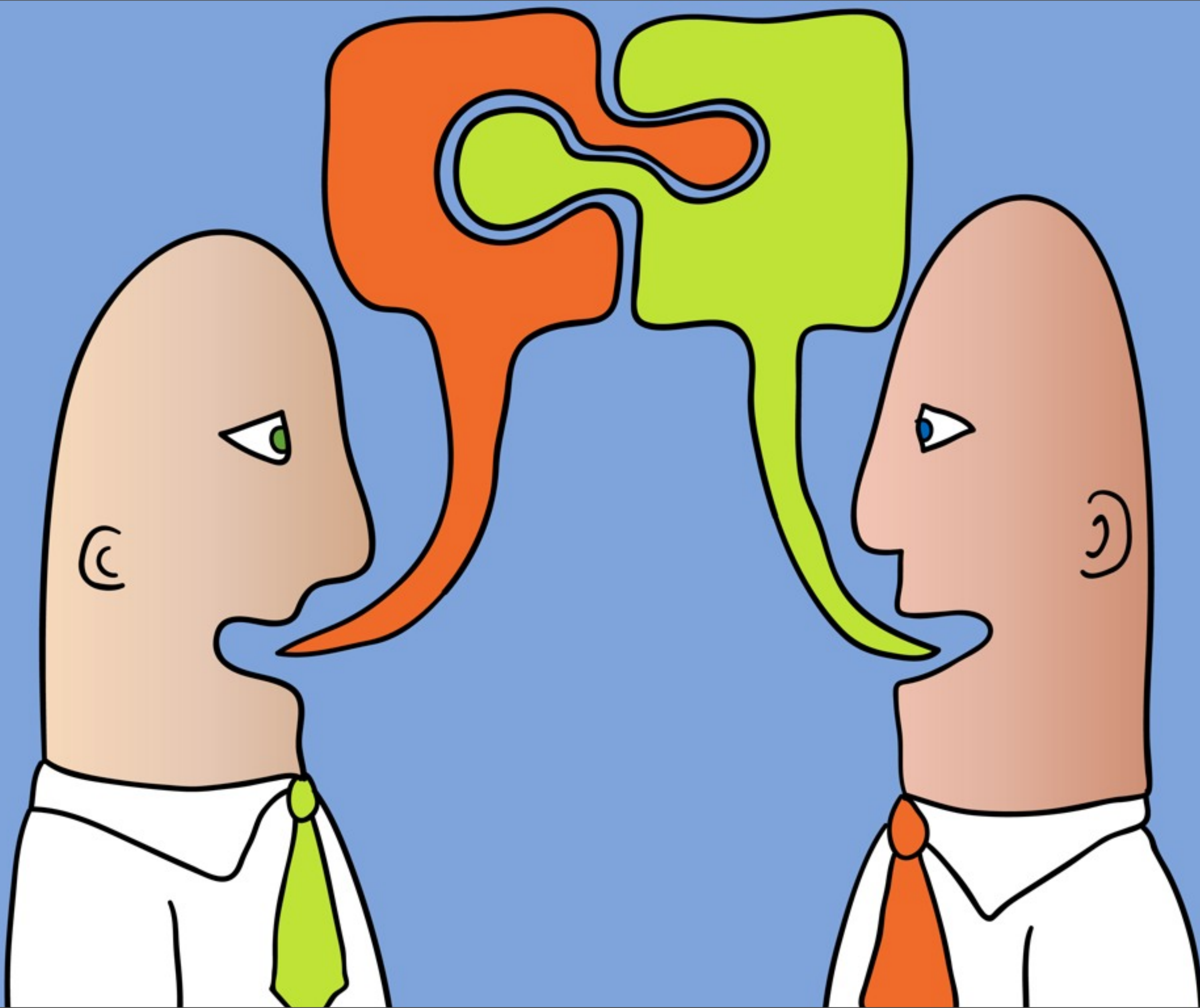


Communication

Press Play: Interactive Device Design | April 25, 2011



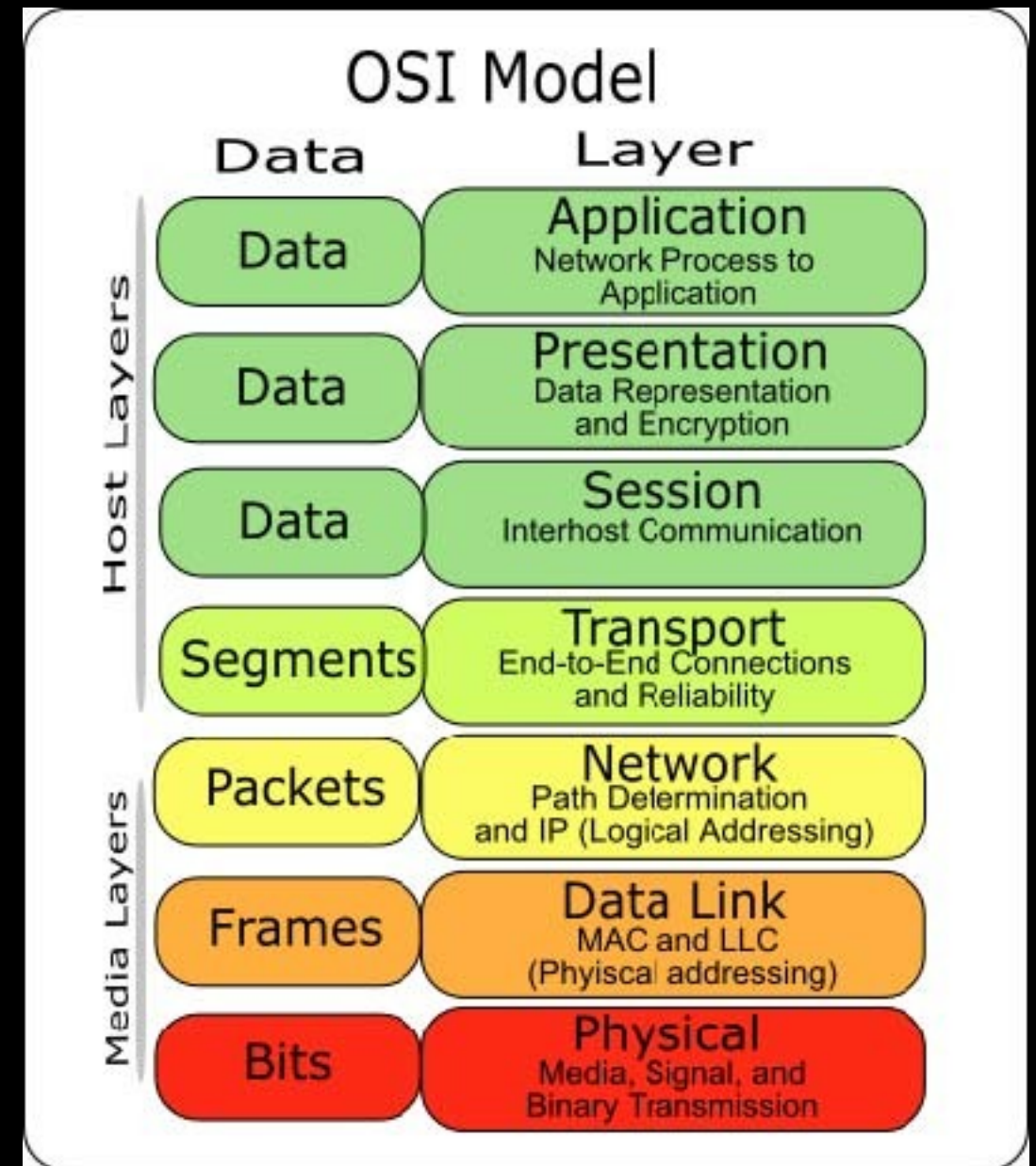
Context of Communication

Conversation - Rules of Conduct

- Communication is holding a conversation
 - Interprocessor communication is **peer-to-peer**
 - Processor to device conversation is **master-slave**
- A protocol is a set of rules of conduct that we agree to uphold during the conversation
 - They govern how we start a conversation, who speaks when, how fast, how often, etc.

Context of Communication

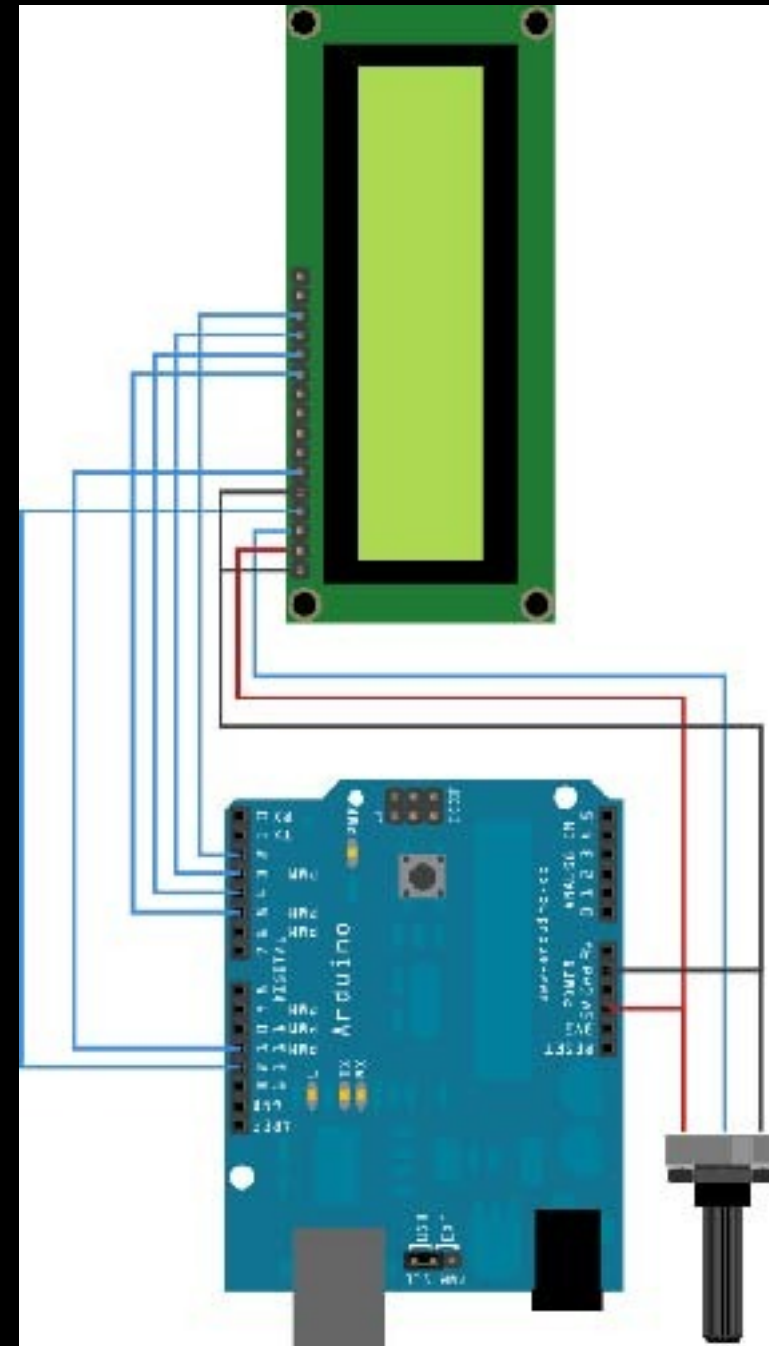
Open Systems Interconnection



Types of Interface

Parallel

- Examples
 - Graphical LCD
 - SCSI, Firewire
- Advantages
 - Faster in Theory
- Drawbacks
 - Crosstalk
 - Clock Skew
 - Wire per Bit



Types of Interface

Serial

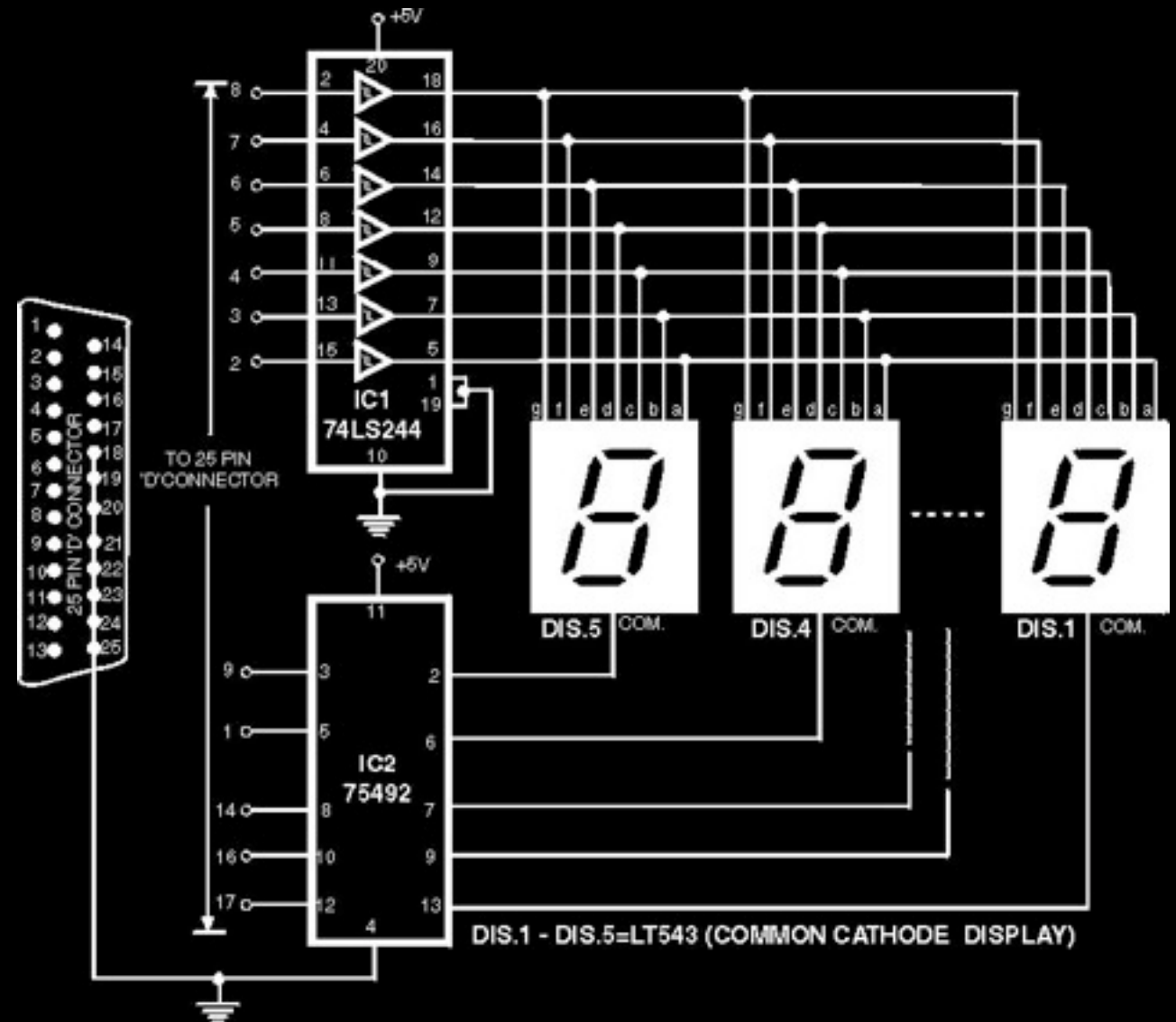
- Examples
 - USB
 - SPI and I²C
- Advantages
 - Clock Faster
 - Fewer Wires
- Drawbacks
 - Overhead of Negotiation



Conserve Resources

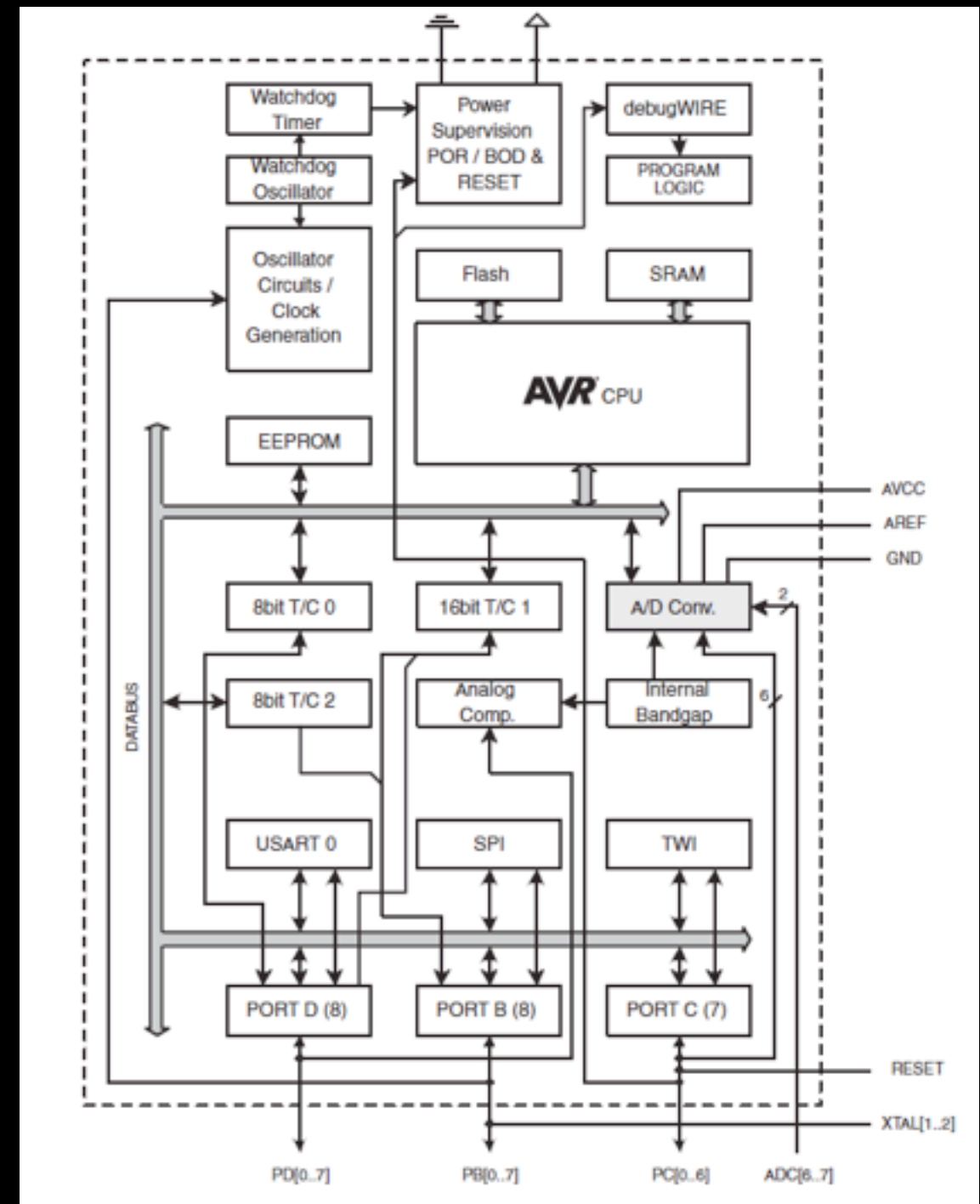
Ride the Bus

- Types of Interface
 - Both Serial and Parallel
 - SPI and I²C
- Internal or External
 - 1 line per device
 - Chip select



Microprocessor Communication

- Atmega 328 supports:
 - Digital and Analog I/O
 - Master/Slave SPI interface
 - 2 wire serial interface bus (I2C)
 - Programmable Serial USART



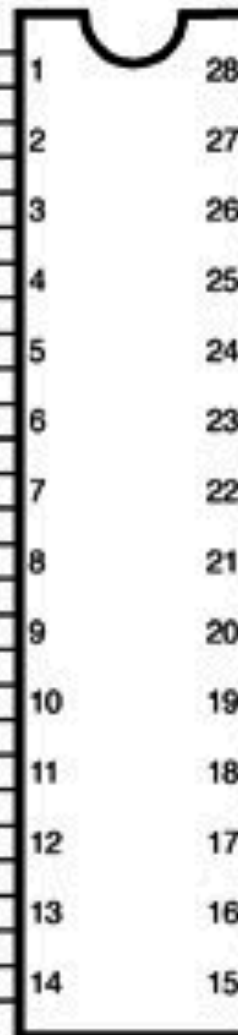
Microprocessor Communication

Atmega168 Pin Mapping

Arduino function

reset
digital pin 0 (RX)
digital pin 1 (TX)
digital pin 2
digital pin 3 (PWM)
digital pin 4
VCC
GND
crystal
crystal
digital pin 5 (PWM)
digital pin 6 (PWM)
digital pin 7
digital pin 8

(PCINT14/RESET) PC6
(PCINT16/RXD) PD0
(PCINT17/TXD) PD1
(PCINT18/INT0) PD2
(PCINT19/OC2B/INT1) PD3
(PCINT20/XCK/T0) PD4
VCC
GND
(PCINT6/XTAL1/TOSC1) PB6
(PCINT7/XTAL2/TOSC2) PB7
(PCINT21/OC0B/T1) PD5
(PCINT22/OC0A/AIN0) PD6
(PCINT23/AIN1) PD7
(PCINT0/CLKO/ICP1) PB0



PC5 (ADC5/SCL/PCINT13)
PC4 (ADC4/SDA/PCINT12)
PC3 (ADC3/PCINT11)
PC2 (ADC2/PCINT10)
PC1 (ADC1/PCINT9)
PC0 (ADC0/PCINT8)
GND
AREF
AVCC
PB5 (SCK/PCINT5)
PB4 (MISO/PCINT4)
PB3 (MOSI/OC2A/PCINT3)
PB2 (SS/OC1B/PCINT2)
PB1 (OC1A/PCINT1)

Arduino function

analog input 5
analog input 4
analog input 3
analog input 2
analog input 1
analog input 0
GND
analog reference
VCC
digital pin 13
digital pin 12
digital pin 11 (PWM)
digital pin 10 (PWM)
digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Microprocessor Communication

SS
MOSI
MISO



SCK

How does a MCU communicate?

Bits & Bytes

1 Bit

- 0 = LOW
- 1 = HIGH

• 2 Bits

- 00 = 0
- 01 = 1
- 10 = 2
- 11 = 3

• 3 Bits

- 000 = 0
- 001 = 1
- 010 = 2
- 011 = 3
- 100 = 4
- 101 = 5
- 110 = 6

How does a MCU communicate?

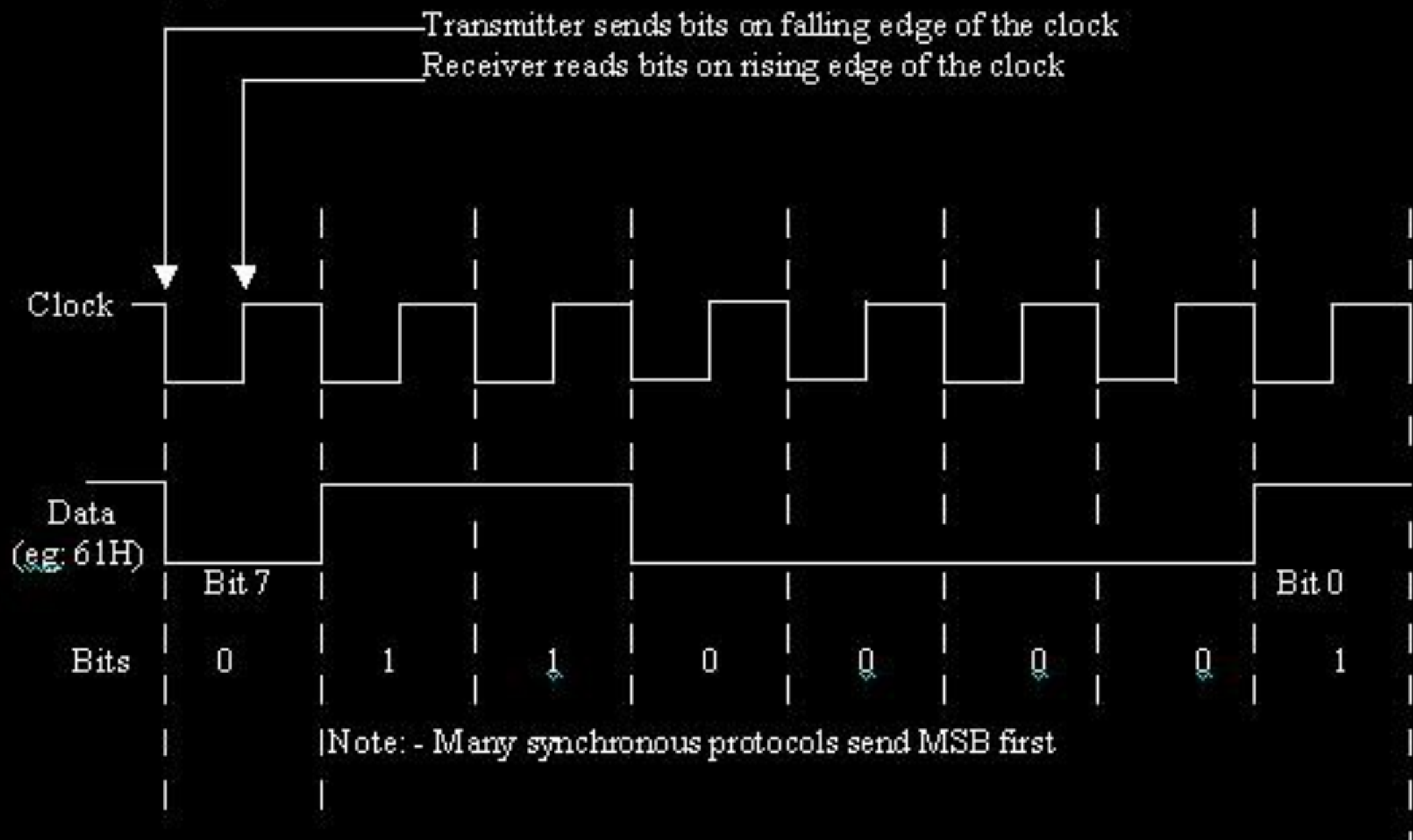
Parallel vs. Serial



Serial Communication

Synchronous

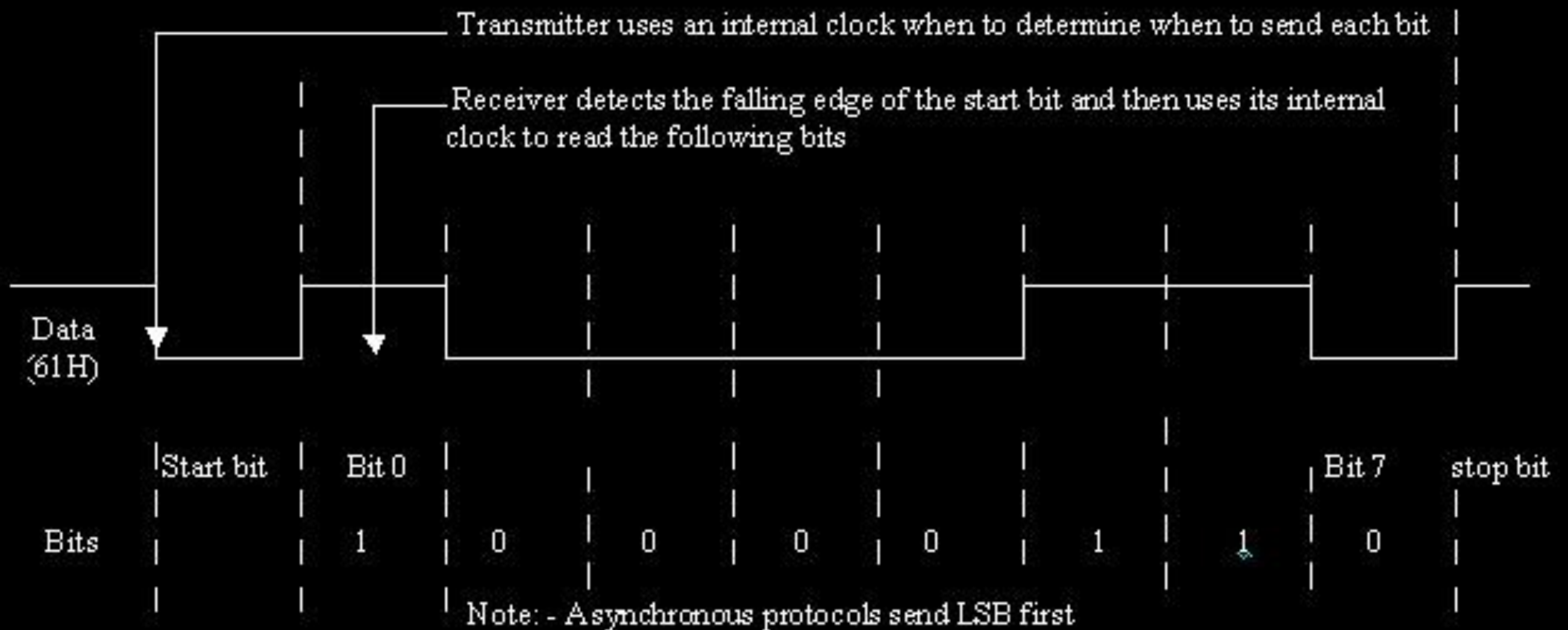
1) Synchronous Transmission: -



Serial Communication

Asynchronous

2) Asynchronous Transmission: -



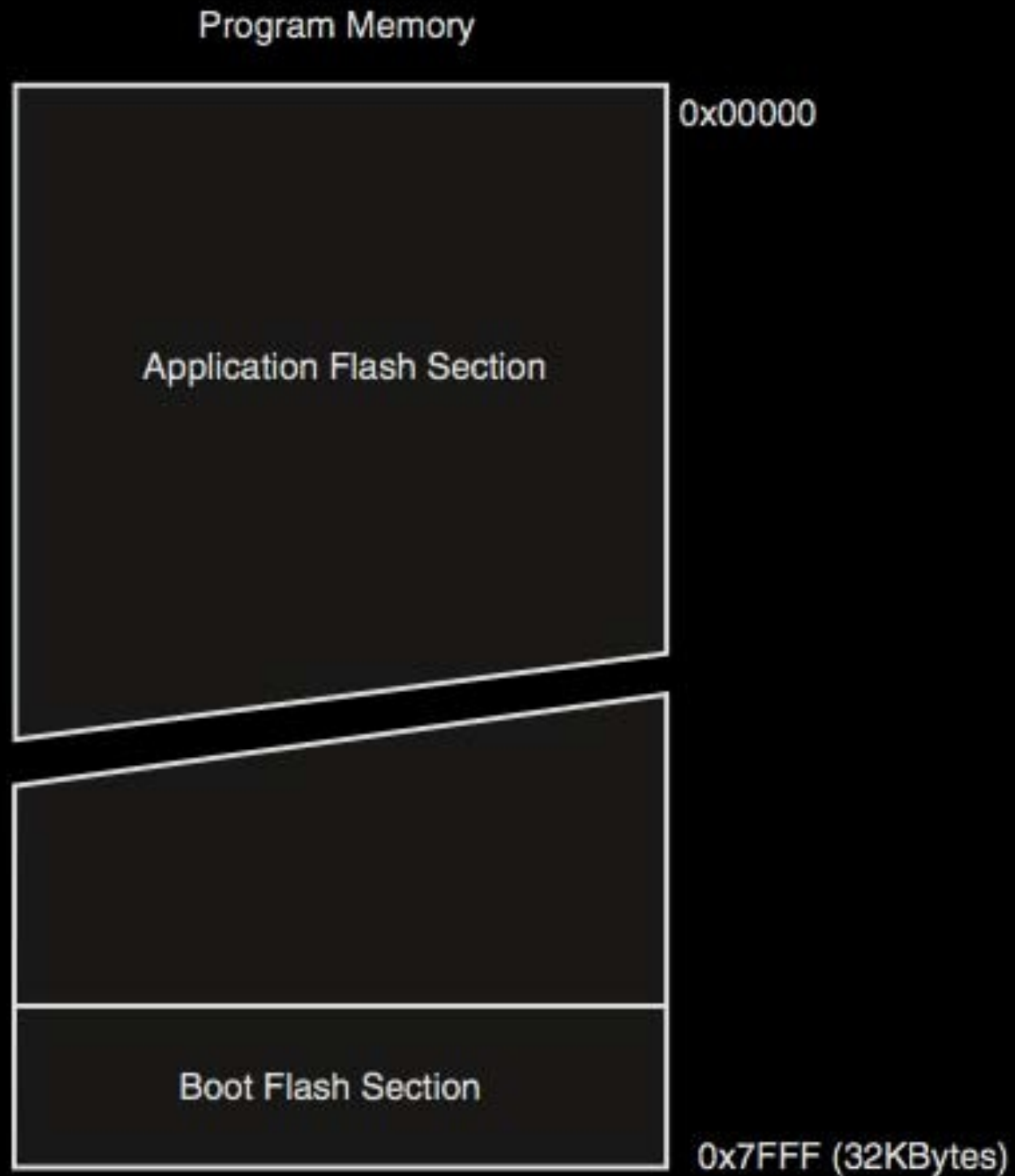
Serial Peripheral Interface

Configuration and Use

- How do we configure the microcontroller (uC) for SPI?
- How do we use SPI to communicate?

Atmega 328 Memory Map

Program Memory

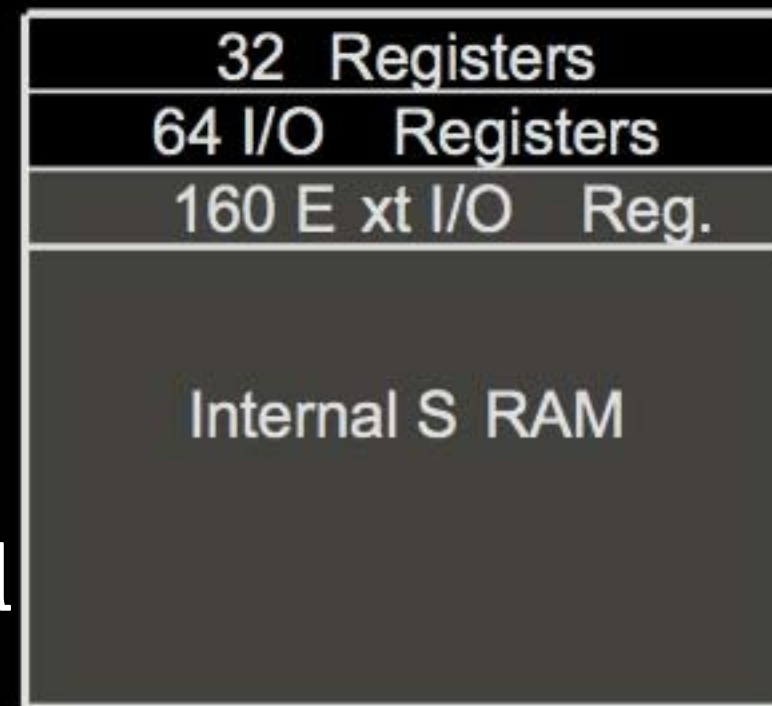


Atmega 328 Memory Map

Access Data & Settings

- A register is a small amount of storage available on the CPU whose contents can be accessed more quickly than storage available elsewhere.
- Certain registers are reserved for μ C operating settings.
- Writing values into these registers changes the settings.

Data Memory



\$0000 - \$001F

\$0020 - \$005F

\$0060 - \$00FF

ISRAM start : \$0100

ISRAM end : \$05FF /

\$FFFF

Serial Peripheral Interface

Configuration

SPI Control Register – SPCR

Bit	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- ☐ Set SPCR = 0101 0000
- ☐ SPE – “Enable SPI mode”
- ☐ MSTR – “I control the clock”

Settings Registers

Bitwise Operators

- How to set bits without over-writing an entire register?
- Set SPCR |= 0101 0000

- Bitwise operators

- AND = "&"
- OR = "|"
- NOT = "~"
- XOR = "^"

- Bitwise AND

- $0 \& 0 = 0$
- $0 \& 1 = 0$
- $1 \& 0 = 0$
- $1 \& 1 = 1$

- Bitwise OR

- $0 | 0 = 0$
- $0 | 1 = 1$
- $1 | 0 = 1$
- $1 | 1 = 1$

Serial Peripheral Interface Configuration

- SCLK, MOSI, & MISO are pre-assigned
- We only connect them

A screenshot of the Arduino IDE interface. The window title is "sketch_apr28a | Arduino 0018". The menu bar includes File, Edit, Tools, and Help. The toolbar contains icons for running, stopping, saving, opening, and other functions. The main text area shows a C++ sketch for SPI configuration. The code defines a chip select pin (SS) as 0, sets the SPI speed and mode in the setup function, and implements a transmit function (tx) that sends a byte through the SPDR register. The loop function enables the remote device by pulling SS low, sends 8 bytes of data, and then pulls SS high. A status bar at the bottom indicates "Done compiling." and "Binary sketch size: 2244 bytes (of a 32256 byte maximum)".

```
sketch_apr28a | Arduino 0018
sketch_apr28a §

#define SS 0 // chip select on pin 0

void setup() {
  SPCR |= 01010000; // set SPE & MSTR
}

// transmit one byte through the SPDR
void tx(byte d) {
  SPDR = d;
  loop_until_bit_is_set(SPSR, SPIF);
}

char data[8]; // the 8 bytes of data

// enable remote device, send 8 bytes
void loop() {
  digitalWrite(SS, LOW);

  for (int i=0; i<8; i++) {
    tx(data[i]);
  }
  digitalWrite(SS, HIGH);
}
```

Done compiling.

Binary sketch size: 2244 bytes (of a 32256 byte maximum)