1. Difference Between Microprocessor and Microcontroller.

|  |  |
| --- | --- |
| Microprocessor | Microcontroller |
| Contains only CPU: RAM, ROM, and I/O timer are separately provided | CPU, RAM, ROM,I/O timer are all on a single chip |
| The designer decides on the amount of ROM, RAM, and I/O ports | Fix the amount of on-chip ROM, RAM, and I/O ports |
| Doesn't support bit addressability | Supports bit addressability |
| Better for multi-tasking | Weak for multi-tasking |
| General Purpose | Single Purpose |
| Architecture: Von Neumann | Architecture: Harvard |
| High Speed and High Cost | Low Speed and Low Cost |
| MP based system requires more hardware to be interfaced | The MC-based system requires less hardware to be interfaced |
| 8085, 8086, core2 duo, and core i7 are examples of Intel processor | AVR, PIC, ARM, and 8051 are some of the microcontroller family |

1. Criteria to Choose a Microcontroller.
2. It must meet the task at hand efficiently and cost-effectively based on the application Decide 8, 16, or 32-bit controller for the computing task.

* Other considerations in this category are
* Speed
* Packaging (Dual in-line or Flat package)
* Power consumption
* On-chip RAM and ROM
* Number of I/O pins and timer on chip
* Ease of Upgradation

1. How easy it is to develop products around it

* Availability of assembler, debugger, compiler, third-party vendor support

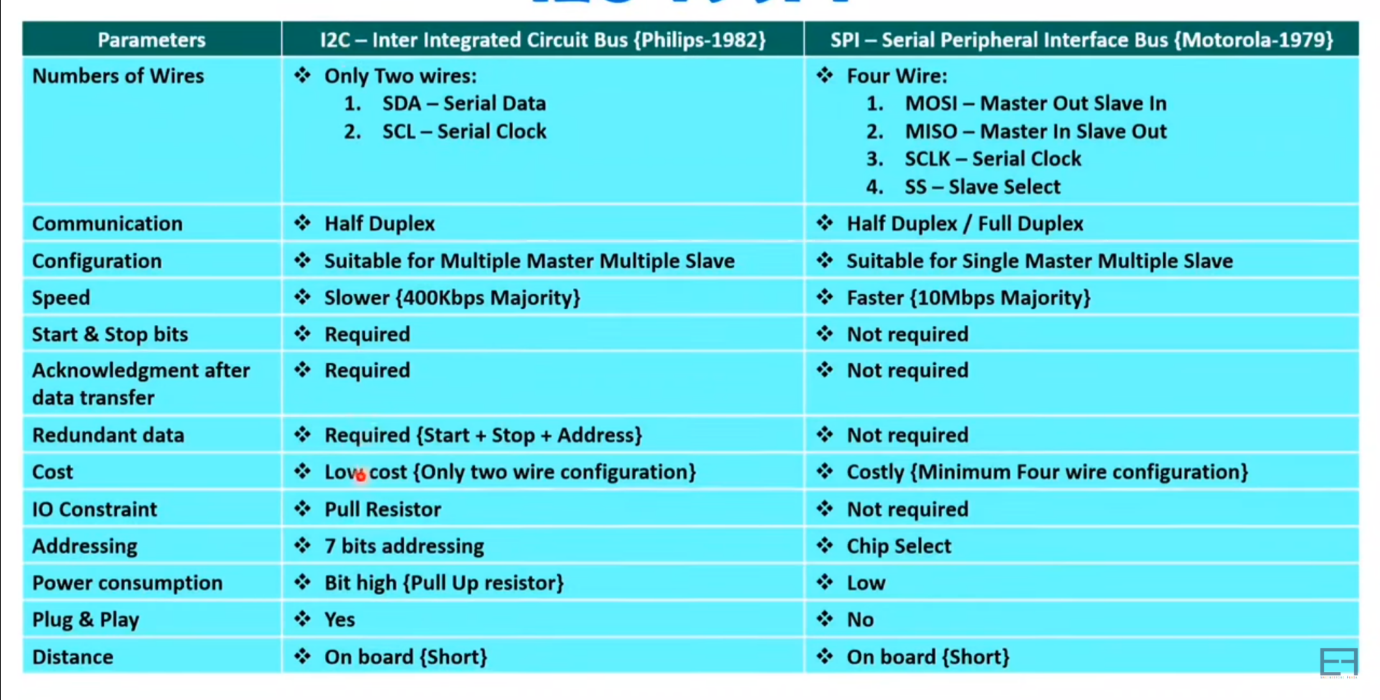
1. MC should be available in needed quantities both now and in the future.
2. Von Neumann Vs Harvard Architecture

|  |  |  |
| --- | --- | --- |
| Parameter | Von Neumann | Harvard |
| Memory | Data and Program (Code) stored in the same memory | Data and Program (Code) stored in different memory |
| Memory Type | RAM for Data & Code | RAM for Data, ROM for Code |
| Buses | Common bus for Address & Data/Code | Separate bus for Address & Data/Code |
| Program Execution | Code executed serially takes more cycles | Code executed in parallel with data takes fewer cycles |
| Data/Code Transfer | Data or Code in one cycle | Data and Code in One Cycle |
| Control Signals | Less | More |
| Space | Needs less space | Needs more space |
| Cost | Less | Costly |

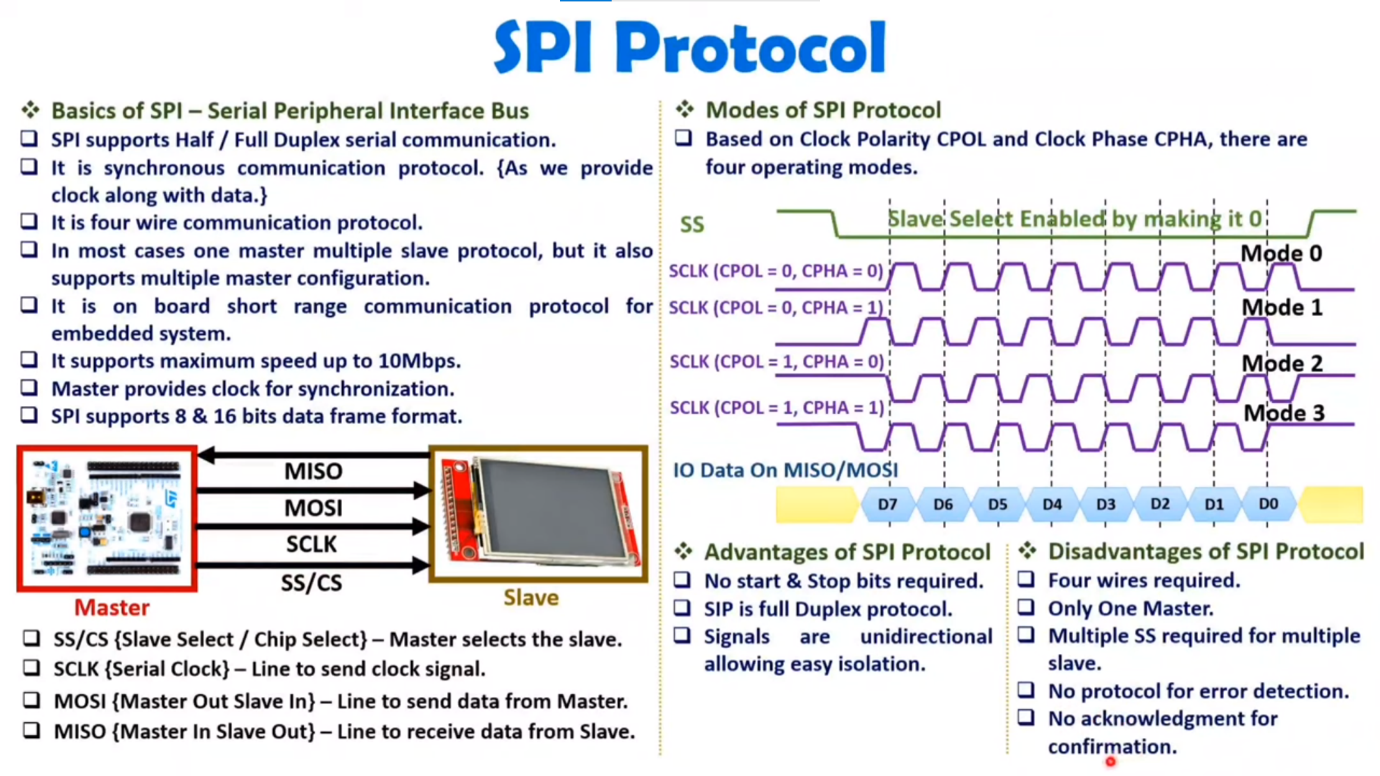
1. RISC vs CISC

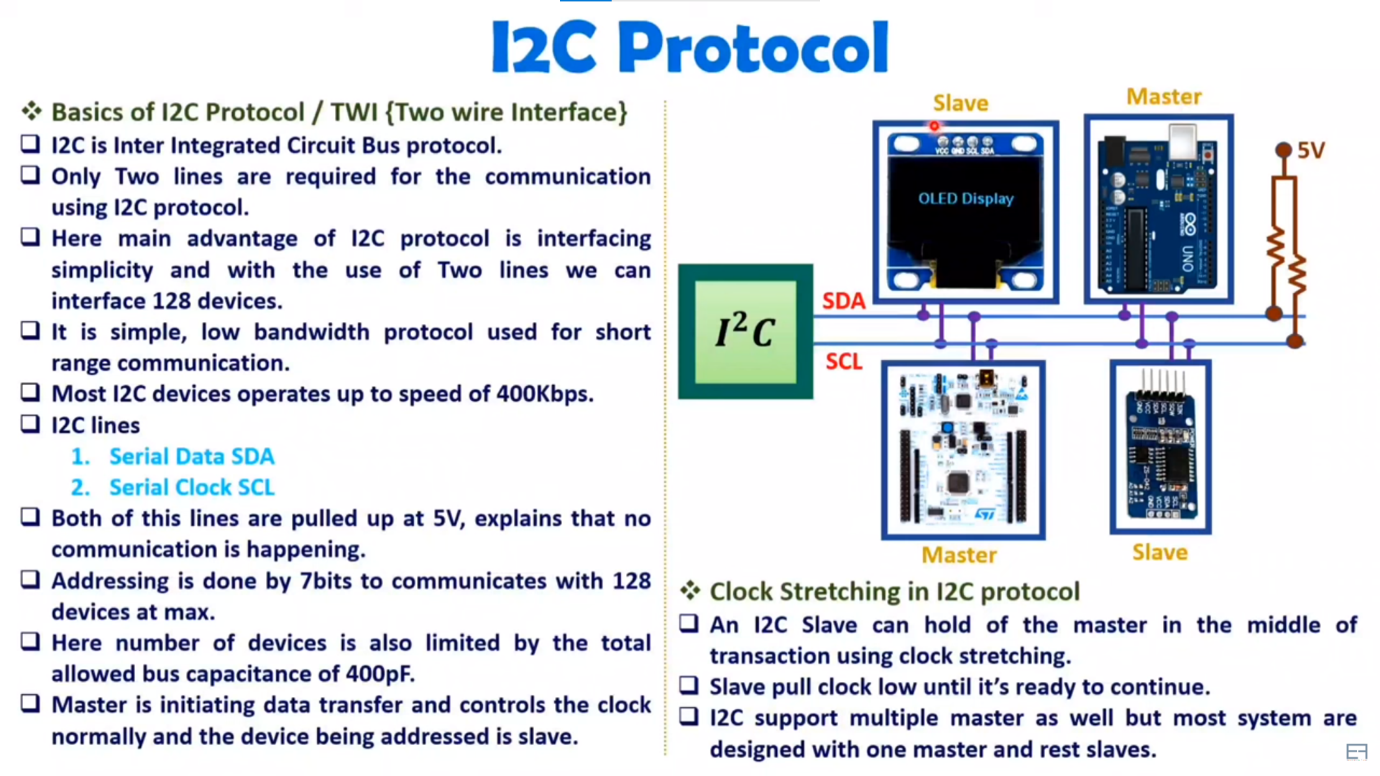
|  |  |  |
| --- | --- | --- |
| Parameter | RISC | CISC |
| Instruction Complexity | Simpler instruction set, simple instruction decoding | Complex instruction set, complex instruction decoding |
| Instruction Size | Single-word size | Larger than one-word size |
| Execution Time | Single clock cycle | May take more than a single clock cycle |
| General-purpose Registers | More | Less |
| Addressing Modes | Simple | Complex |
| Data Types | Fewer | More |
| Code Size | More | Reduced |
| Memory Efficiency | May require more memory | More memory efficient |
| Power Consumption | Lower | Higher |
| Development Cost | Lower | Higher |
| Pipeline | Pipeline can be achieved | Pipeline may be more challenging to implement |

1. I2C vs SPI



1. SPI



1. I2C

