ECE 442/510
Internet of Things and Cyber Physical Systems

Lecture 2: Introduction to Embedded Systems
Summer 2022

Jafar Saniie & Won-Jae Yi

Embedded Systems

- Some combination of computer hardware and software
- Programmable and/or reconfigurable
- Designed for specific functions and/or features
- We are living in the world flooded with embedded systems

Examples of Embedded Systems

- Home appliances
  - A/C, refrigerator, microwave oven
- Mobile devices
  - smartphone, tablet, smart watch
- Automotive
  - Advanced Driver Assistant System(ADAS), cruise control
- Military
  - Missiles, satellites
- Gaming Consoles
  - XBOX, PlayStation
- Almost all electronics are equipped with embedded systems

Embedded System Development

- Use microcontrollers/microcomputers
  - More keen to software development compared others
  - Adding hardware is somewhat limited
  - Programming language depends on the platform
- Use FPGA (Field Programmable Gate Array)
  - Capable of becoming *any* digital circuit (within allowed space)
  - Hardware reconfiguration possible
  - VHDL/Verilog, C/C++/System C by HLS (High level synthesis)
  - MATLAB, LabVIEW
- Use ASIC (Application Specific Integrated Circuit)
  - Customized hardware configuration (VLSI techniques)
  - Full-custom design starting from RTL design

Microcontroller/Microcomputer

- A small computer on a single chip
  - Processor, control unit, input/output, memory, clock
- Typically “embedded” inside some device that they control

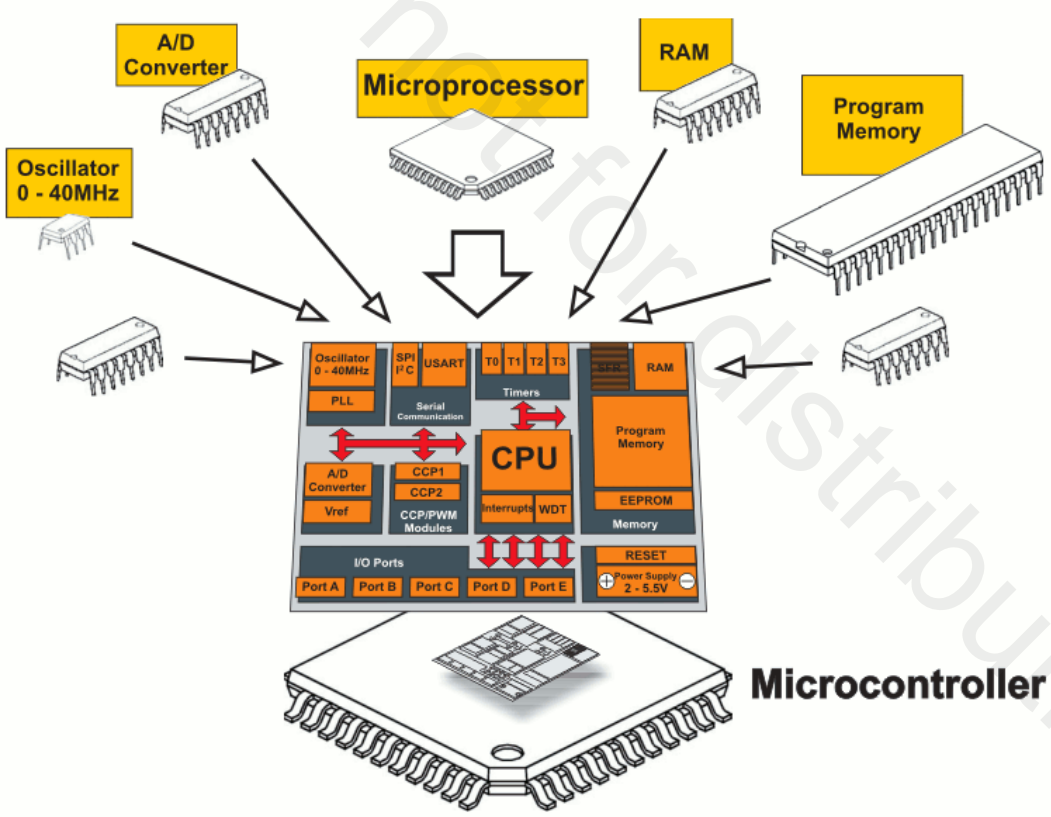


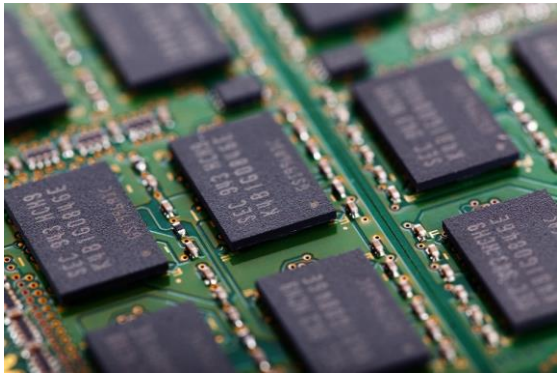
Fig. 0-1 Microcontroller versus Microprocessor

Microcontroller Components

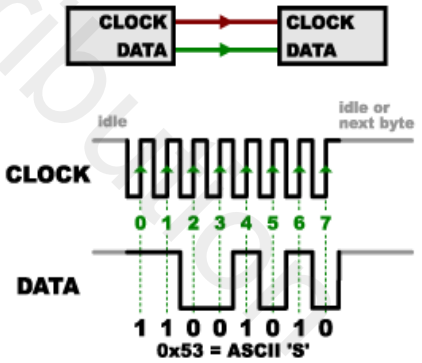
- Processor (CPU)
  - Execute programs with serial execution of instructions
  - Addition, subtraction, bit-wise AND/OR, shift operations
  - Registers (fast storage for load/store data from memory)
- Control Unit (Send/receive signals to control components)
  - Send/receive control signals to other components
  - Flow of the data between other units and operations

Microcontroller Components

- Memory
  - Program code (instruction) and data are stored
  - Addressable locations to load/store data
  - RAM, ROM, SD card, HDD, SSD



- Clock
  - Periodic signal to all components in microcomputer
  - Synchronization of data flow

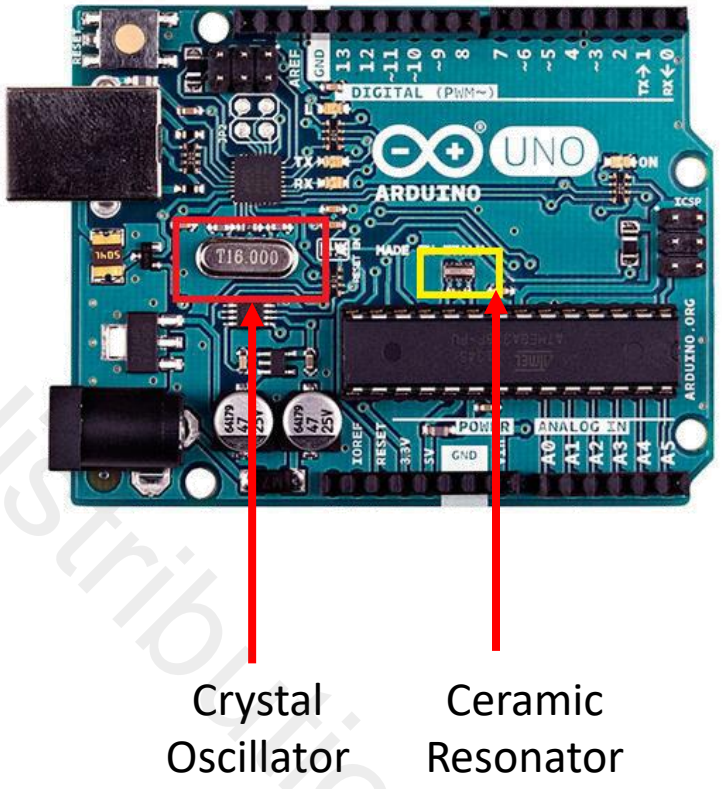


Microcontroller Components

- Input/Output (I/O)
  - Interface between internals of a microcontroller and outside world in the form of the pins of a microcontroller via special variables called “registers”
  - Registers are actual hardware memory locations inside the microcontroller
  - When assign a value to these registers, actual value in the hardware changes, and can be changed multiple times
  - Keyboard, LED/LCD Display, printers, USB drives, sensors, actuators, etc.

Microcontroller Components

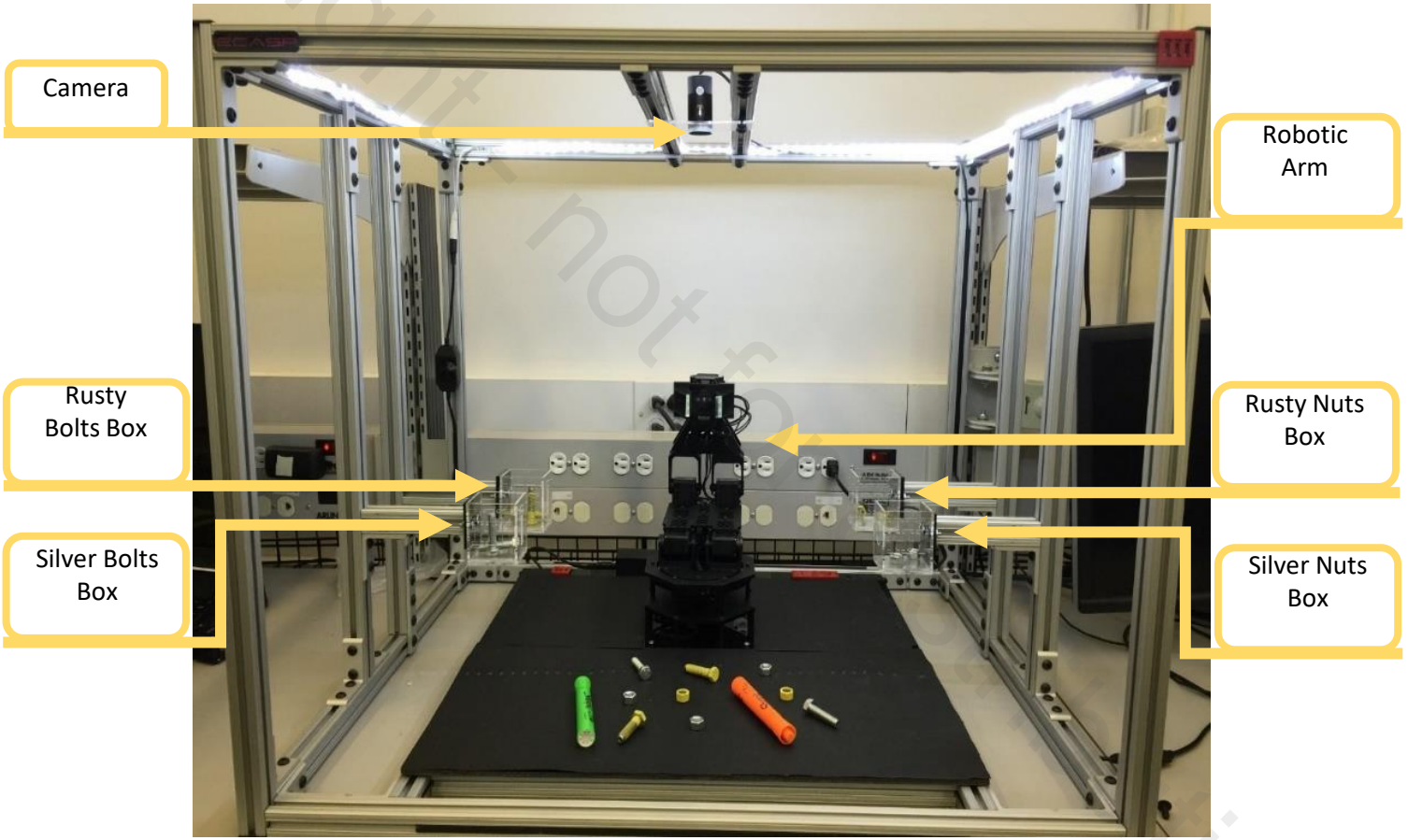
- Clock Signal Generator
  - Microcontrollers require clock cycles to execute instructions, which is generated by an oscillator circuit
  - Used for synchronous functions within the embedded systems
  - e.g., crystal oscillator(quartz crystals), ceramic resonator



Embedded Systems

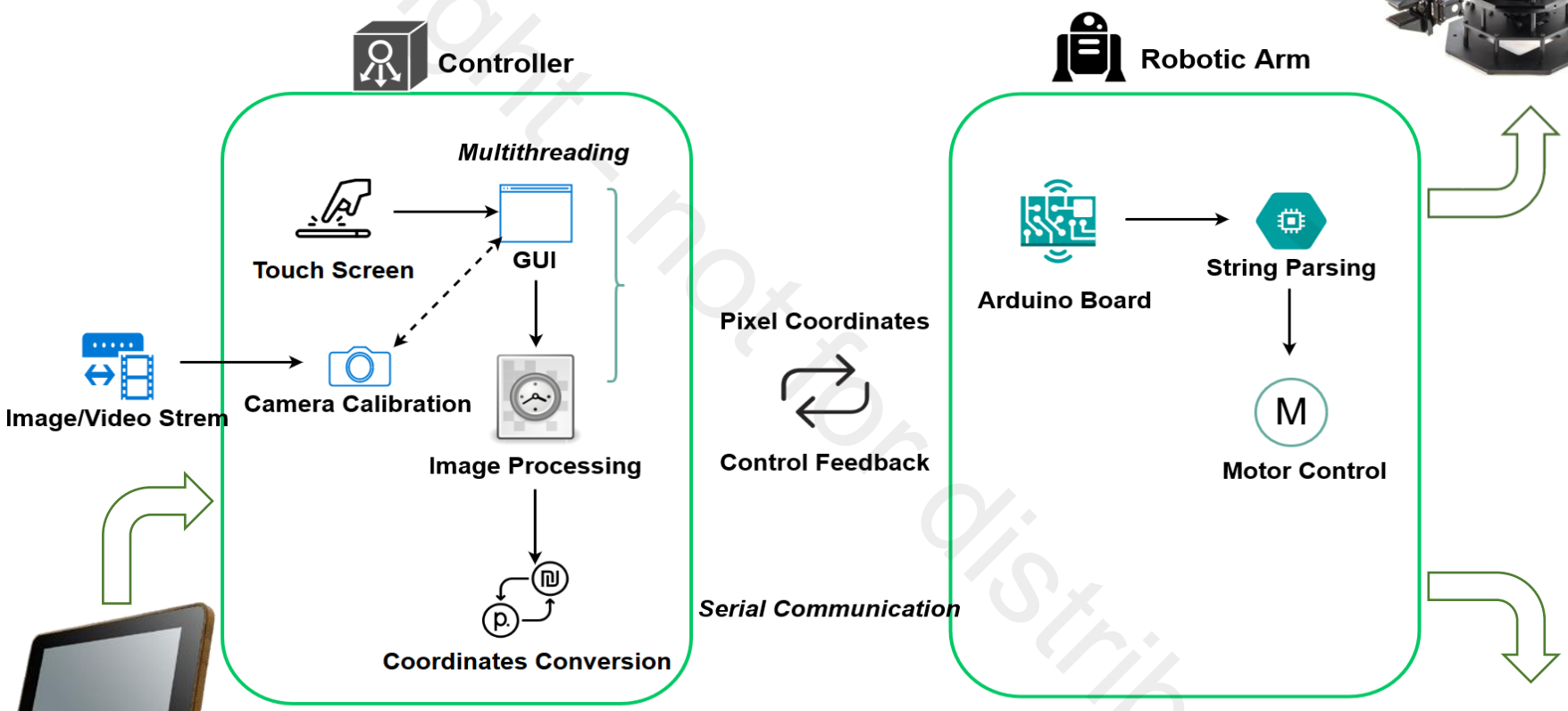
- Application-specific, acts as a dedicated system to certain or multiple purposes
- Limited processing capabilities, requiring a small amount of power
- Easy Hardware and software reconfiguration on a small platform
- Can have significant real-time constraints, act on inputs very quickly and generate high-frequency outputs
- Cost-effective, easy to manage, spend less resources, often a higher expectation of reliability

Sorting Robotic Arm using Machine Vision – System Overview



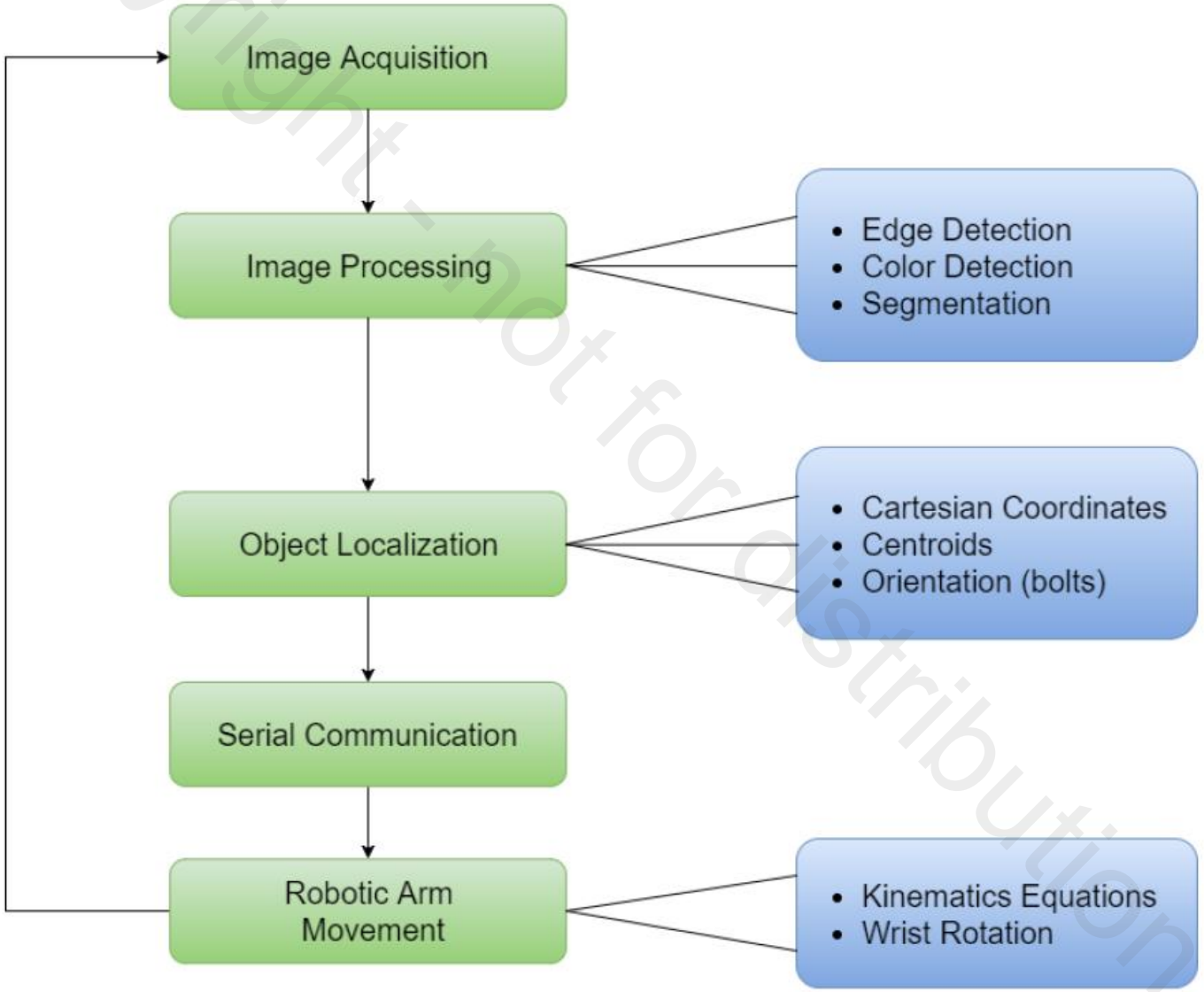
LINK : http://ecasp.ece.iit.edu/video/bsmp2016/ecasp_robotic_arm.mp4

Sorting Robotic Arm using Machine Vision – System Overview



Architecture of Robotic Arm

Sorting Robotic Arm using Machine Vision – Design Flow



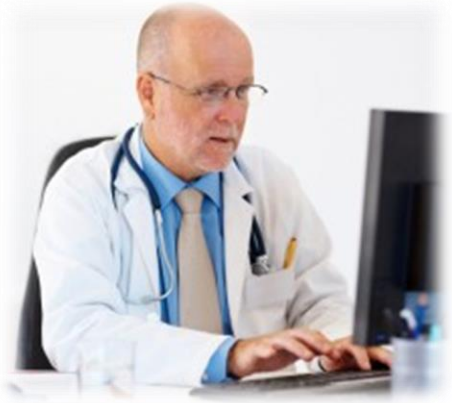
Sorting Robotic Arm using Machine Vision

- Applied machine vision for sorting robotic arm
  - Webcam recognizes objects (bolts, nuts, anomalies)
  - C++ using OpenCV libraries on Raspberry Pi
  - Application sends coordinates of the object to arm controller
  - Raspberry Pi sends command to Arduino
  - Arm controller (Arduino) moves the object
  - C program on Arduino interfacing with the robotic arm
- Can be improved for more sophisticated application
  - What do you think??

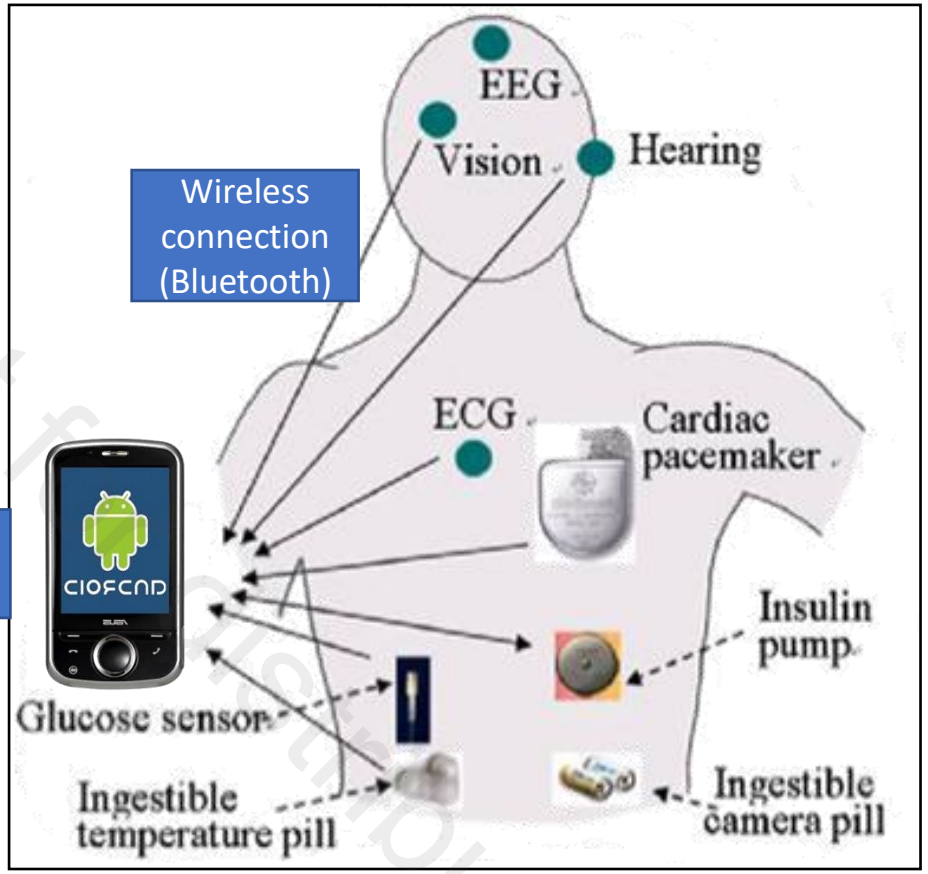
Smartphone is also a microcomputer (very advanced)

- Smartphone has the components as a microcomputer
- Typical Android smartphone has… (Samsung Galaxy S8)
  - ARM Cortex Octa-core processor up to 2.3 GHz + on-board GPU
  - 4GB LPDDR4X SDRAM with 64GB on-board memory
  - Bluetooth, Wi-Fi, Cellular for data connectivity
  - Iris scanner, fingerprint, accelerometer, gyro, proximity, compass, barometer, heart rate, SpO2
  - Android: open-source operating system (reconfigurable!)
- Programming language for Android app development
  - Java with SDK (Software Development Kit)
    - Android Studio (dedicated development tool from Google)
  - C with NDK (Native Development Kit, not used often)

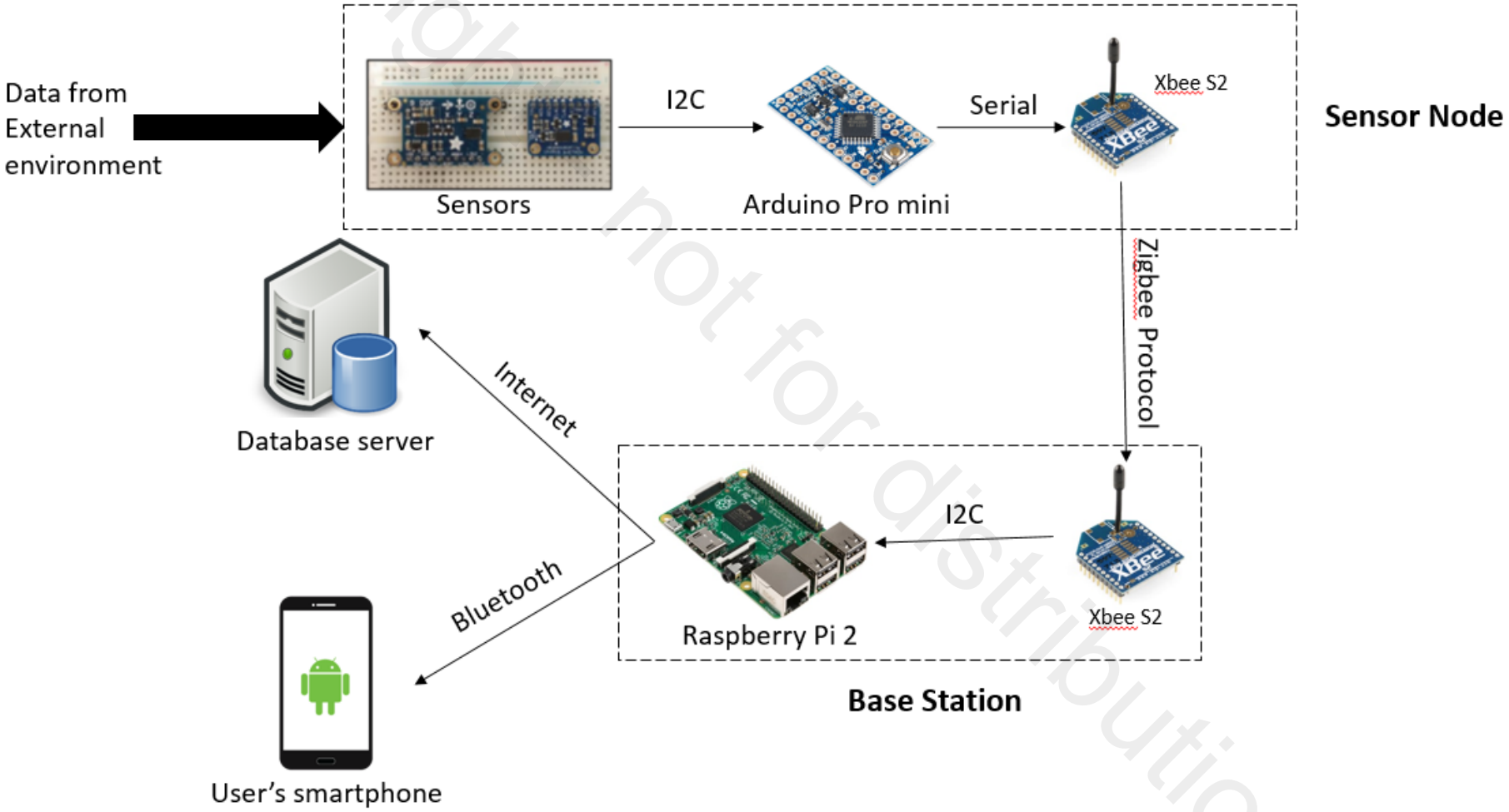
Wireless Body Sensor Network



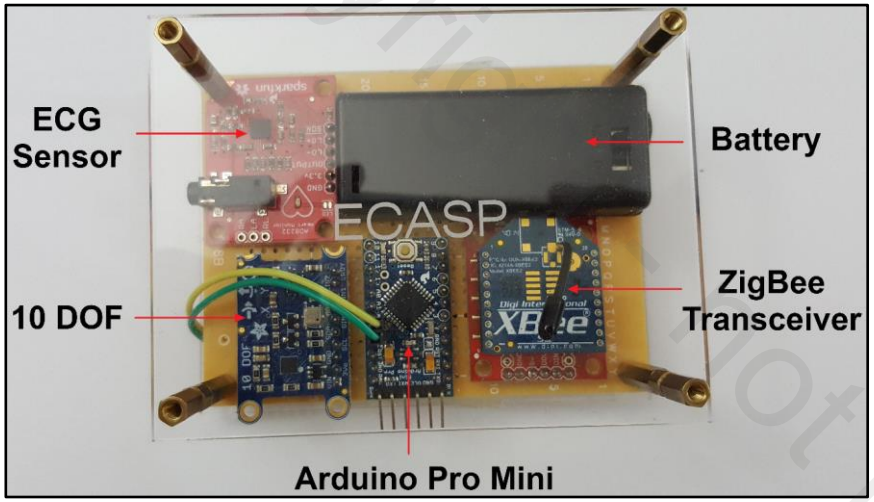
Sensors data though internet



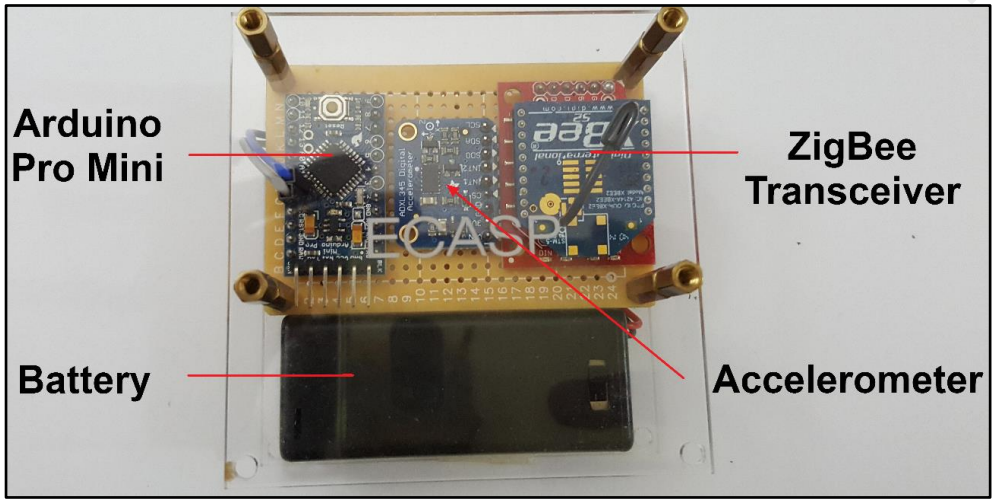
Fall Detection System using Sensors, Raspberry Pi & Smartphone



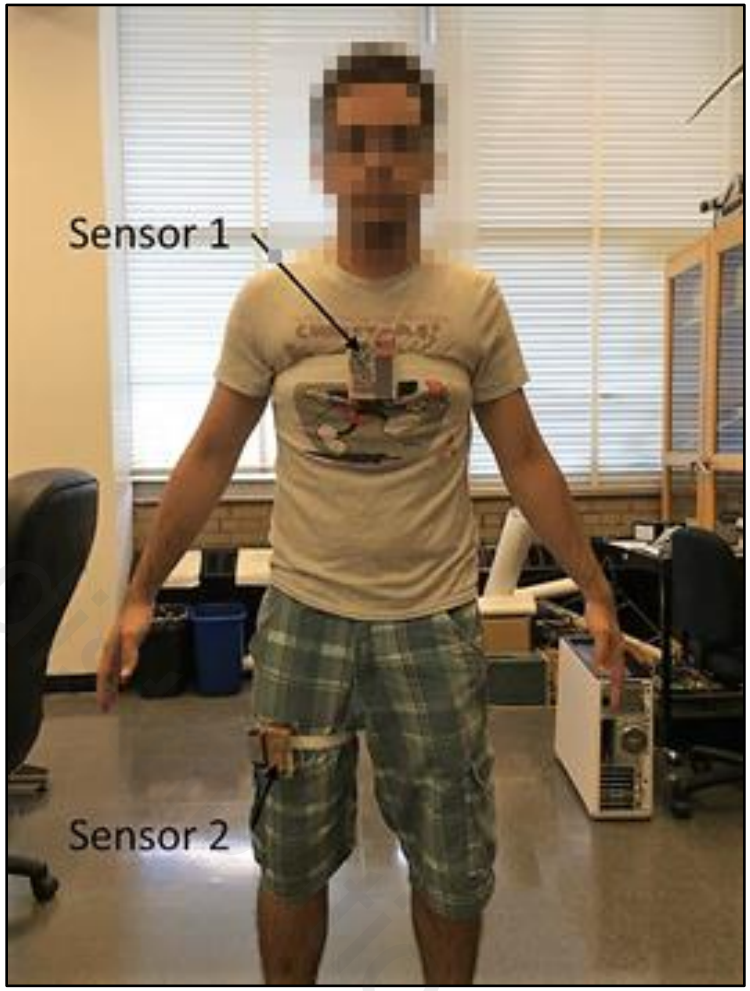
Fall Detection System using Sensors, Raspberry Pi & Smartphone



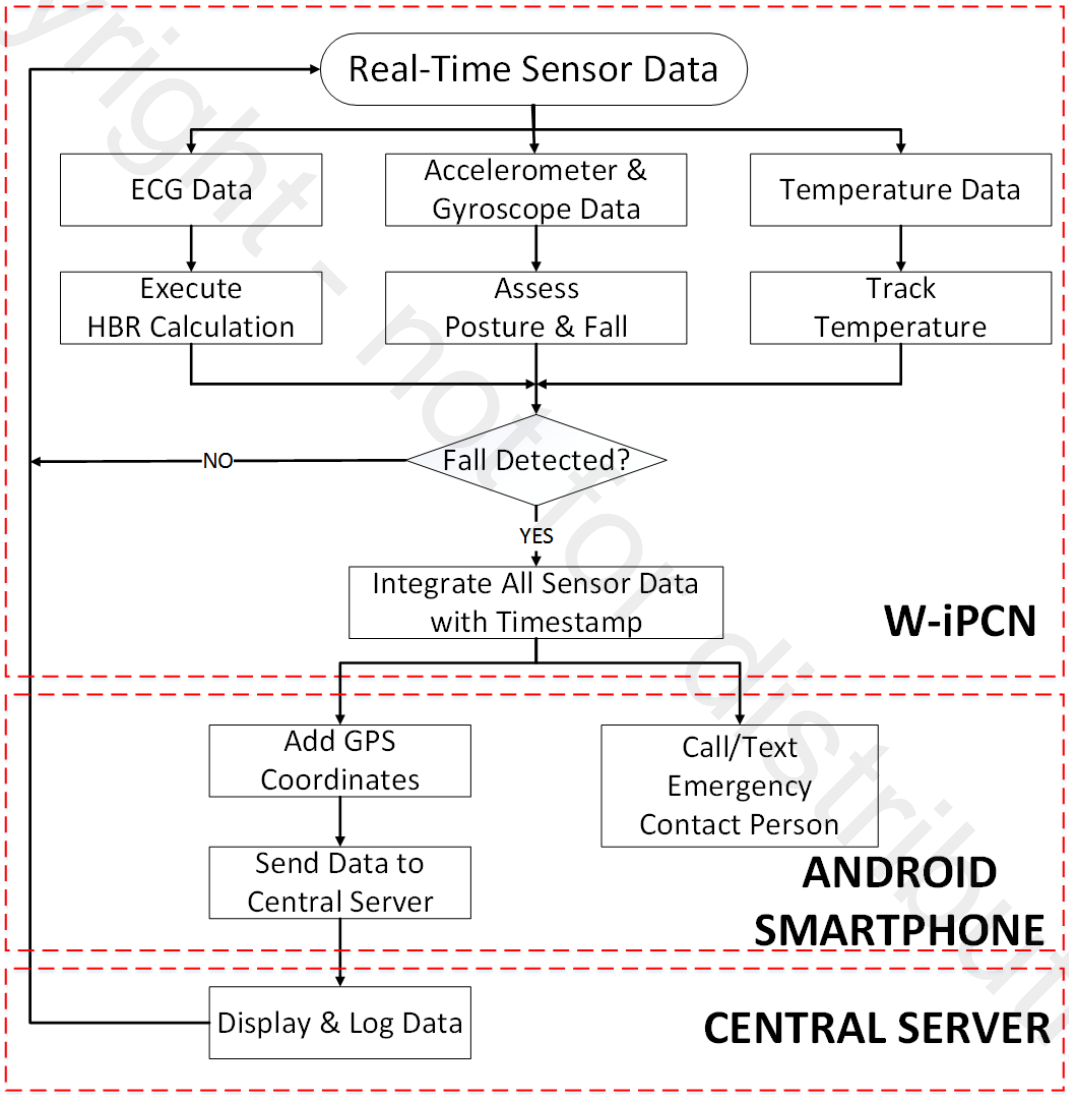
Chest Node (Sensor 1)



Thigh Node (Sensor 2)



Fall Detection System using Sensors, Raspberry Pi & Smartphone



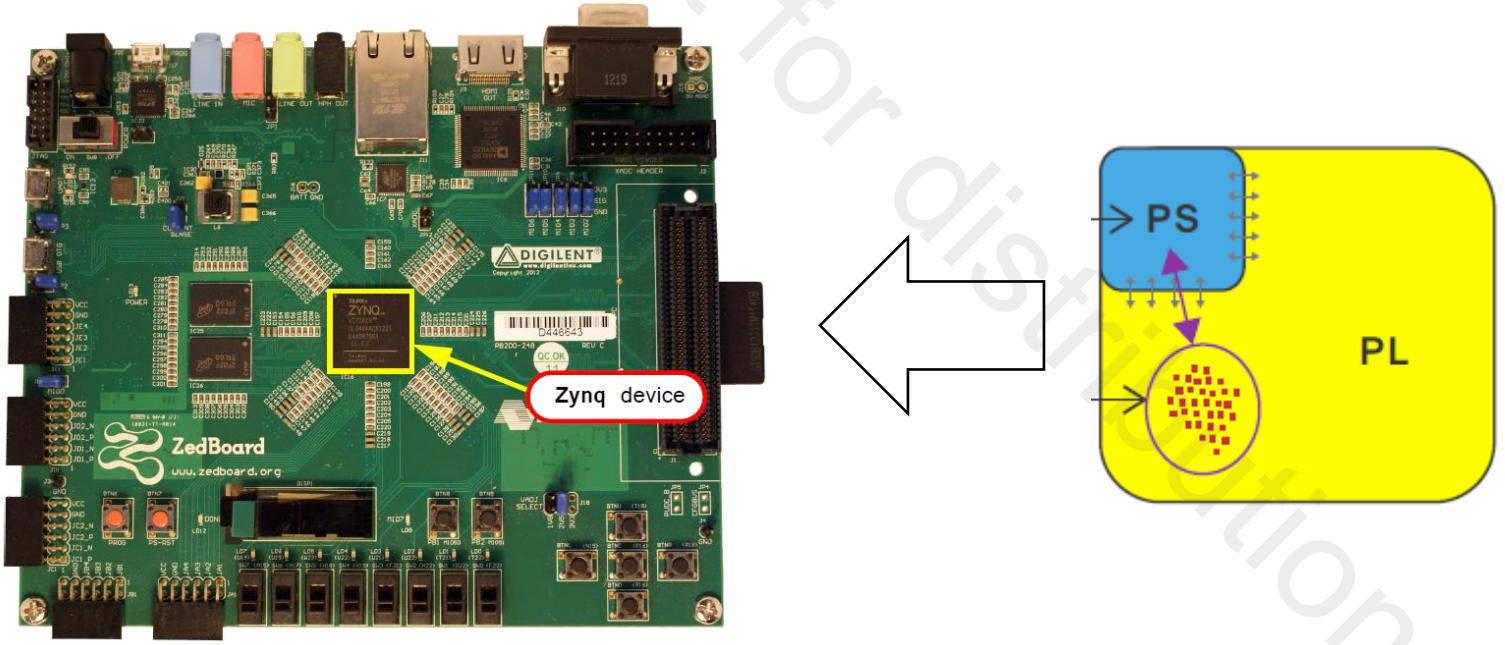
LINK : http://ecasp.ece.iit.edu/video/bsmp2016/ecasp_fall_detection.mp4

Fall Detection System using Sensors, Raspberry Pi & Smartphone

- Sensors, ZigBee transceiver connected(wired) to Arduino Mini
  - Programmed to obtain sensor data through I^2C bus, transmit sensor data to ZigBee transceiver through Serial/Digital bus
- ZigBee transceiver connected(USB) to Raspberry Pi
  - Programmed to receive incoming data from ZigBee transceiver through USB
  - Customized application processes body posture & fall detection in real-time, and sends analyzed information to the Android smartphone through Bluetooth transceiver (USB/On-board)
- Android smartphone utilizing Bluetooth and Wi-Fi/Cellular
  - Customized application to receive incoming data from Raspberry Pi through Bluetooth
  - Displays information on screen, sends to the server through Wi-Fi/cellular
  - Alerts sent via SMS to designated person

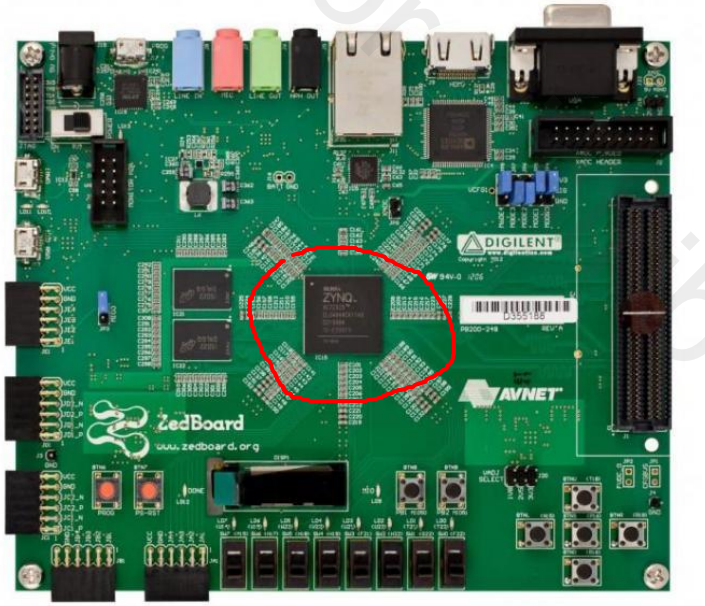
Hybrid Embedded System

- Xilinx ZYNQ consists of an ARM processor as well as FPGA
  - Hardware/software development on the same die



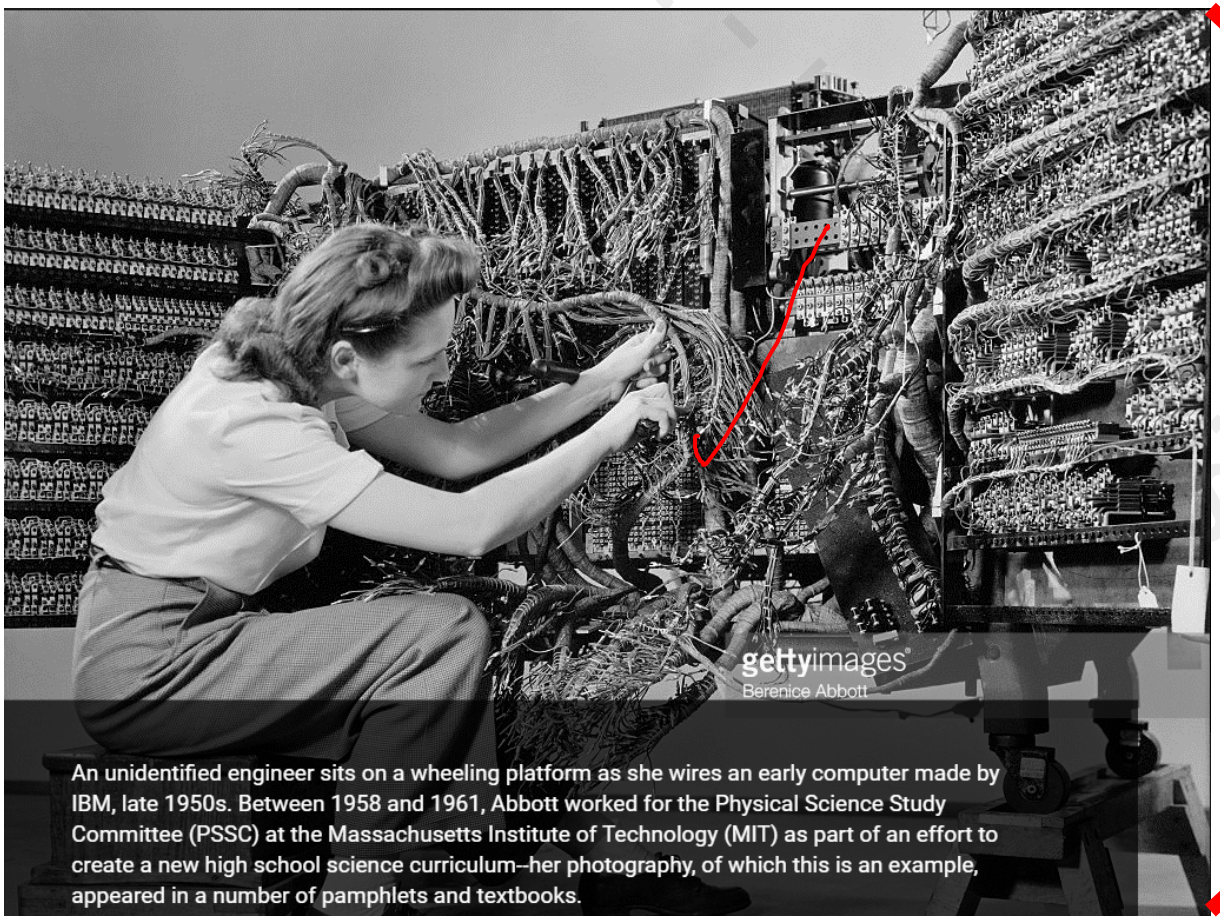
FPGA

- **F**ield **P**rogrammable **G**ate **A**rray
  - **Field**: Using field transistors to store the configuration
  - **Gate Array**: Many logic blocks that can be connected to each other
  - **Programmable**: Connection between blocks, inside blocks can be configured at will



FPGA

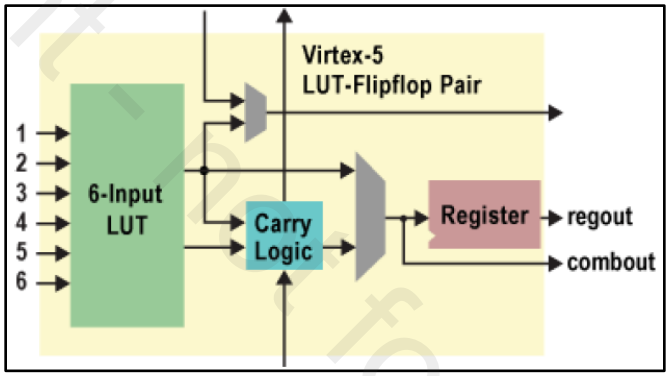
- To put many logics into one single chip
  - Enhancing the **space**, **speed**, and **power consumption**



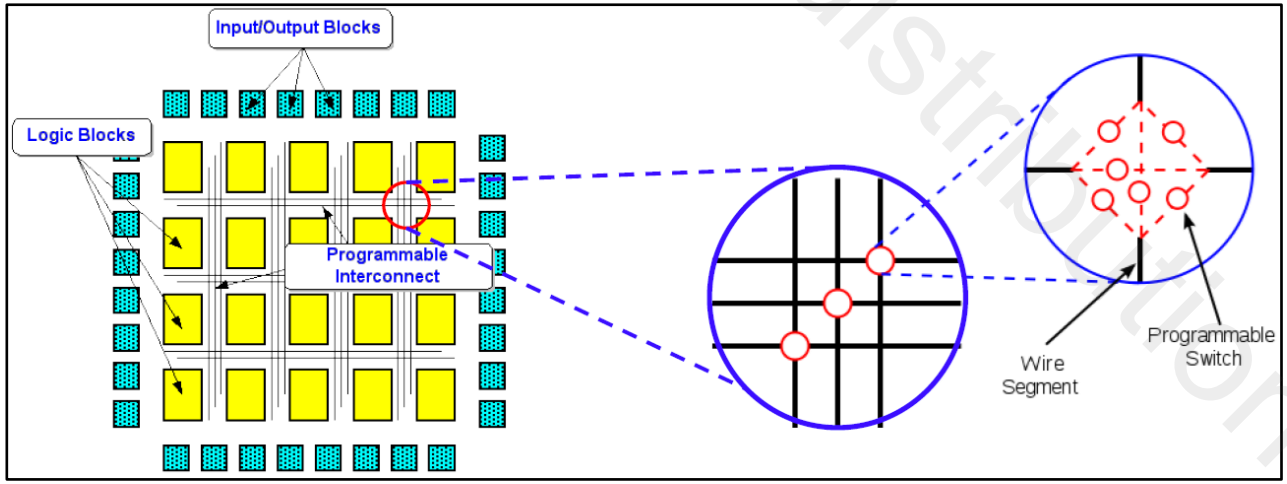
An unidentified engineer sits on a wheeling platform as she wires an early computer made by IBM, late 1950s. Between 1958 and 1961, Abbott worked for the Physical Science Study Committee (PSSC) at the Massachusetts Institute of Technology (MIT) as part of an effort to create a new high school science curriculum--her photography, of which this is an example, appeared in a number of pamphlets and textbooks.

Inside of FPGA

- Many logic blocks of basic logic



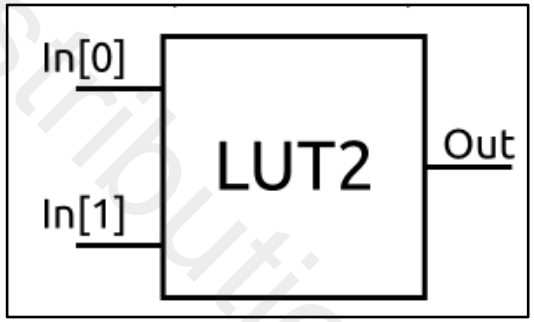
- Interconnected by configurable connections



LUT (Look Up Table)

- A block that is implementing a logic function storing the output states for each input state
- A table determines what the output is for any given input(s)
  - Truth table for combinatorial logic behaviors (AND, OR gates)
  - e.g., AND gate

Address (In [1:0])	Value (Out)
00	0
01	0
10	0
11	1



Advantage of FPGA Development

- Time to market
  - Completely **reconfigurable**
  - Many libraries already available (IP cores)
  - Provides some hardware implementation for some application specific designs
    - Digital Signal Processing (DSP48 on Xilinx)
    - Communications (High speed serial, ethernet, …)
- Performance
  - **Hardware parallelism**, overwhelms computing power of sequential execution
  - Controlling inputs and outputs at the hardware level provides faster response times
- Cost
  - No need to rebuild a chip per modification
    - Non-recurring engineering cost ↓

Advantage of FPGA Development

- Reliability
  - Processor-based systems involve several abstraction layers for task scheduling and shares resources among multiple processes
    - Driver layer controls hardware resources, OS manages memory and processor bandwidth
    - Continuously at risk of time-critical tasks (Only one instruction can be executed)
  - FPGAs (No OS!) minimize reliability concerns with true parallel execution and deterministic hardware dedicated to every task
- Long-term maintenance
  - Do not require the time and expense involved with ASIC redesign
  - Specifications can be changed over time (e.g., digital communication protocols, other peripherals to be attached)
  - Functional enhancements possible without spending time redesigning hardware or modifying the board layout

Programming FPGA

- Configured using Hardware Description Language (HDL)
  - VHDL (VHSIC(Very-High-Speed Integrated Circuits) Hardware Description Language)
    - More verbose, more natural than Verilog, non-C like syntax
  - Verilog
    - More compact, fewer code lines, better grasp on hardware modelling but a lower level of programming constructs

VHDL:

```
process (S0,S1,A,B,C,D)
begin
   case (S0,S1), is
      when "00" => Y <= A;
      when "01" => Y <= B;
      when "10" => Y <= C;
      when "11" => Y <= D;
      when others => Y <= A;
   end case;
end process;
```

Verilog:

```
always @((S0,S1), A, B, C, D)
      case ({S0,S1})
         2'b00: Y = A;
         2'b01: Y = B;
         2'b10: Y = C;
         2'b11: Y = D;
      endcase
```

Programming FPGA

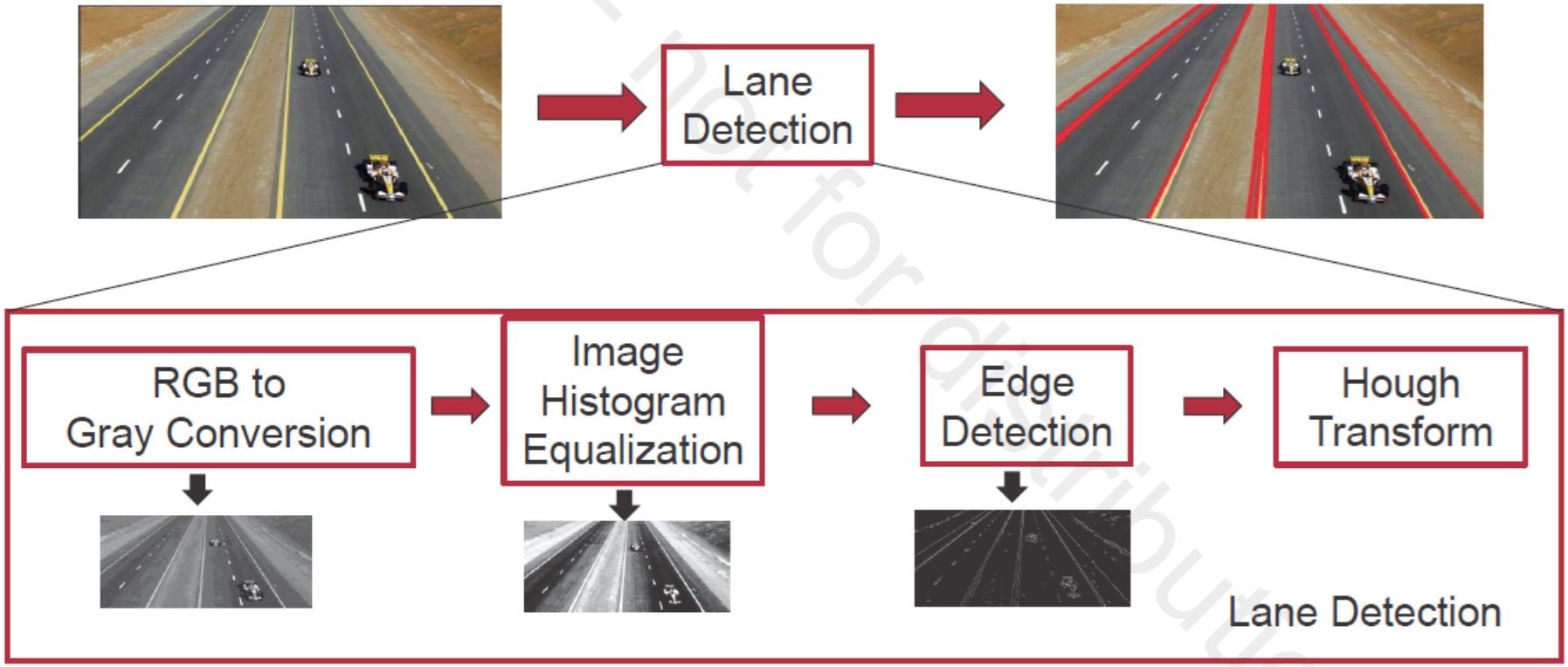
- Synthesizer generates a netlist, known as RTL (Register-Transfer Level). Design is represented as a set of elementary gates and links between them
- Once RTL is obtained, the router allocates FPGA resources and routes different nets (Place and Route)
- Previous step generates a binary file to be sent into the FPGA
  - Configure connections, LUTs, initial memory, etc.
- These processes are done by a software tool provided by the FPGA manufacturers (e.g., Xilinx, Altera)
  - ISE (**Integrated** Synthesis **Environment**), Vivado depending on the FPGA board platform

Programming FPGA

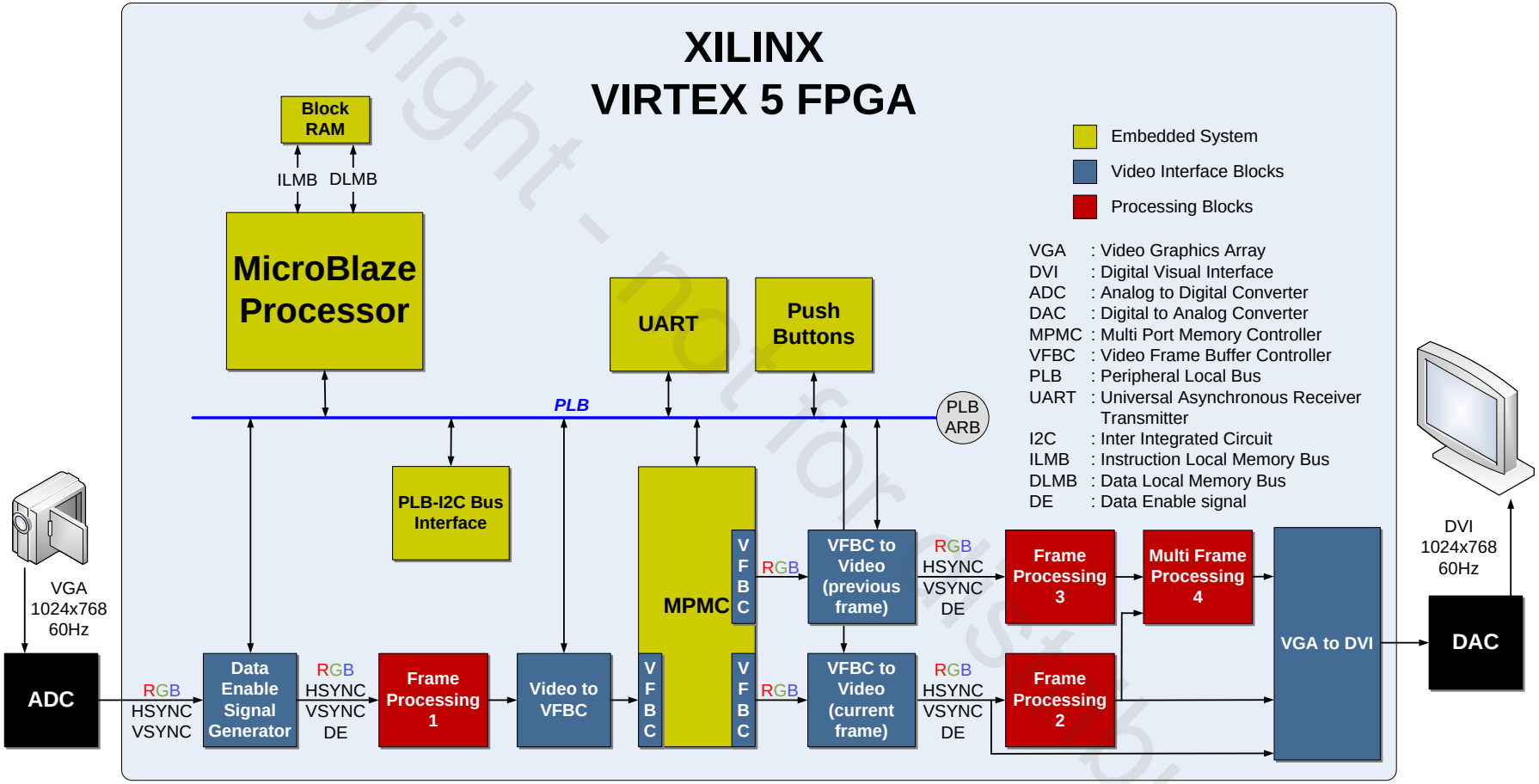
- High-level synthesis (HLS)
  - Automated design process of algorithmic description interpretations and creations of digital hardware for a desired behavior
  - Designer describes the design at a higher abstract level, the tool does the RTL implementation
  - Modeling complex designs, reduce design efforts, fast turnaround time
- C-Based System level languages (Handel C, Impulse C, Streams C)
- MATLAB-based (System Generator for DSP, Xilinx)
- GUI Data-Flow based (Corefire)
- Java-based (Forge, JHDL)

FPGA Real-world Utilization

- Lane Detection
  - Analyze a video frame to detect road lane markings



Embedded Image and Video Processing Application on FPGA



LINK : http://ecasp.ece.iit.edu/video/research/ecasp_virtex5.mp4

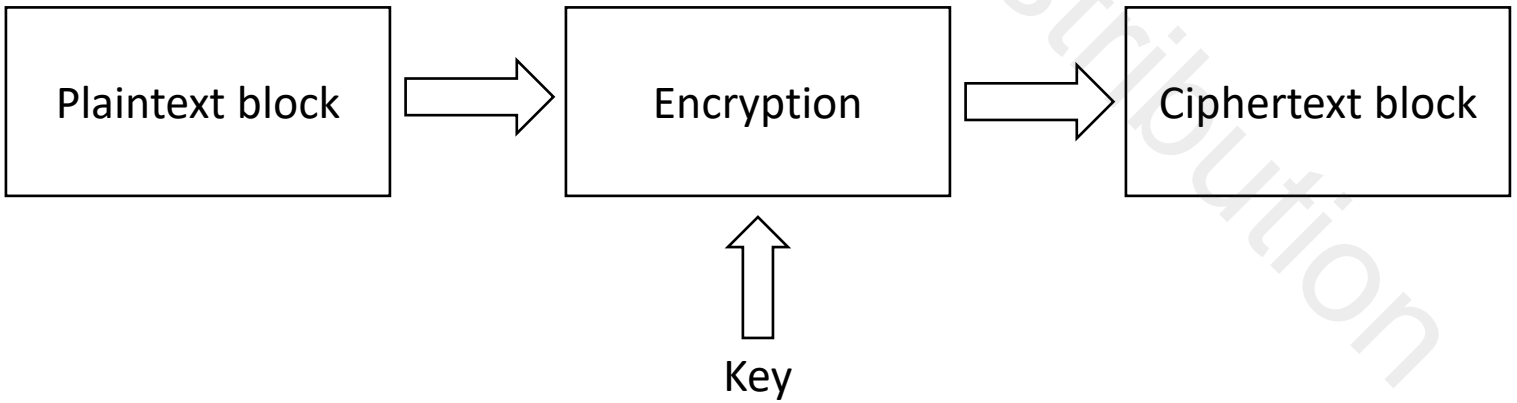
LINK : http://ecasp.ece.iit.edu/video/research/ecasp_traffic.mp4

High Performance Cryptology System using FPGA

- Encryption/decryption on real-time image transmission
- Parallel processing faster on hardware (FPGA) implementation than software (ARM) implementation
- AES-128 (cipher algorithm, AES (Advanced Encryption Standard), Key Lenght=128 bits) implemented using VHDL
- Input/Output controlled by ARM processor

Block Cipher

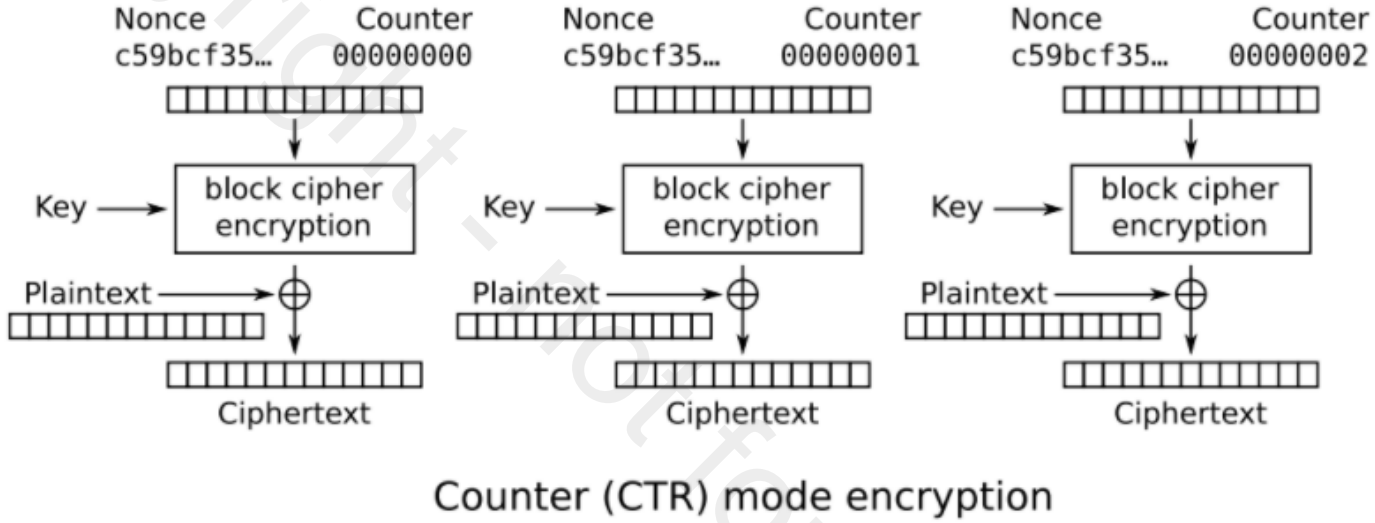
- Used when the plaintext length is longer than the “block” size to cipher/encrypt data
- Block cipher works in certain length (block size), thus requires to separate plaintext (in various sizes) into the block sizes
- Uses a symmetric key throughout the encryption/decryption
- Symmetric key is used for both encryption/decryption, and it is a private/secret key that only sender and receiver should know



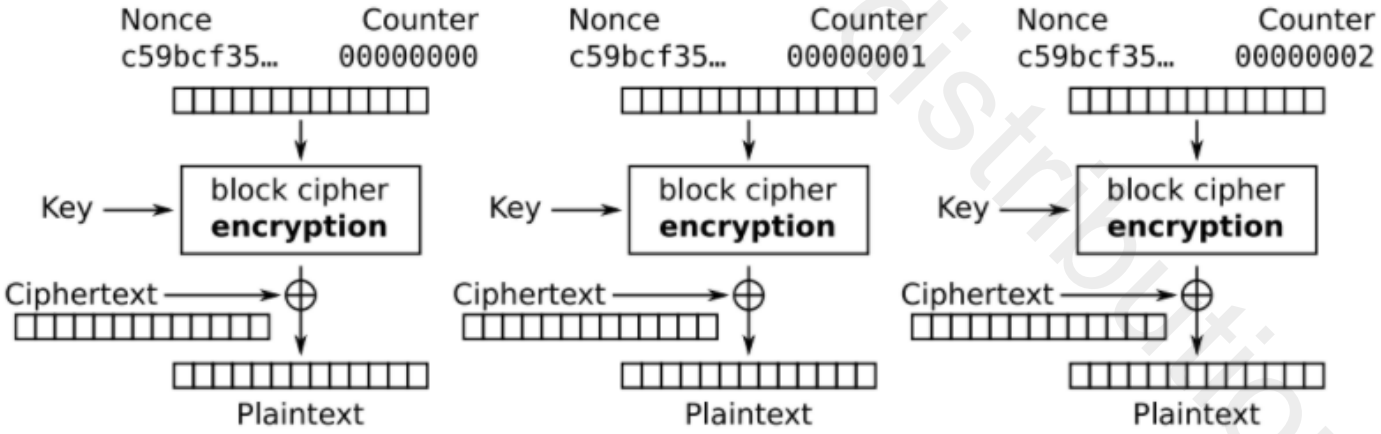
Block Cipher – CTR (Counter) Mode

- Each block gets its own “counter”
- Combination of this “counter” and nonce (pseudo-random number) per block is used as an input to the block cipher
- Encryption/decryption algorithm takes above combination and a key (symmetric, private) as inputs
- Plaintext blocks are XORed with the output from the encryption/decryption algorithm
- Independent of the previous blocks for processing encryption/decryption, thus possible to run in parallel
- Possible to only decrypt desired block of data

High Performance Cryptology System using FPGA

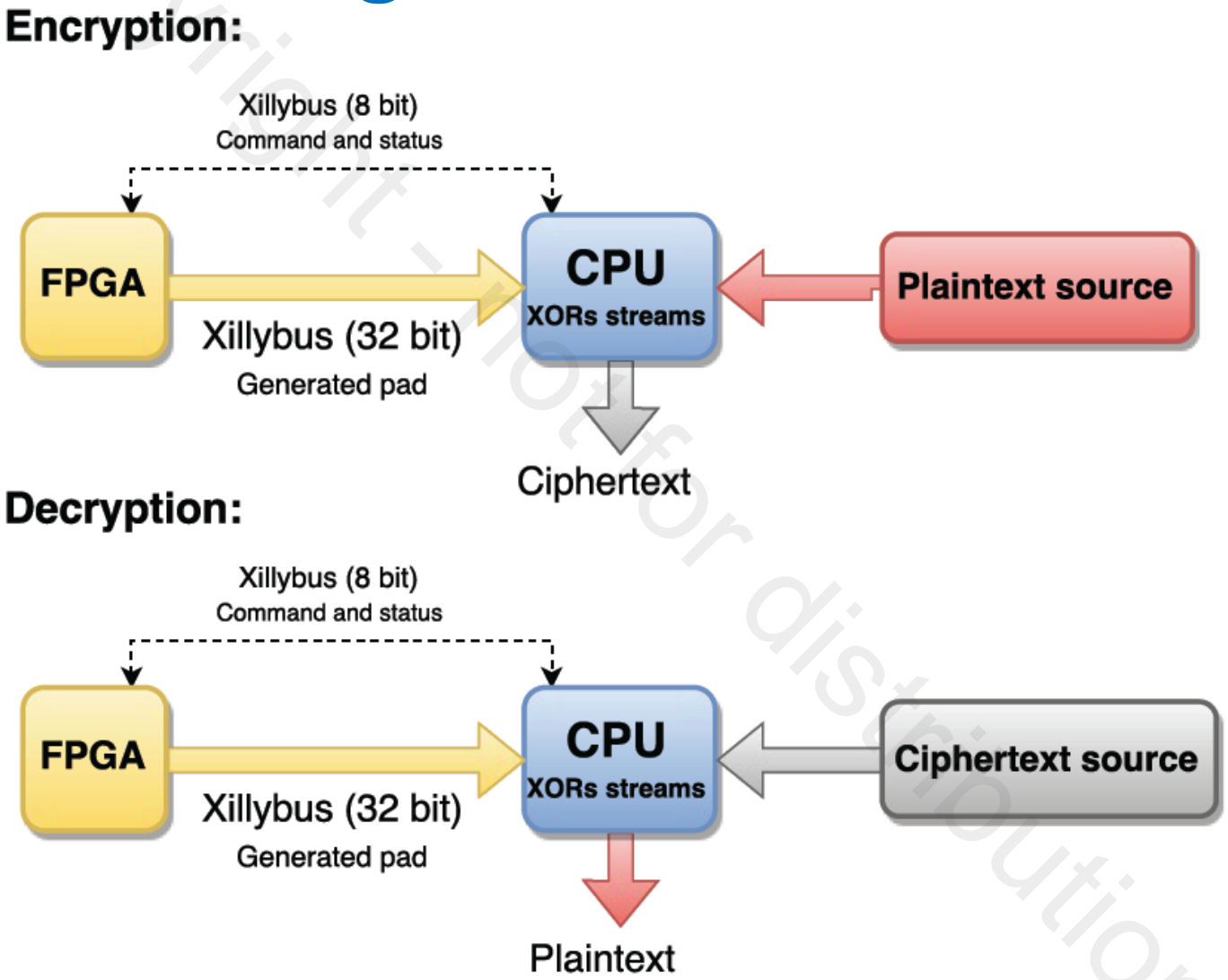


Counter (CTR) mode encryption



Counter (CTR) mode decryption

High Performance Cryptology System using FPGA



LINK : http://ecasp.ece.iit.edu/video/bsmp2016/ecasp_cryptology.mp4

High Performance Cryptology System using FPGA

- For encryption…
  - CPU loads expanded key on FPGA
  - IV (Initialization Vector)/Counter value also loaded; unique for a given key for security
  - Counter value fed as input block to the AES cipher
  - Counter increases after each block, creating a pseudorandom stream of data, called the pad
  - CPU can start/stop pad production on FPGA
  - CPU reads pad stream from FPGA, XORs with plaintext resulting ciphertext output
- For decryption…
  - Symmetrical process as above
  - Same pad generated from FPGA, XORs with ciphertext resulting plaintext output

A Few Cryptology Terms

Certificate

- A binary file that can be used for encrypting files, email, signing data, establishing secure communication over the Internet and authentication. Certificates include, at a minimum, details about the subject, and the public key that can be used for encryption purposes. Certificates can be generated by anyone, or issued by a certificate authority.

Ciphertext

- Data that has been encrypted

Cleartext

- Data that is not encrypted, and is readable by anyone

Cryptanalysis

- Study of cryptographic algorithms or encrypted data in an effort to decrypt it, circumventing the protections afforded by the encryption

Cryptography/cryptology

- Study of techniques for secure communications

A Few Cryptology Terms

DLP

- Data Loss Prevention. Technology designed to prevent confidential data from being released to unauthorized users. Frequently used with email systems to detect and prevent confidential data from being set out through email.

DRM

- Digital Rights Management. Technology used to help protect copyright and ensure that only authorized users have access to data. Uses encryption/decryption to validate authorized access, and frequently restricts what authorized users can do.

Encryption

- Process of encoding data in such a way that only authorized users or intended recipients can access the data.

A Few Cryptology Terms

Hash

- A one-way mathematical function that creates a unique, fixed length value from data of an arbitrary length. 128-bit hashes are 128 bit values that can be computed on data that is much smaller/larger (64 bit password or 4.7GB ISO file). Hashes can be computed by anyone, but cannot be reversed to determine any property of the data used to create the hash. Hashes are used to verify data integrity but not to secure the data

OTP

- One Time Password. A key that can be used only once. In theory, using OTPs means that an encryption scheme is practically unbreakable, but the logistics behind creating a workable OPT system and securely distributing it to the other party in a data exchange scheme makes OTPs very difficult to manage.

PIN

- Personal Identification Number. Used to seed encryption or to unlock encryption keys for use.

A Few Cryptology Terms

Key

- A mathematical value used in encryption and decryption. A key can be a simple password, or a complex derivative of complex mathematical and computations. The key is not the lock; that is the algorithm used for encryption.

Private Key

- One half of a key pair, and can be used to digitally sign communications to prove authenticity, or to decrypt data that was encrypted using the corresponding public key. As the name implies, you keep your private key private, and may further protect it by assigning a PIN that must be used to access the private key.

Public Key

- One half of a key pair, and can be used to verify a digital signature, or to encrypt data destined for the owner of the public key. As the name implies, your public key can be shared with others, and is often published in online directories or in certificates. The more widespread your public key, the more useful it is.

A Few Cryptology Terms

Salt

- A unique numeric value used to make an encryption scheme even more difficult to break as it makes the values that go into encryption even more unique than they might be on their own. Weak encryption systems can be made stronger by salting hashes.

SSL

- Secure Sockets Layer. Part of the transport layer of the TCP/IP suite that provides encryption of data in motion. Whenever you use HTTPS in a web browser, you are using HTTP over an SSL connection to ensure that no one can intercept your data on the wire and decrypt it.

TLS

- A newer form of SSL, Transport Layer Security provides stronger encryption than SSL can by using stronger algorithms. Often the two are analogous and used interchangeably.

Fabricating Your Own Chip

- ASIC (Application Specific Integrated Circuit)
  - Customized for a specific purpose
- Test on FPGA if it works fine, then make your own chip
  - You'll need to design EVERYTHING (transistor level)
- You need to know what is VLSI, CMOS, MOSFETs and more…
  - Will be taught in ECE VLSI courses…

Building Your Own Embedded Systems

- Fundamental concepts of electronics needed for building any embedded system project
- Communication protocols and their uses with respect to communication among electronic components
  - UART, SPI, I^2C Protocols
  - GPIO essentials
- How to utilize these through programming language
  - e.g., System C, C/C++, Python, Java, etc.

Electronics and Embedded Systems

- General Definition of ELECTRONICS
  - Hardware/circuit made up of several Integrated Circuits (ICs) with different resistors, capacitors, inductors and other components
- Programmable ICs
  - When powered, device follows steps written in the program
  - Robots, washing machines, vacuums, other home appliances
- Embedded System Developer
  - Must decide function written in software or implemented on hardware connecting logical ICs
  - Analyze speed, size, complexity, other parameters

Electronics Basics - Review

- Voltage
  - Allows flow of charges (electrons) from higher to lower potential
  - How much of voltage are you connecting to the device?
    - e.g., AA battery cell (3V), cellphone charger (5V), laptop adapter (20V)
- Current
  - Flow in a fixed direction from the positive to the negative terminal
  - How much of amperes are you sending to the device?
    - Too much of current could damage your device!
    - e.g., cellphone charger (0.7A – 2.5A), laptop adapter (5A – 7A)
- Resistor
  - “to try to stop or to prevent” flow of current
  - Used to safeguard from excessive current

Electronics Basics - Review

- Capacitor
  - Filter voltage supplied in filter circuits, to generate clear voice in amplifier circuits
  - A surge suppressor as power smoothing, decoupling, DC blocking…
  - With sensors, to hold voltage level for some time to give time for microprocessor to read the voltage value
    - Needs to be a stable voltage value until microprocessor reads
    - Avoiding erroneous calculations
  - Holding time depends on an RC time constant

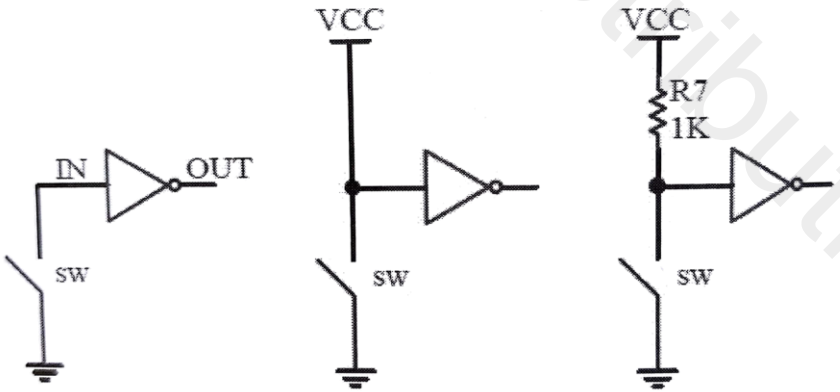
Electronics Basics - Review

- Open Circuit: No current flow
- Closed Circuit: two terminals connected
  - Other terms: short circuit, connected circuit, closed circuit

- DO NOT SHORT(directly connect) two terminals of a power supply such as batteries, adaptors and chargers
  - Cause serious damages including fire damage, component failure
  - Theoretically, think about Ohm's law where R is 0!

Electronics Basics - Review

- Pull-up/down resistors
  - Pull-up resistor pulls the terminal to the voltage supplied
  - Pull-down resistor pulls the terminal to the ground or common line
- Why use it?
  - To bring back the logic level to the default value when no input is present on that particular terminal
  - To make the circuitry susceptible to noise
  - To avoid *floating* state
  - Logic level (1 or 0) cannot be changed from a small variation in terms of voltages (due to noise)

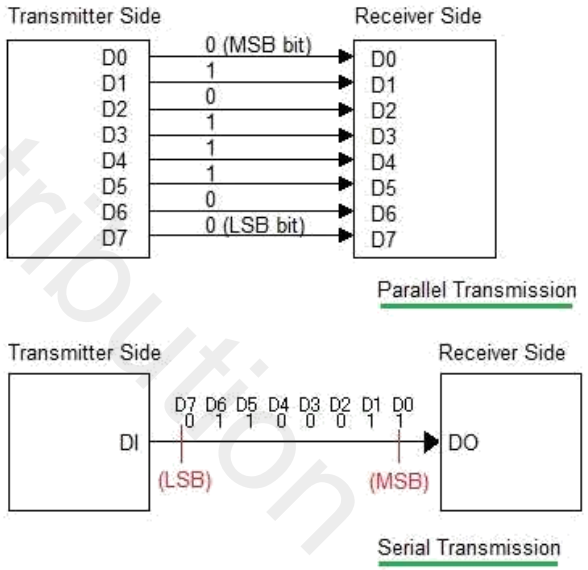


Electronics Basics - Review

- Floating state
  - When there is no connection on the input terminal
  - Not ON, not OFF, then what is it?? @_@
- Connecting input directly from VCC to GND
  - Will definitely damage your circuit for sure
- Must use a resistor to drop some voltage
  - Recommended to use throughout any sensors and ICs
  - Regardless of the datasheets/technical manuals

Communication Protocols

- Establishing communication between ICs/sensors as peripherals of a microprocessor
- Serial or Parallel Data-line connection between ICs and a microprocessor
  - Parallel: Faster than serial, less preferred (8,16+ lines required)
    - Can't have many free pins/lines to connect many peripherals
  - Serial: Slower than parallel but only requires **up to four lines**
    - Preferred to reduce space (low pin counts)
- Which one is better?
  - Depends on the specific application/system

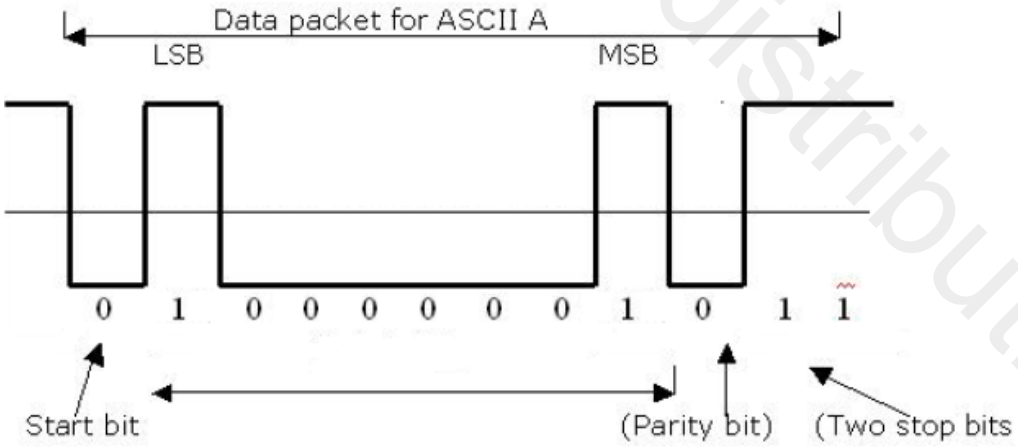


Communication Protocols

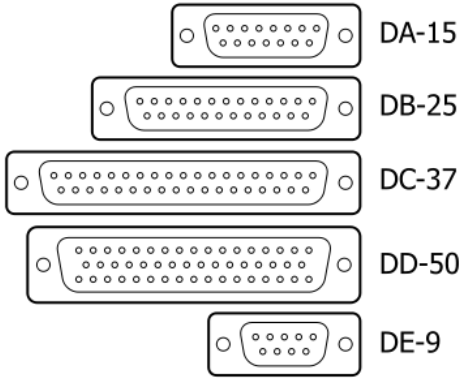
- How Serial Communication is established?
  - Data sent over frames or packets
    - Large data broken into chunks, sent over lines by a frame/packet
- Protocol: A set of rules for interfacing ICs to microprocessors
  - Data Frame Structures, Frame Lengths, Voltage Levels, Data Types, Data Rates, and more…
- Synchronous serial communication
  - **SPI**: Serial Peripheral Interface
  - **I^2C**: Inter-Integrated Circuit
- Asynchronous serial communication
  - **RS-232**
  - **UART**: Universal Asynchronous Receiver and Transmitter

Communication Protocols – RS-232

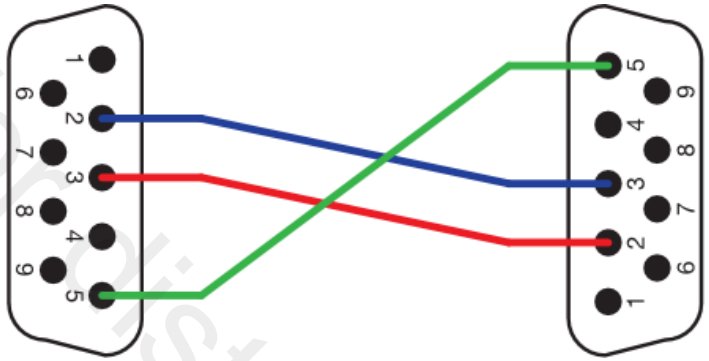
- One to one serial communication
- TX (transmit) and RX (receive) lines
- Transmission process (9600 baud, 1 bit=1/9600 s, approx. 0.1ms)
  - Transmit idles high (when no communication)
  - Goes low for 1 bit (0.1ms)
  - Sends out data, LSB first (7 or 8 bits)
  - May be a parity bit (even/odd for error detection)
  - May be a stop bit (or two)

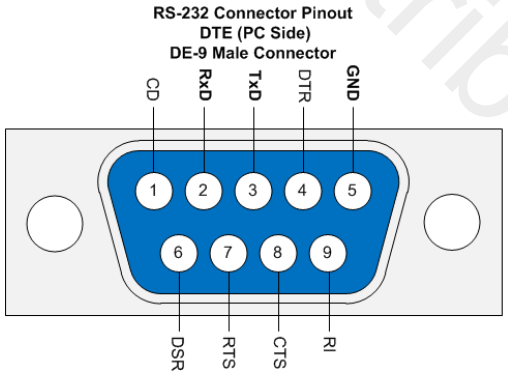


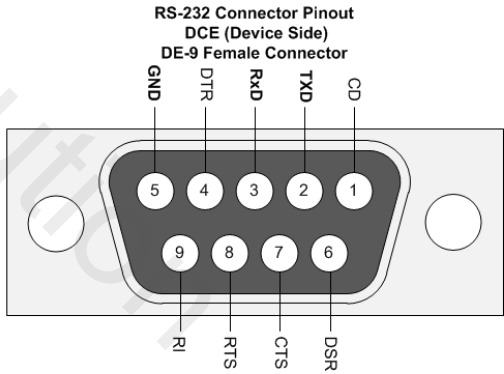
Communication Protocols – RS-232



- If don't need to worry about level conversions
  - NULL UART connection
- Example: DE-9 connector for serial connection
  - Pin 1: DCD (Data Carrier Detect)
  - Pin 2: RxD (Receive Data)
  - Pin 3: TxD (Transmit Data)
  - Pin 4: DTR (Data Terminal Ready)
  - Pin 5: GND
  - Pin 6: DSR (Data set Ready)
  - Pin 7: RTS (Request to Send)
  - Pin 8: CTS (Clear to Send)
  - Pin 9: RI (Ring Indicator)







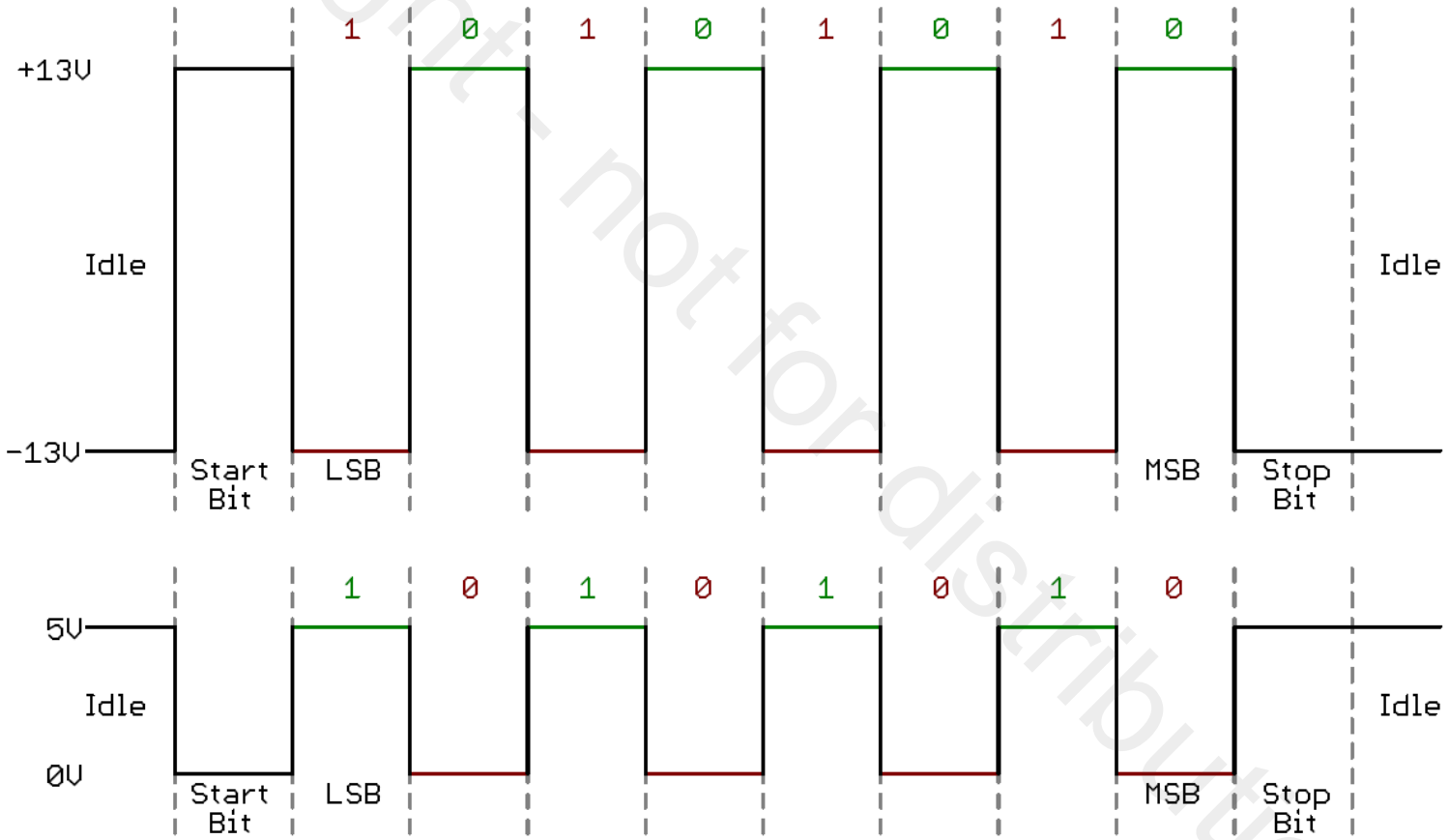
Communication Protocols – RS-232

- Some RS-232 connections using handshaking lines between DCE (Data Communications Equipment) and DTE (Data Terminal Equipment)
  - RTS (Request To Send)
    - Sent by the DTE to signal the DCE, that it is requesting to send
  - CTS (Clear To Send)
    - Sent by the DCE to signal the DTE, that it is ready to receive
  - DTR (Data Terminal Ready)
    - Sent by the DTE to signal the DCE, that it is ready to connect
  - DSR (Data Set Ready)
    - Sent by the DCE to signal the DTE, that it is ready to connect

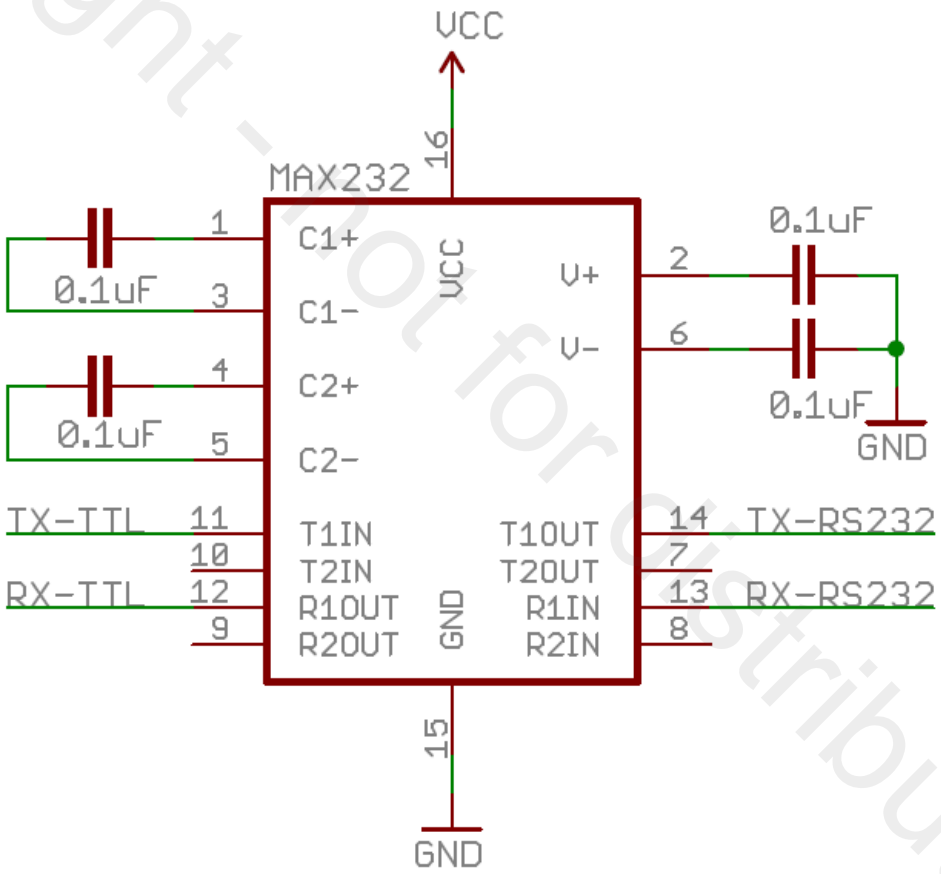
Communication Protocols - UART

- UART or RS-232 Standard
  - Full duplex, a complete standard including electrical, mechanical, and physical characteristics for a particular instance of communication
  - Data sent over a bus, varying data levels (voltage)
    - *(voltage)* > +3V : Logic '0'
    - *(voltage)* <= -3V: Logic '1'
    - -3V < *(voltage)* <= +3V: 'Undefined States'
- Voltage conversion needed from TTL to RS-232
  - TTL: Transistor-transistor logic (0V to 5V)
  - RS-232: (-13V to +13V)
  - Need a level shifter (e.g. MAX232)

Communication Protocols - UART



Communication Protocols - UART

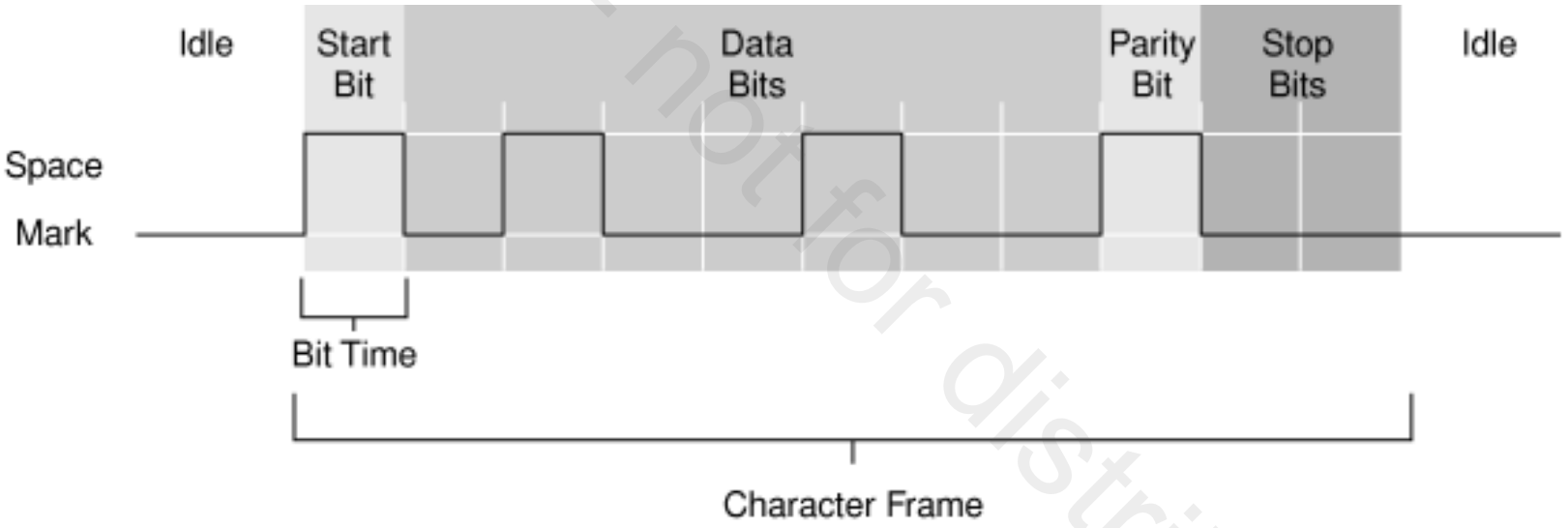


MAX232

Communication Protocols - UART

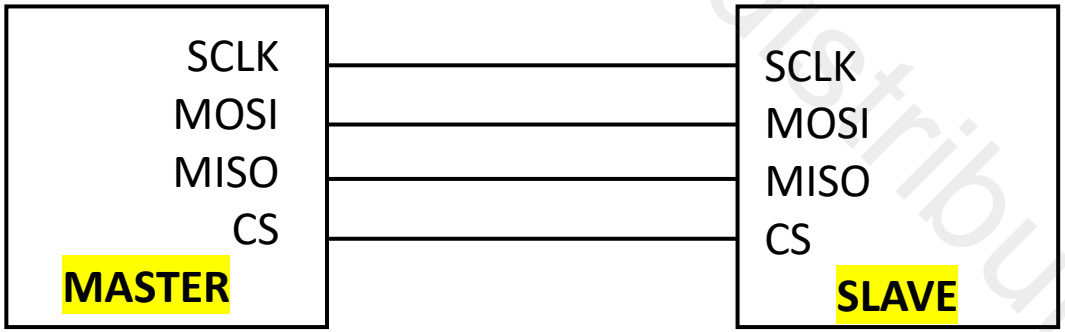
- Useful in short-distance communication
- Baud rate (symbols per second), Bit rate (bits per second)
  - Specifies how fast data is sent over a serial line
  - Must be matched between the transmitter and the receiver
  - e.g., 9600 bps, 9600 baud
- Typical UART frame consists:
  - Start bit: tells receiver that the data stream is about to start
  - Data bit: contains data (generally up to 8bits)
  - Parity bit: low-level error checking (odd or even)
  - Stop bit: tells receiver that the transmission is over
- Programs with UART driver to obtain data from peripherals

Communication Protocols - UART



Communication Protocols - SPI

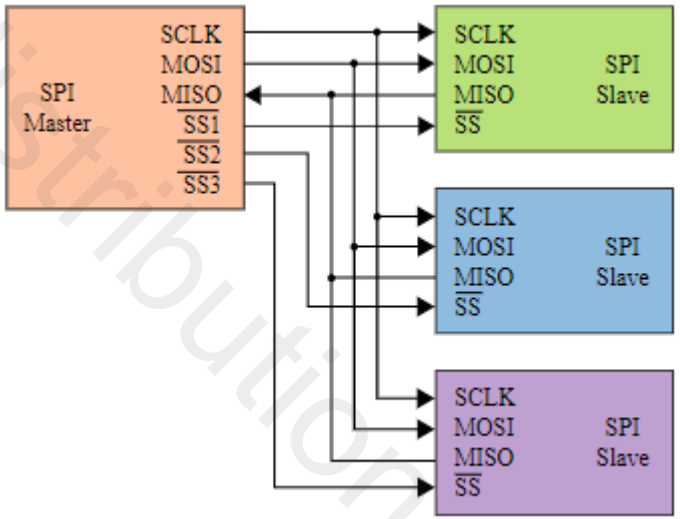
- Serial Peripheral Interface (SPI)
  - Full-duplex, short-distance, single-master protocol
  - Synchronous communication protocol
- Simple master-slave SPI connection
  - SCLK: Serial Clock (output from Master)
  - MOSI: Master Out Slave In (output from Master)
  - MISO: Master In Slave Out (output from Slave)
  - CS: Chip Select (also known as Slave Select, output from Master)



Simple master-slave SPI connections

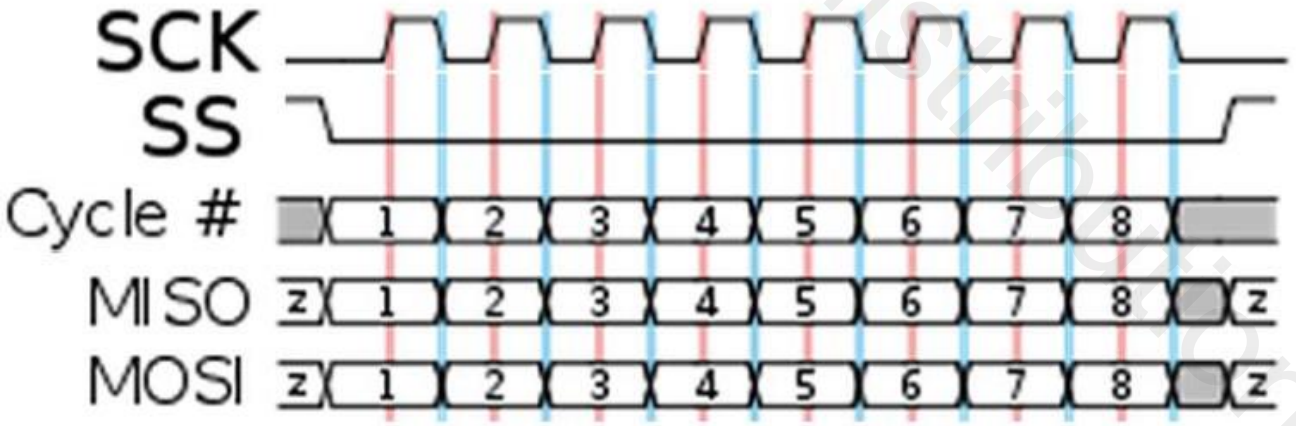
Communication Protocols - SPI

- Master ALWAYS initiates data frame and clock
- Clock frequencies vary, depending on master/slave
  - Typically from 1 MHz to 40 MHz or higher
- Some slave devices trigger on active low input
  - Logic zero signal from master → slave chip is ON → accepts clock and data
- Multiple slaves can be connected to a master device
  - Additional CS(SS) lines from master to multiple slaves
- NO ACK available
  - No way to know if data is received correctly



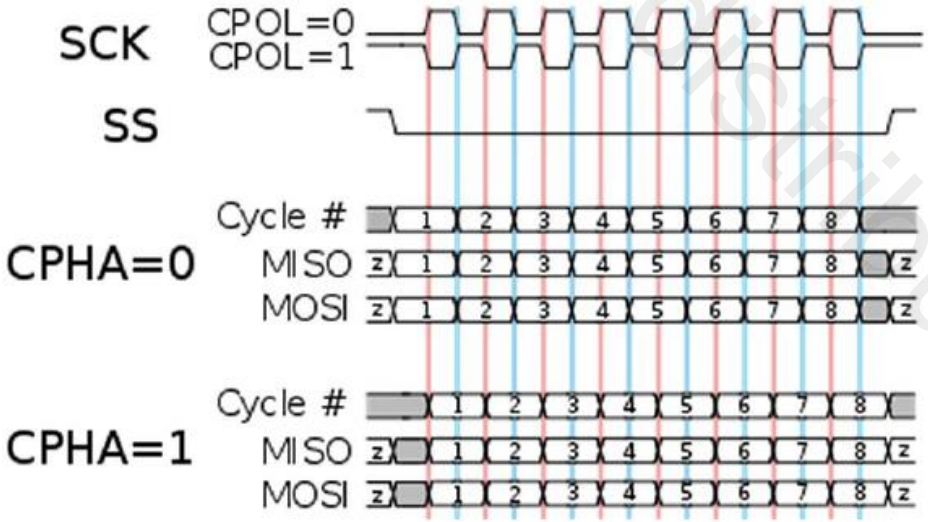
Communication Protocols - SPI

- Synchronized communications using clock
- Pull slave select line low to select device
- 1st bit of data gets put on MISO and MOSI (so a byte goes both ways)
- Data gets shifted out (typically 8 bits, but not necessarily)
  - Data gets put on bus on falling edge of clock
  - Data gets read on the rising edge of clock



Communication Protocols - SPI

- Clock can be set in different ways, determined by clock polarity and phase
- CPOL=0: Base value of the clock is 0
  - CPHA=0: Data read on rising edge, put on bus on falling edge
  - CPHA=1: Data read on falling edge, put on bus on rising edge
- CPOL=1: Base value of the clock is 1
  - CPHA=0: Data read on falling edge, put on bus on rising edge
  - CPHA=1: Data read on rising edge, put on bus on falling edge



Communication Protocols - SPI

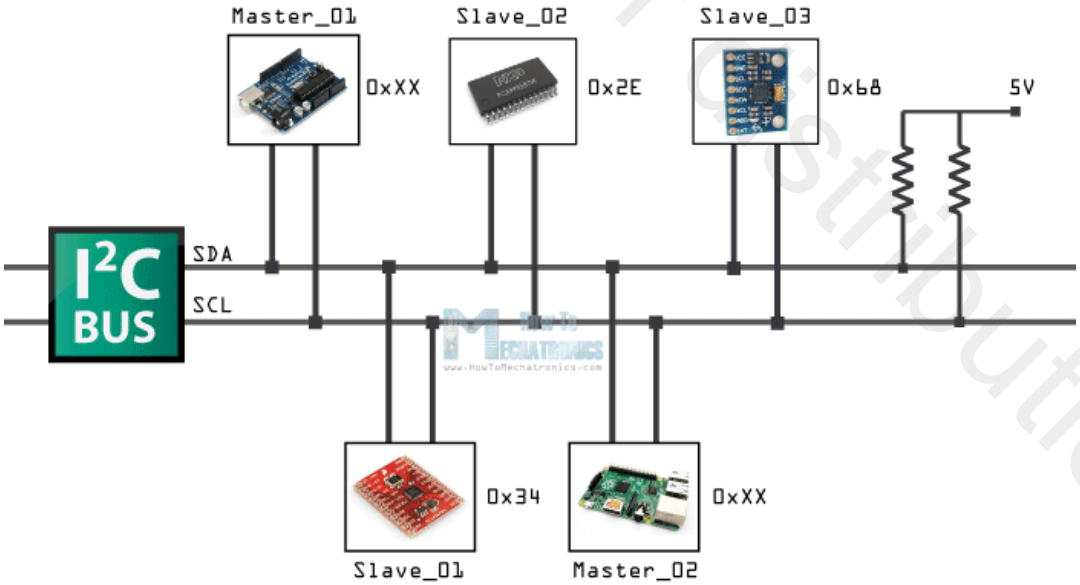
- Pros
  - Fast for point-to-point connections
  - Easily allows streaming/constant data inflow
  - No addressing in protocol, simple to implement
  - Supported broadly
- Cons
  - Chip select makes multiple slaves more complex
  - No ACK (can't tell if data was received)
  - No inherent arbitration
  - No flow control (must know slave speed)

Communication Protocols – I^2C

- Inter-Integrated Circuit Bus
- Developed and patent by Philips Semiconductors
  - Original purpose: Connect a CPU to peripheral chips in a TV-set
- Peripheral devices in embedded systems
  - Connected to microcontroller as memory-mapped I/O devices
- Requires only two wires, data (SDA) and clock (SCL)
  - 2-wire communication bus to provide communication link between integrated circuits
  - Always pulled up via resistors to the input voltage
- Half duplex: sender sends the command, the receiver just listens and cannot transmit anything; and vice versa
- Three speeds available
  - High speed (3.4 Mbps)
  - Fast speed (400 Kbps)
  - Slow (<100 Kbps)

Communication Protocols – I²C

- Most Widely used protocol for sensor interfacing in embedded systems
  - Accelerometer, gyroscope, temperature, humidity sensors
  - Stepper motor control, proximity sensor and more
- Useful for low-pin-count devices
  - All sensors can be tied to a single I²C bus
  - How differentiated?



Communication Protocols – I^2C

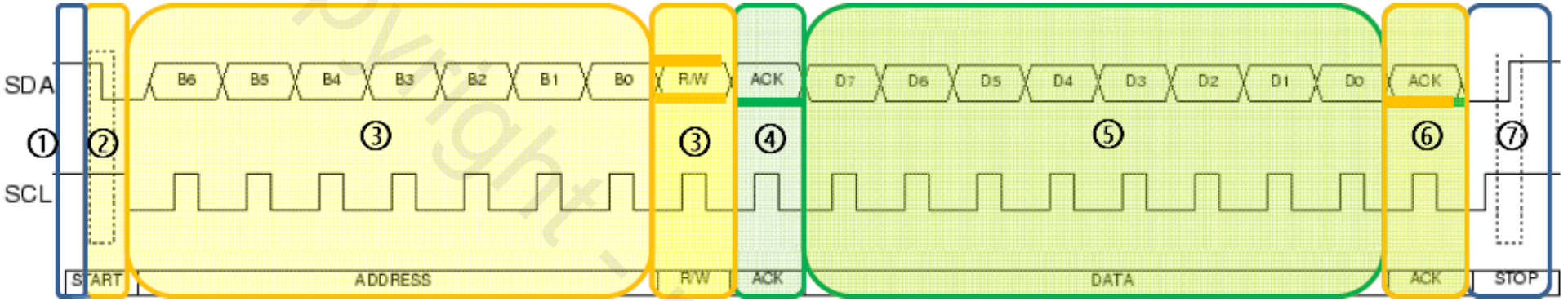
- Each I^2C device has an address of 7 or 10 bits
  - I^2C lines can be shared with other I^2C peripherals
  - Can be differentiated by identifying address
  - Master can always connect and send data meant for specific slave
  - Slave device manufacturer needs to provide address to use
  - If address conflicts with other devices, it will not work properly as you hoped to be! Address must be unique to a device within same I^2C bus
- Data is received at every slave, but only that slave can take data for which it is made
- Master reads data available in predefined data registers in the device

Communication Protocols – I²C

- 7-bit addressing
  - Slave address is transferred in the first byte after the Start condition
  - First 7 bits represents the slave address, 8th bit is R/W* bit

- 8 bit addressing (not recommended)
  - If vendor has specified an address outside of 0x08 to 0x77
  - LSB can be mistakenly read as R/W*
  - May be two different addresses for Read and Write
- 10 bit addressing
  - Designed to be compatible with 7-bit addressing
  - Special indicator used for 10-bit addressing

Example – I²C Read/Write



For WRITE

1. All: allow SDA, SCL start high
2. Master: SDA low to signal start
3. Master: Send SCL, 7 bit device address followed by 0 (W*) on SDA
4. Slave: Pull SDA low to ACK
5. Master: Send one byte data on SDA
6. Slave: Pull SDA low to ACK
7. All: allow SDA to go high when SCL is high (stop)

For READ

3. Master: Send SCL, 7 bit device address followed by 1 (R) on SDA
5. Slave: Send one byte data on SDA
6. Master: Pull SDA low to ACK

Useful Tips and Precautions

- Avoid touching electronic components on the device
  - Sweat, static charge from your body can damage boards
  - Touch devices from the corners or use a casing to cover the device
- Put your device on the table, avoid laying on metal
  - It can be shorted…
- NEVER connect higher voltage than its original configuration
  - If a voltage pin is designed for 3.3V, then use only 3.3V!
  - Be careful not to connect voltage to ground, ground to voltage!

Acronyms

Acronym	Meaning	Acronym	Meaning
ACK	Acknowledgment	DRAM	Dynamic RAM
ADAS	Advanced Driver Assistance Systems	DSP	Digital Signal Processing
CPHA	Clock Phase	DSR	Data Set Read
CPOL	Clock Polarity	DTE	Data Terminal Equipment
CS	Chip Select	DTR	Data To Read
CTS	Clear To Send	EEPROM	Electrically Erasable Programmable Read Only Memory
DCE	Data Communications Equipment	FPGA	Field Programmable Gate Array

Acronyms

Acronym	Meaning	Acronym	Meaning
GFLOPS	Giga Floating Point Operations per Second	IoT	Internet of Things
GPIO	General Purpose Input Output	LCD	Liquid Crystal Display
HDD	Hard Disk Drive	LED	Light Emitting Diode
HLS	High Level Synthesis	LUT	Look Up Table
HUD	Heads Up Display	MISO	Master Input Slave Output
I2C	Inter-Integrated Circuit	MOSI	Master Output Slave Input
IDE	Integrated Development Environment	NVMe	Non-Volatile Memory Express

Acronyms

Acronym	Meaning	Acronym	Meaning
OpenCV	Open Source Computer Vision	SATA	Serial ATA (AT Attachment)
PCIe	Peripheral Component Interconnect Express	SCLK	Serial Clock
PWM	Pulse Width Modulation	SD Card	Secure Digital Card
RAM	Random Access Memory	SDA	Serial Data
ROM	Read Only Memory	SoC	System on Chip
RTL	Register-Transfer Level	SPI	Serial Port Interface
RTS	Ready To Send	SRAM	Static RAM

Acronyms

Acronym	Meaning	Acronym	Meaning
SS	Slave Select	XOFF	Transfer OFF
SSD	Solid State Disk	XON	Transfer ON
UART	Universal Asynchronous Receiver Transmitter		
USB	Universal Serial Bus		
VHDL	VHSIC Hardware Description Language		
VHSIC	Very High Speed Integrated Circuit		
VLSI	Very Large Scale Integration		