# IO Interfacing

C R Sarma
28th June 2010

# Input Output Ports-1

- Microprocessors can have Two addresses spaces
  - using the same address bus
  - using a special control signal to identify the selected address space
- Memory Mapped IO Ports
  - common address space to both IO & Memory mapping
  - Does not need special instructions
  - contiguous memory is broken
- IO Mapped IO Ports
  - special instructions are needed eg. Out dx,al
  - special signal for decoding the memory and io devices which share the same address lines
  - The break in memory continuity is avoided

# Input Output Ports-2

- Simple IO
  - Registers are associated with ports for data transfer
  - control of port direction is done by one register only
  - Direction of IO is byte oriented (eg. 8255 PPI)

- Bit Oriented IO
  - Registers are associated with ports for data transfers
  - control of port direction is done with Data Direction Registers for each port and each of its bits (eg. AVR & PIC micons)

- Direct IO
  - Here each port is associated with a register
  - SET the bit which is to be input – Read it
  - Output to bit if it is to be output. ( eg. 80x51 & 89x51)

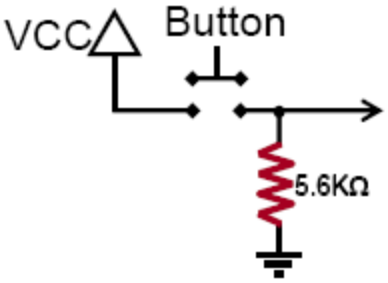
# Input Output Ports-3

Simple IO

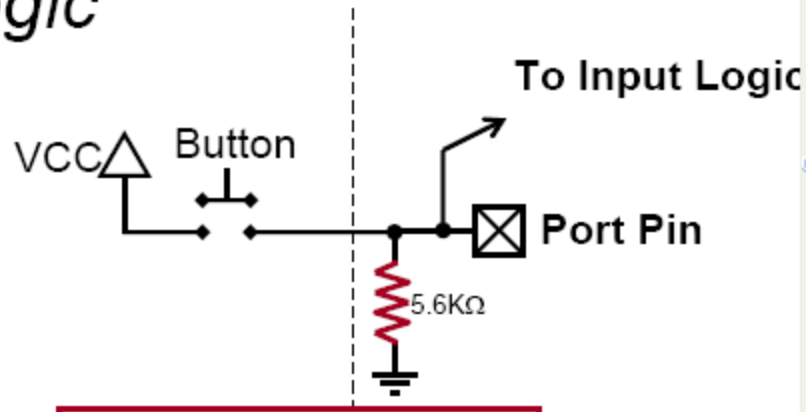
- Input ports are in tristate condition
- – so they are floating
- – if they are read they will not give a steady value
- - never leave them free use a pullup (logic 1) or

*Use a pull-up/down resistor, GND,*

*or internal programmable logic*




Button produces either Vcc or Floating input. Adding a pull-down resistor fixes it.



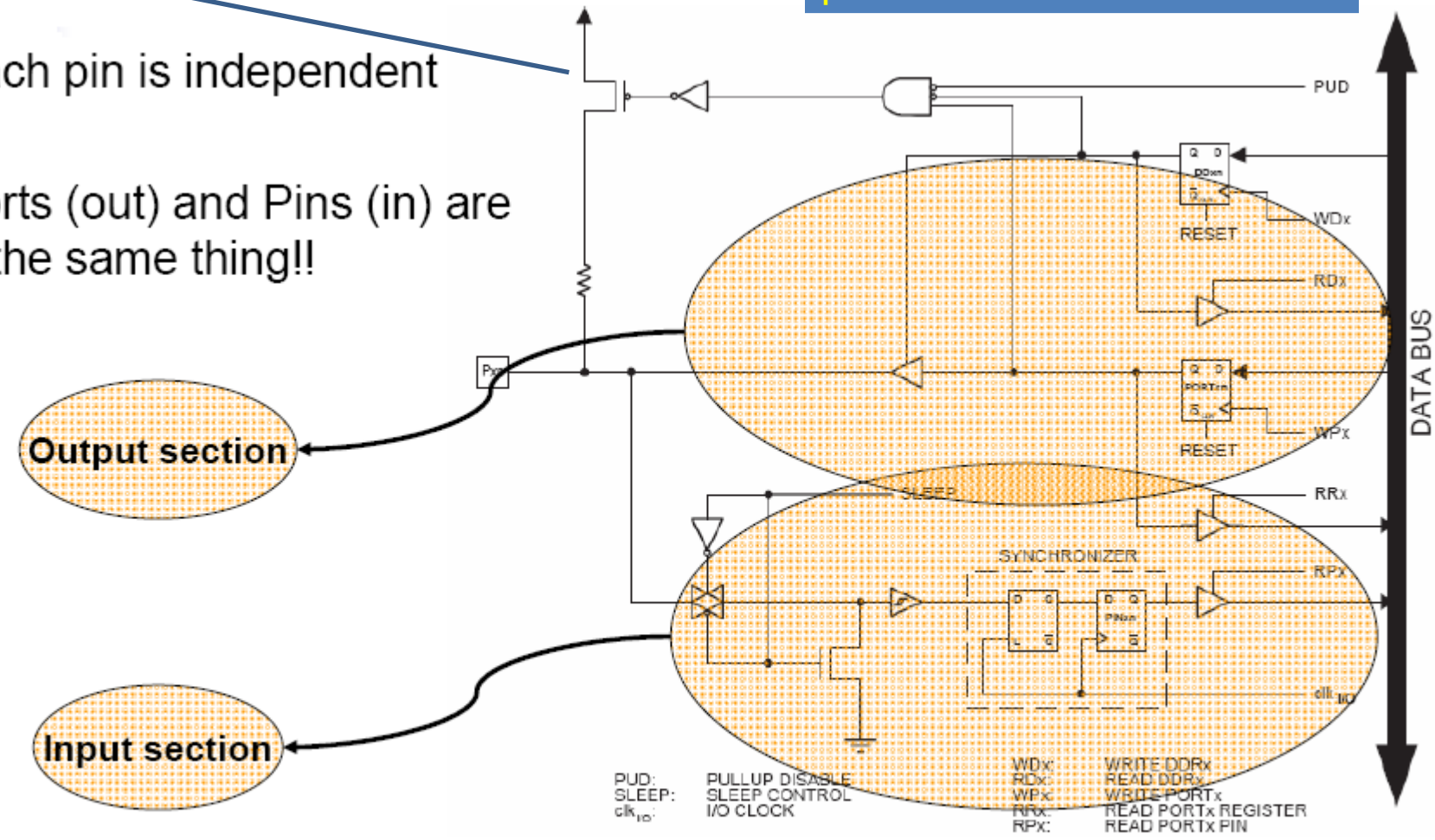

Some ports have internal programmable resistors

# Input Output Ports-4

Active Pull Up
Can be used to tri-state the bit

DDR used in AVR & PIC micons control the Direction and logis state of individual pin

- Each pin is independent
- Ports (out) and Pins (in) are not the same thing!!



# LCD Module Interfacing -1

The Pin assignment – for back lighting 15 is Anode 16 is Kathode

| Pin number | Symbol | Level | I/O | Function |
|---|---|---|---|---|
| 1 | Vss | - | - | Power supply (GND) |
| 2 | Vcc | - | - | Power supply (+5V) |
| 3 | Vee | - | - | Contrast adjust |
| 4 | RS | 0/1 | I | 0 = Instruction input<br>1 = Data input |
| 5 | R/W | 0/1 | I | 0 = Write to LCD module<br>1 = Read from LCD module |
| 6 | E | 1, 1->0 | I | Enable signal |
| 7 | DB0 | 0/1 | I/O | Data bus line 0 (LSB) |
| 8 | DB1 | 0/1 | I/O | Data bus line 1 |
| 9 | DB2 | 0/1 | I/O | Data bus line 2 |
| 10 | DB3 | 0/1 | I/O | Data bus line 3 |
| 11 | DB4 | 0/1 | I/O | Data bus line 4 |
| 12 | DB5 | 0/1 | I/O | Data bus line 5 |
| 13 | DB6 | 0/1 | I/O | Data bus line 6 |
| 14 | DB7 | 0/1 | I/O | Data bus line 7 (MSB) |

**Command RS=0, Data RS=1**

# LCD Module Interfacing -2 The Instruction Set

| Instruction | Code | | | | | | | | | | Description | Execution time** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | | |
| Clear display | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Clears display and returns cursor to the home position (address 0). | 1.64mS |
| Cursor home | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | * | Returns cursor to home position (address 0). Also returns display being shifted to the original position. DDRAM contents remains unchanged. | 1.64mS |
| Entry mode set | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S | Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write. | 40uS |
| Display On/Off control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | Sets On/Off of all display (D), cursor On/Off (C) and blink of cursor position character (B). | 40uS |
| Cursor/display shift | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | * | * | Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM contents remains unchanged. | 40uS |
| Function set | 0 | 0 | 0 | 0 | 1 | DL | N | F | * | * | Sets interface data length (DL), number of display line (N) and character font(F). | 40uS |
| Set CGRAM address | 0 | 0 | 0 | 1 | CGRAM address | | | | | | Sets the CGRAM address. CGRAM data is sent and received after this setting. | 40uS |
| Set DDRAM address | 0 | 0 | 1 | DDRAM address | | | | | | | Sets the DDRAM address. DDRAM data is sent and received after this setting. | 40uS |
| Read busy-flag and address counter | 0 | 1 | BF | CGRAM / DDRAM address | | | | | | | Reads Busy-flag (BF) indicating internal operation is being performed and reads CGRAM or DDRAM address counter contents (depending on previous instruction). | 0uS |
| Write to CGRAM or DDRAM | 1 | 0 | write data | | | | | | | | Writes data to CGRAM or DDRAM. | 40uS |
| Read from CGRAM or DDRAM | 1 | 1 | read data | | | | | | | | Reads data from CGRAM or DDRAM. | 40uS |

**Remarks:**
- DDRAM = Display Data RAM.
- CGRAM = Character Generator RAM.
- DDRAM address corresponds to cursor position.
- * = Don't care.

**Command RS=0, Data RS=1**

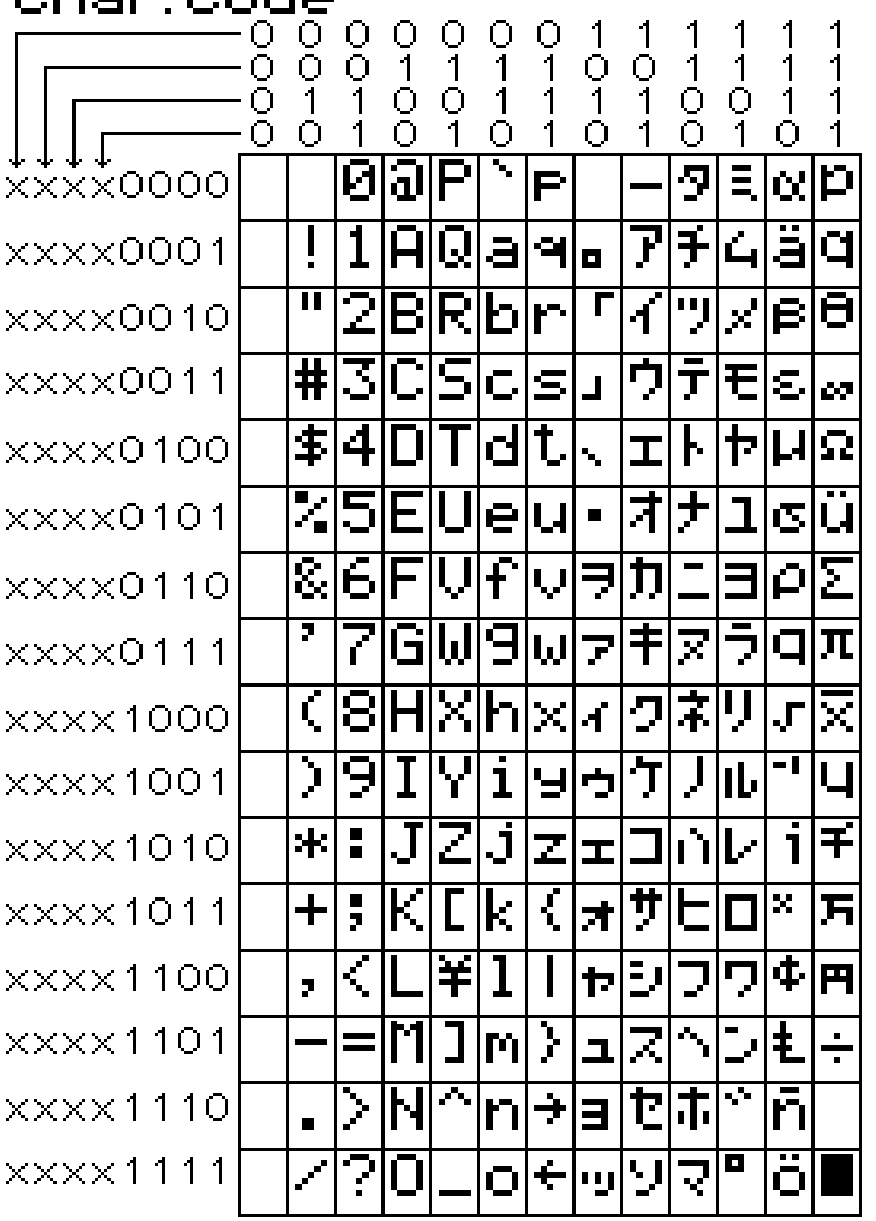
Table 2.4: Bit names

| Bit name | Setting / Status | |
|---|---|---|
| I/D | 0 = Decrement cursor position | 1 = Increment cursor position |
| S | 0 = No display shift | 1 = Display shift |
| D | 0 = Display off | 1 = Display on |
| C | 0 = Cursor off | 1 = Cursor on |
| B | 0 = Cursor blink off | 1 = Cursor blink on |
| S/C | 0 = Move cursor | 1 = Shift display |
| R/L | 0 = Shift left | 1 = Shift right |
| DL | 0 = 4-bit interface | 1 = 8-bit interface |
| N | 0 = 1/8 or 1/11 Duty (1 line) | 1 = 1/16 Duty (2 lines) |
| F | 0 = 5x7 dots | 1 = 5x10 dots |
| BF | 0 = Can accept instruction | 1 = Internal operation in progress |

# LCD Module Interfacing -3 Data Ram and Character Set

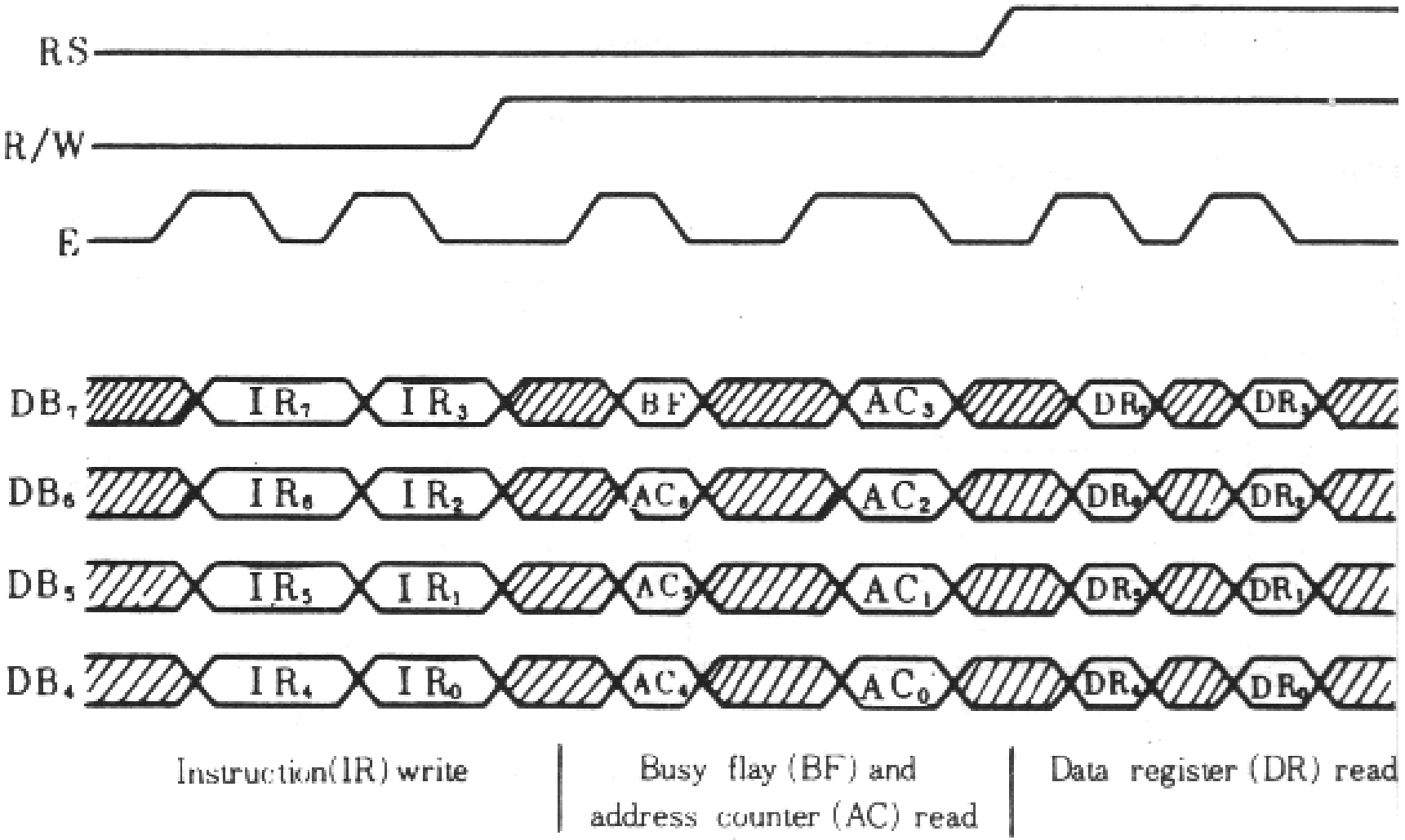
DDRAM address usage for a 2-line LCD

| Display size | Visible Character positions | DDRAM addresses |
|---|---|---|
| 2*16 | 00..15 | 0x00..0x0F + 0x40..0x4F [1] |
| 2*20 | 00..19 | 0x00..0x13 + 0x40..0x53 |
| 2*24 | 00..23 | 0x00..0x17 + 0x40..0x57 |
| 2*32 | 00..31 | 0x00..0x1F + 0x40..0x5F |
| 2*40 | 00..39 | 0x00..0x27 + 0x40..0x67 |

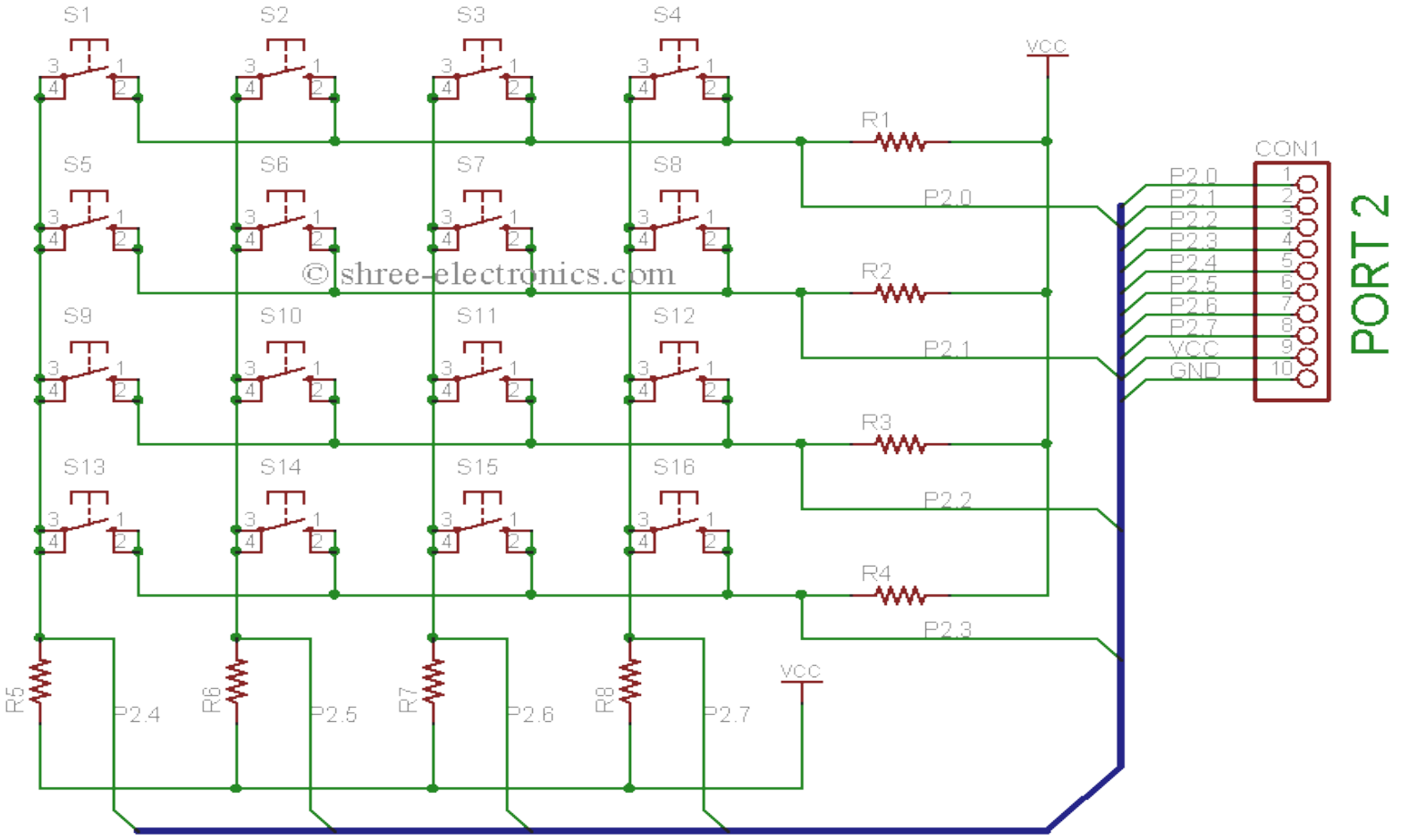


# LCD Module Interfacing -4 Interface Signals

**4 bit interface**



# KeyBoard Interfacing




P2.0-P2.3 scan, P2.4-P2.6 sense lines.

Write 0xFE to P2 – read P2 –Mask 0xf0 – SHR 4 times – test if 0x0f-yes then do the next scan row

Else copy to register and add row val now 0

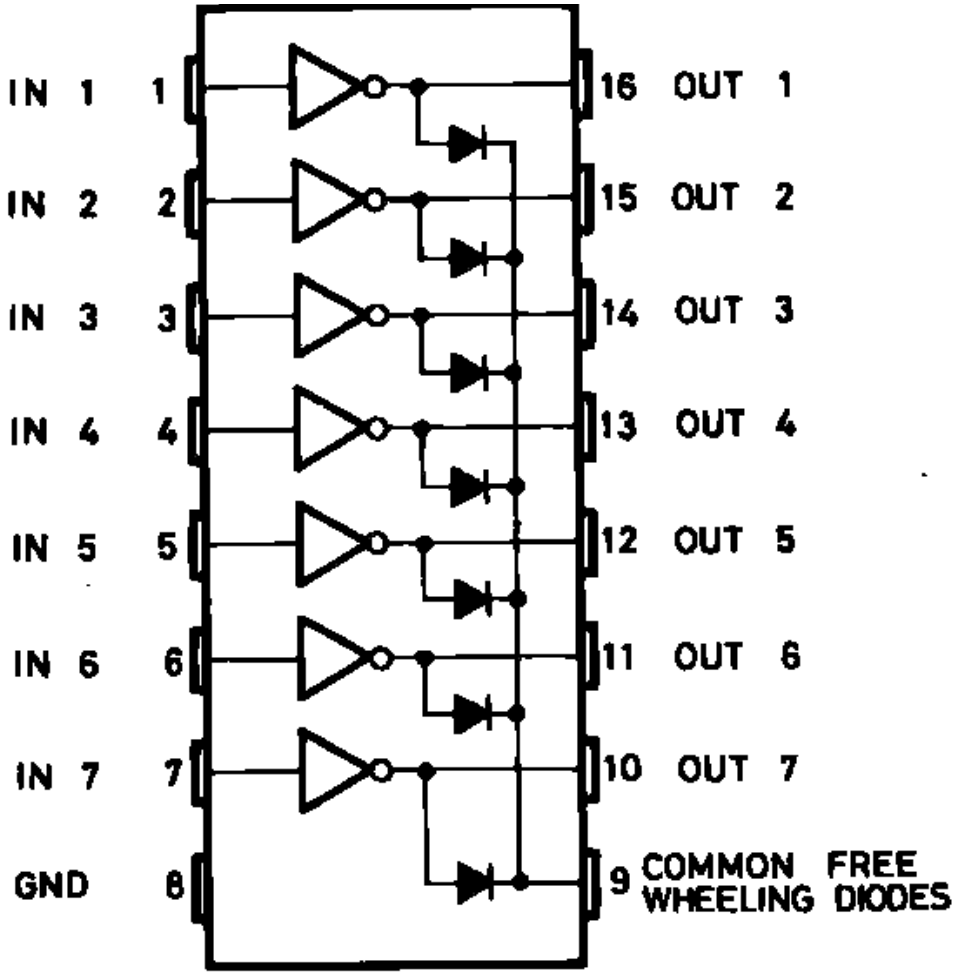
For next row add row val 4 and scan with P2<-0xFD, Read if key pressed - to key pressed add key val.

# Port Drive Boost-1

- Interfaces are not just Keyboards and LCD Modules
- Power Drive is necessary also
- IO ports cannot drive High Power
- High Power is something which needs more than 500 mA
- The IO ports can supply atmost 1 TTL load- less than 10mA
- SO there is a need for Power Drivers
- Transistors and VMOSFETS could be used
- They are discrete -take lots of space- so here it is

# Port Drive Boost-2
# ULN 2003A- Darlington solenoid driver – 7 way

•.SEVEN DARLINGTONS PER PACKAGE
• .OUTPUT CURRENT 500mA PER DRIVER
(600mA PEAK) .
•OUTPUT VOLTAGE 50V .
•INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS .
•OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
• .TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS .
•INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT

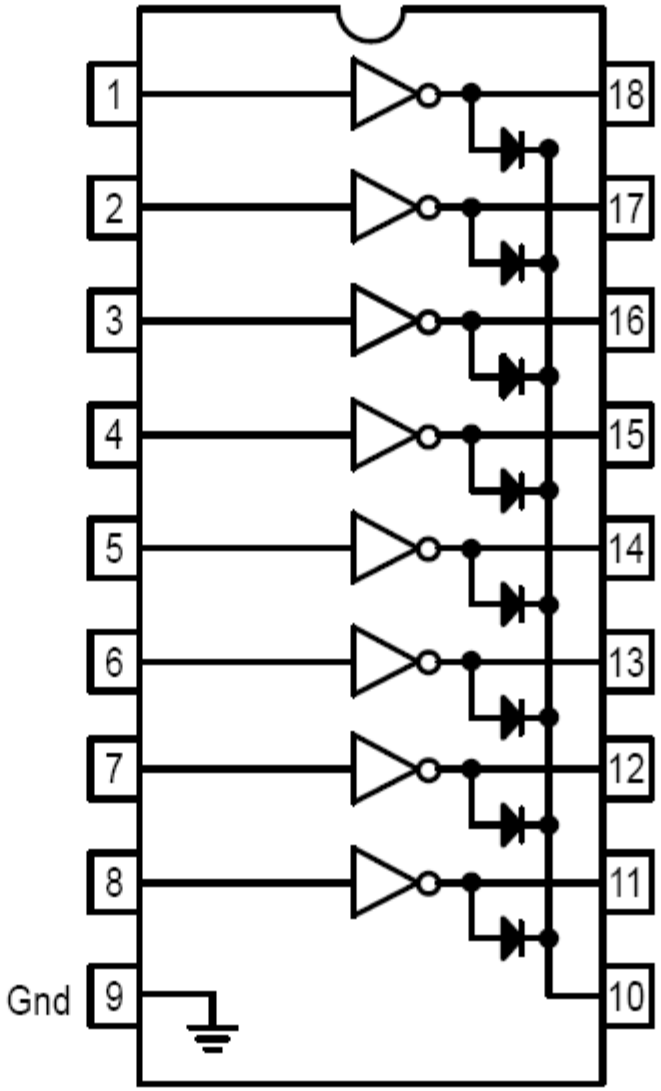



These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays,filament lamps, thermal printheads and high power buffers.

# Port Drive Boost-3
# ULN 2803A- Darlington solenoid driver – 8 way

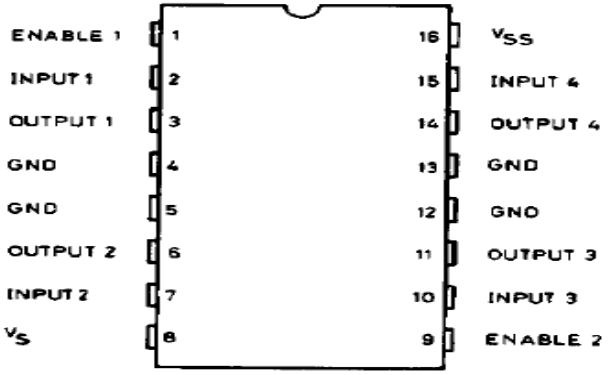
The eight NPN Darlington connected transistors in this family of arrays are ideally suited for interfacing between low logic level digital circuitry (such as TTL, CMOS or PMOS/NMOS) and the higher current/voltage requirements of lamps, relays, printer hammers or other similar loads for a broad range of computer, industrial, and consumer applications. All devices feature open–collector outputs and free wheeling clamp diodes for transient suppression.

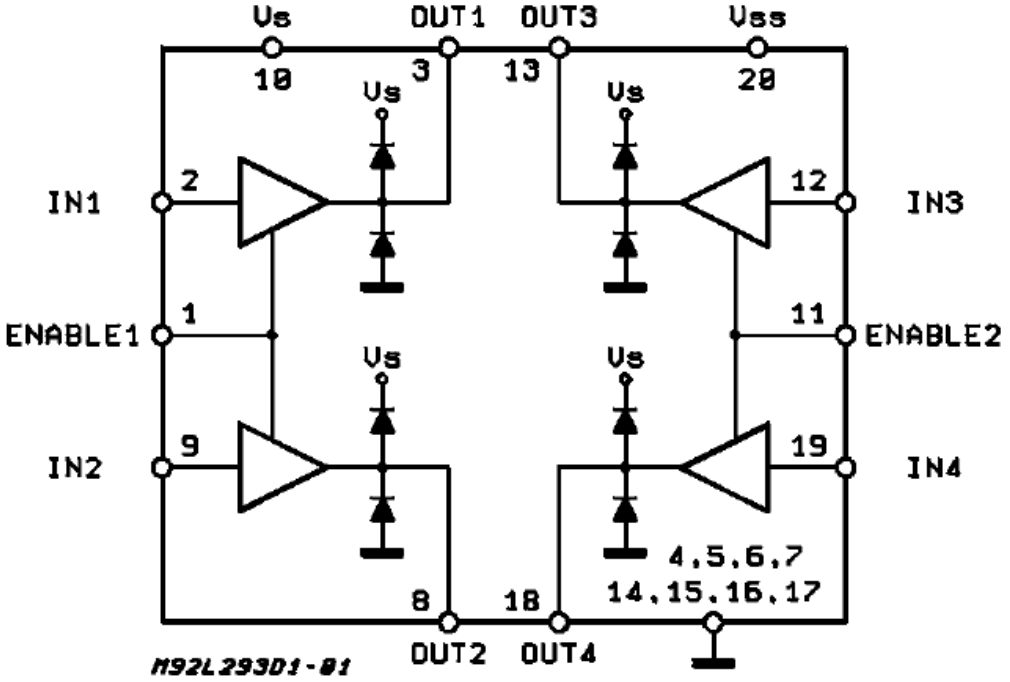
The ULN2803 is designed to be compatible with standard TTL families while the ULN2804 is optimized for 6 to 15 volt high level CMOS or PMOS.



# Port Drive Boost-4
# L293D Motor Driver

- PER CHANNEL 1.2A PEAK OUTPUT CURRENT (non repetitive)
- PER CHANNEL ENABLE FACILITY
- OVERTEMPERATUREPROTECTION
- LOGICAL ”0” INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES



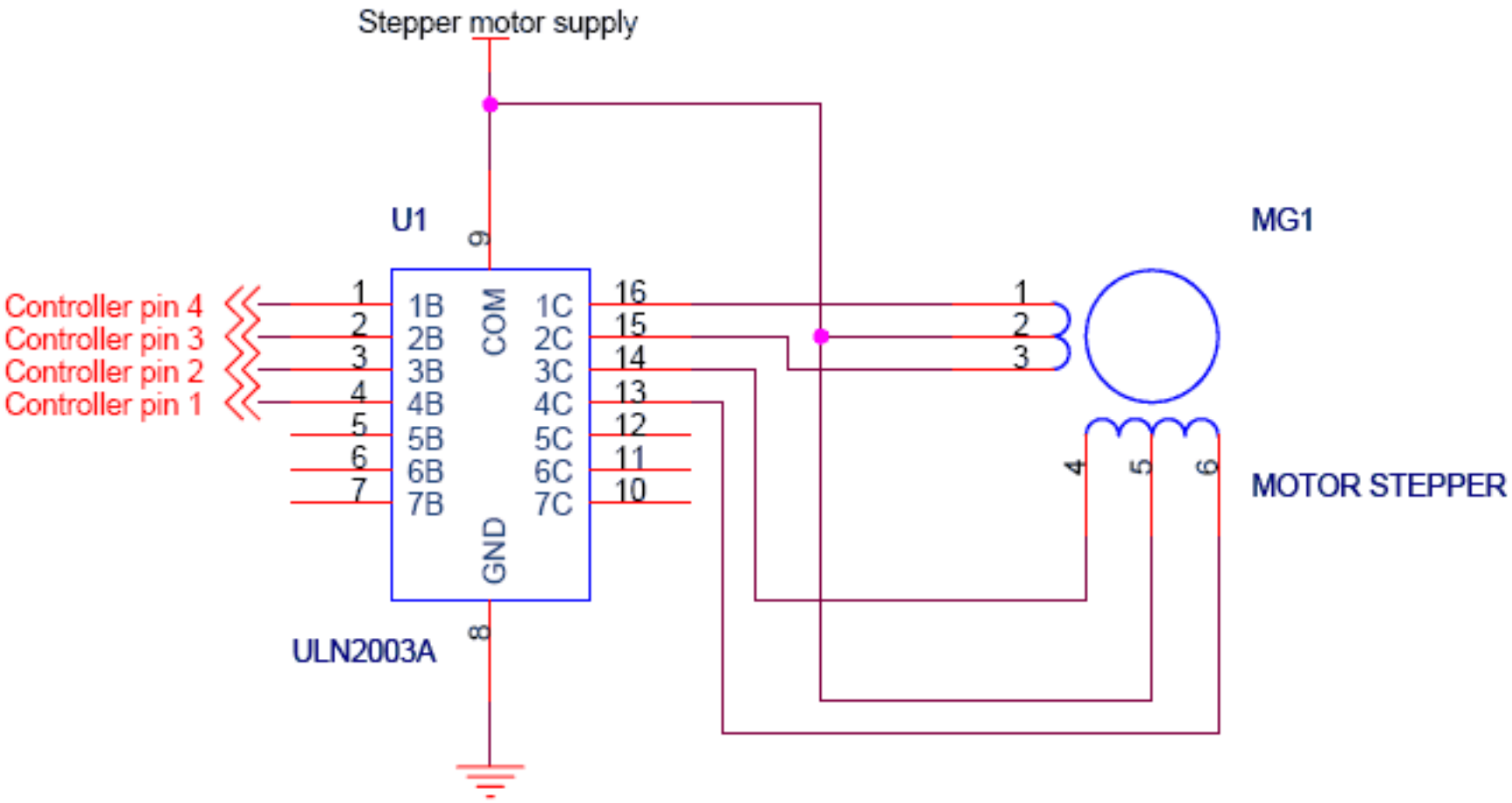




It is a monolithic integrated high voltage, high current four channel driver - accepts standard DTL or TTL logic levels and driveinductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

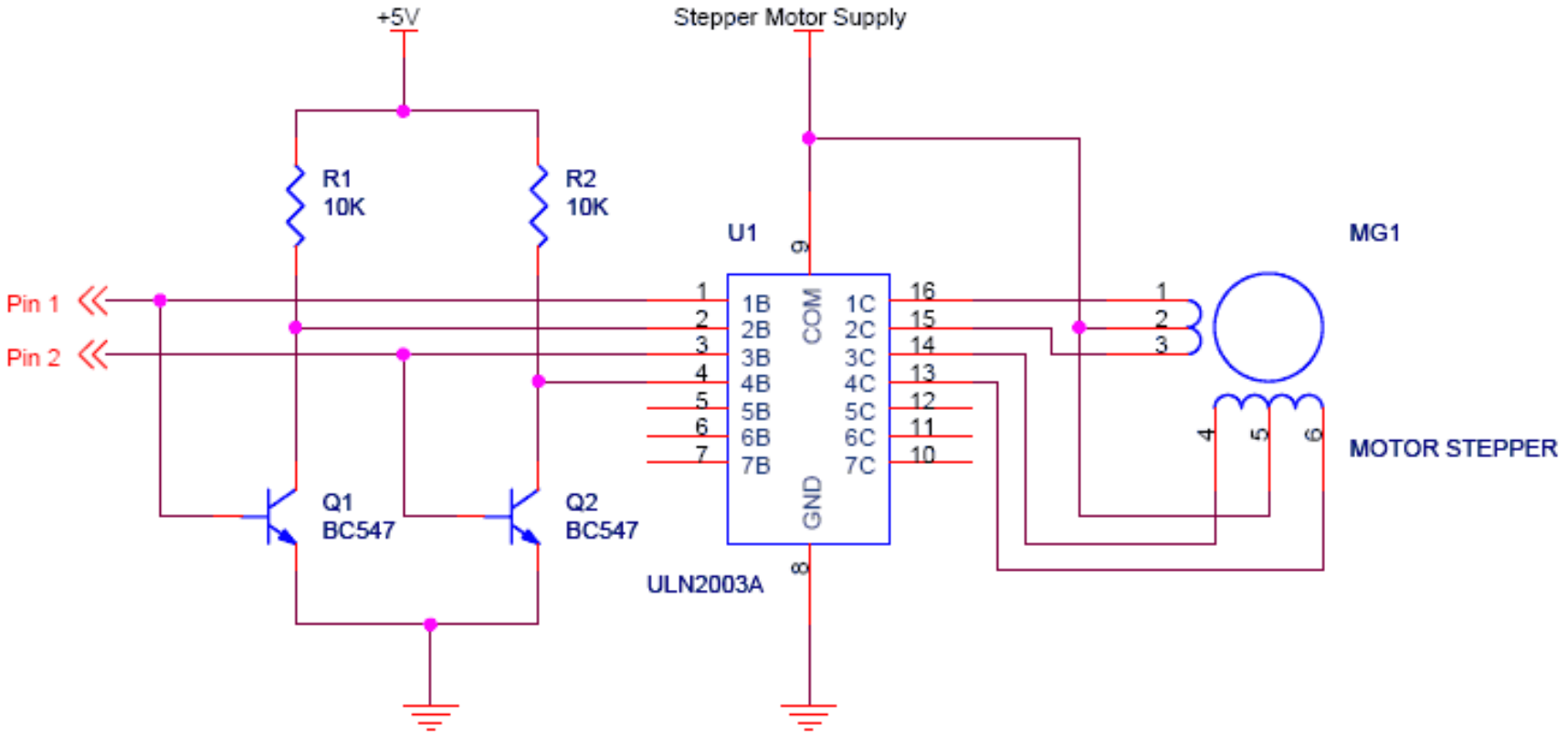
This device is suitable for use in switching applications at frequencies up to 5 kHz

As already discussed in case of L293D, Here in this circuit too the four pins "Controller pin 1",2,3 and 4 will control the motion and direction of the stepper motor according to the step sequnece sent by the controller.



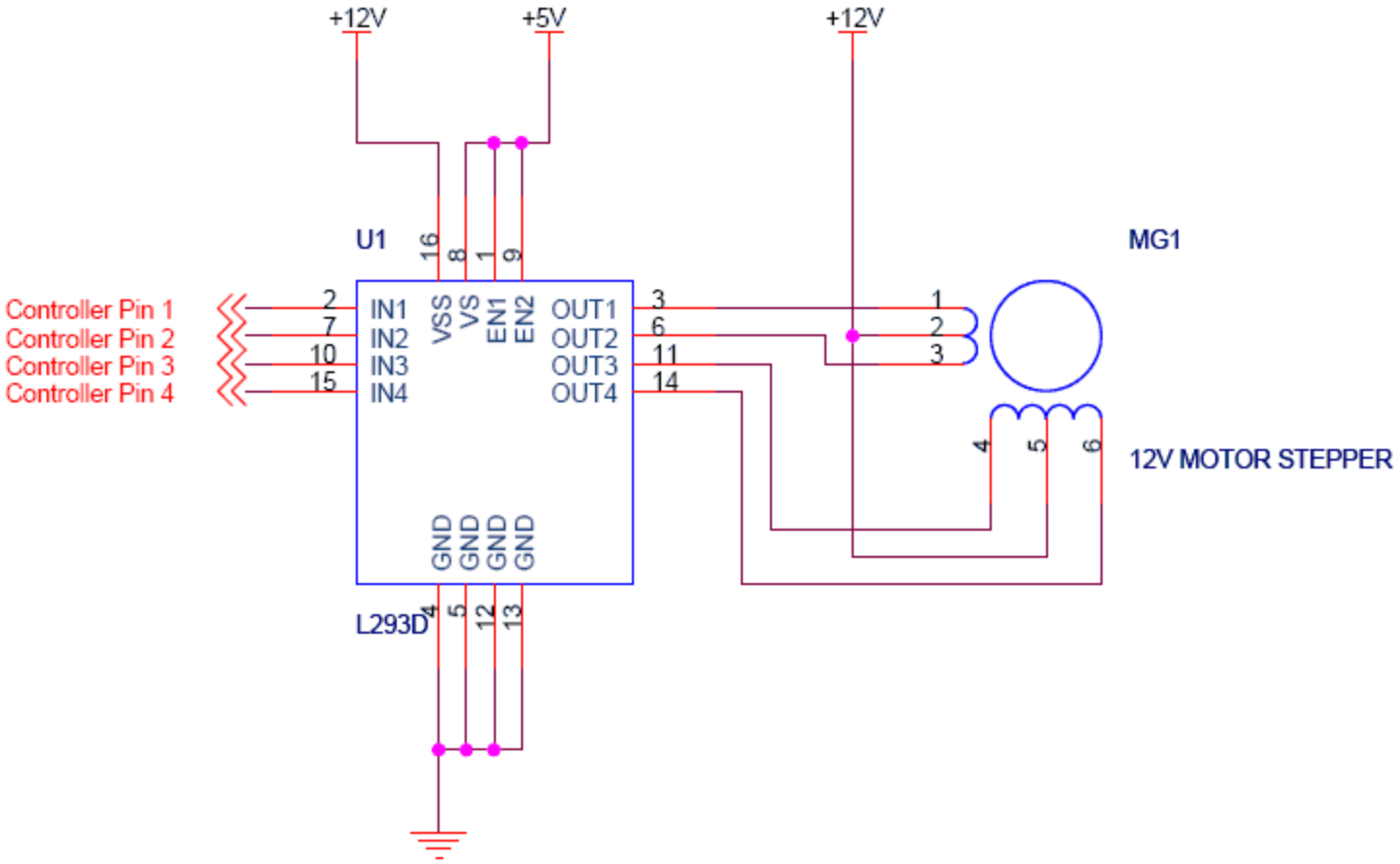
# 2-wire connection for Unipolar Stepper Motor

We have seen the generally used 4-wire connection method for interfacing unipolar stepper motor, but we can simplify the design to make controller use less pins with the help of 2-wire connection method. The circuit for 2-wire connection is shown below.

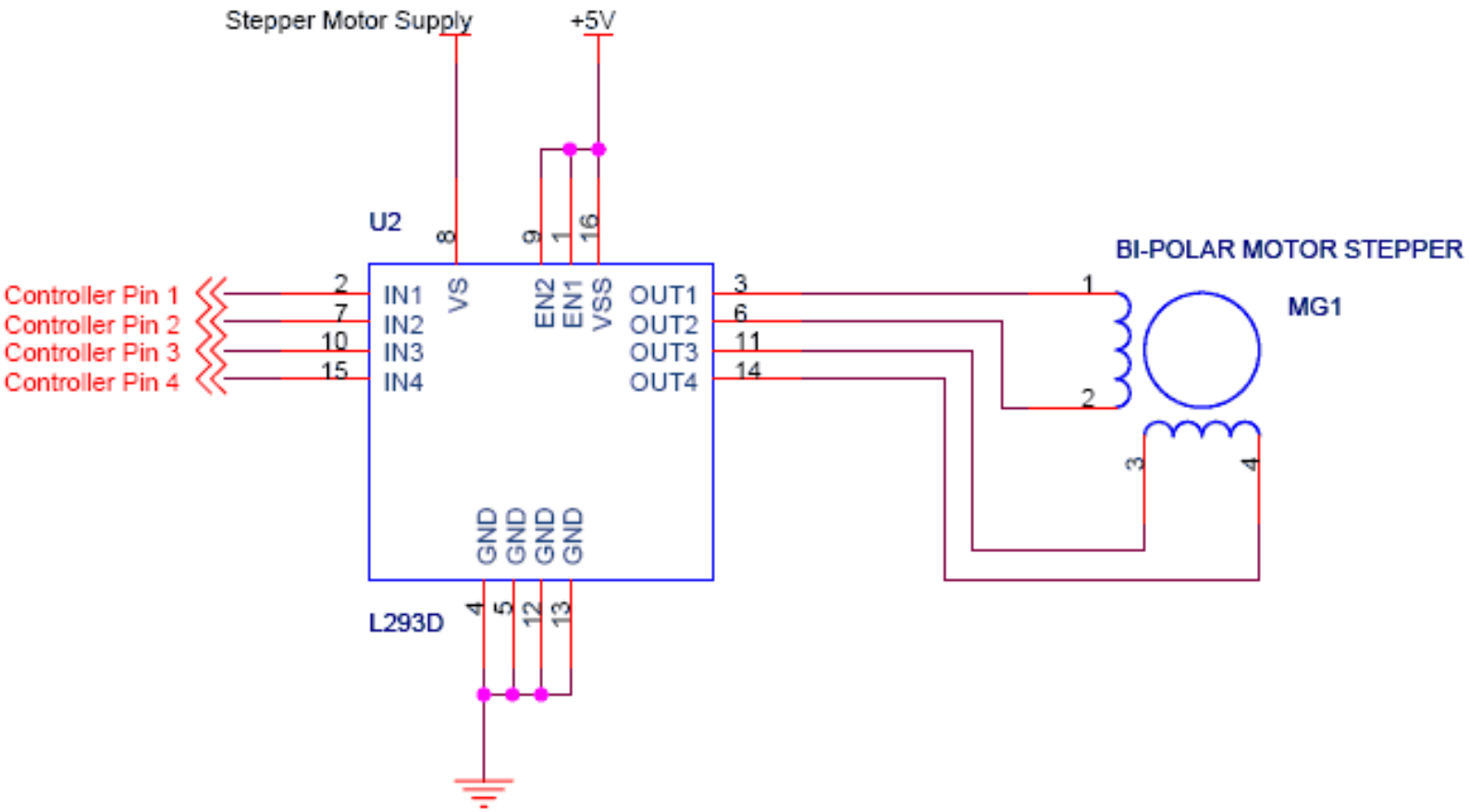


# Connecting Unipolar stepper using L293D

In the circuit - the four pins "Controller pin s”1,2,3 and 4 will control the motion and direction of the stepper motor according to the step sequence programmed in the controller.



As we have studied that, Bi-polar stepper motors has 2 different coils. The step sequence for Bipolar stepper motor is same as that of unipolar stepper motors. The driving circuit for this require an H-Bridge as it allows the polarity of the power applied to be controlled independently. This can be done as shown in the figure below:



# Serial Busses - !RS232

- Displays and Power Drivers are not enough
- Small foot print devices are needed
- Less complicated bussing – avoid a crows nest

## Purpose of Serial Interconnect Buses

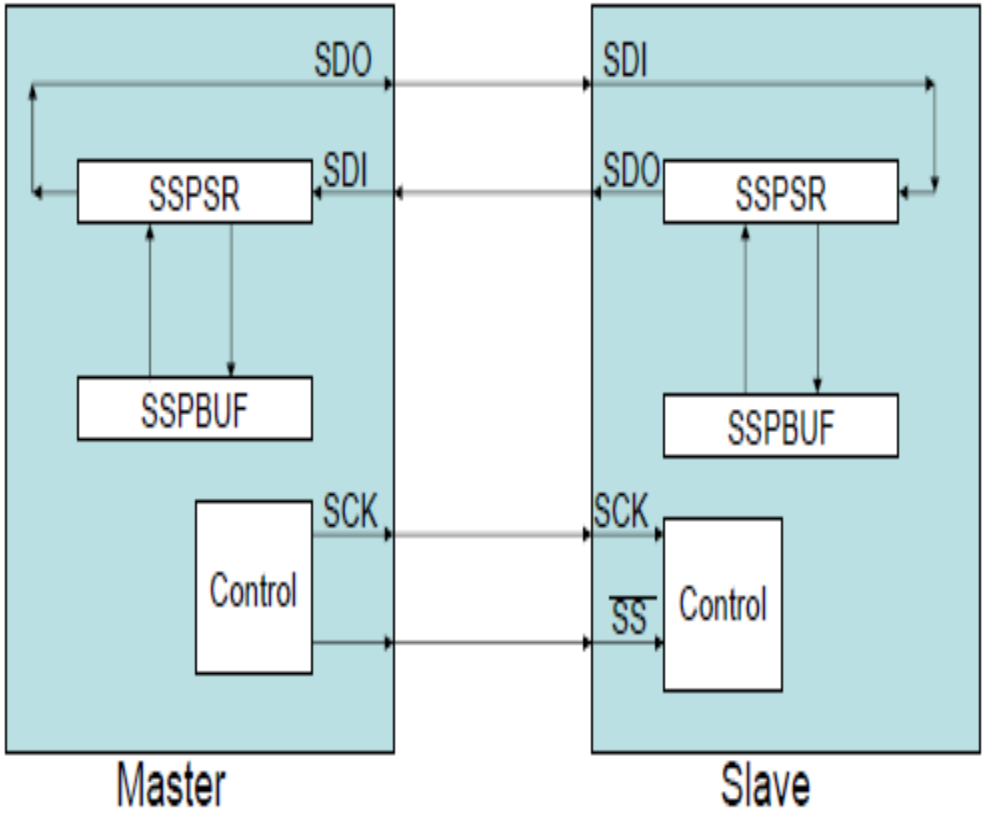
- Provide low-cost—i.e low wire/pin count—connection between IC devices
- There are lots of serial bus "standards"
  - $I^2C$
  - SMB
  - SPI
  - Microwire
  - Maxim 3-wire
  - Maxim/Dallas 1-wire
  - etc.

# SPI-1

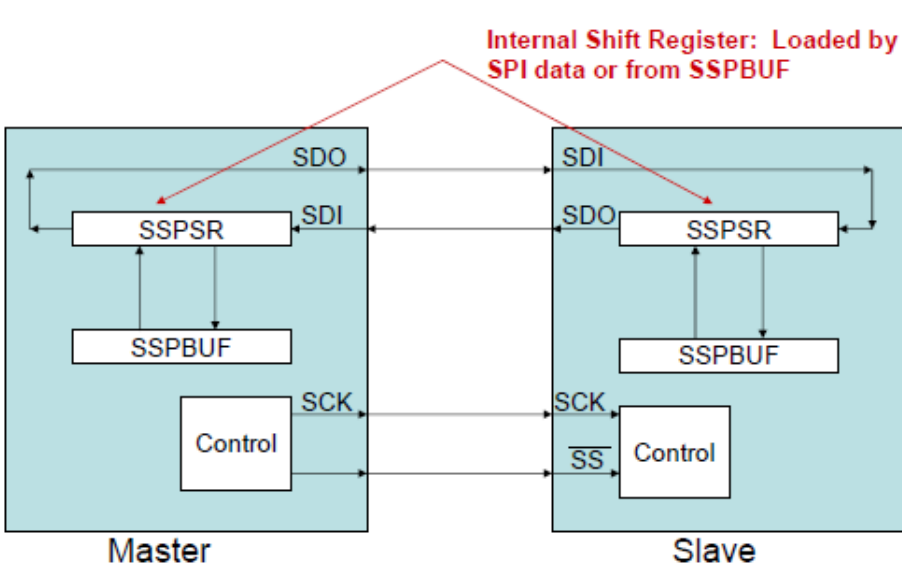
## SPI signals

- $\overline{SS}$ ($\overline{CS}$) (Slave Select, Chip Select)
  - When SS is low the slave is enabled
- SCK (Serial Clock)
  - Controls the sending and reading of data
- SDO (Serial Data Out)
  - Carries data OUT of the device
- SDI (Serial Data In)
  - Carries data INTO the device

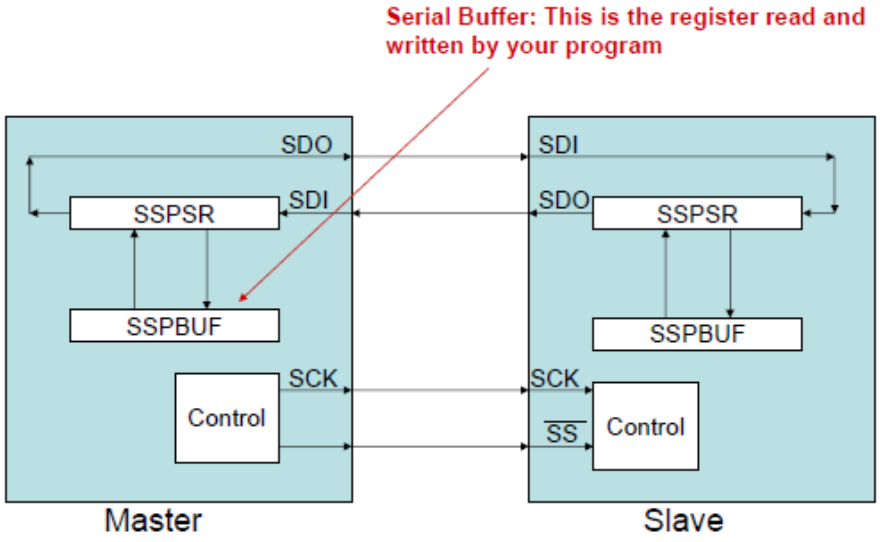
## SPI Data Loop



# SPI Data Loop

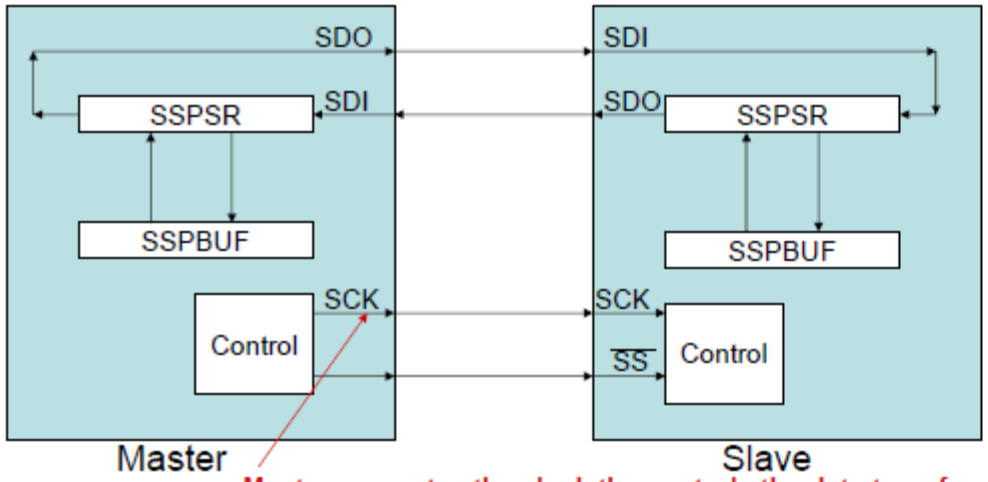



# SPI Data Loop



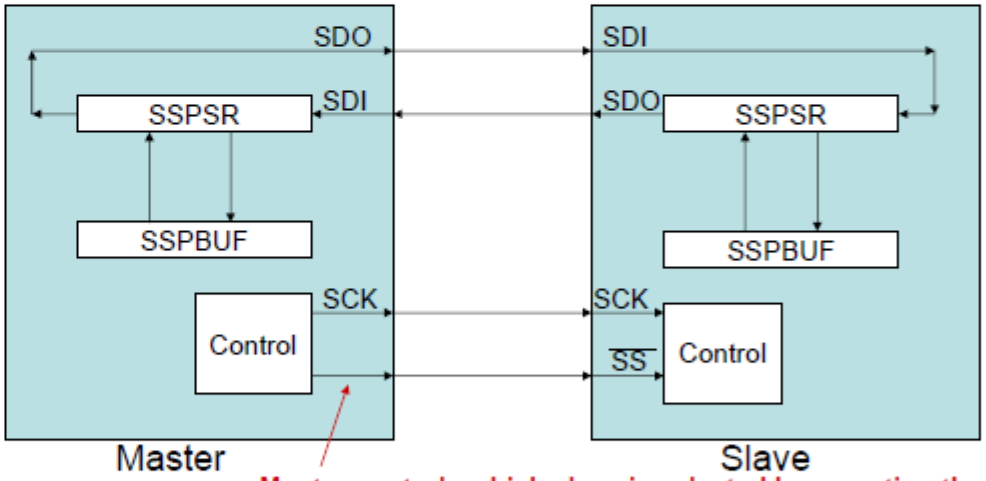
# SPI-3

## SPI Data Loop




Master generates the clock the controls the data transfer

## SPI Data Loop




Master controls which slave is selected by asserting the slave's **SS**

# SPI-4
# An ADC example

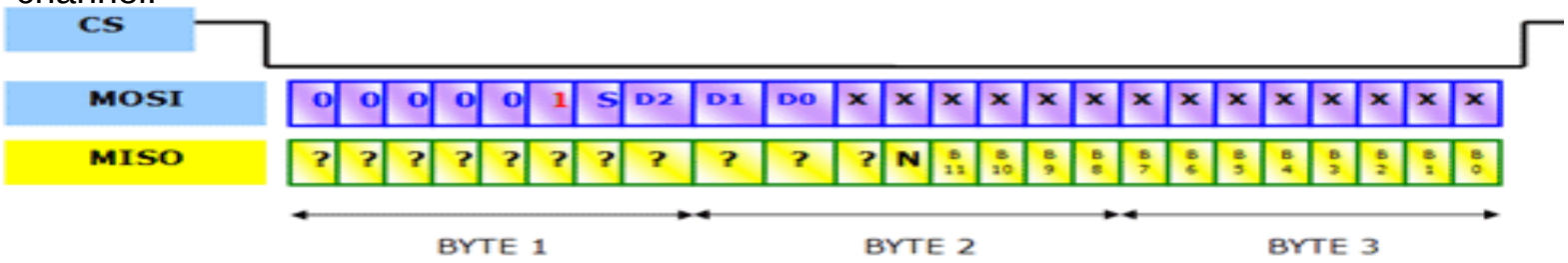
The SPI Packet for one ADC Conversion is made up of 3 bytes. The complete transaction is shown below.

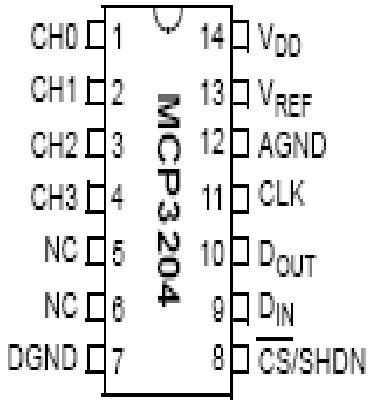
**Byte I**

In this byte the Master Writes a bit sequence as shown below
'0' '0' '0' '0' '0' '1' 'S' 'D2'
Where S is the bit which selects between differential and single ended operation. In our example we need single ended operation and for that this bit must be 1.

**Byte II**

In this byte MASTER Writes the following sequence 'D1' 'D0' 'X' 'X' 'X' 'X' 'X 'X‘ Where D2,D1,D0 selects the input channel.

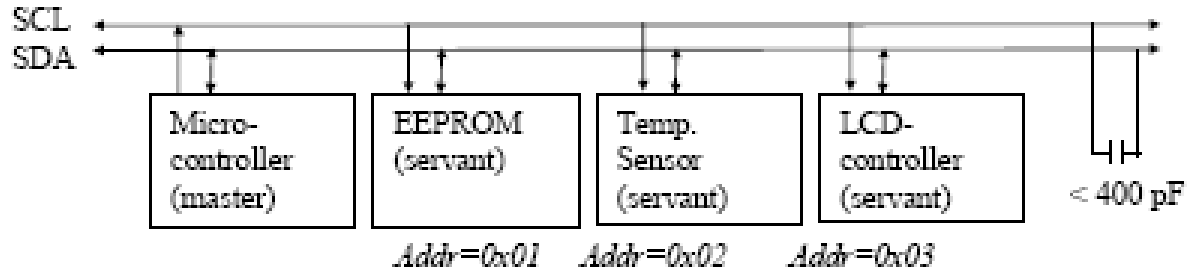


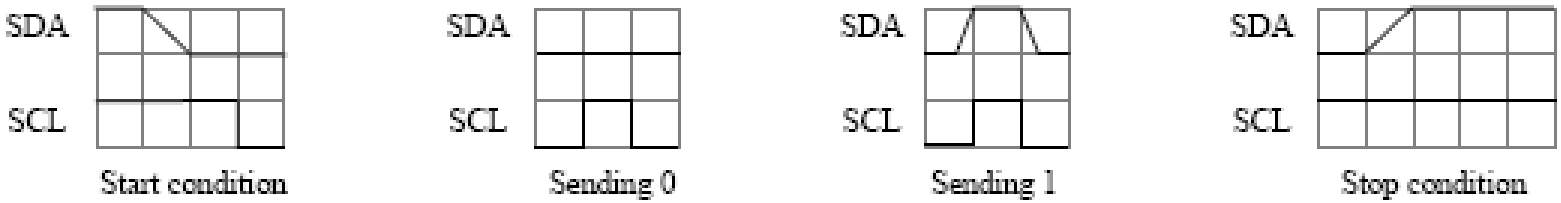



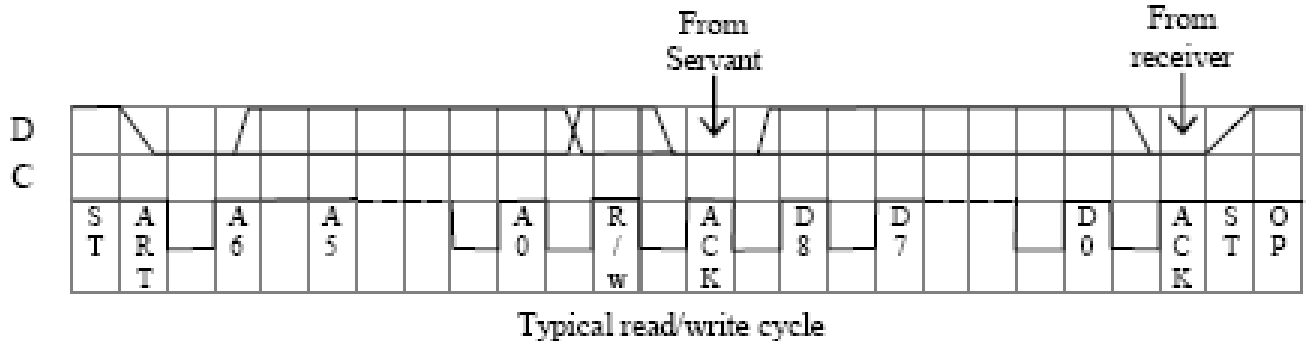

**Byte III**

In this byte the Master writes a DON'T CARE Byte to slave. You can write any value it does not matters.
In the same time Slave returns bits B7 to B0 of conversion.
So Bit B11 to B0 forms the 12bit result of Analog to Digital Conversion. Following Code Example demonstrate the complete transaction.

# I2C bus structure



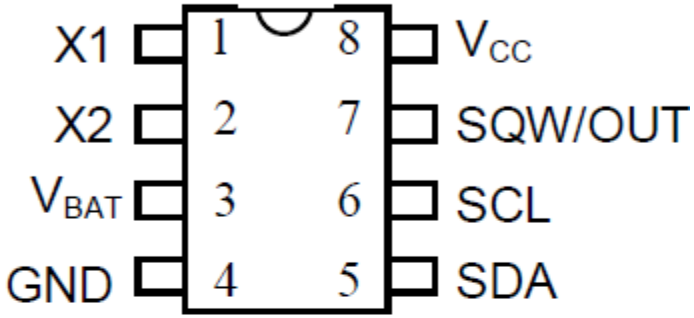

Typical read/write cycle

# I2C devices

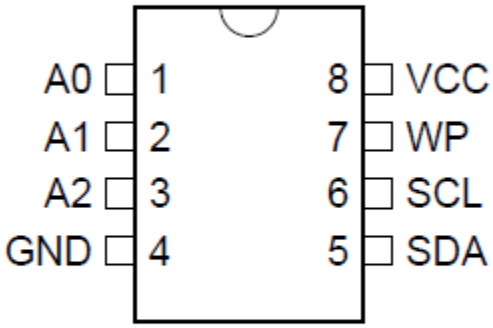
- DS1307
- Real time clock counts seconds, minutes,
- hours, date of the month, month, day of the
- week, and year with leap year compensation
- valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2-wire serial interface
- Programmable squarewave output signal

## 24C256 – 64KX8 bit EEPROM

**Bidirectional Data Transfer Protocol**
**• 100 kHz (1.8V) and 400 kHz (2.5V) Clock Rate for AT24C32A**
**• 400 kHz (1.8V) Clock Rate for AT24C64A**
**• Write Protect Pin for Hardware Data Protection**
**• 32-Byte Page Write Mode (Partial Page Writes Allowed)**
**• Self-Timed Write Cycle (5 ms Max)**
**• High Reliability**
**– Endurance: 1 Million Write Cycles**
**– Data Retention: 100 Years**




DS1307 8-Pin DIP (300 mil)



# JTAG Interfacing

- **IEEE Std 1149.1-1990** JTAG (Joint Test Action Group); Test Access Port and Boundary-Scan Architecture.
- This is a serial bus with four signals: **Test Clock** (TCK), **Test Mode Select** (TMS), **Test Data Input** (TDI), and **Test Data Output** (TDO). To use JTAG, during the design, you most select JTAG compatible devices. ICs supporting JTAG will have the four additional pins listed above. Boundary-scan tests can be used to check continuity between devices. Continuity checks on PWB nets may be performed by sending out a know pattern and receiving that same pattern at the input to another IC(s).

# JTAG-2

TCK: [Test Clock] has noting to do with the board or system clock. The Test Clock is used to load the test mode data from the TMS pin, and the test data on the TDI pin [on the rising edge]. On the falling edge test clock outputs the test data on the TDO pin.

TMS: [Test Mode Select Input] controls the operation of the test logic, by receiving the incoming data].

TDI: [Test Data Input] receives serial input data which is either feed to the test data registers or instruction register, but depends on the state of the TAP controller. The TDI line has an internal pull-up, so the input is high with no input.

TDO: [Test Data Output] outputs serial data which comes from either the test data registers or instruction register, but depends on the state of the TAP controller.

TRST: [Test Rest] will asynchronously reset the JTAG test logic.

20 Pin JTAG PinOut

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | TRST | 2 | GND |
| 3 | TDO | 4 | GND |
| 5 | TDI | 6 | GND |
| 7 | TMS | 8 | GND |
| 9 | TCK | 10 | GND |
| 11 | VPP_E | 12 | GND |
| 13 | A/W | 14 | GND |
| 15 | User 0 | 16 | GND |
| 17 | Rdy/ Bsy | 18 | GND |
| 19 | User 1 | 20 | Vcc |

# JTAG-2

## 10 Pin Altera ByteBlaster II

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | TCK | 2 | GND |
| 3 | TDO | 4 | Power |
| 5 | TMS | 6 | NC |
| 7 | NC | 8 | NC |
| 9 | TDI | 10 | GND |

## 14 Pin Xilinx Cable IV

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | VGND | 2 | VREF |
| 3 | GND | 4 | TMS |
| 5 | GND | 6 | TCK |
| 7 | GND | 8 | TDO |
| 9 | GND | 10 | TDI |
| 11 | GND | 12 | NC |
| 13 | GND | 14 | NC |