# Industrial Embedded Systems
# - Design for Harsh Environment -

Dr. Alexander Walsch
alexander.walsch@ge.com

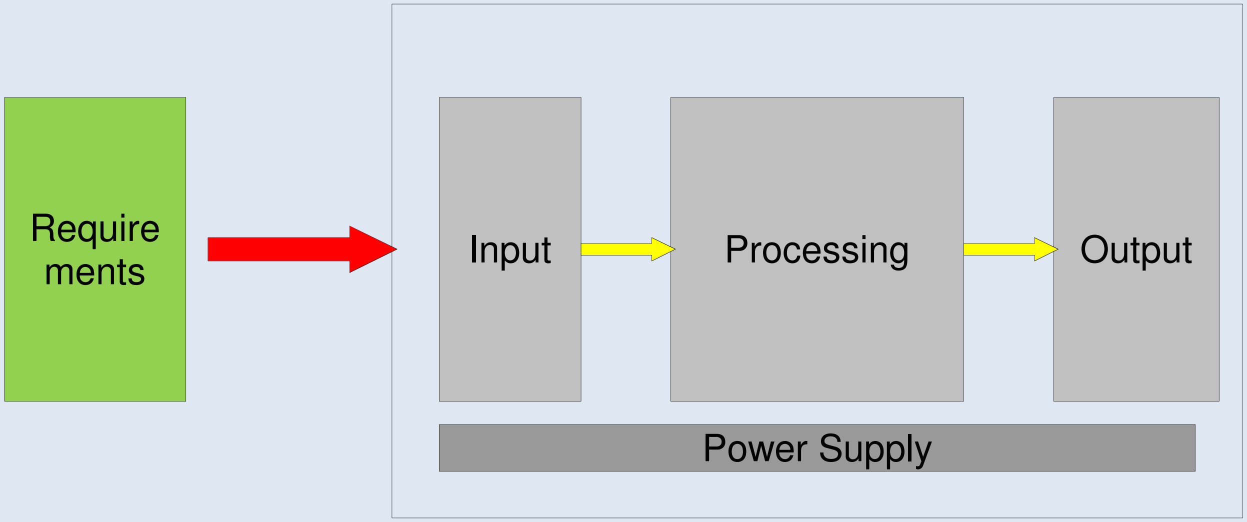
Part V

WS 2012/13

Technical University Munich (TUM)

# Architecture and Detailed Design
# - Overview -

- We do have a requirements specification
- Requirements are mapped to hardware and software
- Compile an „architecture design document“ which is the technical specification of the system as realized
- The „embedded system researcher/developer“ needs a sound understanding of hardware

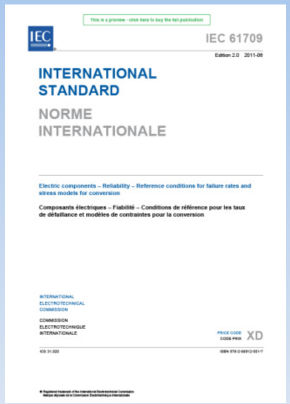


# Approach

- Identify standards (internal, best practice, etc.) which should be used (overlap with requirements analysis)
  - Company internal design practice: packages, traces, component size limits
  - Proven in use components (the latest stuff is not always the best)
  - Standards: architectural and design constraints imposed by certification
- Specify major hardware building blocks and do a rough footprint calculation
- Identify hardware design patterns if applicable (reusable principles, white papers, etc.)
- Specify interfaces between hardware building blocks
- Schematic entry, layout and manufacturing

# Reliability Standards

- There are many possible sources for reliabilty metrics.
- They provide a model to obtain a failure rate for a large number of electronics components
- The models vary and should be understood as a guidline (e.g. putting specific stress on a component may increase its failure rate by 20%)



This is a preview - click here to buy the full publication

IEC 61709

Edition 2.0 2011-06

INTERNATIONAL STANDARD

NORME INTERNATIONALE

Electric components – Reliability – Reference conditions for failure rates and stress models for conversion

Composants électriques – Fiabilité – Conditions de référence pour les taux de défaillance et modèles de contraintes pour la conversion

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

PRICE CODE
CODE PRIX XD

ICS 31.020

ISBN 978-2-88912-551-7

® Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

**IEC 61709:**
Parts count method.

This is a preview - click here to buy the full publication

TECHNICAL REPORT

IEC TR 62380

First edition 2004-08

Reliability data handbook –
Universal model for reliability prediction of electronics components, PCBs and equipment

© IEC 2004 — Copyright - all rights reserved

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

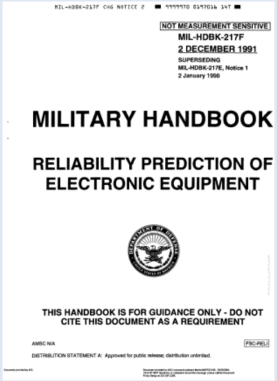
International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XC

For price, see current catalogue

**IEC 62380:**
Parts stress method.



MIL-HDBK-217F CHG NOTICE 2 ■ 9999970 0197016 14T ■

NOT MEASUREMENT SENSITIVE

MIL-HDBK-217F
2 DECEMBER 1991
SUPERSEDING
MIL-HDBK-217E, Notice 1
2 January 1990

MILITARY HANDBOOK

RELIABILITY PREDICTION OF ELECTRONIC EQUIPMENT

THIS HANDBOOK IS FOR GUIDANCE ONLY - DO NOT CITE THIS DOCUMENT AS A REQUIREMENT

AMSC N/A

FSC-RELI

DISTRIBUTION STATEMENT A: Approved for public release; distribution unlimited.

**MIL-HDB-217:**
Parts stress method.

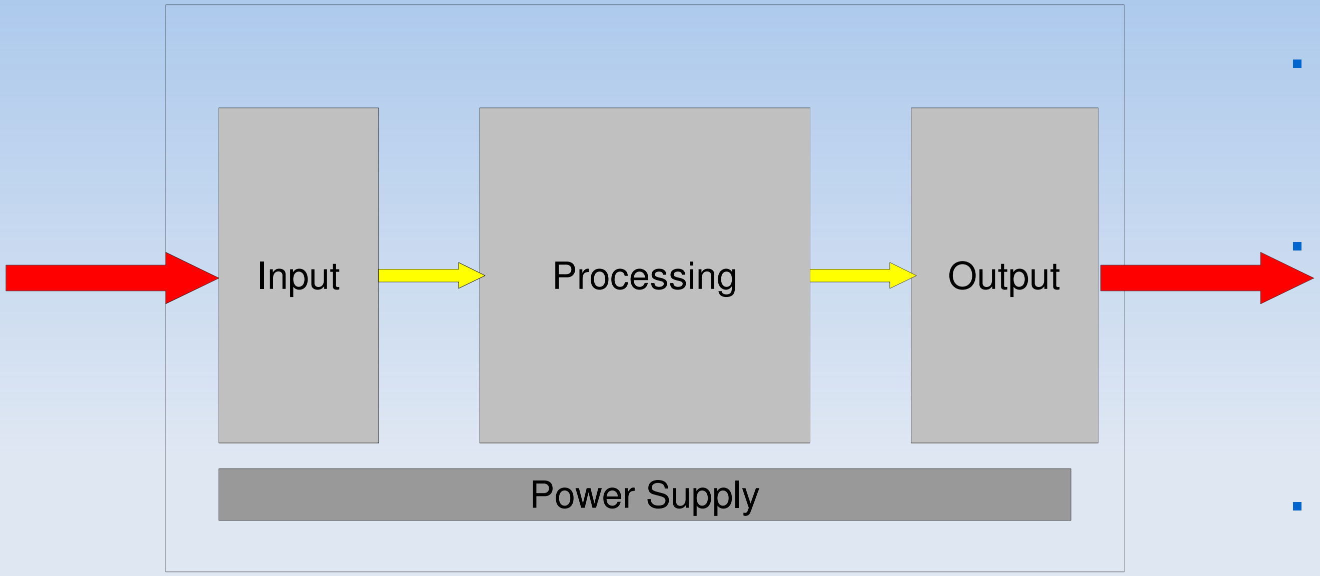
# Reliability Standards II (from wikipedia)

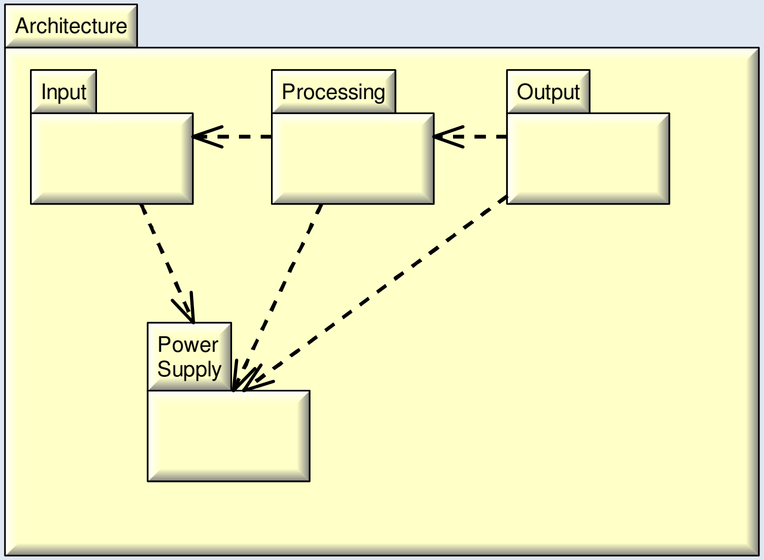
- All the models use basically the same process, with detailed variations.
  - Identify the components in the system
  - Such as R123, 10kOhm carbon film resistor
  - For each component, determine the component model to use from the standard
  - Such as "resistor, film, < 1 Megohm" or "Connector, multi-pin"
  - From the standard's component model, discover what, if any, complexity parameter is needed, and find the value of that parameter for this component (pin count, gate count e.g.)
  - From the standard's component model, discover what thermal stress curve applies, and find the value of the temperature in operation for this component

# Reliability Standards III (from wikipedia)

- From the standard's component model, discover what, if any, part stress parameter is needed, what part stress curve applies, and find the value of that part stress parameter for this component in this application (e.g. applied power relative to rated power)
- From the standard's component model, find the base failure rate for this component, and modify that according to the complexity parameter, the operating temperature and thermal stress curve, the part stress parameter and part stress curve, with arithmetic specified by the standard. This now is the expected failure rate for this component in this application
- Add up all the results for every component in the system to find the overall failure rate for all components in this system.

# Graphical Representation





- Block diagram widely used in HW design
- Easy to understand and sufficient for architectures
- UML has no advantage since hardware needs to be described in CAD tool from scratch anyways
- Block diagram a signal-oriented approach (also widely used in modeling control and signal processing systems)

# Major System Hardware Components

- We will look into the following components and introduce design patterns and/or explain basic principles:
  - Processors  - uC, DSP, general purpose
  - Reconfigurable electronics - CPLD, FPGA
  - Analog Input – Op Amps, INAs, filters, ADCs
  - Analog Outputs – DACs
  - Power supplies – step down/up converters
  - Connectors, Communication
- Afterwards we will try to take the patterns and apply it to the PMU

# Processors

- Processor (CPU) itself contains data path, control
- Embedded processors usually contain
  - Memory (SRAM, EEPROM, Flash)
  - Communication (UART, I2C, SPI, CAN, etc.)
  - Internal watchdog
  - Peripherals (ADC, DAC, PWM, Timer, Capture-Compare, etc.)
- Embedded processors are System-on-Chips and do not need any general hardware configuration (it is vendor specific). There are two add-ons that we want to highlight:
  - Reset circuitry (power on reset, brown out detection)
  - External watchdogs (program flow supervision)

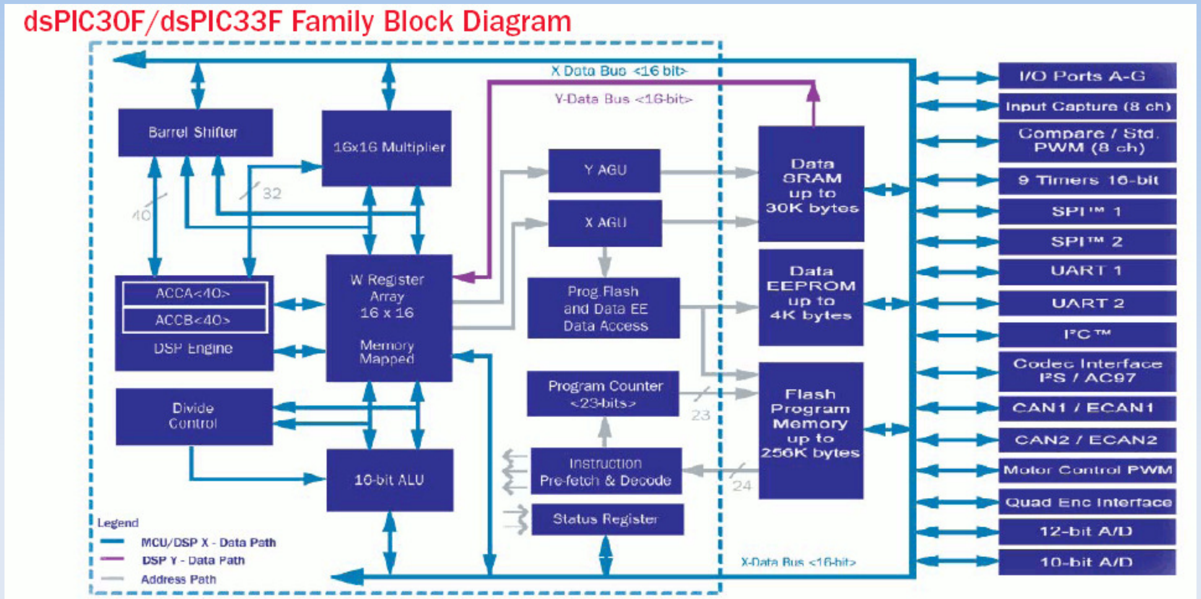
# CPU

- Often the CPU is not a free choice:
  - Prior usage
  - Tools already available
  - Long term availability or scalability (CPU roadmap)
  - Communication interfaces in package
  - Multiple suppliers (e.g. IP cores like ARM/MIPS)
  - Legacy code
- However, the CPU performance highly (only) depends on the function you want it to execute and on the machine code

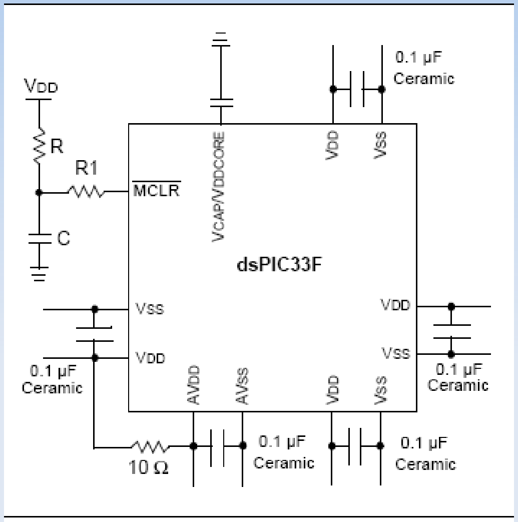
# Memory

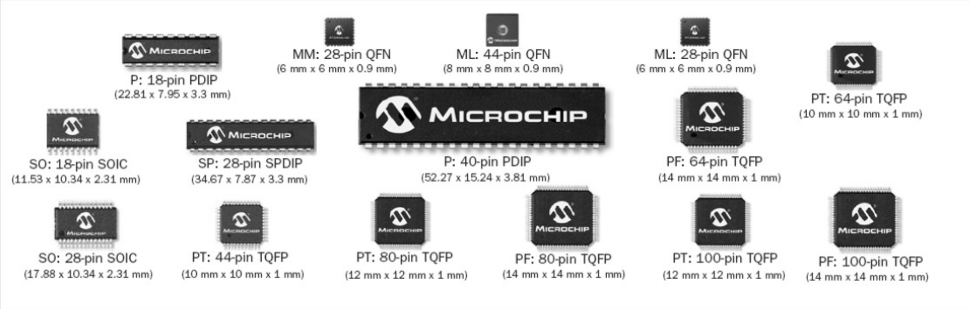
- Parameter memory (non volatile)
  - EEPROM – byte wise read and write – holds e.g. configuration parameters, run-time parameters (hour meter, status)
- Program memory (non volatile)
  - Flash (NOR)– word wise read, write requires a block erase - holds executable (XIP – execute in place)
- Data memory (volatile)
  - RAM (SRAM) – word wise read and write addressable - holds data and stack

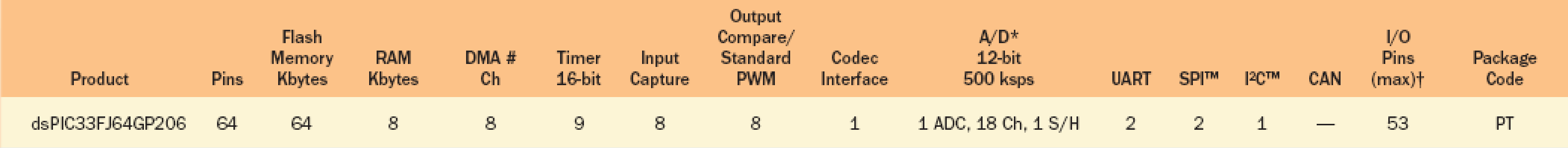
# Example: Microchip dsPIC



Source: microchip.com



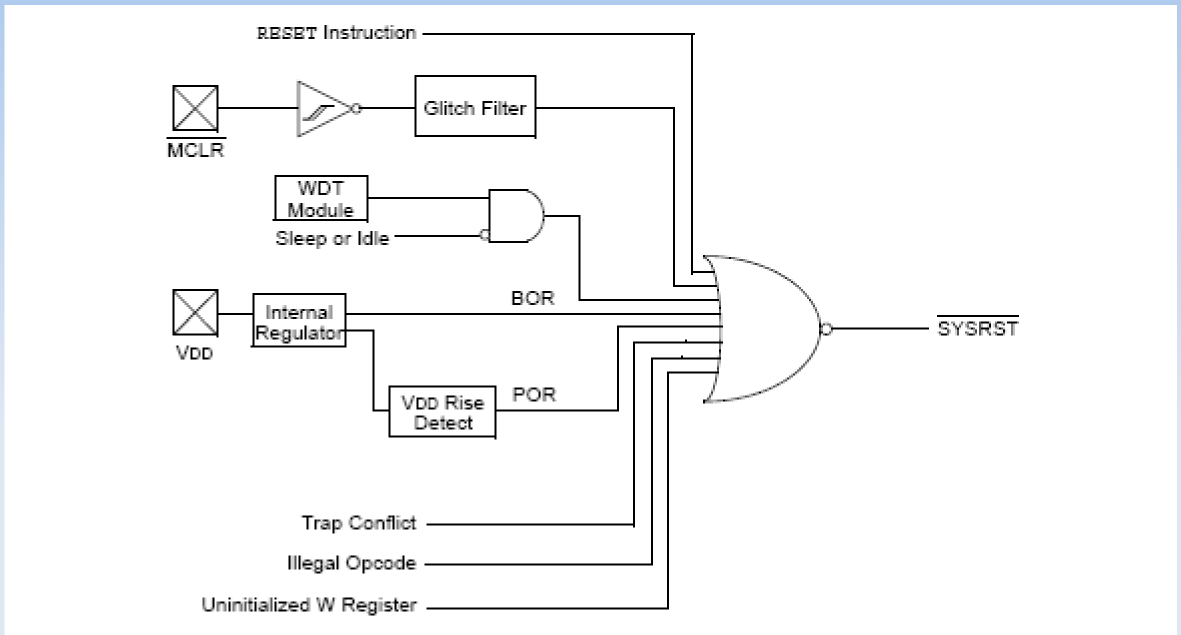




| Product | Pins | Flash Memory Kbytes | RAM Kbytes | DMA # Ch | Timer 16-bit | Input Capture | Output Compare/ Standard PWM | Codec Interface | A/D* 12-bit 500 ksps | UART | SPI™ | I²C™ | CAN | I/O Pins (max)† | Package Code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dsPIC33FJ64GP206 | 64 | 64 | 8 | 8 | 9 | 8 | 8 | 1 | 1 ADC, 18 Ch, 1 S/H | 2 | 2 | 1 | — | 53 | PT |

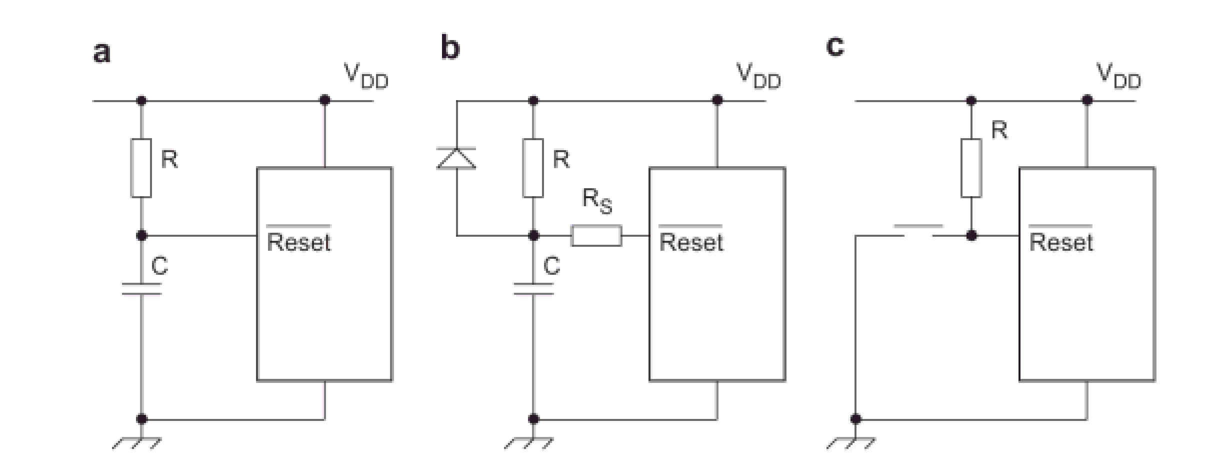
# Reset Circuits

- Reset: puts the system into a defined state
- A reset can be requested for many reasons:
  - Instruction
  - External event or fault
  - Internal Fault
- Reset stops the processor immediately. After release the reset vector is executed.
- The cause of a reset is usually indicated by a flag internal to the processor



Source: microchip.com

# Reset Circuits II
# - Power on Reset -

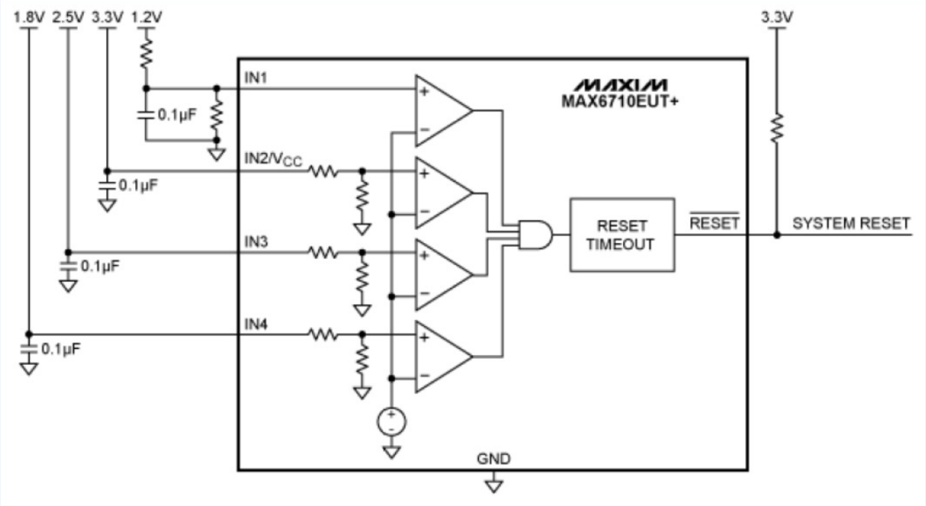

a
$V_{DD}$
R
C
$\overline{\text{Reset}}$
b
$V_{DD}$
R
$R_S$
C
$\overline{\text{Reset}}$
c
$V_{DD}$
R
$\overline{\text{Reset}}$

Source: Wilmshurst, Designing Embedded Systems with PIC Microcontrollers

- Example circuits for active low resets (processor is reset if voltage is low, operational if voltage is high)
  - a) simple power on reset circuit
  - b) reset pin current limitation, additional current path to allow quick power cycle with subsequent power on reset
  - c) reset button (only for lab tests – not populated in products)

# Reset Circuits III
# - Voltage Supervision -

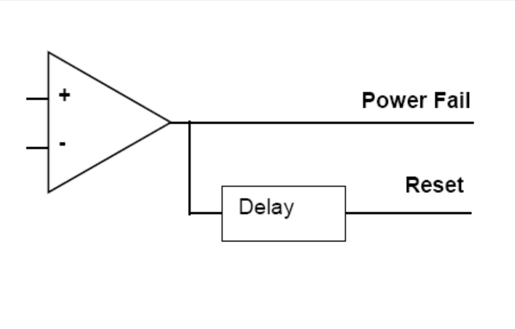
- Commercial products require supervision of several board level voltages
- Voltage violations other than the processor voltage can cause erronous results
- Supervisory and reset circuits are available from all major IC companies (Maxim, Analog Devices, ...). There are programmable ones where limits can be set by software.



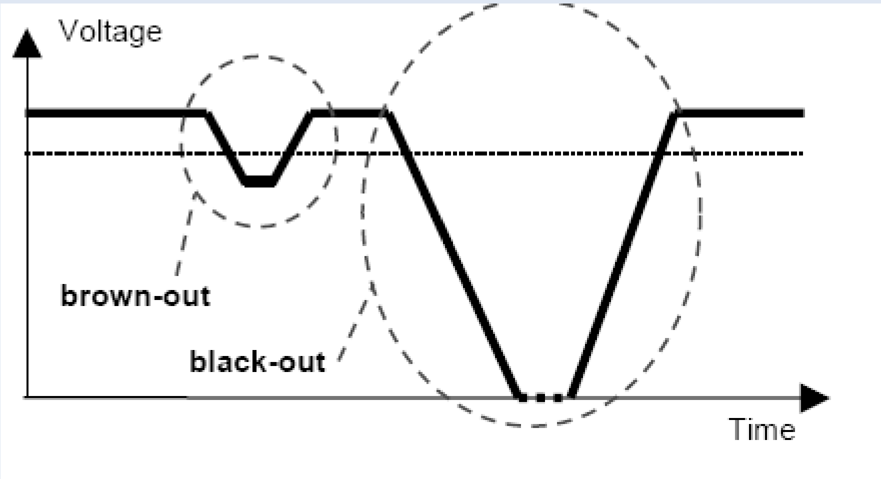
Source: Maxim AN540

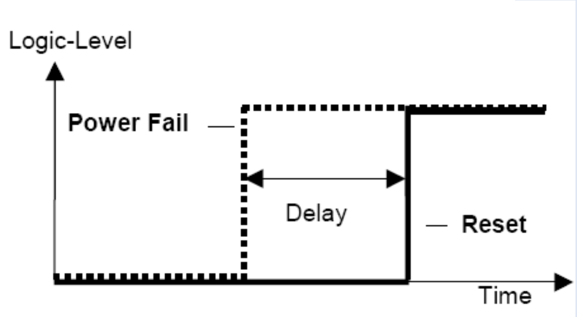
# Reset Circuits IV
# - Brown Out Detection -

- Brown out detection (BOR) important for battery operated systems
- Voltage glitch or gradually decreasing voltage
- Some data might to be stored in non-volatile memory (e.g. EEPROM)
- Routing of a delayed reset to an NMI triggers a data storage routine



Source: Philips AN468

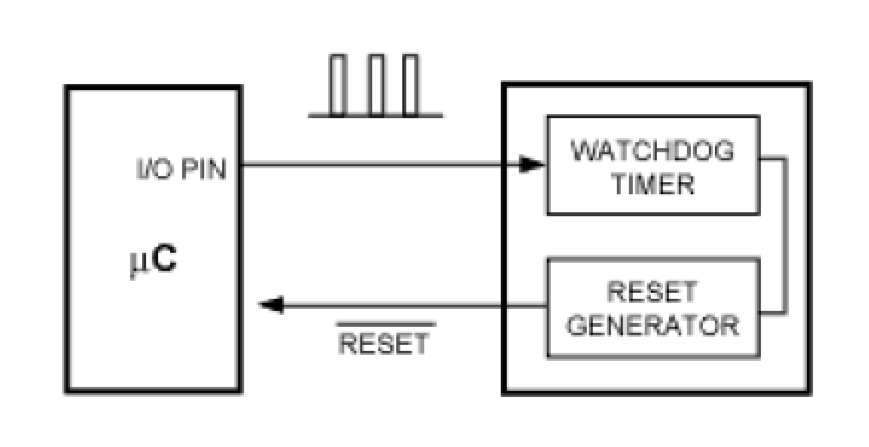


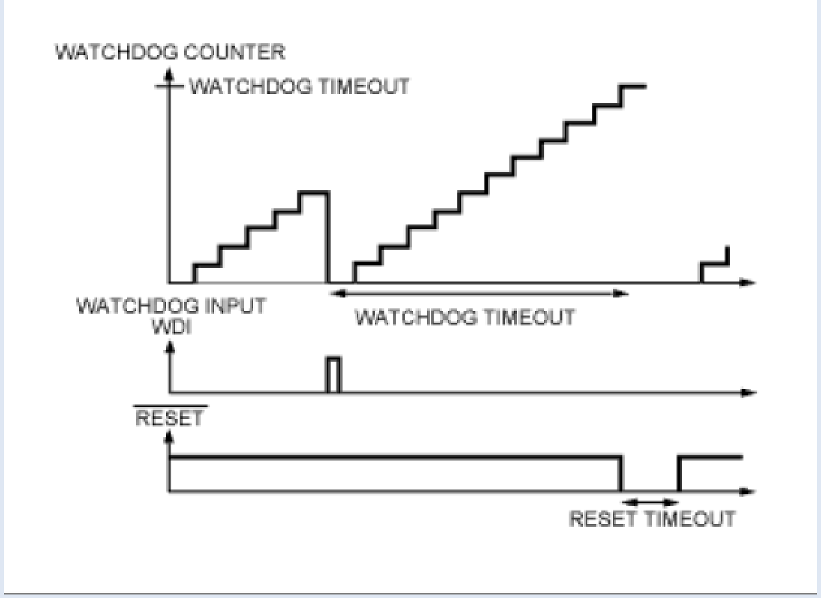


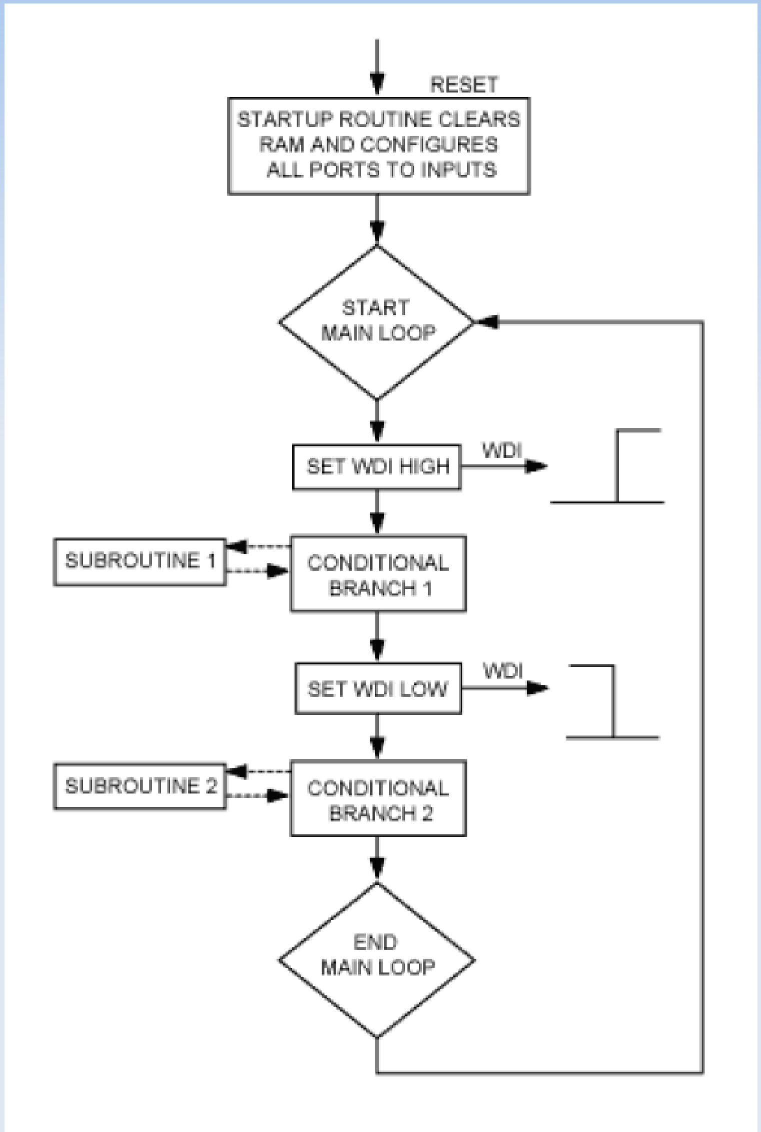
# Watchdog Circuits

- A watchdog timer is a supervisory component which must be triggered in regular intervals in order to avoid system reset
- Embedded processors usually come with internal watchdog circuits.
- A failure mode (drift) of the oscillator (account for in FMEA) makes a second external one with a separate clock source highly advisable for robust systems.
- Internal watchdogs can be disabled accidentally by software
- Set and reset the watchdog in different parts of the software to disallow stuck-at watchdog pulse loops

# Watchdog Circuits II







Source:
Maxim AN1926

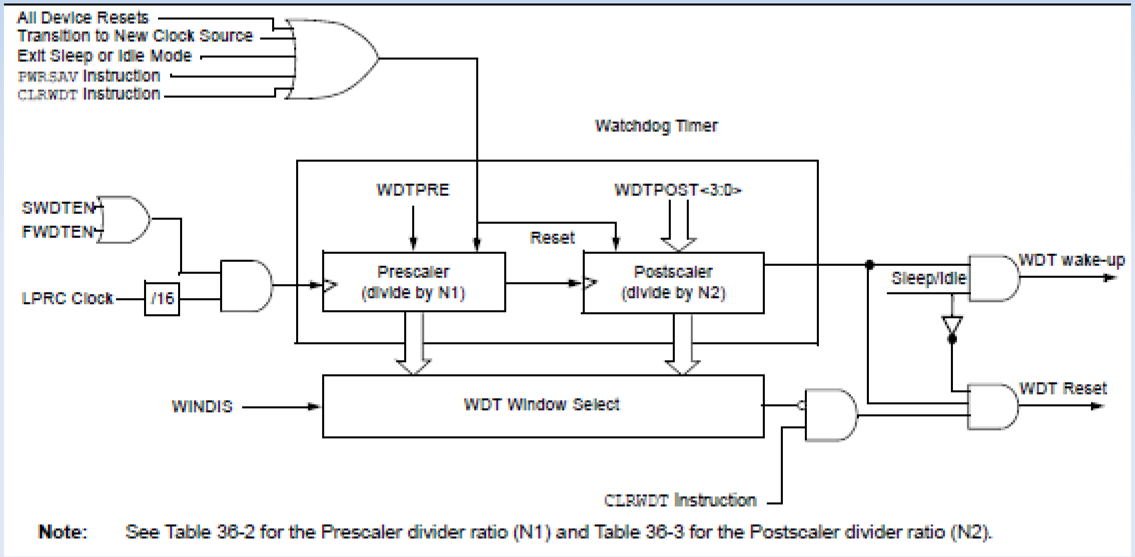
# Watchdog Circuits III
# - 1oo1D architecture -

- Embedded processor is supervised by an external watchdog and a separate independent output is provided
- A single channel can fail dangerously and safely: $\lambda_d$, $\lambda_s$
- Dangeros failures (different failure modes) can be detected and undetected: $\lambda_{dd}$, $\lambda_{du}$
- Dangerous detected failures are „converted“ into safe failures (the response is the safe state).
- An external watchdog detects deviations in software execution sequence and timeliness and thus detects possible dangerous failures.

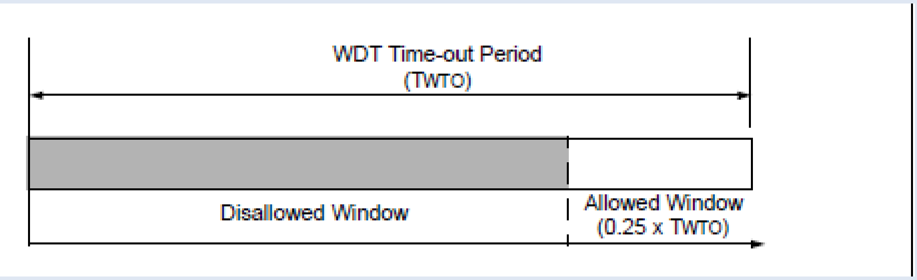
# Watchdog Circuits IV

- Standard watchdog
- Windowed watchdog

Source:
Microchip, dsPIC30F



**Note:** See Table 36-2 for the Prescaler divider ratio (N1) and Table 36-3 for the Postscaler divider ratio (N2).

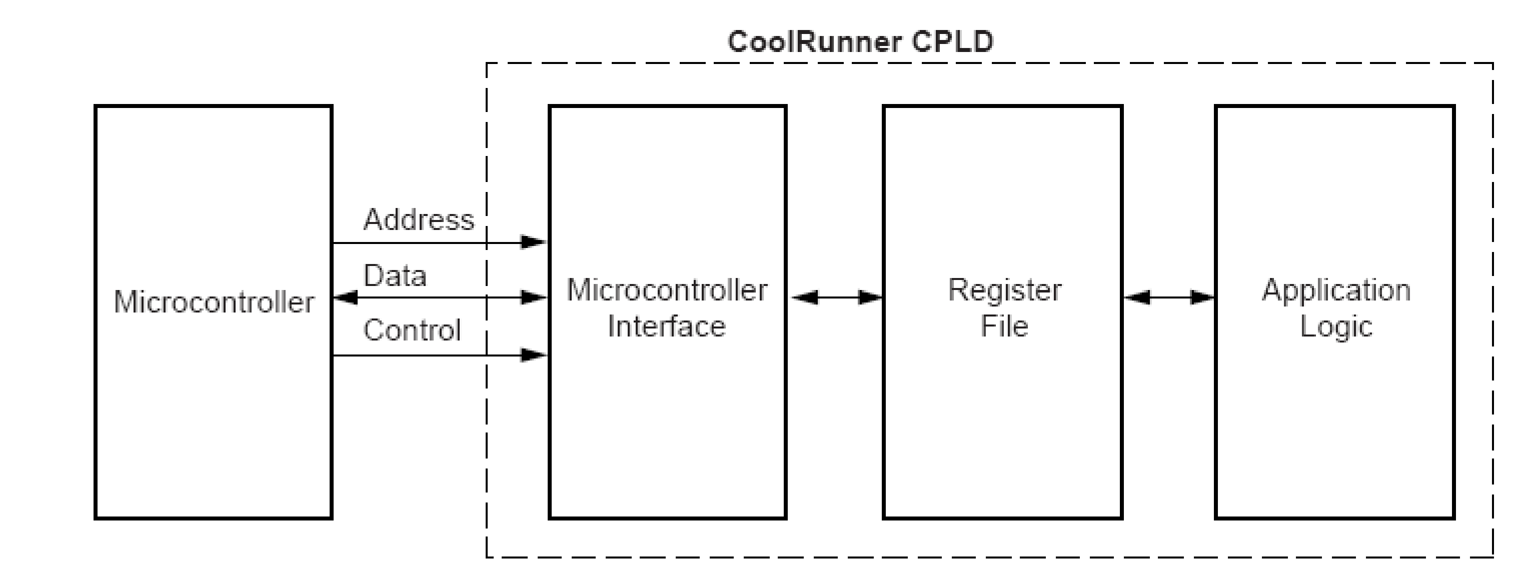


# Reconfigurable Electronics

- FPGAs, CPLDs widely used. More exotic reconfigurable components out there – e.g. analog FPAAs
- Functionality is specified by high-level languages (e.g. VHDL, Verilog) or schematic entry.
- Coprocessors for time-critical functions, I/O or signal processing. Connected via parallel or serial standard interface (digital) or configurable part of conditiong electronics.
- FPGAs mostly SRAM based (volatile) while CPLDs mostly Flash or EEPROM based (non-volatile).
- CPLDs are less complex and used for the realization of combinatorial and sequential electrical circuits with a deterministic pin-to-pin latency by design.
- FPGAs at the high end are very costly and hold complex circuitry.

# Reconfigurable Electronics II

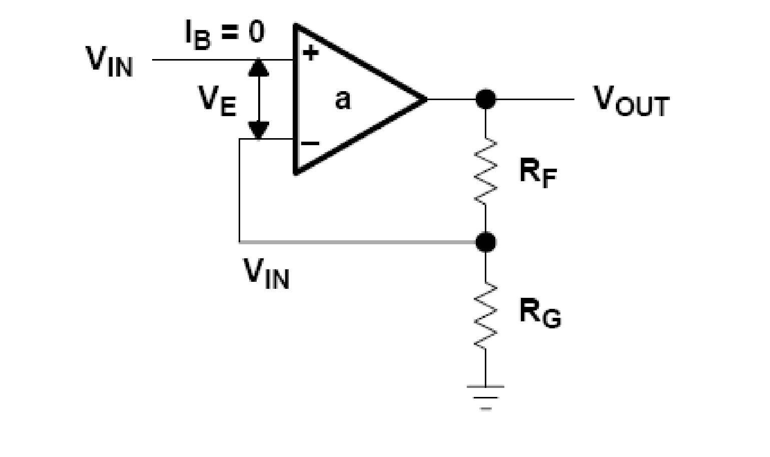
- FPGAs and CPLDs can be connected to embedded processors using either the system bus or a communication interface (SPI e.g.).



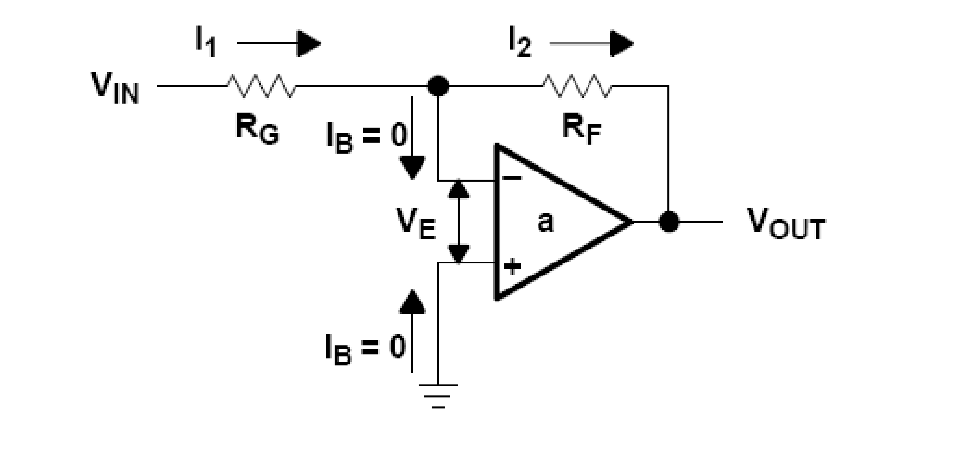
Source:
Xilinx XAPP349

# Analog Electronics

- Non-inverting amplifier: $V_{out} = V_{in}\left(1 + \frac{R_F}{R_G}\right)$



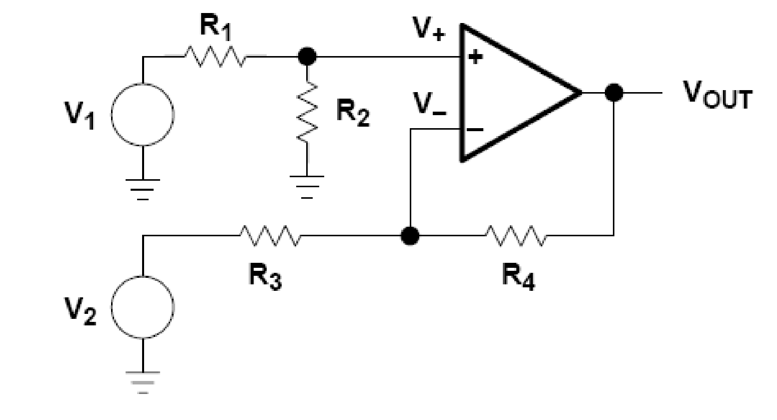
- Inverting amplifier: $V_{out} = -V_{in}\frac{R_F}{R_G}$



Source:
TI, Op Amps for Everyone

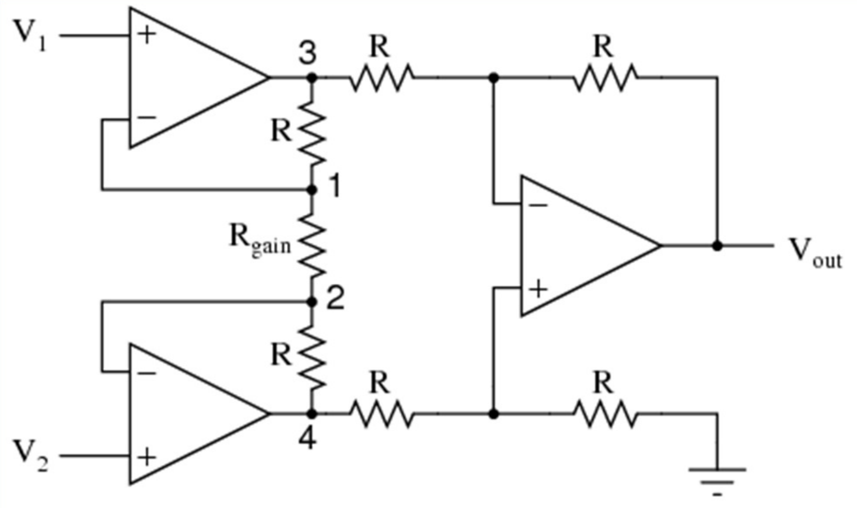
# Analog Electronics II

- Differential amplifier: $V_{out} = (V_1 - V_2)\frac{R_4}{R_3}$



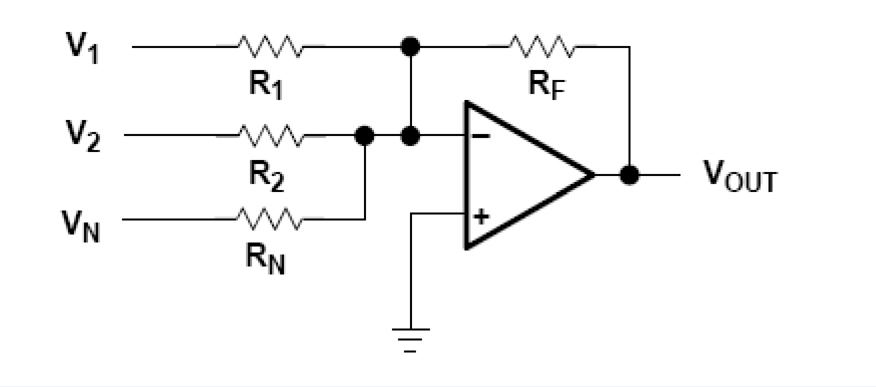
Source:
TI, Op Amps for Everyone

- Instrumentation amplifier: $V_{out} = (V_1 - V_2)\left(1 + \frac{2R}{R_{gain}}\right)$



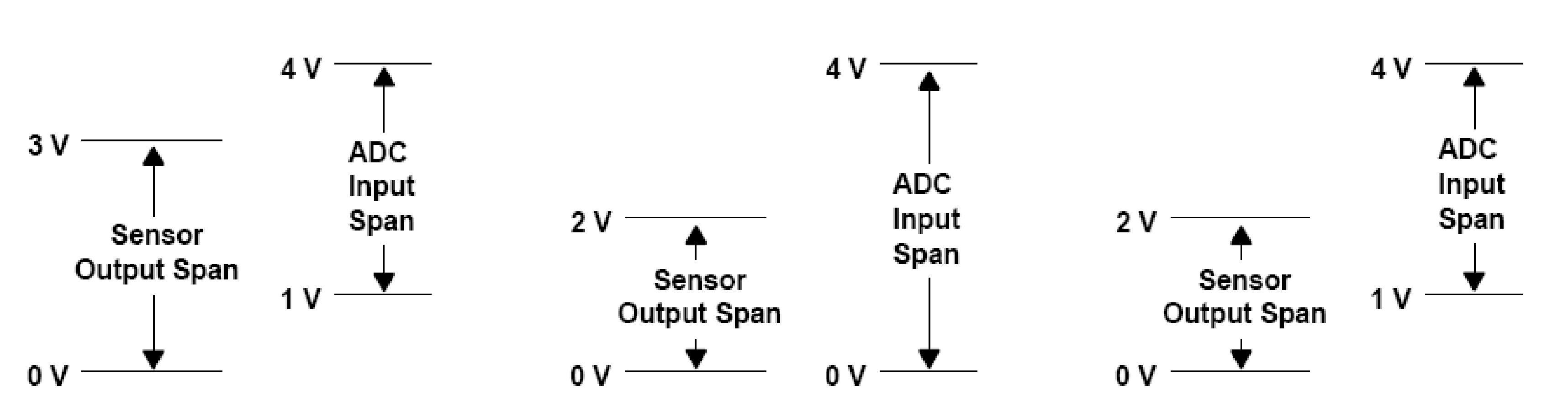
# Analog Electronics III

- Adder: $V_{out} = -\left(\frac{R_F}{R_1}V_1 + \frac{R_F}{R_2}V_2 + \frac{R_F}{R_N}V_N\right)$



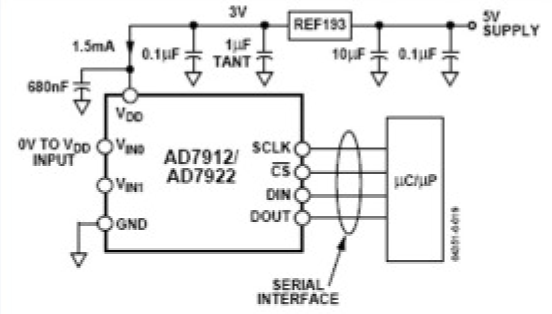
Source:
TI, Op Amps for Everyone

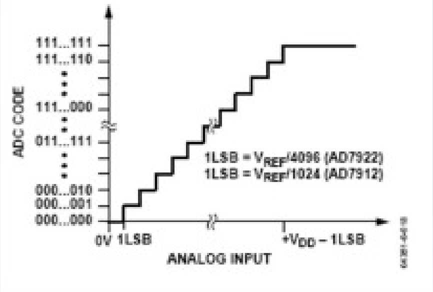
- What is that good for? - Adjust sensor output span to ADC input span (level shift + amplify):



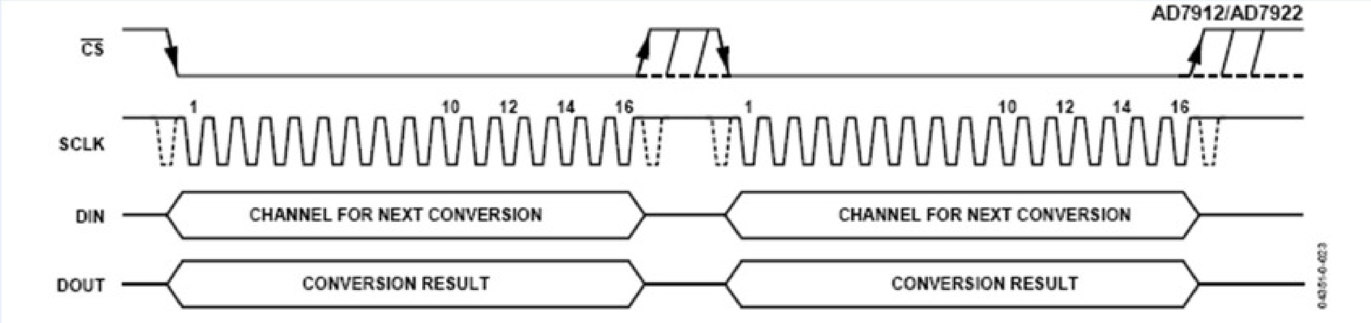
# Mixed-signal Electronics
# - ADC -

- ADCs: analog to digital conversion (Internal to the embedded processor or external)
- Decision criteria (focus on low-bandwidth like monitoring)
  - Resolution: # of output bits (resolution of amplitude)
  - Sampling frequency (resolution in time)
  - #channels: e.g. 8 (sampling of 8 channels simultaneously)
  - Input dynamic range: ratio between the largest and the smallest input
- Example: AD 7192



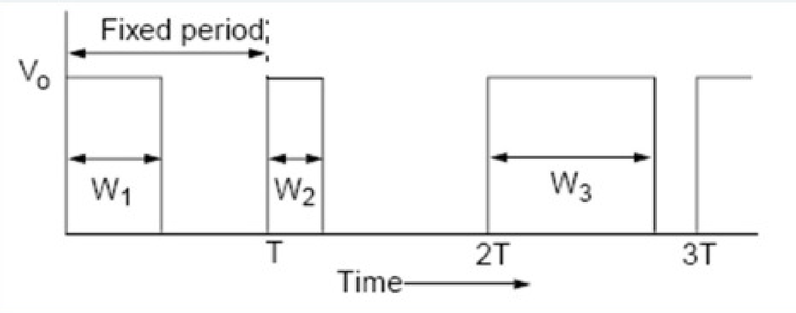


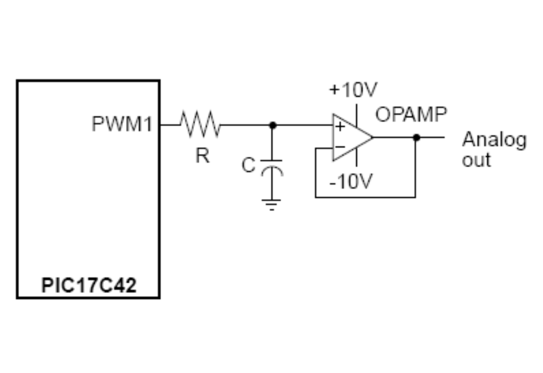
Source:
Analog Devices, AD7192



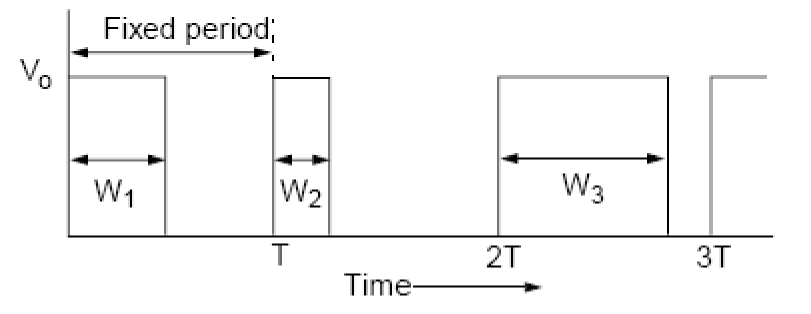
# Mixed-signal Electronics
# - DAC -

- DACs: digital to analog conversion (ICs available)
  - Can be realized by PWM peripheral
- PWM
  - Base frequency fixed, pulse width is variable
  - Pulse width proportional to the amplitude of the unmodulated signal (a digital value provided to the PWM logic)
  - Use external low pass filter (RC) to extract analog information



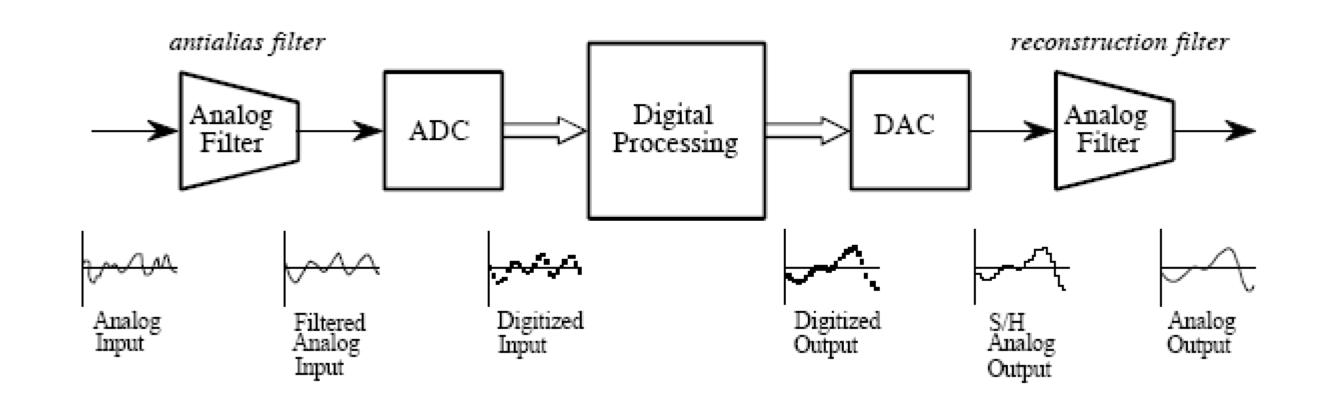


Source:
Microchip, AN538



# Mixed-signal Electronics

- General signal chain:

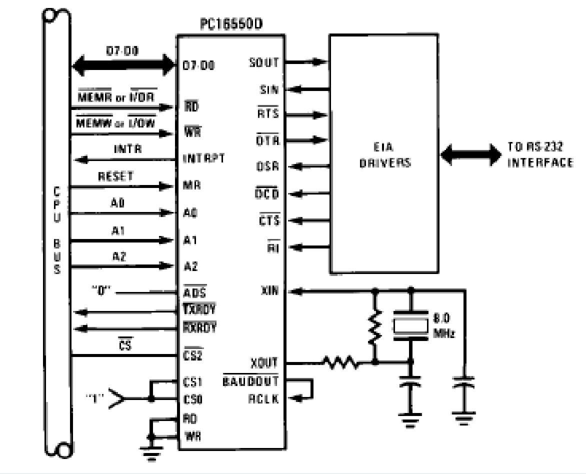


Source:
Smith, dspguide.com

- Analog filter: RC, Sallen Key
- ADC/DAC to processor interface: mostly standard serial for low bandwidth circuits (e.g. I2C, SPI) or integrated into embedded processor

# Communication
# - UART -

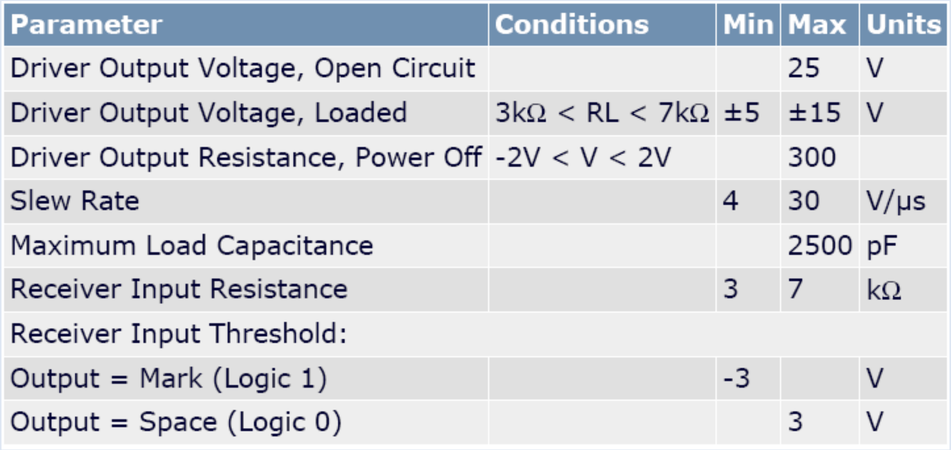
- UART: Universal Asynchronous Receiver/Transmitter
  - Literally any embedded processor comes with built in UART (sometimes we need more and need to connect an additional one)
  - Full-duplex asynchronous protocol which translates data from serial to parallel and vice versa
  - Defines start (synchronization), data, stop, parity bit(s) per frame
  - kbps to Mbps transmission rate, e.g. 9600baud
  - Physical link to other UARTs: TTL (native) , RS232 (single-ended), RS422/485 (differential) for off-board data transmission



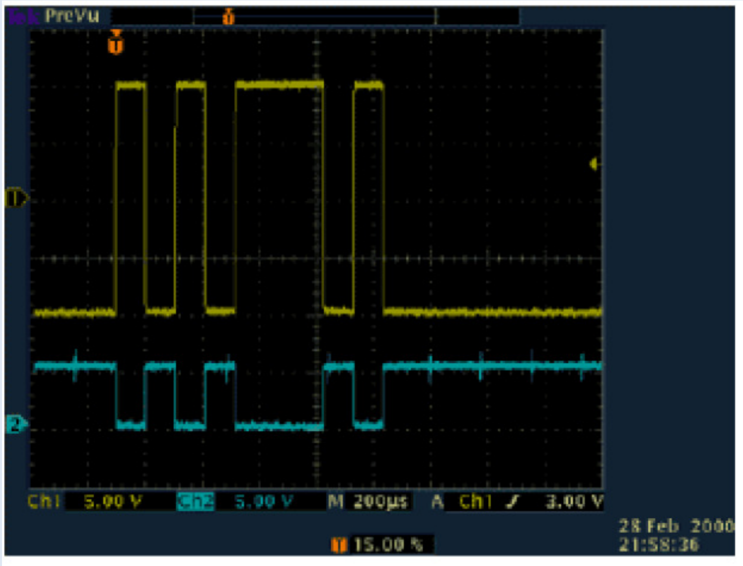
Source:
TI, PC16550D

# Communication
# - RS232 -

- Standard serial interface on older PCs. Now replaced by USB.
- Physical layer single-ended data transmission (usually point-to-point)
- Often used as debug interface in embedded systems, not used as regular data transmission line.
- e.g. MAX232 (Maxim), LTC2801 (LT)



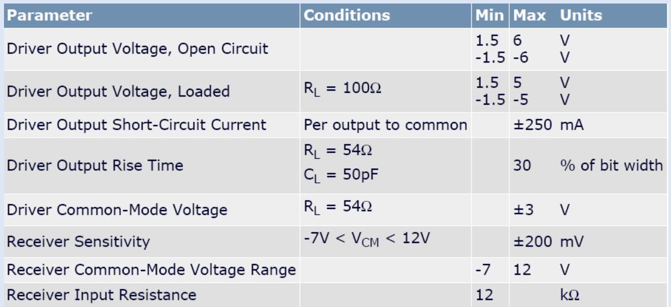
| Parameter | Conditions | Min | Max | Units |
|---|---|---|---|---|
| Driver Output Voltage, Open Circuit | | | 25 | V |
| Driver Output Voltage, Loaded | 3kΩ < RL < 7kΩ | ±5 | ±15 | V |
| Driver Output Resistance, Power Off | -2V < V < 2V | | 300 | |
| Slew Rate | | 4 | 30 | V/µs |
| Maximum Load Capacitance | | | 2500 | pF |
| Receiver Input Resistance | | 3 | 7 | kΩ |
| Receiver Input Threshold: | | | | |
| Output = Mark (Logic 1) | | -3 | | V |
| Output = Space (Logic 0) | | | 3 | V |



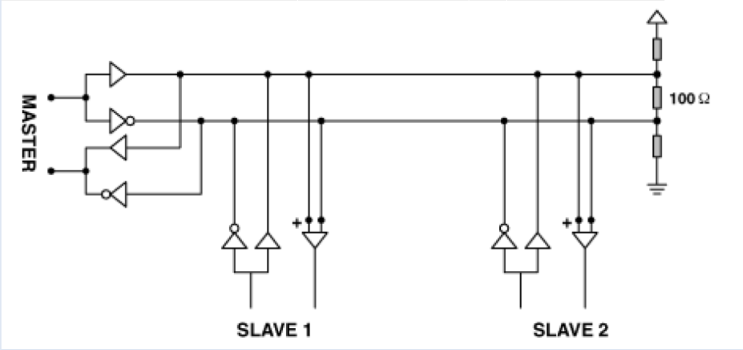
Source:
Maxim, AN723

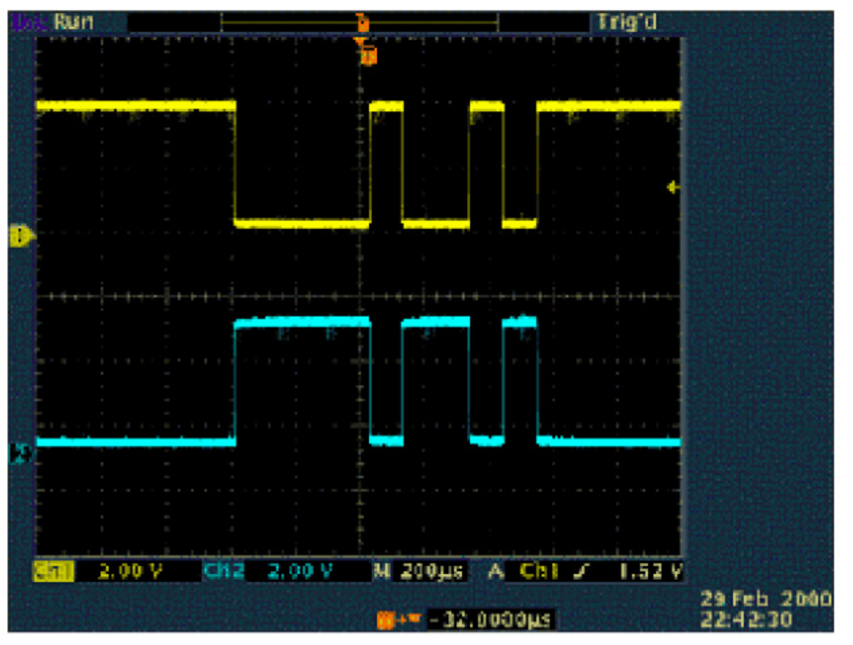
# Communication
# - RS422/485 -

- Physical layer differential data transmission (multi-drop)
- Often used as regular data transmission line since robust and inexpensive.
- RS422 and RS485 differ in increased common mode range and input impedance (RS485).



| Parameter | Conditions | Min | Max | Units |
|---|---|---|---|---|
| Driver Output Voltage, Open Circuit | | 1.5<br>-1.5 | 6<br>-6 | V<br>V |
| Driver Output Voltage, Loaded | $R_L$ = 100Ω | 1.5<br>-1.5 | 5<br>-5 | V<br>V |
| Driver Output Short-Circuit Current | Per output to common | | ±250 | mA |
| Driver Output Rise Time | $R_L$ = 54Ω<br>$C_L$ = 50pF | | 30 | % of bit width |
| Driver Common-Mode Voltage | $R_L$ = 54Ω | | ±3 | V |
| Receiver Sensitivity | -7V < $V_{CM}$ < 12V | | ±200 | mV |
| Receiver Common-Mode Voltage Range | | -7 | 12 | V |
| Receiver Input Resistance | | 12 | | kΩ |

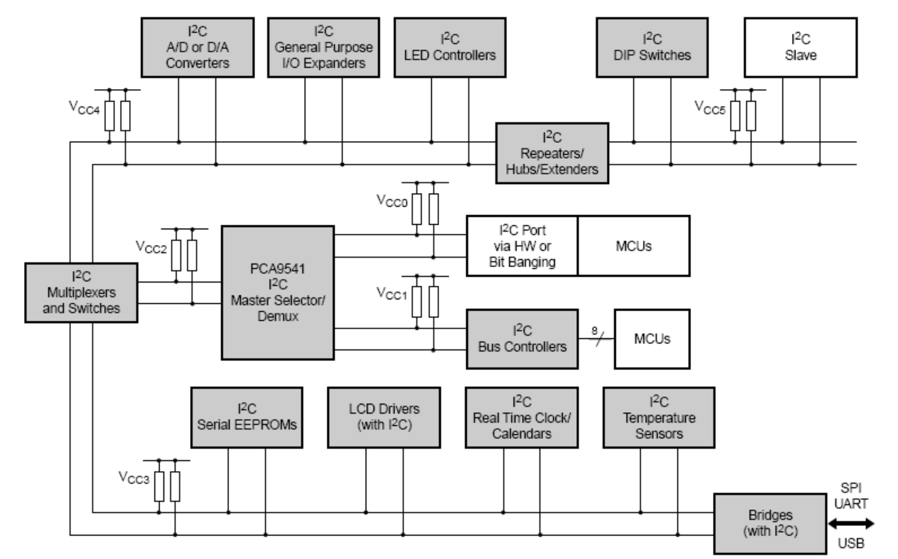


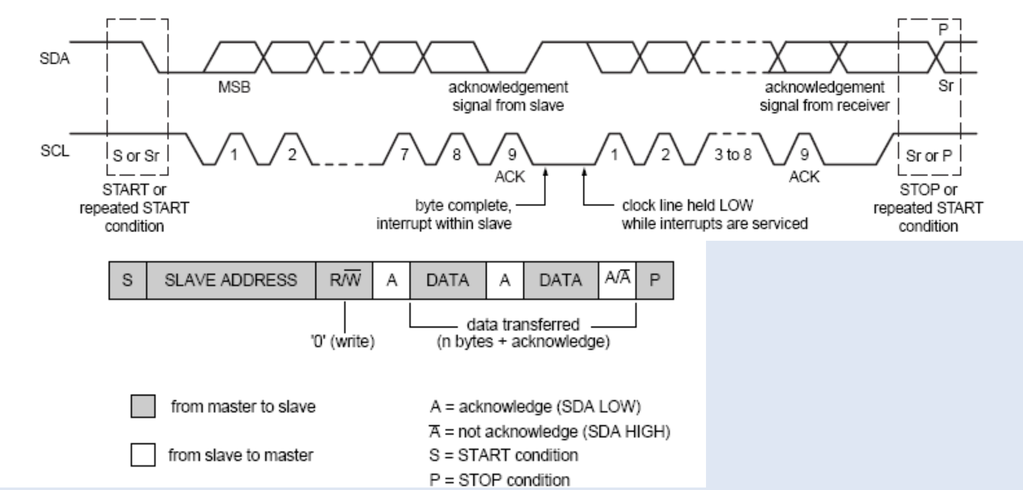


Source:
Maxim, AN723

# Communication
# - I2C -

- I2C: Inter-IC Bus, inexpensive, only two lines (SDA, SCL), master usually in embedded processor
- Each device is addressable by software
- Master/slave protocol, serial 8-bit oriented, multi-master bus, 100kbit/s in standard mode
- Protocols for additional low-speed peripheral connection

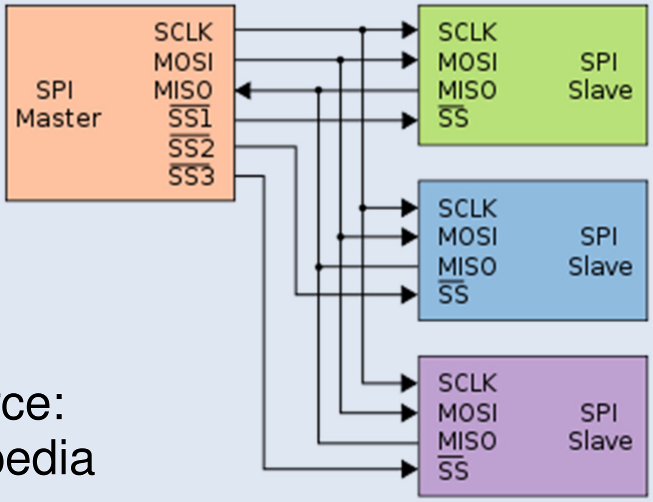

I²C A/D or D/A Converters
I²C General Purpose I/O Expanders
I²C LED Controllers
I²C DIP Switches
I²C Slave
VCC4
VCC5
I²C Repeaters/ Hubs/Extenders
VCC0
VCC2
I²C Port via HW or Bit Banging
MCUs
PCA9541 I²C Master Selector/ Demux
I²C Multiplexers and Switches
VCC1
I²C Bus Controllers
8
MCUs
I²C Serial EEPROMs
LCD Drivers (with I²C)
I²C Real Time Clock/ Calendars
I²C Temperature Sensors
VCC3
Bridges (with I²C)
SPI
UART
USB


SDA
SCL
MSB
acknowledgement signal from slave
acknowledgement signal from receiver
P
Sr
S or Sr
1
2
7
8
9
ACK
1
2
3 to 8
9
ACK
Sr or P
START or repeated START condition
byte complete, interrupt within slave
clock line held LOW while interrupts are serviced
STOP or repeated START condition
S
SLAVE ADDRESS
R/W̄
A
DATA
A
DATA
A/Ā
P
'0' (write)
data transferred (n bytes + acknowledge)
from master to slave
from slave to master
A = acknowledge (SDA LOW)
Ā = not acknowledge (SDA HIGH)
S = START condition
P = STOP condition

Source:
NXP, UM10204

# Communication
# - SPI -

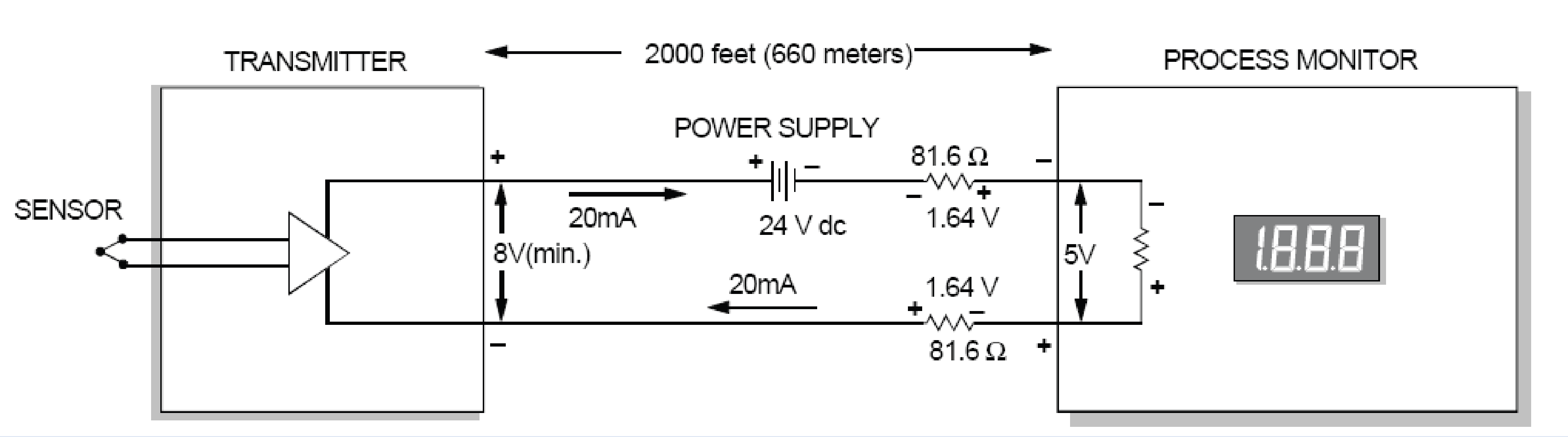
- SPI: Serial Peripheral Interface, common peripheral in embedded processors
- Used to move streams of data, bit rates in the MHz range
- Synchronous, master-slave
- Protocol for high-speed devices (Ethernet, ADC/DAC, …)
- As data is clocked out, new data is clocked in (data exchange)
- Clock (SCLK), Slave Select (SS)
- Master-out-slave-in (MOSI)
- Master-in-slave-out (MISO)



Source: wikipedia

# Communication
# - Current Loop (long distance)-

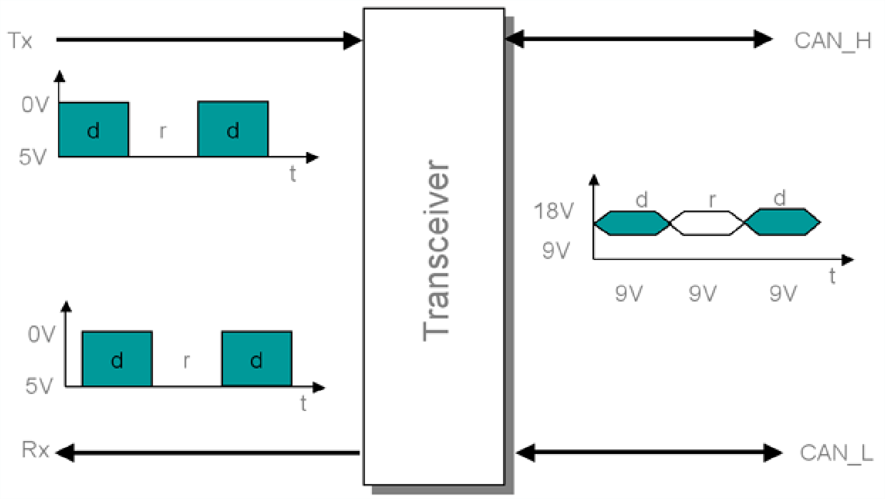
- 4-20 mA interface
- Robust interface for sensor data transmission
- A sensor converts its reading into a current (4 mA being the zero level and 20 mA the FS of the sensor e.g.)
- A receiver converts the current into a voltage for further usage
- Advantage: loop voltage drops (line resistance) can be compensated, less sensitive to noise



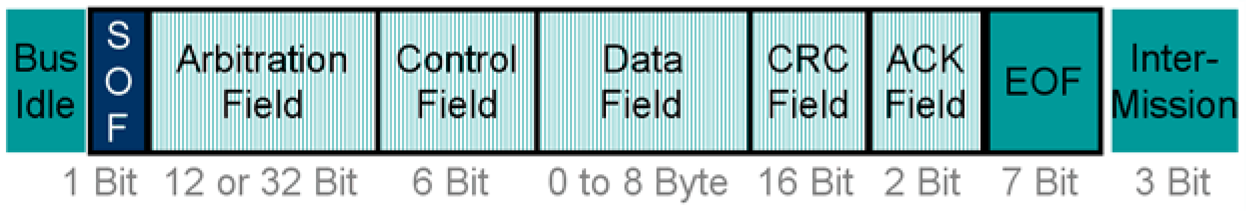
Source: Murata, DMS-AN-20

# Communication
# - CAN -

- CAN: Controller Area Network, ISO 11898 (PHY, DLL)
- Protocol controller available as peripheral of embedded processors, line driver external (creates differential signals, adds protection circuits)
- Serial protocol, up to 1 Mbit/s
- Bit-wise arbitration
- Error detection



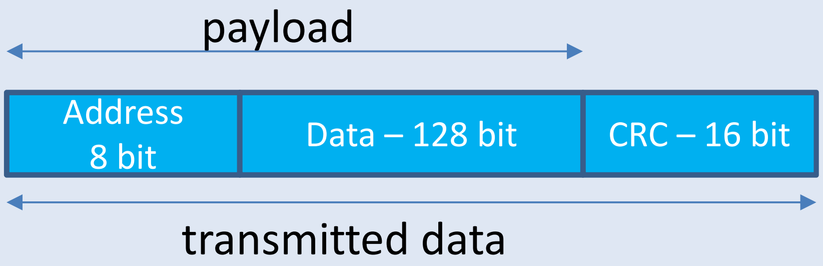
Source: Softing

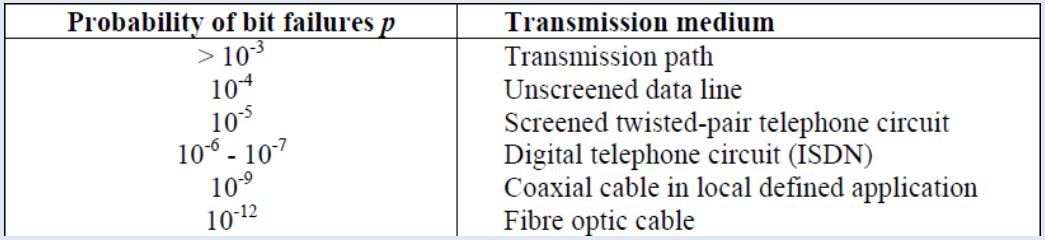


| Bus Idle | SOF | Arbitration Field | Control Field | Data Field | CRC Field | ACK Field | EOF | Inter-Mission |
|---|---|---|---|---|---|---|---|---|
| | 1 Bit | 12 or 32 Bit | 6 Bit | 0 to 8 Byte | 16 Bit | 2 Bit | 7 Bit | 3 Bit |

# Communication
# - error detection -

- We usually use standard protocols to transmit data. Correctness is guaranteed by by error detection mechanisms (e.g. parity, CRC)
- Sometimes error detection capability not sufficient
  - Hamming distance of n: n-1 bit errors can be detected.
  - Residual error: If we do know the Hamming distance and do know the bit error rate (bit flips are statistically independent) we can calculate a residual error.
  - CRC: an additional peace of data is added to the existing bit stream. The additional peace of data allows error detection





| Probability of bit failures *p* | Transmission medium |
|---|---|
| $> 10^{-3}$ | Transmission path |
| $10^{-4}$ | Unscreened data line |
| $10^{-5}$ | Screened twisted-pair telephone circuit |
| $10^{-6} - 10^{-7}$ | Digital telephone circuit (ISDN) |
| $10^{-9}$ | Coaxial cable in local defined application |
| $10^{-12}$ | Fibre optic cable |

Source: Börcsök, HIMA

# Questions?