Savitribai Phule Pune University
Final Year of E & Tc Engineering (2019 Course) EPD (Elective-II) 404185

Unit II: Circuit Design & Testing Methods

Agenda

- Teaching and Exam Scheme
- Prerequisite Courses, if any
- Course Objectives
- Course Outcomes
  - From Symbols To Substance
  - Convert Requirements Into Design
  - Reliability
  - Fault Tolerance
  - High Speed Design
  - Low Power Design,
  - Noise And Error Budget
  - Standard Data Buses And Networks
  - Reset And Power Failure Detection
  - Interface: Inputs and Outputs,
  - Breadboards, Evaluation Boards and Prototypes.

Teaching and Exam Scheme:

- Teaching Scheme: Theory: 03 hrs. / week
- Credit : 03
- Examination Scheme
  - In-Sem (Theory): 30 Marks
  - End Sem (Theory): 70 Marks

Prerequisite Courses, if any:

1. Digital Logic Design
2. Electronic Components and Hardware
3. Basics of Programming languages like C Language, JAVA.

Course Objectives:

During the course study students will be able to

- To understand the **stages of product (hardware/ software) design and development.**
- To learn the **different considerations** of analog, digital and mixed circuit design.
- To be acquainted with **methods of PCB design and different tools** used for PCB Design.
- To understand the importance of **testing in product design cycle.**
- To understand the **processes and importance** of documentation.

Course Outcomes:

On completion of the course, learner will be able to -

CO1: Understand and explain design flow of design of electronics product

CO2: Associate with various circuit design issues and testing

CO3: Inferring different software designing aspects and the Importance of product test & test specifications.

CO4: Summarizing printed circuit boards and different parameters

CO5: Estimating assorted product design aspects

CO6: Exemplifying special design considerations and importance of documentation

Unit II Circuit Design & Testing Methods

1. From Symbols To Substance
2. Convert Requirements Into Design
3. Reliability
4. Fault Tolerance
5. High Speed Design
6. Low Power Design,
7. Noise And Error Budget
8. Standard Data Buses And Networks
9. Reset And Power Failure Detection
10. Interface: Inputs and Outputs,
11. Breadboards, Evaluation Boards and Prototypes.

1. From Symbols To Substance

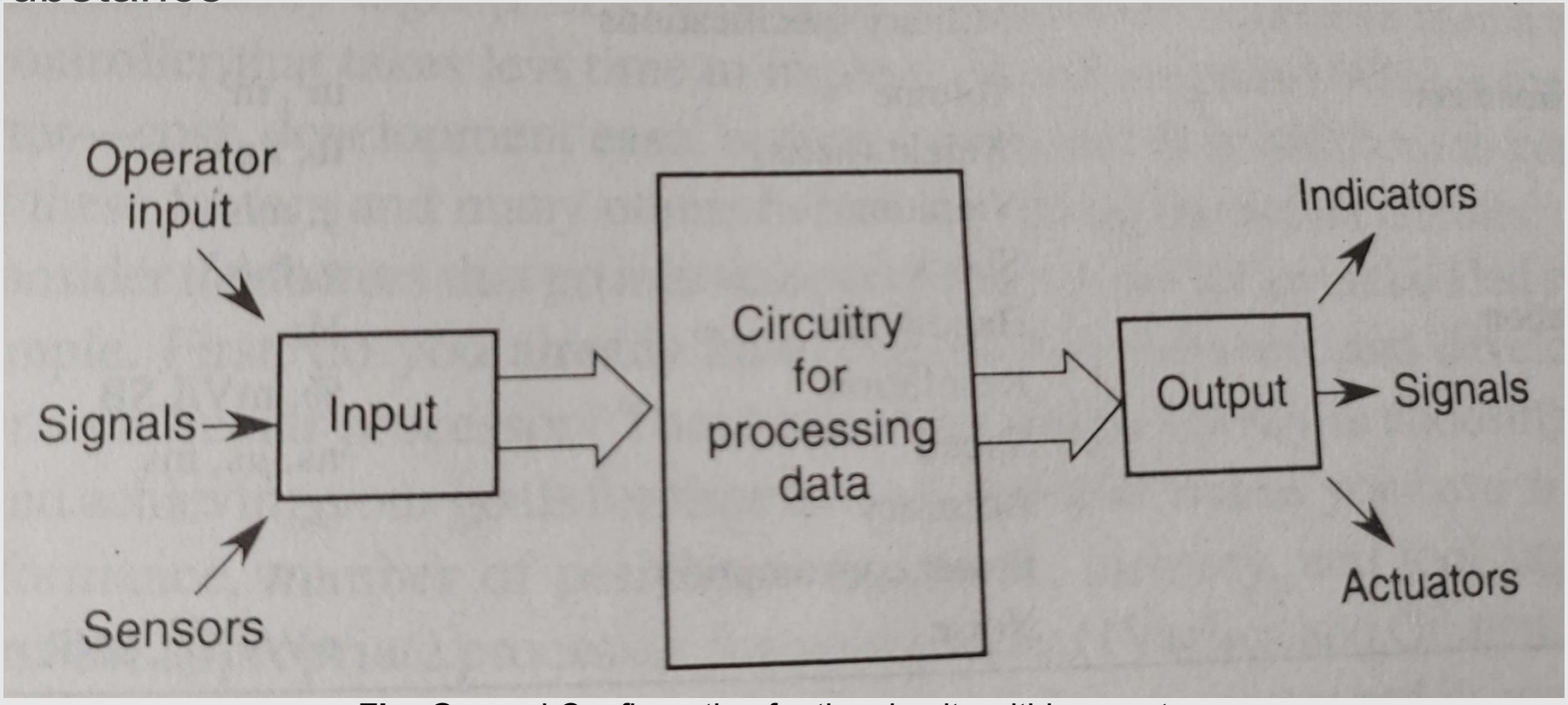


Fig: General Configuration for the circuits within a system.

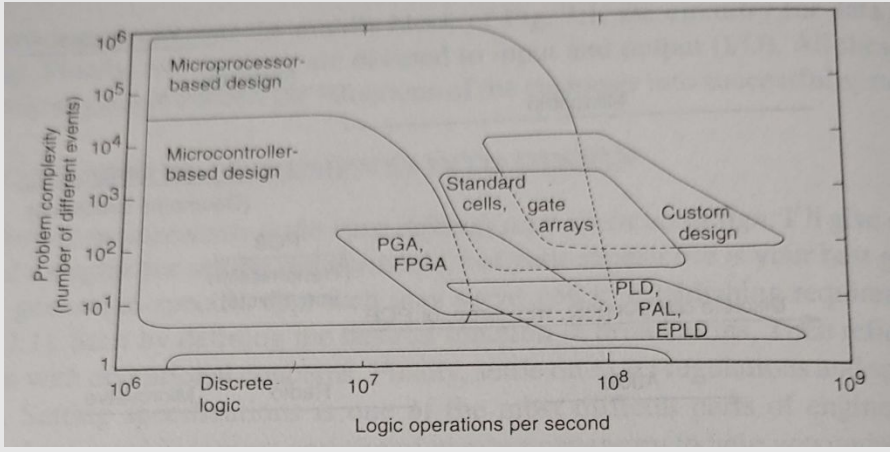
Convert Requirements Into Design

System concern	Requirement	Parameters
Function	Response times	s, min, hr
	Data rates	Mbytes/s, kbits/s
	I/O drive	A, V
	Reliability—MTBF	hr
Regulations	FCC	
	UL	
	Military specifications	
Environment	Volume	in^3, m^3
	Weight (mass)	lb, kg
	Vibration	g, m/s^2
	Shock	g, m/s^2
Operation	Bandwidth	Hz
	Resolution	%, mV/LSB
	Speed	ns, μs, ms
	Accuracy	%
	Power consumption	mW
	Noise	$nV/\sqrt{Hz}$, SNR

Fig: Design Driving requirements

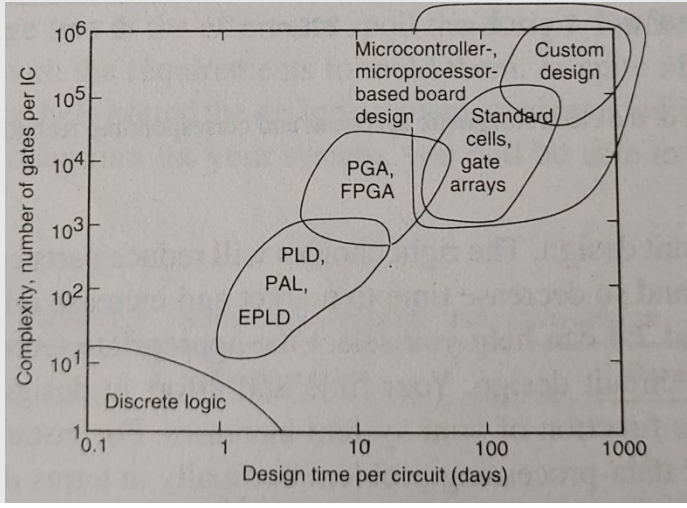
Convert Requirements Into Design

- Throughput
- Resolution and dynamic range
- Address space and available memory
- Language choice: code size, speed (compilation and actual execution)
- Predominant types of calculations: integer or floating point



Peripheral functions also drive the selection of a processor. These functions include the following:

- Math coprocessors
- Graphics accelerators
- Interrupt handlers
- Data transfer and communications: DMA, small computer system interface (SCSI), serial I/O ports
- Timers
- Analog-to-digital converters (ADCs)
- Digital-to-analog converters (DACs)
- Power drivers
- Watchdog timing



Convert Requirements Into Design

On the other hand, the architecture of microprocessors does not include memory or peripherals, so you will have to include these components in your design, thereby increasing the parts count and complexity. You can estimate the amount of memory you need by adding up the possible sizes of the following:

1. Data Arrays
2. Stack
3. Temporary and permanent variables
4. Compiler overhead
5. I/O buffers

Finally, and probably most important,
what experience do you have, and what tools are available to support your development?
Do hardware emulators exist that will help you debug both circuits and code?
What software tools are available?
Do they provide good support?
Does the vendor have a good reputation?

Development tools that target specific microcontrollers or microprocessors can markedly affect either for good or bad, you design effort.

RELIABILITY

1. **Complexity:** Fewer parts are almost always better. Design margin: You must allow for stressing of components.
2. **Design Margin:** You must allow for stressing of components

$$\lambda = \lambda_b \pi_e \pi_q \pi_a$$

λ = failure rate of component
λ_b =base failure rate
π_e = environmental factor
π_q = quality factor
π_a = acceleration factor

Reliability of a component is defined as a function of failure rate:

$$R(t) = e^{-\lambda t}$$

$R(t)$= reliability
λ= failure rate
t= time

Reliability of a system is the product of all component reliabilities:

$$R_{system} = \prod_{i=1}^{n} R_i$$

R_{System} = reliability of the system
R_i = reliability of component i

RELIABILITY

We must consider the application and how some of these stresses and susceptibility factors might affect reliability:

- Corrosion
- Thermal cracks
- Electromigration
- Secondary diffusion
- Ionizing radiation
- Vibration
- High-voltage breakdown
- Aging

Fault Tolerance

Fault tolerance goes beyond the design and analysis for reliable operation and reduces the possibility of dysfunction or damage from abnormal stresses and failures. It allows a measure of continued operation in the event of a problem. Fault tolerance is primarily a philosophy of system design and architecture.

It has three distinct considerations:

a. **Careful Design**

b. **Testable Architecture**

c. **Redundant Architectures**

Fault Tolerance

Careful Design

We can avoid many failures from abnormal stresses by conservative and careful design. Here are some design techniques that can reduce the probability of failure

1. **Reduce overstress from heat** with cooling and lower-dissipation design.
2. Use **opto-isolation or transformer coupling** to stop overvoltage and leakage current.
3. Implement **ESD protection.**
4. **Mount for shock** (from accidental drops) and **vibration.**
5. **Tie down** wires and cables that flex frequently and use **strain relief.**
6. **Prevent incorrect hook-up**; use keyed connectors.

All these techniques fortify a design, but blind application of them leads to costly "gold-plated" or overly conservative design. Analyse the function of $y_S o_h u_{as} r_{ha} p_{nk} ro_{D.} d_B u_{ir} c_{ad} t_{ar} a_{,\,V} n_P d_{KB} i_I t_E s_{T.}$ use before picking the appropriate techniques to reduce the danger of abnormal stresses in the application.

Fault Tolerance

Testable Architecture

A testable architecture has two possible configurations.

- The simpler configuration provides probe points for a technician or instrument to stimulate circuits and record responses. Usually, trained personnel must disassemble the system and remove the circuit for testing.

- The more complex configuration has dedicated internal circuitry called built-in test (BIT) that tests the system and diagnoses problems without disassembly of the equipment.

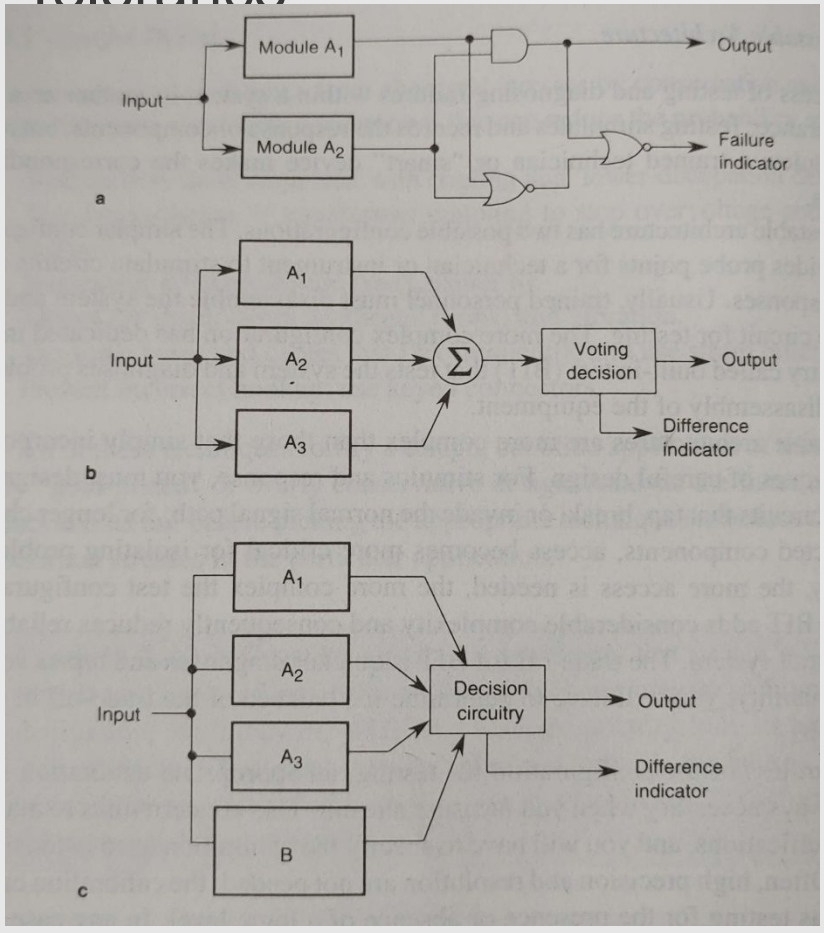
Appropriate Calibration

Fault Tolerance

Redundant Architectures

- Most complex and fault-tolerant architectures
- Use multiple copies of circuitry and software to self-check between functions
- Justified only when downtime for repair and maintenance cannot be tolerated
    - A doubly redundant architecture -- indicates a failure in one of the sub systems; this allows for quick repair.
    - A triply redundant architecture -- uses voting between the outputs of three identical modules to select the correct value. It can have a failure and still operate correctly.
    - Dissimilar redundance architecture -- compares the output from modules with different software and hardware to select the correct output.

Fault Tolerance



a. A doubly redundant architecture

b. A triply redundant architecture

c. Dissimilar redundance architecture

High Speed Design

- **Harmonics Generated – Factor of Concern**

Two conservative criteria may be used to estimate when transmission line effects begin:

1. **circuit dimensions versus signal wavelength**
2. **rise time versus propagation delay.**

1. $l > \lambda/20$

l = length of the signal path
λ = Max. wavelength of the signal

2. $t_r = 4t_p$

t_r = rise time of the signal
t_p = propagation delay of the signal path

Careful Design : with

- transmission lines,
- including bandwidth limitation,
- decoupling,
- ground bounce,
- crosstalk,
- impedance mismatch,
- and timing skew or delay.

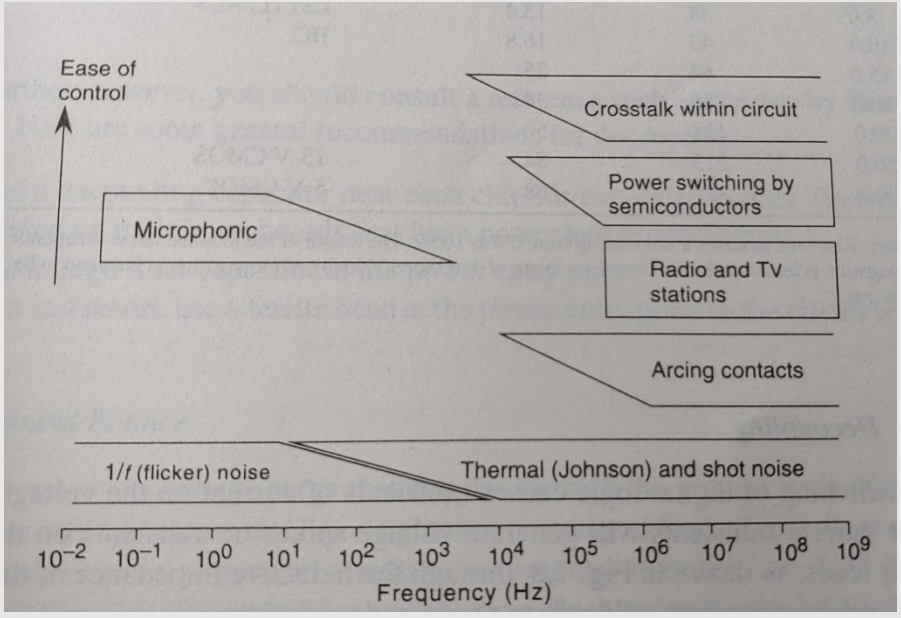


Fig: some sources of noise and the ease of remedies.

High Speed Design : Considerations

1. **Bandwidth**
2. **Decoupling**
3. **Ground Bounce**
4. **Crosstalk**
5. **Impedance Matching**
6. **Timing**

Guidelines	Reduces			
	Signal reflections	Power-distribution noise	Ground bounce	Crosstalk
Control the rise time of signal pulses by selecting the appropriate logic family and technology.	√		√	√
Reduce signal frequency.	√			√
Use decoupling capacitors.		√		
Use multilayer printed circuit boards with power and ground planes.	√	√	√	√
Terminate transmission lines where $t_r/t_p < 4$.	√			
Keep stubs short.	√			
Keep signal lines short and perpendicular on adjacent layers. Control spacing between traces.	√			√
Group circuits into separate areas: analog circuits, high-speed logic, and high-current switching.		√		√
Avoid parallel asynchronous lines, wire wrap, and sockets.			√	√

Low – Power Design

$$P = fCV^2$$

P = Power
f = Frequency
C = load capacitance
V = DC Supply Voltage

Guidelines in design to minimize power:

- Lower clock frequency.
- Lower supply voltage to digital logic.
- Shut down unused circuits.
- Put controller into sleep mode when not needed.
- Terminate all unused inputs. Don't allow any to float.
- Avoid slow signal transitions.

NOISE AND ERROR BUDGETS

Three types of errors occur in electronics.
Two of them are variations in parameters due to **production and environment**, the third is **noise** that is described statistically.
Anytime you design a system, you should know these fundamental limits.

$$V_{noise(rms)} = \sqrt{4kTRW}$$

$$I_{noise(rms)} = \sqrt{2qI_{DC}W}$$

$$V_{noise(rms)} = V_i[0.392 + \log_{10}(\frac{f_{high}}{f_{low}})]$$

$$V_{total} = \sqrt{V_1^2 + V_2^2 + \cdots + V_n^2}$$

$$SNR = 10\log_{10}(V_{signal}^2 / V_{noise}^2)$$

Section	Component	Description of possible errors
Input	Sensor	Inherent signal noise
	Transmission line	Interference, crosstalk
	Amplifier	Temperature dependence, linearity, CMRR, SNR
	Filter	Variations in component values, bandwidth, pass-band and stop-band ripple, phase linearity
Conversion and processing	Sample-and-hold	Aperture jitter, droop
	Analog-to-digital conversion	Quantization error, differential nonlinearity, integral nonlinearity, aliasing
	Digital filtering	Round-off error in calculations
Output	Digital-to-analog conversion	Quantization error, differential nonlinearity, integral nonlinearity, aliasing
	Level shifting amplifier	Temperature dependence, linearity, CMRR, SNR
	Filter	Variations in component values, bandwidth, pass-band and stop-band ripple, phase linearity

Standard Data Buses And Networks

a. Bus Architecture Concerns

b. Serial Communications

c. Instrumentation and I/O Buses

d. Back-Plane Buses

e. Local Area Networks

Standard Data Buses And Networks

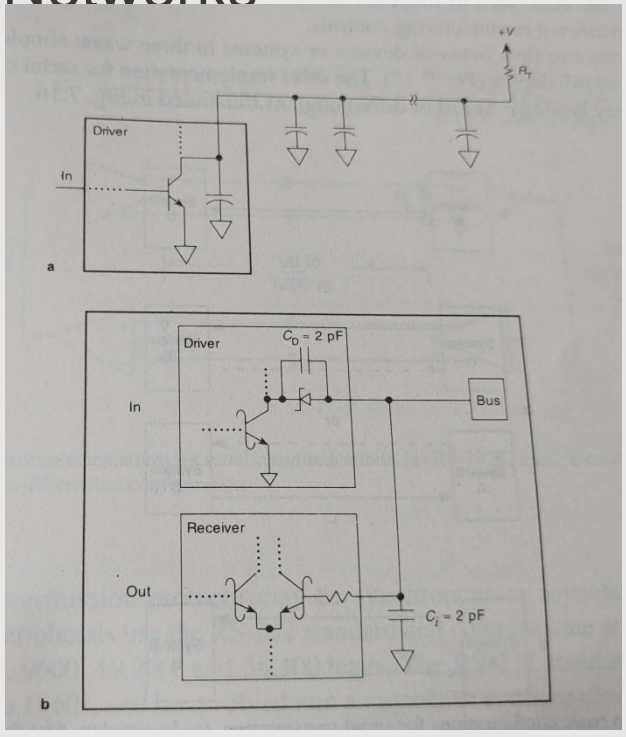


Fig: Bus Driving Circuits

a. TTL Driving Capacitive load on Bus
b. Back-Plane transceiver logic interface circuit isolate cap. Load from the bus

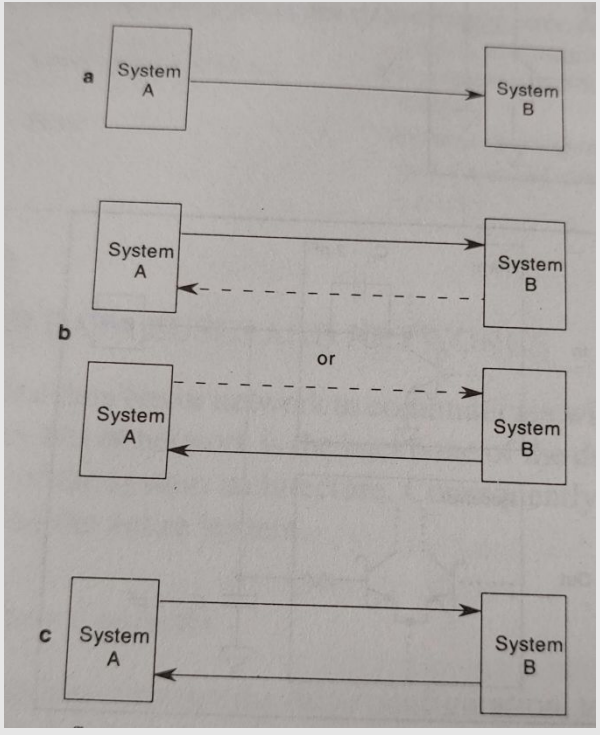


Fig: 3 basic config. for serial Tx

a. Simplex
b. Half duplex
c. Full duplex

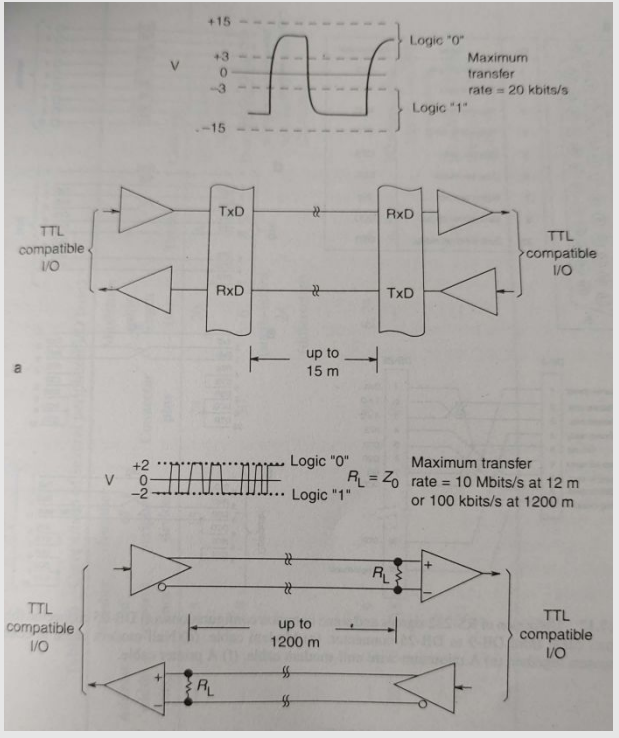


Fig: 2 implementations for serial Tx

a. RS-232C single ended config.
b. RS-422A differential config.

Standard Data Buses And Networks

Comparison of several peripheral I/O buses

Bus	Maximum data transfer (Mbytes/s)	Data width (bits)	Address width (bits)	Number of control lines	Number of peripheral devices	Connector pins	Maximum cable length (m)	Timing	Control
IEEE-488	1	8	—	8	15	24	20	A	Instrumentation networks
SCSI	2 (asynchronous) 5 (synchronous)	8 + parity	—	9	7	50	6 (single-ended) 24 (differential)	A, S	Primary to support computer peripherals like disk drives and CD-ROMs
SCSI-2	10, 40	16, 32	—	9	7	50	6, 24	A, S	
Sbus	100	32	32	18	—	96	—	A, S	I/O coprocessing, supports only three circuit boards
Turbochannel	100	32	32	11	—	96	—	S	I/O address = 27 bits, board size = 11.6 × 14.4 × 3.3 cm.

Note: A = asynchronous; S = synchronous.

Fig: Comparison of several peripheral I/O Buses

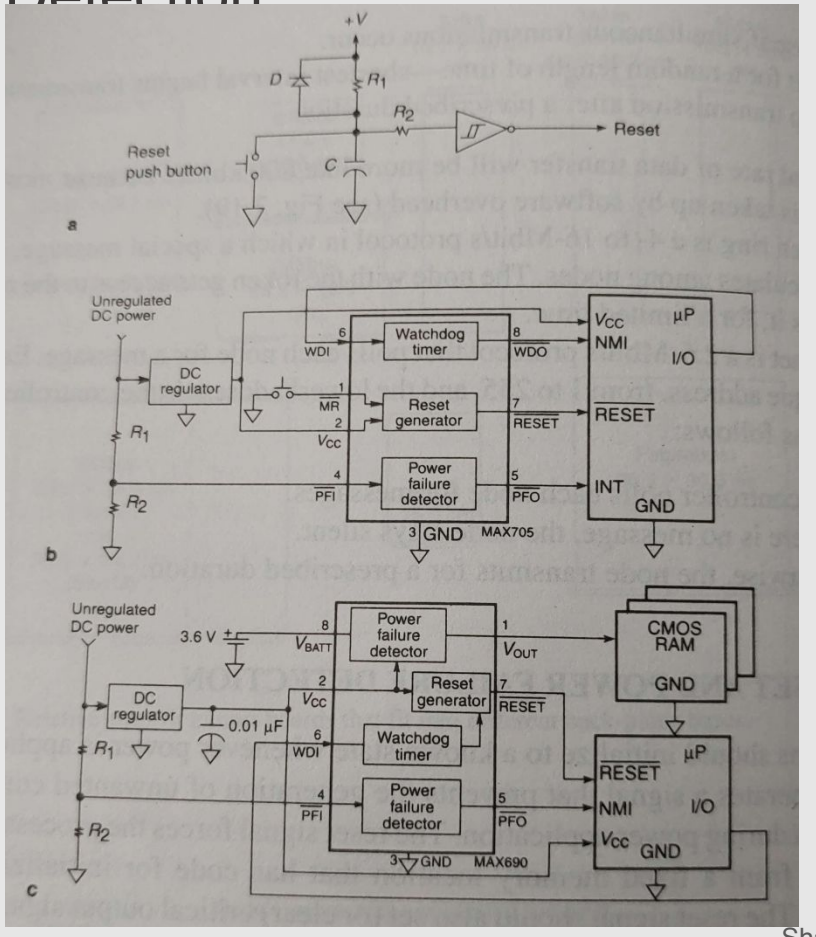
Standard Data Buses And Networks

Back-plane bus	Maximum data transfer (Mbytes/s)	Data width (bits)	Address width (bits)	Multiplexed data/ address	Timing	Drive type	Connector type	Connector pins	Interrupts	Physical dimensions W × L (cm)	Comments
STD 32	20	8, 16, 32	16, 32	—	A	TTL	CE	108	1	11.4 × 16.5	Wide industrial base; robotics process control, data acquisition
G-64, G-96+	2, 40	8, 16, 32	24	—	A	TTL	DIN	96	6	9.9 × 16.0	Single Eurocard
VMEbus	40	16, 32	16, 32	√	A	TTL	DIN	96	7	23.3 × 16.0	Single or double Eurocard instrumentation computing
VXIbus	1000	8, 16, 32	16, 24, 32	—	A	TTL	DIN	192	7	23.3 × 34.0 or 36.6 × 34.0	Superset of VME instrumentation
Multibus II	40	16, 32	32	√	S	TTL	DIN	96	1	21.6 × 25.7	Instrumentation
Nubus	40	32	32	√	S	TTL	DIN	96	1	10.2 × 30.0	One interrupt per slot
EISA	33	8, 16, 32	16, 24	—	S	TTL	CE	198	11	10.7 × 34.3	PCs
Micro Channel	17	8, 16, 32	16, 24, 32	—	A	TTL	CE	198	11	7.9 × 29.2	IBM PS/2
Futurebus+	400	32, 64	32, 64	√	A	BTL	NA	192	NA	26.5 × 30.0	

Note: NA = not available.

Fig: Comparison of several back - plane bus configs.

Reset And Power Failure Detection



a. A simple RESET circuit

b. MAX705
(debounced manual reset and watchdog timer)

c. MAX690
(battery backup and watchdog timer)

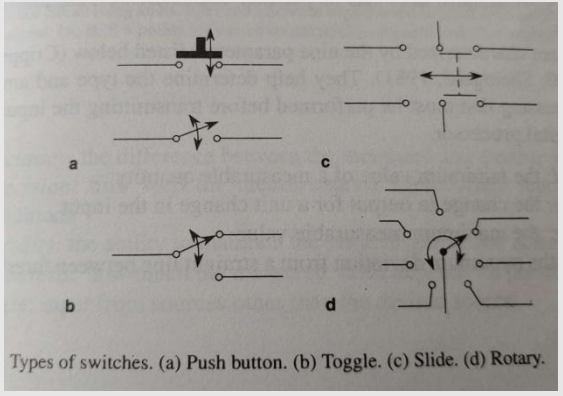
Interface: Inputs

a. Switches

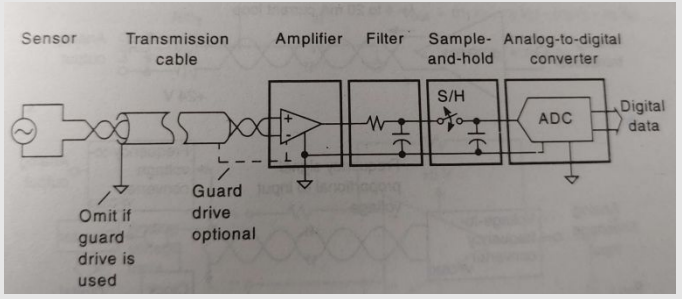
b. Sensors

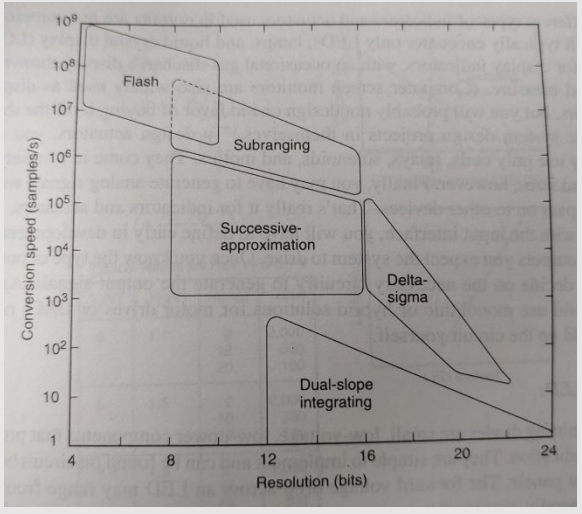
c. Analog Pre-processing

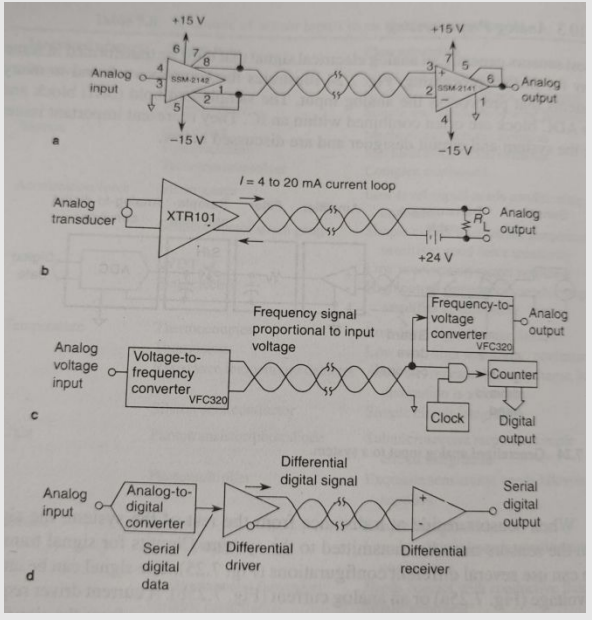
d. ADCs



Types of switches. (a) Push button. (b) Toggle. (c) Slide. (d) Rotary.







Interface: Outputs

a. LEDs

b. LCDs

c. Lamps

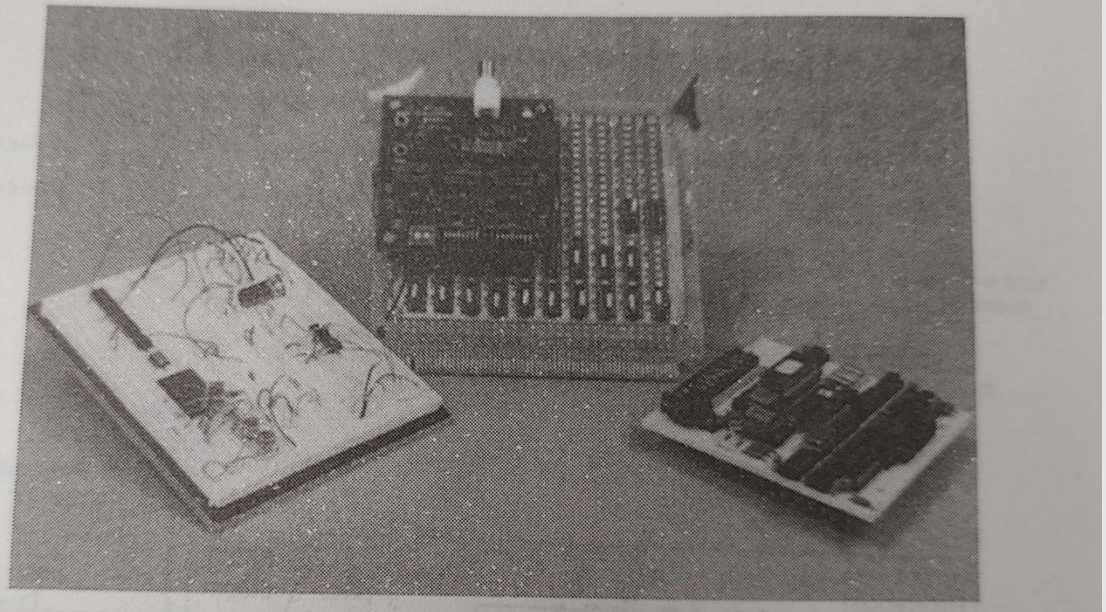
d. Relays and Solenoids

e. Motors

f. DACs

g. Analog Drive

Breadboards, Evaluation Boards and Prototypes



Examples of a breadboard, an evaluation board, and a wire-wrapped prototype board.

Resources

Textbooks:

1. Kim Fowler, “Electronic Instrument Design”, Oxford university press.
2. Robert J. Herrick, “Printed Circuit board design Techniques for EMC Compliance”, Second edition, IEEE press.

Reference Books:

1. James K. Peckol, “Embedded Systems – A Contemporary Design Tool”, Wiley publication
2. J C Whitakar, “The Electronics Handbook”, CRC press.

Thank You