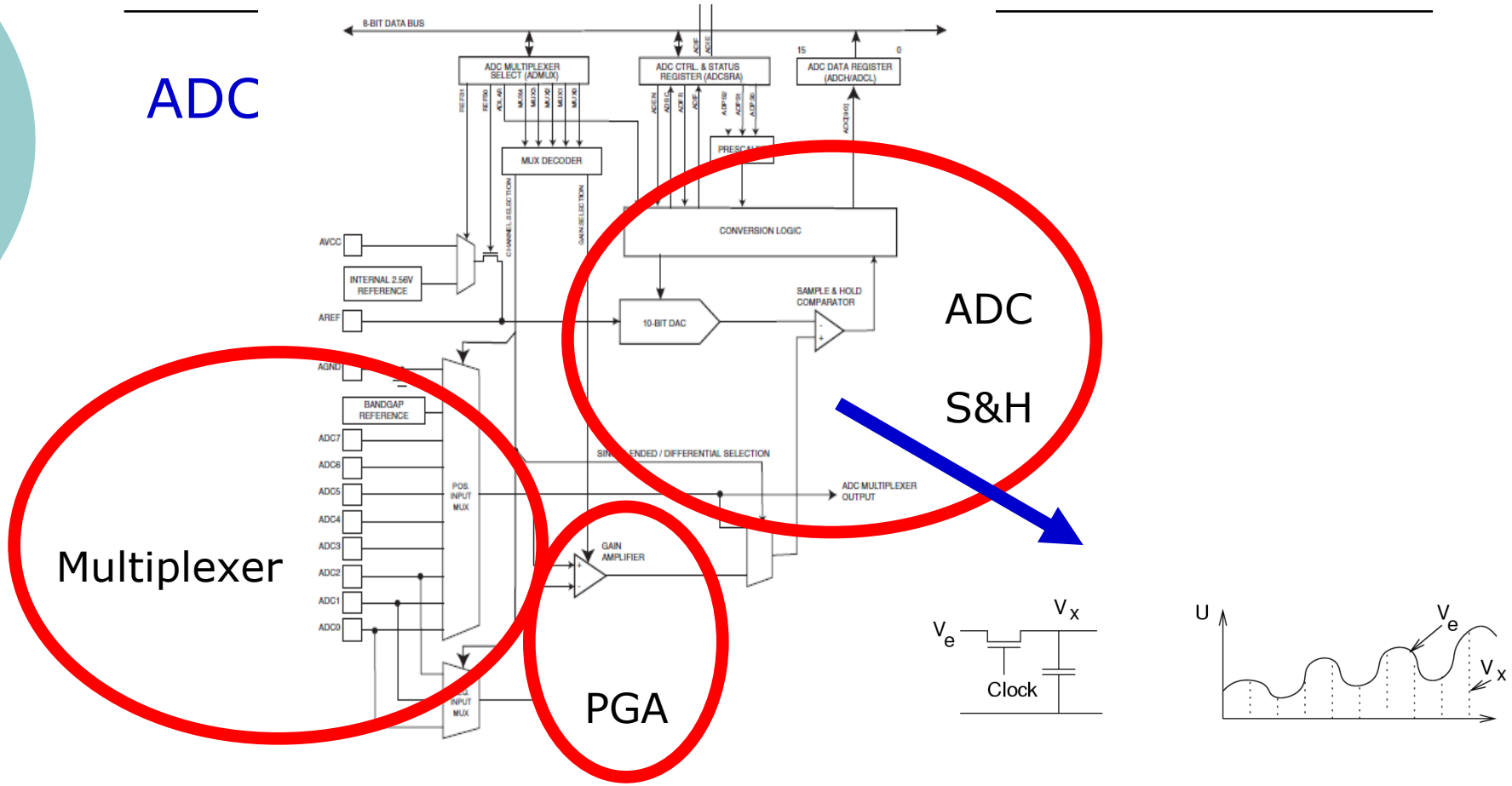
Microcontrollori – Analogical Device

ADC

ADC architecture	Resolution	Conversion rate	Advantages	Disadvantages
SAR	≤ 18 bit	< 5 Msps	Zero-cycle latency Low latency-time High accuracy Low power Simple operation	Sample rates 2-5 MHz
SD	≤ 24 bit ≤ 16-18 bit	< 625 ksps < 10 Msps	High resolution High stability Low power Moderate cost	Cycle-latency Low speed
Pipeline	≤ 16 bit	< 500 Msps	Higher speeds Higher bandwidth	Lower resolution Delay/Data latency Power requirements

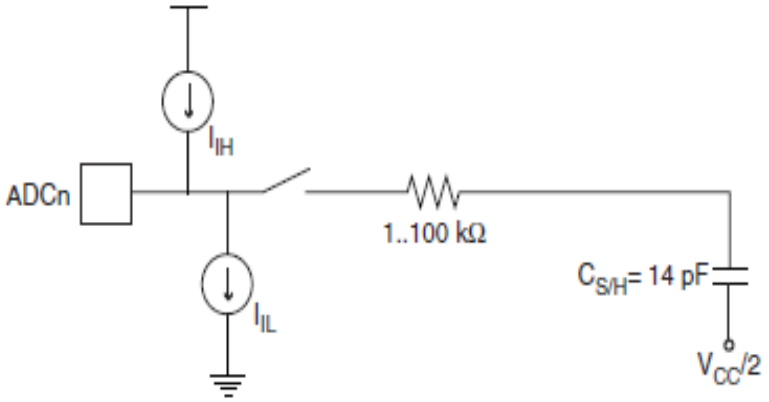
Microcontrollori – Analogical Device

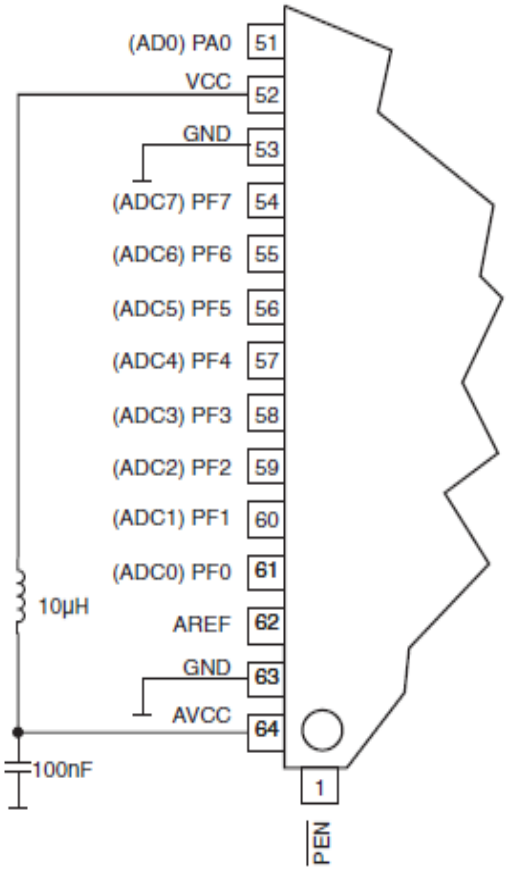
ADC



Microcontrollori – Analogical Device

ADC – Single Ended Input

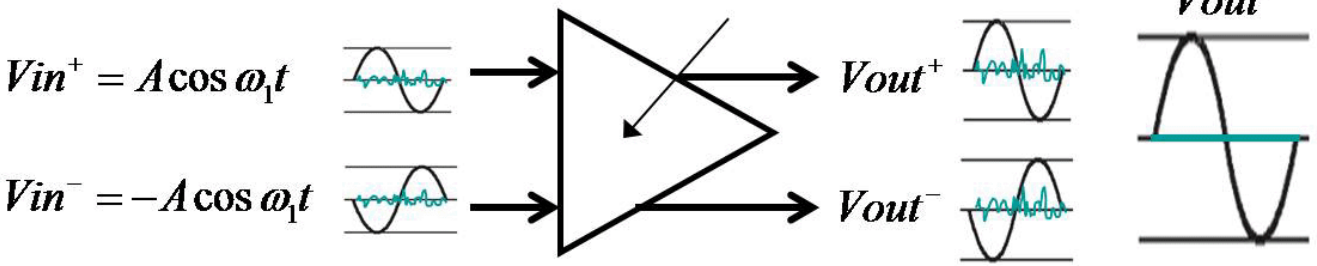




Microcontrollori – Analogical Device

ADC – Differential Input

$$Vout = \sum C_n Vin^{+n} - C_n Vin^{-n}$$

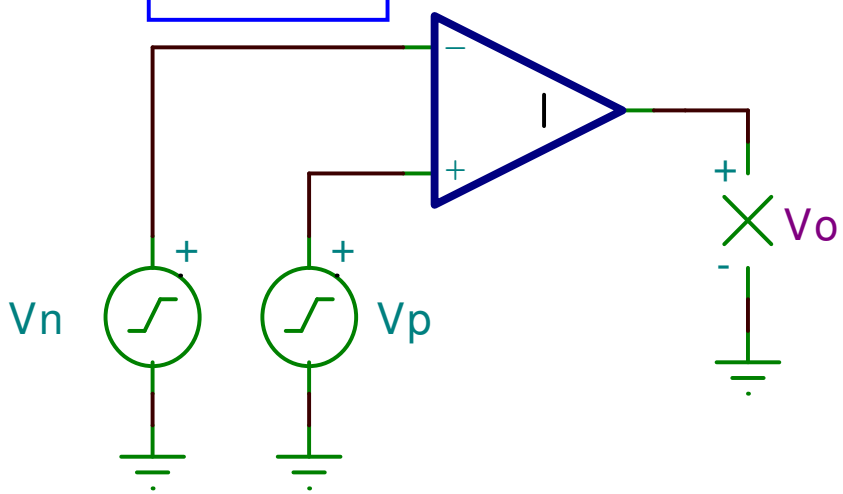


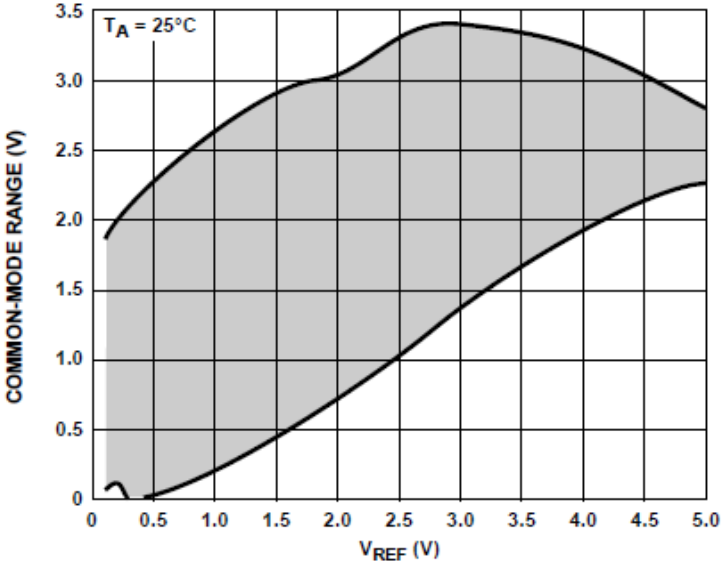
$$Vout^+ - Vout^- = (0) + \left(0 - A^3C_4/2\right)\cos\omega_1 t + (0)\cos 2\omega_1 t - \left(A^3C_4/2\right)\cos 3\omega_1 t + \ldots$$

Microcontrollori – Analogical Device

ADC – Differential Input

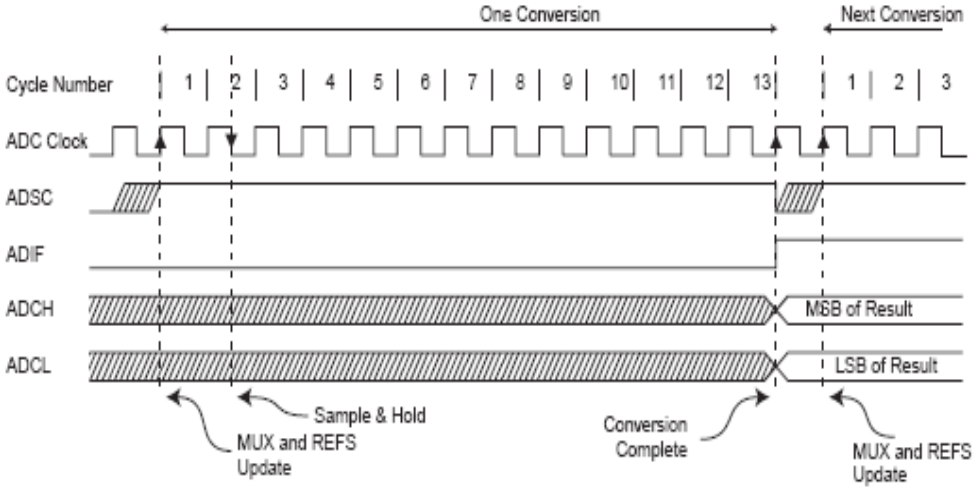
$$V_{cm} = \frac{V_p + V_n}{2}$$

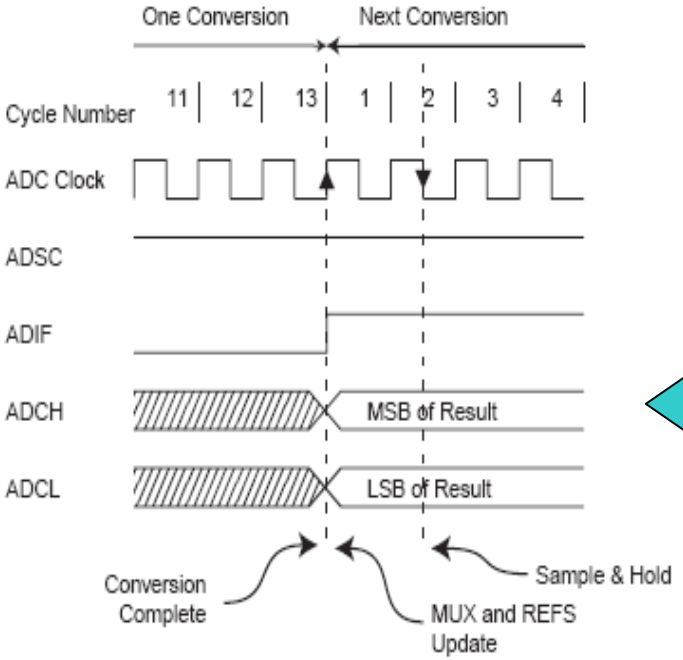




Microcontrollori – Analogical Device

Single Conversion

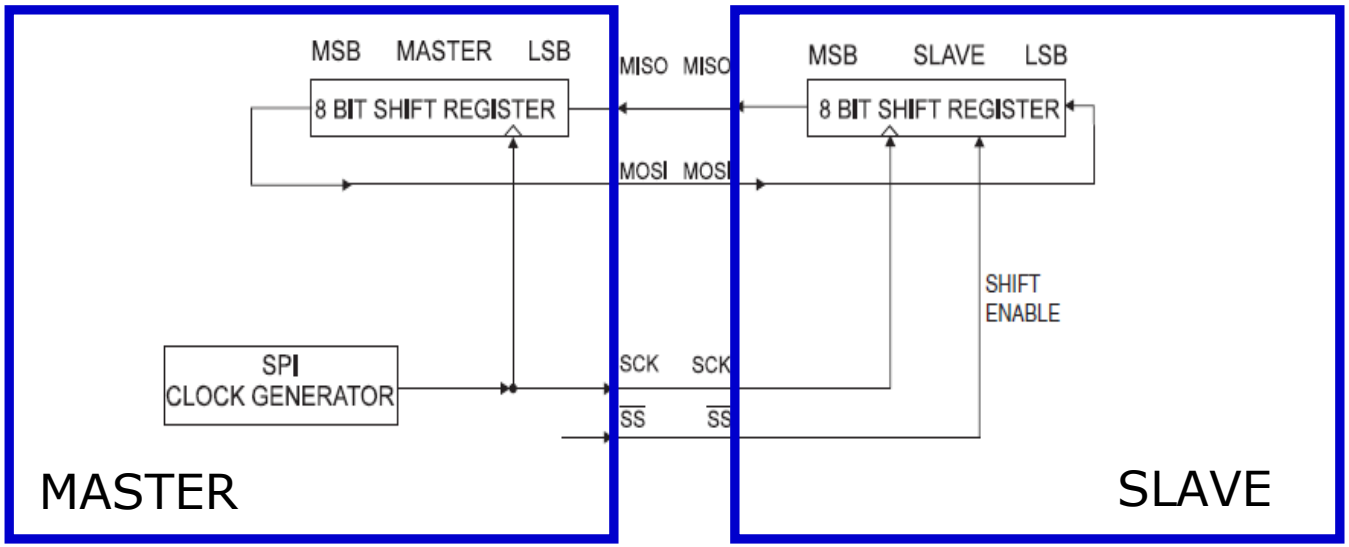




Continuous Conversion

Microcontrollori – Comm. Interface

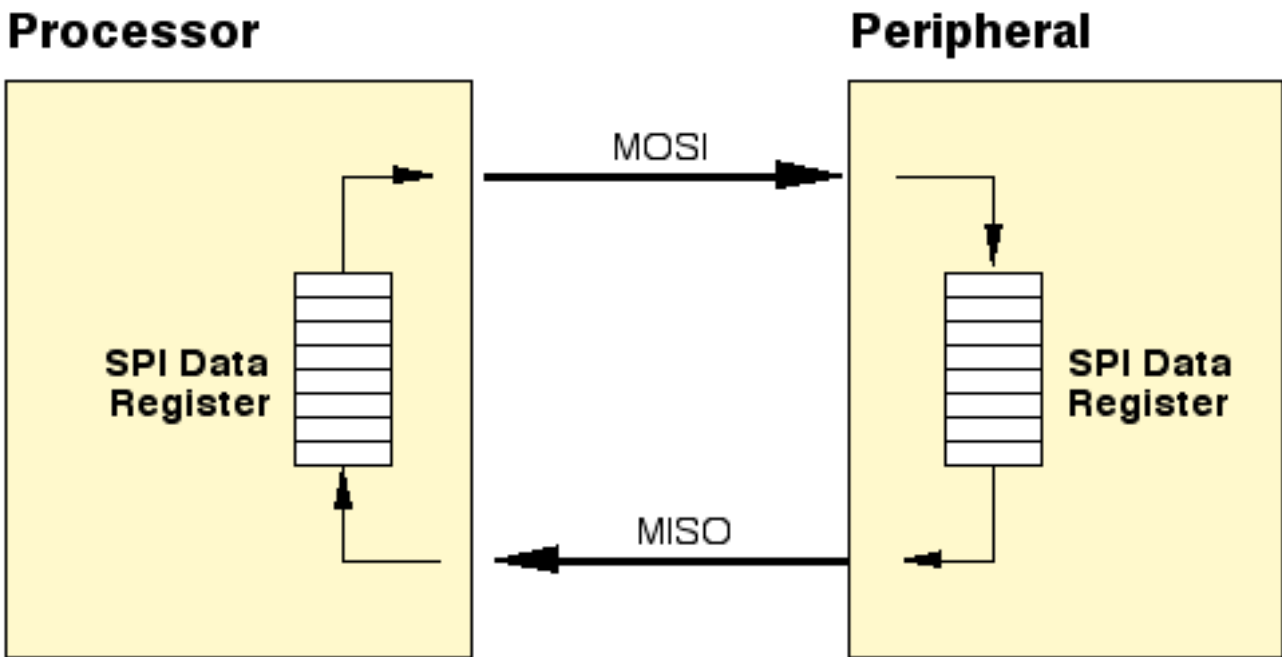
SPI



Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

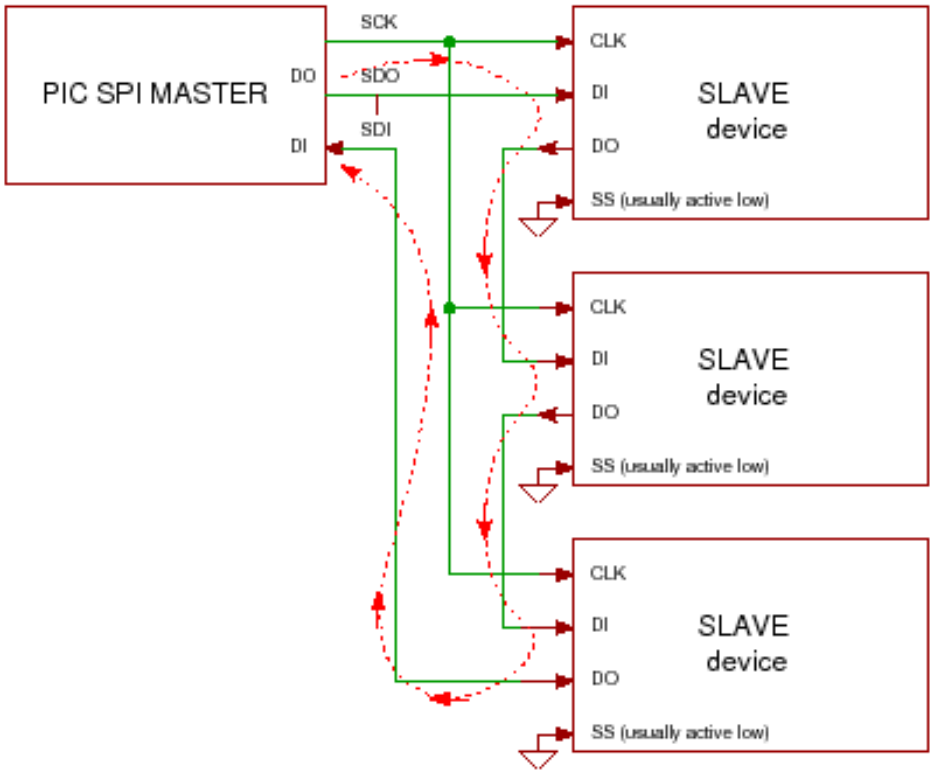
Microcontrollori – Comm. Interface

SPI



Microcontrollori – Comm. Interface

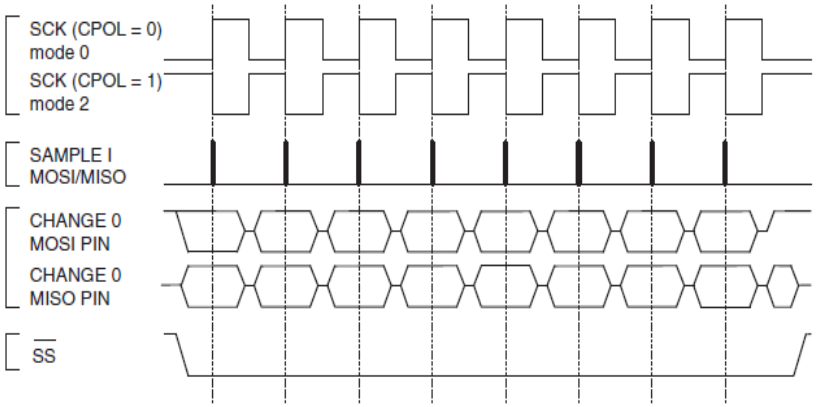
SPI

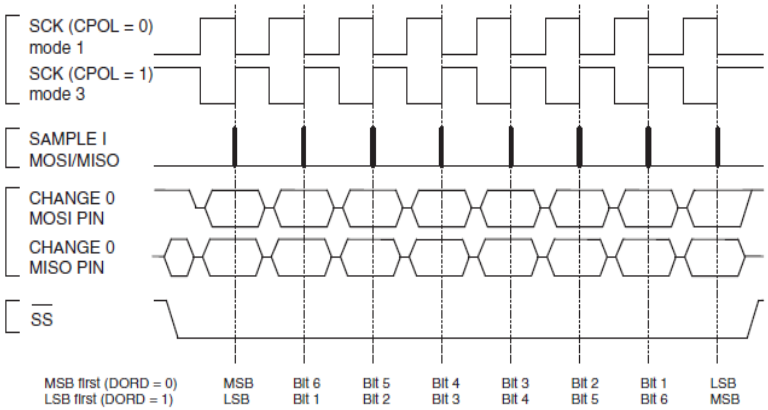


Microcontrollori – Comm. Interface

SPI

POLARITY!!!!!!

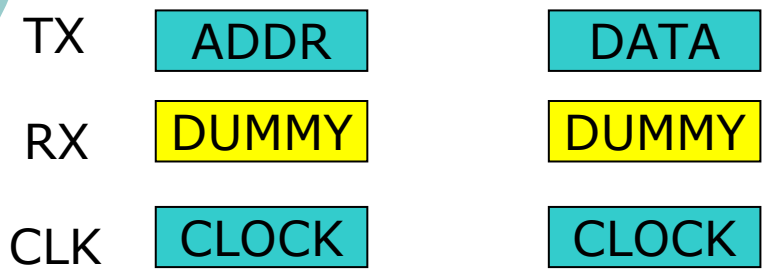




Microcontrollori – Comm. Interface

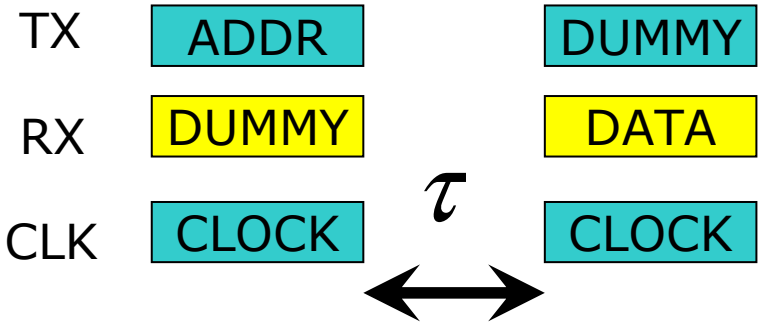
SPI Code

WRITE



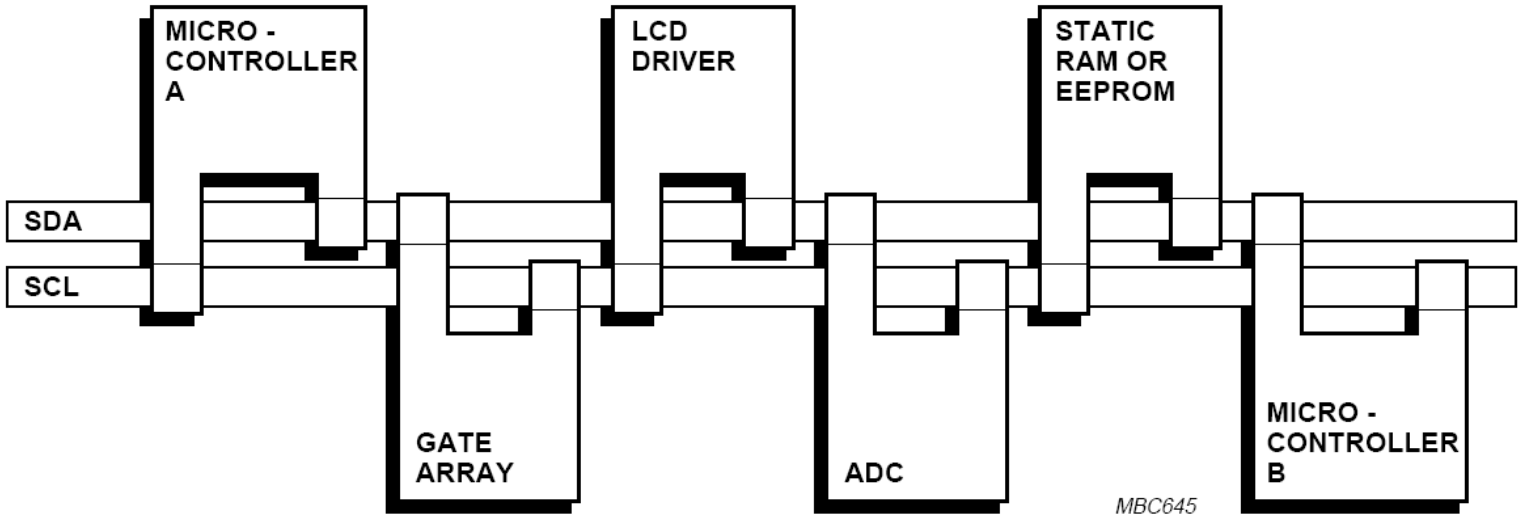

```
packet.chip = TARGET_ADDRESS;
packet.addr[0] = 27;//(VIRTUALMEM_ADDR >> 16) & 0xff;
packet.addr_length = TARGET_ADDR_LGT;
write1_data[0]=0x18;        //scala 500
// Where to find the data to be written
packet.buffer = (void *) write1_data;
// How many bytes do we want to write
packet.length = PATTERN_TEST_LENGTH;
//print_dbg ("Writing data 27\r\n");
// Write data to TARGET
status = twi_master_write (TWIM, &packet);
```

READ

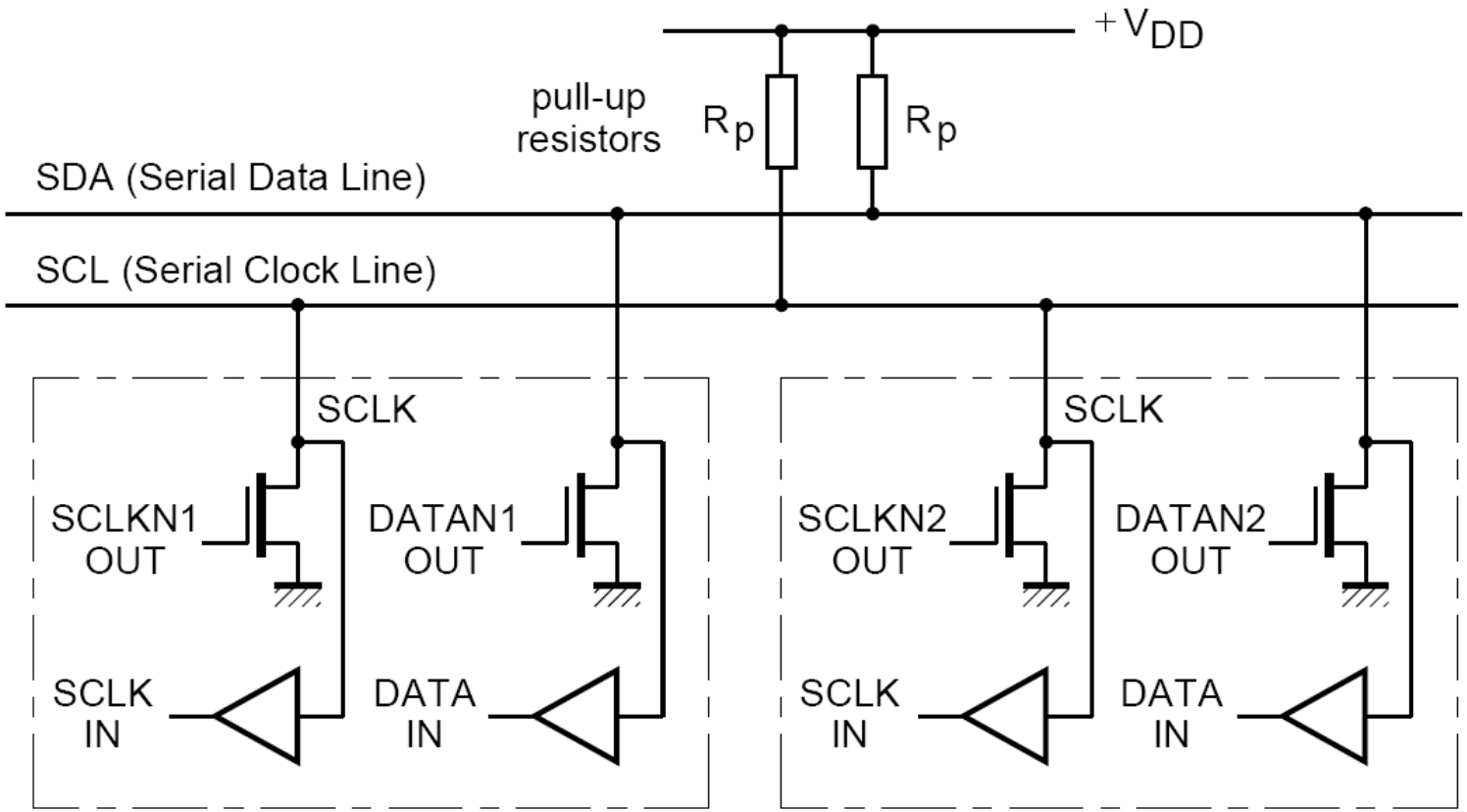



```
packet_received.chip = TARGET_ADDRESS;
packet_received.addr_length = TARGET_ADDR_LGT;
packet_received.length = 6;//PATTERN_TEST_LENGTH;
packet_received.addr[0] = 59;//(VIRTUALMEM_ADDR >> 16) & 0xff;
packet_received.buffer = read1_data;
status = twi_master_read (TWIM, &packet_received);
```

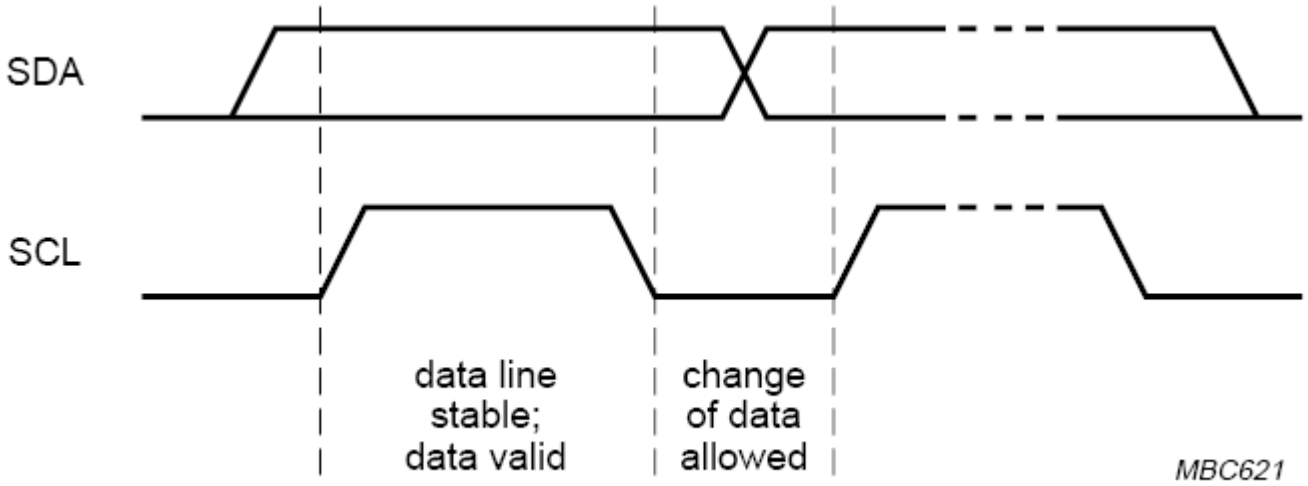
I2C


MICRO - CONTROLLER A
LCD DRIVER
STATIC RAM OR EEPROM
SDA
SCL
GATE ARRAY
ADC
MICRO - CONTROLLER B
MBC645

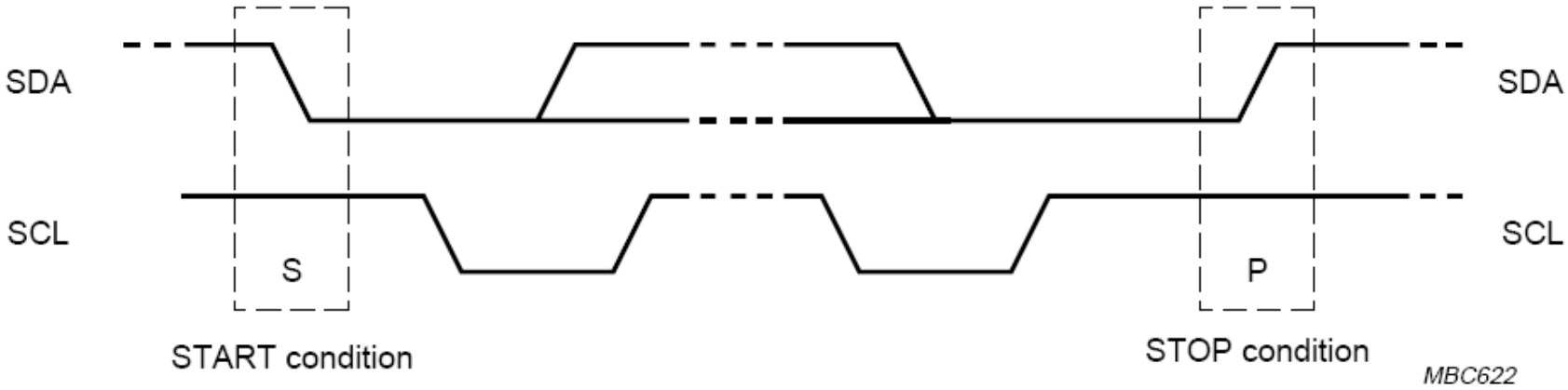
I2C


+VDD
pull-up
resistors
Rp
Rp
SDA (Serial Data Line)
SCL (Serial Clock Line)
SCLK
SCLKN1
OUT
DATAN1
OUT
SCLK
IN
DATA
IN
SCLK
SCLKN2
OUT
DATAN2
OUT
SCLK
IN
DATA
IN

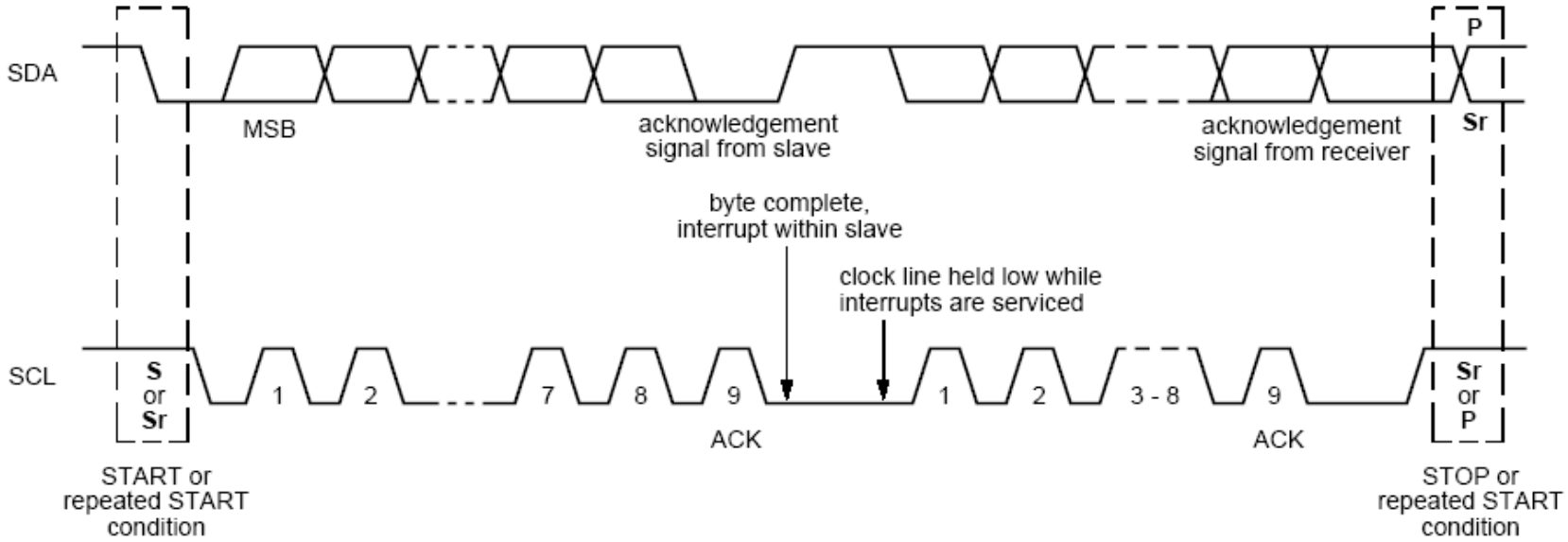
I2C Bit transfer



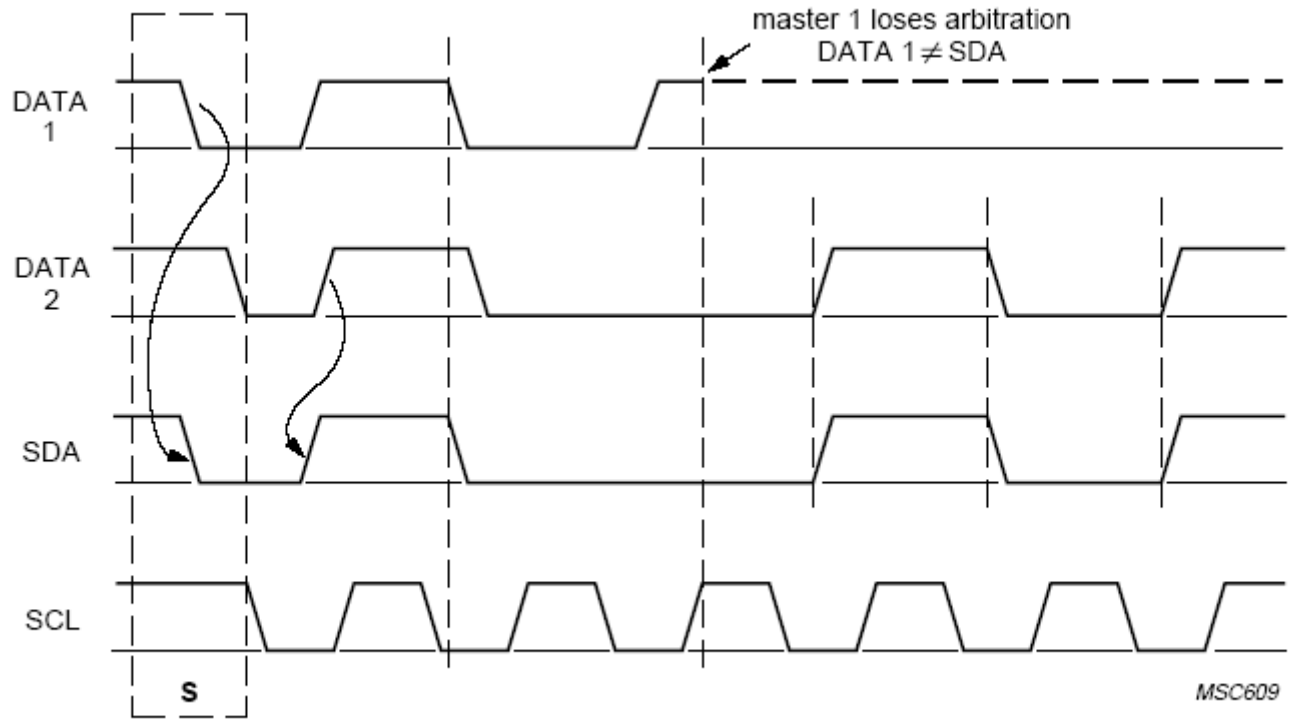
I2C Start Stop



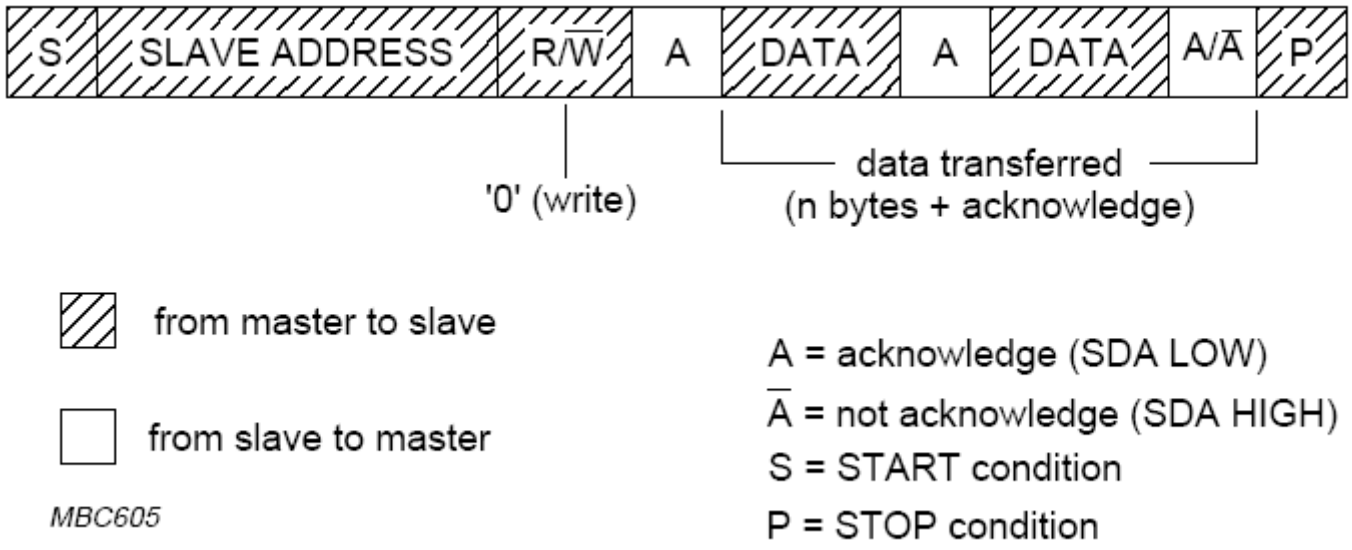
I2C Data Transfer



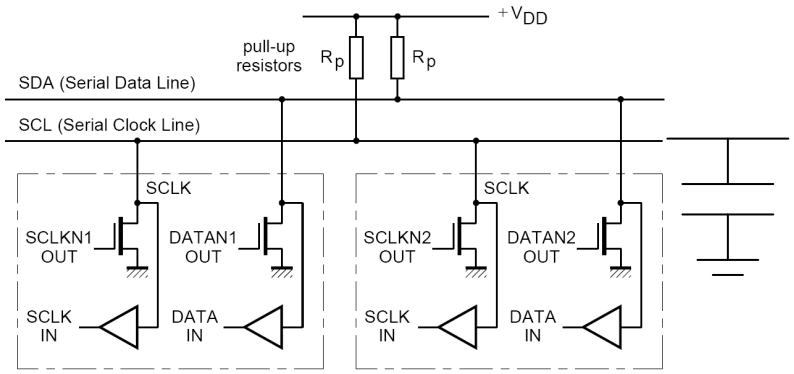
I2C Arbitration



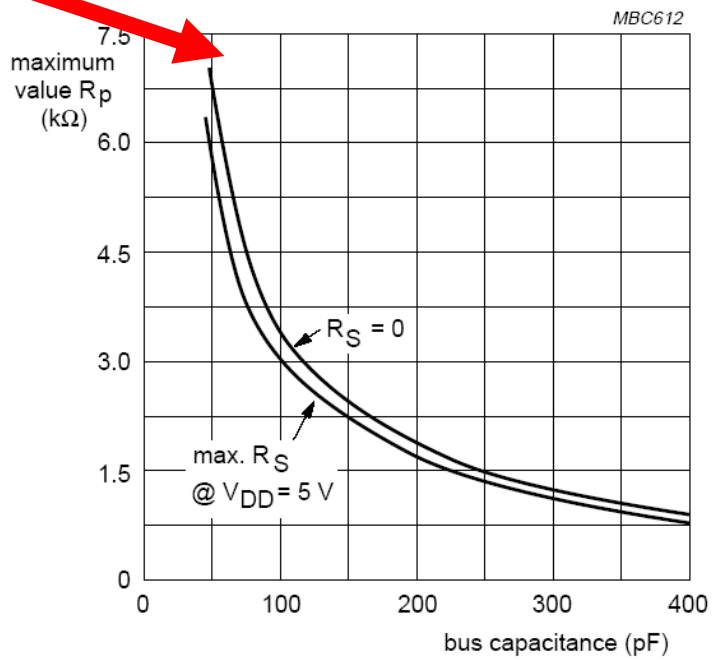
I2C Trasmissione



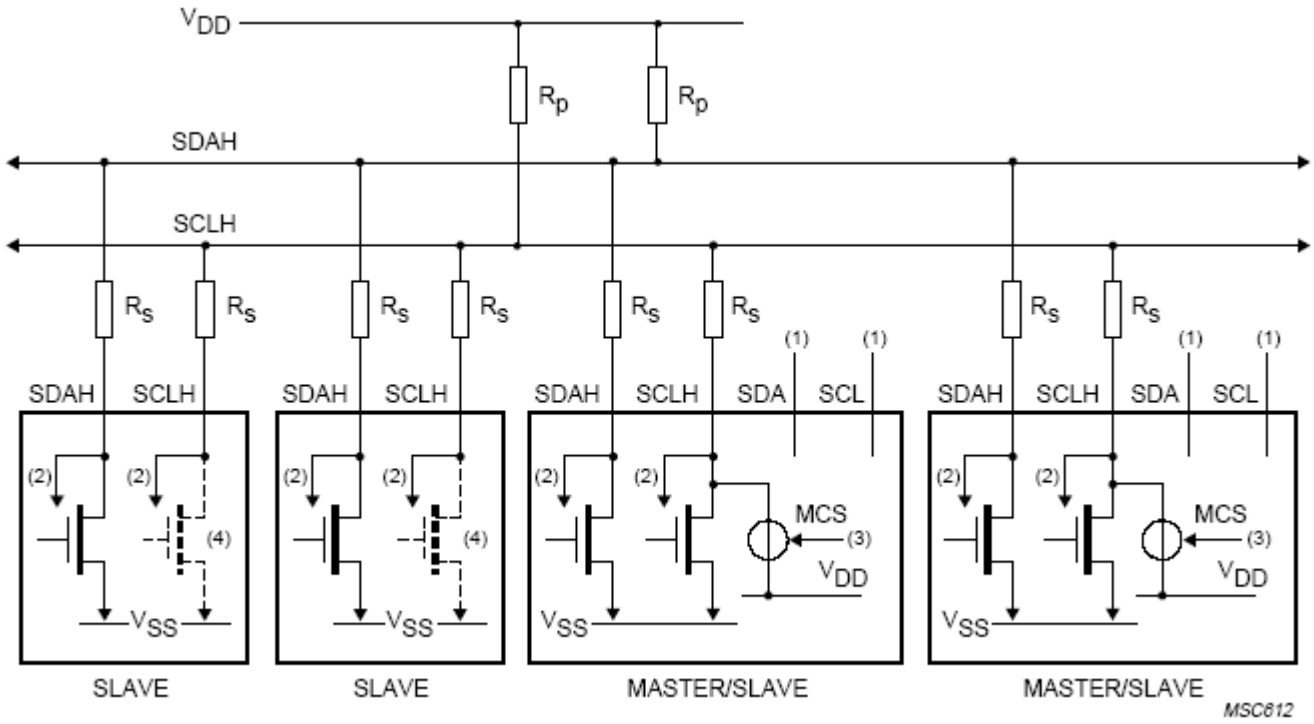
I2C Rp e C



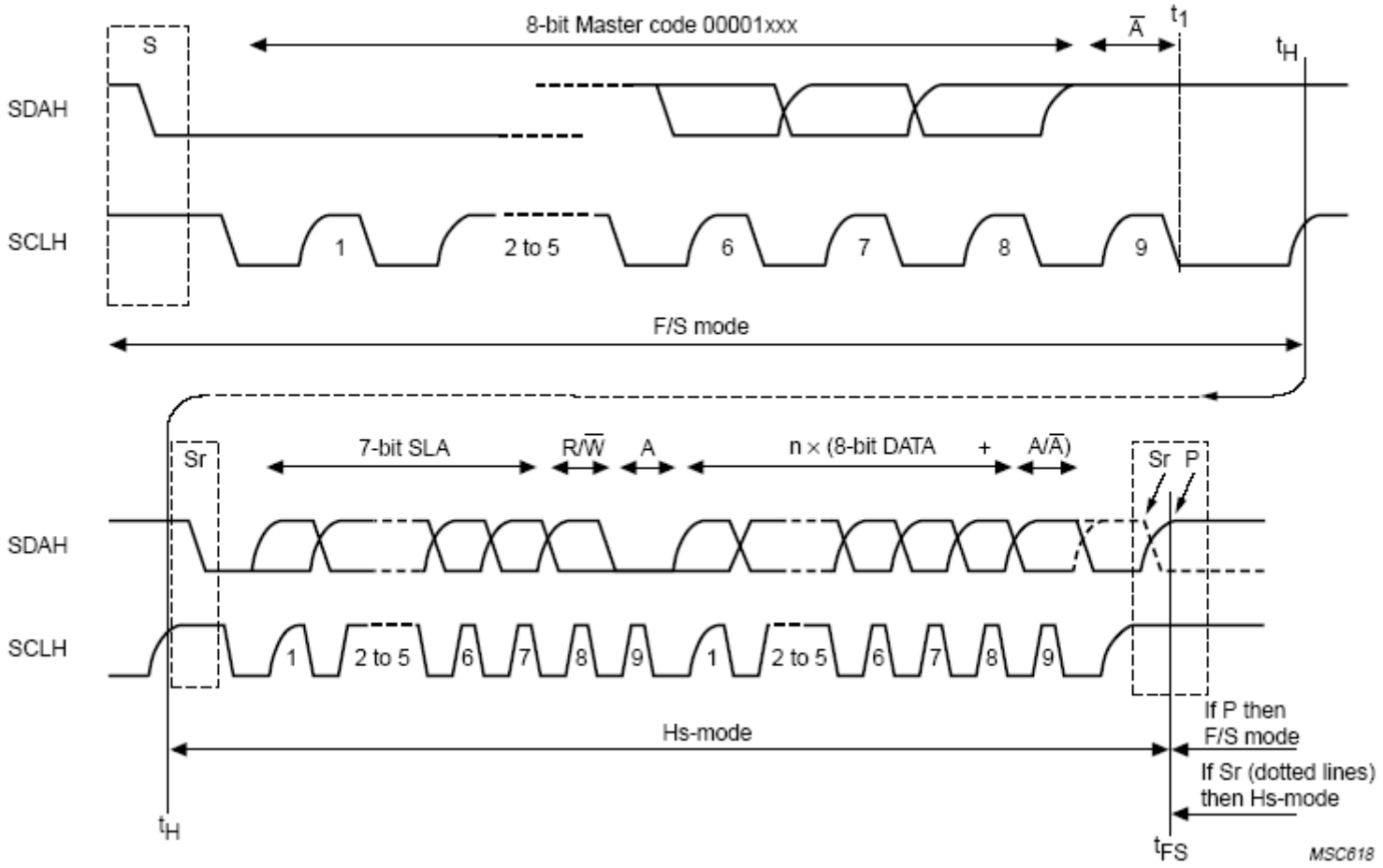
$$C_L \; \alpha \; (L; D)$$



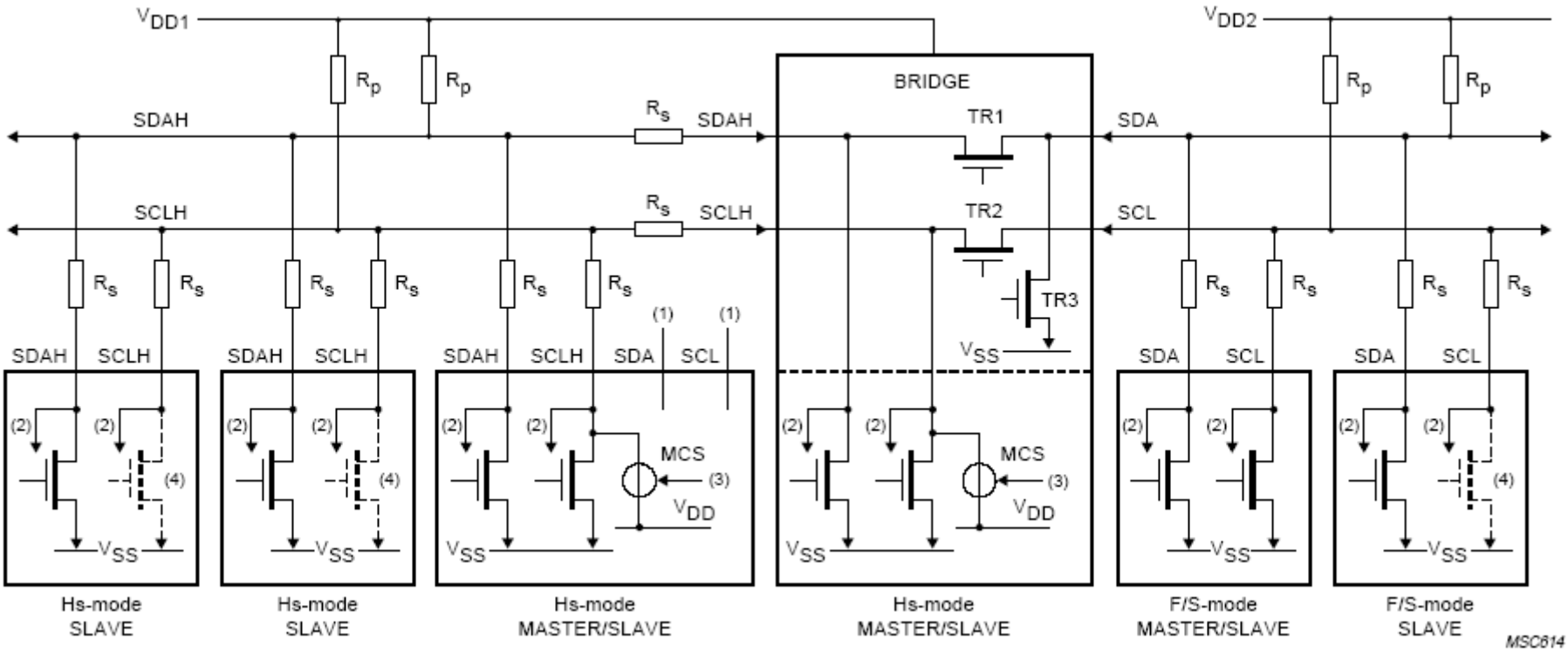
I2C HS mode


VDD
Rp
Rp
SDAH
SCLH
RS
SDAH
SCLH
(2)
(4)
VSS
SLAVE
(1)
SDA
SCL
MCS
(3)
MASTER/SLAVE
MSC812

I2C HS Mode


8-bit Master code 00001xxx
S
Ā
t1
tH
SDAH
SCLH
1
2 to 5
6
7
8
9
F/S mode
Sr
7-bit SLA
R/W
A
n × (8-bit DATA + A/Ā)
Sr P
SDAH
SCLH
1
2 to 5
6
7
8
9
1
2 to 5
6
7
8
9
If P then
F/S mode
Hs-mode
If Sr (dotted lines)
then Hs-mode
tH
tFS
MSC618

I2C HS Mode Bridge



I2C Velocità di comunicazione

TRANSFER BETWEEN	SERIAL BUS SYSTEM CONFIGURATION			
	Hs + FAST + STANDARD	Hs + FAST	Hs + STANDARD	FAST + STANDARD
Hs <–> Hs	0 to 3.4 Mbit/s	0 to 3.4 Mbit/s	0 to 3.4 Mbit/s	–
Hs <–> Fast	0 to 100 kbit/s	0 to 400 kbit/s	–	–
Hs <–> Standard	0 to 100 kbit/s	–	0 to 100 kbit/s	–
Fast <–> Standard	0 to 100 kbit/s	–	–	0 to 100 kbit/s
Fast <–> Fast	0 to 100 kbit/s	0 to 400 kbit/s	–	0 to 100 kbit/s
Standard <–> Standard	0 to 100 kbit/s	–	0 to 100 kbit/s	0 to 100 kbit/s

RS232

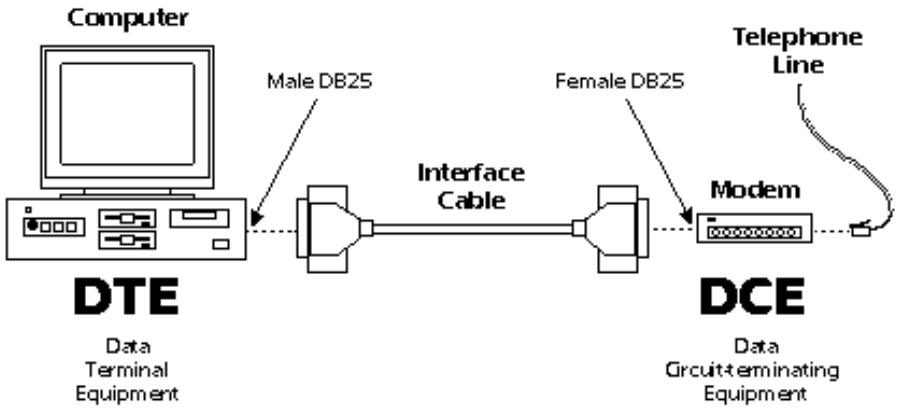
- **Universal asynchronous receiver/transmitter**
- **Transmit bits in a single channel**
  - simplex (one way)
  - half-duplex (one direction at a time)
  - full-duplex (two way)
- **A sequence of bits – packet or character**
  - ASCII code – 7 bits for 128 characters (alphabet, numerical, and control)
  - fixed length or variable length
  - Start, stop, and parity bits

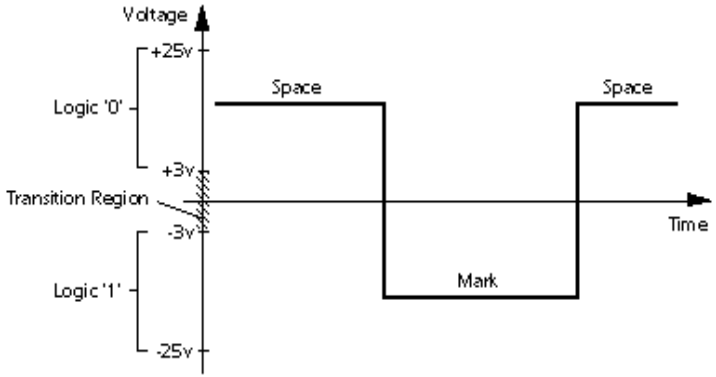
RS232

EIA RS232

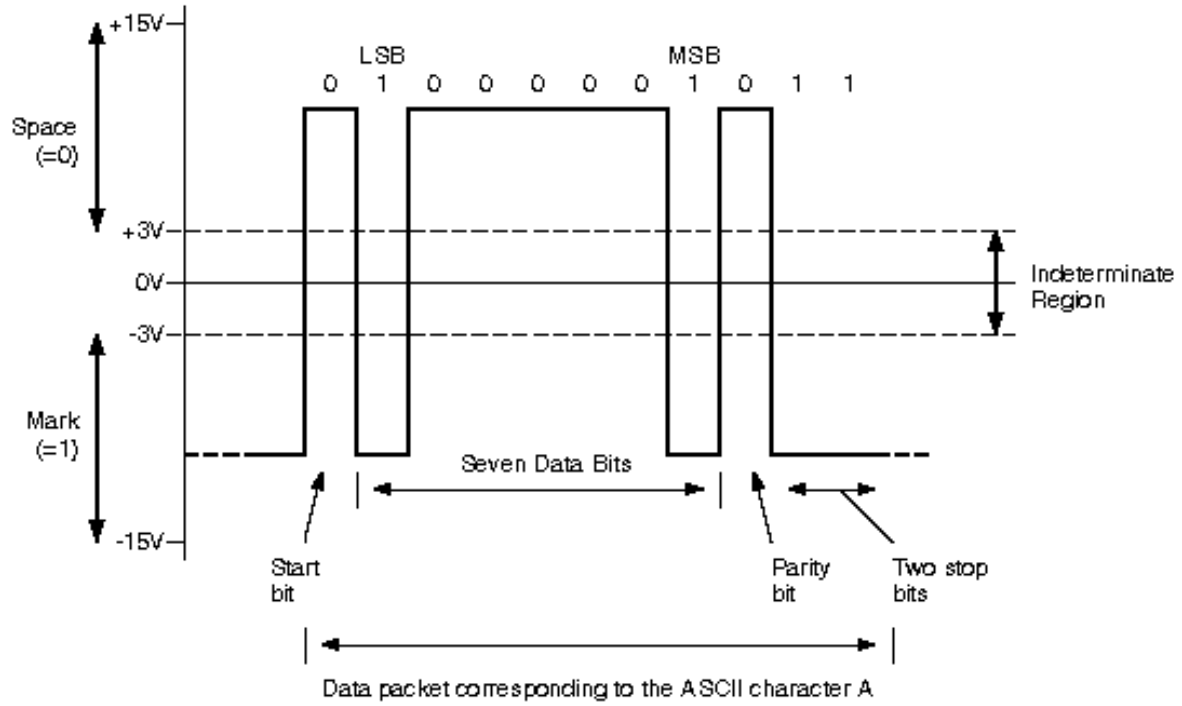
- Connection and signal characteristics
- Data terminal equipment and data communication equipment
- Logic '1' (marking) – -3v to -25v with respect to signal ground
- Logic “0” (spacing) – +3v to +25v
- Not assigned –between -3v and +3v (a transition region)

RS232

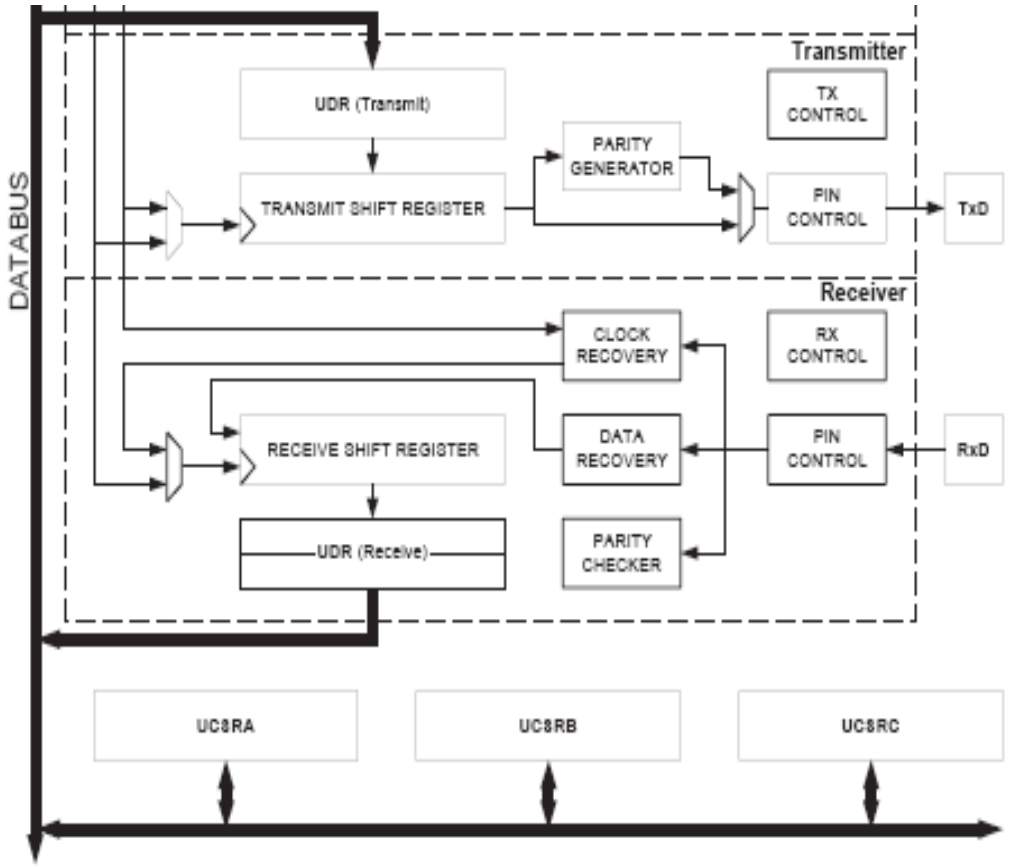

Computer
Male DB25
Female DB25
Telephone Line
Interface Cable
Modem
DTE
Data Terminal Equipment
DCE
Data Circuit-terminating Equipment


Voltage
+25v
Logic '0'
Space
Space
+3v
Transition Region
-3v
Time
Logic '1'
Mark
-25v

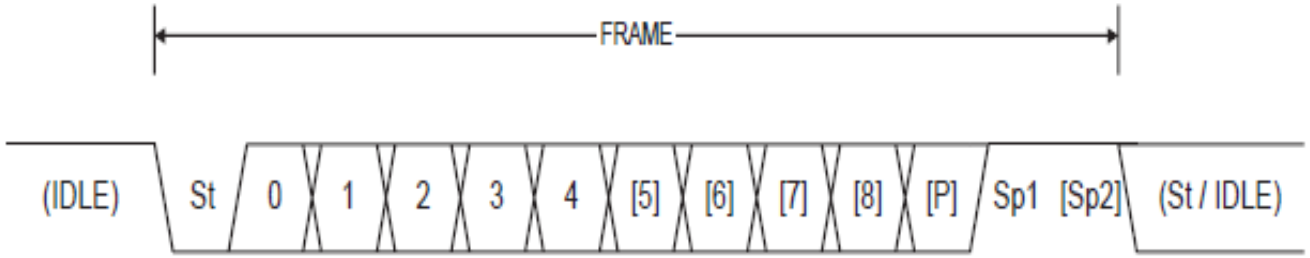
UART – Interface RS232



UART - Interface

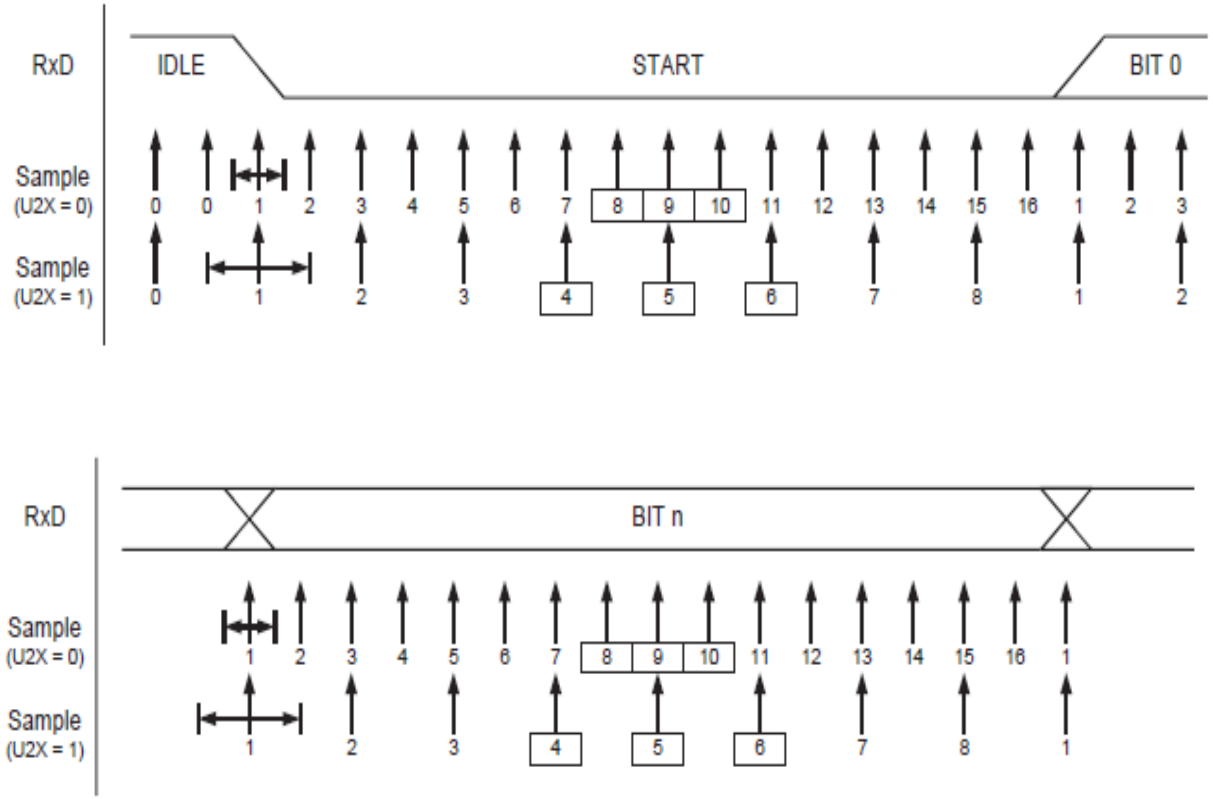


UART - Interface



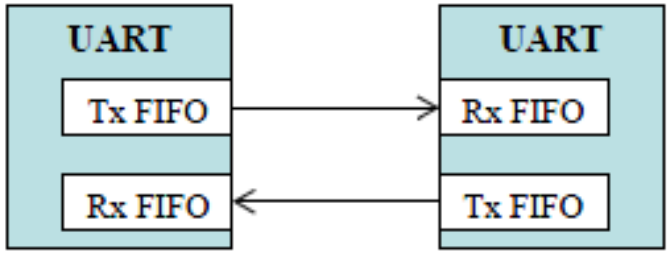
St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high.

UART - Interface



UART – Interface

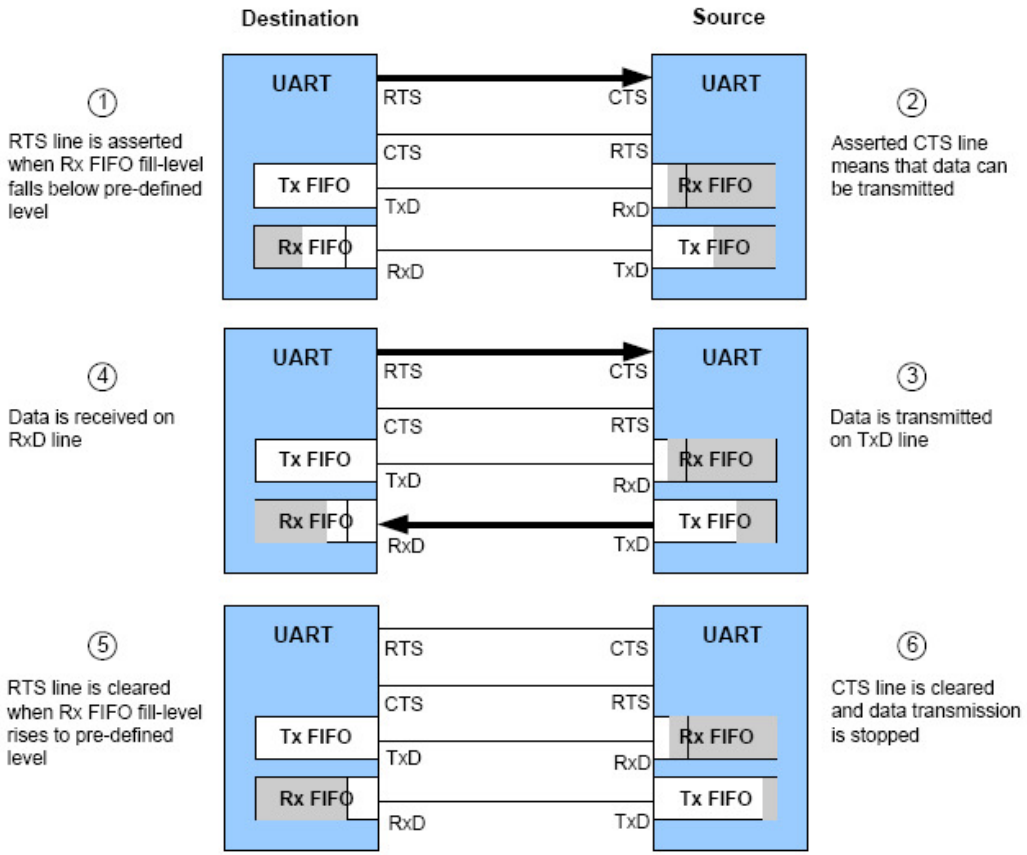
2 Wire (Asincrono)



Nessuna informazione di sincronismo
Il sincronismo viene ricavato direttamente dalla linea dati
Possibilità di errori causati dallo sfasamento fra trasmettitore
E ricevitore

UART – Interface

4 Wire (sincronizzazione)



UART – Interface RS232