CENG 314
Embedded Computer Systems

Lecture 1

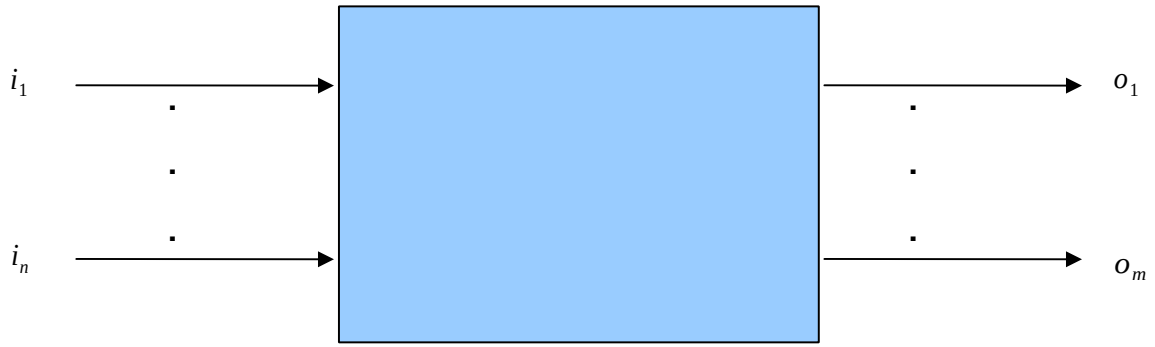
Introduction to Embedded Computer Systems

Asst. Prof. Tolga Ayav, Ph.D.

Department of Computer Engineering
İzmir Institute of Technology

System

A system has a set of one or more inputs entering a black box and a set of one or more outputs exiting the black box.



System with *n* inputs and *m* outputs.

More formally:

$$Let\, i_1 \square I_1 \ldots i_n \square I_n, o_1 \square O_1 \ldots o_m \square O_m.$$

Then the system is a cross product of $I_1 \ldots I_n \square O_1 \ldots O_n$

$$S \square I_1 \times \ldots \times I_n \times O_1 \times \ldots \times O_n$$

$$I_1 \times \ldots \times I_n \, are\, called\, input\, space. O_1 \times \ldots \times O_n \, are\, called\, output\, space.$$

Deterministic System

Definition.

A system is said to be deterministic if for each possible state, and each set of inputs, a unique set of outputs, response times and next state of the system can be determined.

Event determinism.

Next states and outputs of the system are known for each set of inputs which trigger events.

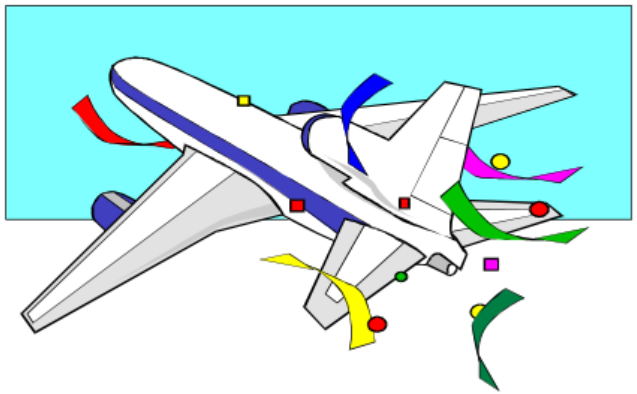
Temporal determinism.

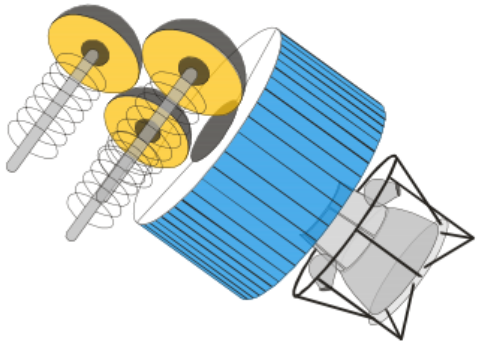
The response time of each set of outputs is known.

Embedded System

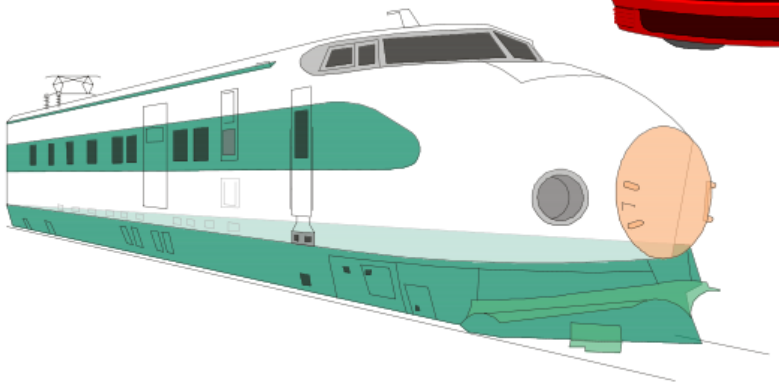
- Definition of the term “Embedded System” is not exact. Here are a few definitions:
- An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, usually with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts.
- A specialized computer system that is part of a larger system or machine. Typically, an embedded system is housed on a single microprocessor board with the programs stored in ROM.
- Computing systems embedded within electronic devices. Nearly any computing system other than a desktop computer

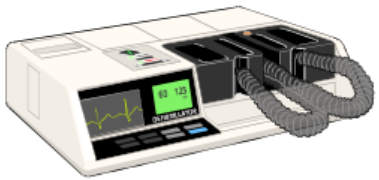
Embedded Systems





WHAT IS AN EMBEDDED SYSTEM?





Less than %1 percent of microprocessors are used in personal computers !

Typical Embedded Systems

Anti-lock brakes
Auto-focus cameras
Automatic teller machines
Automatic toll systems
Automatic transmission
Avionic systems
Battery chargers
Camcorders
Cell phones
Cell-phone base stations
Cordless phones
Cruise control
Curbside check-in systems
Digital cameras
Disk drives
Electronic card readers
Electronic instruments
Electronic toys/games
Factory control
Fax machines
Fingerprint identifiers
Home security systems
Life-support systems
Medical testing systems
Modems
MPEG decoders
Network cards
Network switches/routers
On-board navigation
Pagers
Photocopiers
Point-of-sale systems
Portable video games
Printers
Satellite phones
Scanners
Smart ovens/dishwashers
Speech recognizers
Stereo systems
Teleconferencing systems
Televisions
Temperature controllers
Theft tracking systems
TV set-top boxes
VCR's, DVD players
Video game consoles
Video phones
Washers and dryers

Embedded System Constraints

- **Small Size, Low Weight**
  - Hand-held electronics, Transportation applications (weight costs money)
- **Low Power**
  - Battery power for 8+ hours (laptops often last only 2 hours), limited cooling may limit power even if AC power available
- **Harsh environment**
  - Heat, vibration, shock, Power fluctuations, RF interference, lightning, Water, corrosion, physical abuse
- **Safety-critical operation**
  - Must function correctly, must not function incorrectly
- **Extreme cost sensitivity**
  - $.05 adds up over 1,000,000 units

Real-Time (1)

The Oxford dictionary of Computing offers this definition for real-time systems:

Any system in which the time at which the output is produced is significant. This is usually because the input corresponds to some movement in the physical world, and the output has to relate to that same movement. The lag from input time to output time must be sufficiently small for acceptable timeliness.

Real-Time system is defined as a system where the correctness of the system depends not only the result of computations but also on the time at which it is produced. Therefore the *time* is the most important item to be managed.

Real-Time (2)

Definition in Laplante's book:

A real-time system is a system that must satisfy explicit (bounded) response-time constraints or risk severe consequences, including failure.

Failed System:

A failed system is a system which cannot satisfy one or more of the requirements laid out in the formal system specification.

Real-Time (3)

It can be argued that
all practical systems are real-time!

Hard Real-Time

Systems where failure to meet system response time constraints leads to a system failure are called hard real-time systems.

Soft Real-Time:

Systems where performance is degraded but not destroyed by failure to meet system response time constraints.

Firm Real-Time:

Systems with hard deadlines where some low probability of missing deadline can be tolerated.

Hard, Soft or Firm Real-Time?

- Word-processing program
- ABS in a car
- Ordinary mail posting system
- Video multicasting over the Internet
- Voice over IP
- Telesurgery
- Nuclear power plants
    a) security breach in a plant
    b) over-temperature control

How to Design an Embedded System?

Your design can be based on:

- Standard PC
- Embedded PC (PC104)
- Design with general purpose µPs (80x86)
- Design with µCs (8051, PIC)
- FPGA, ASIC design

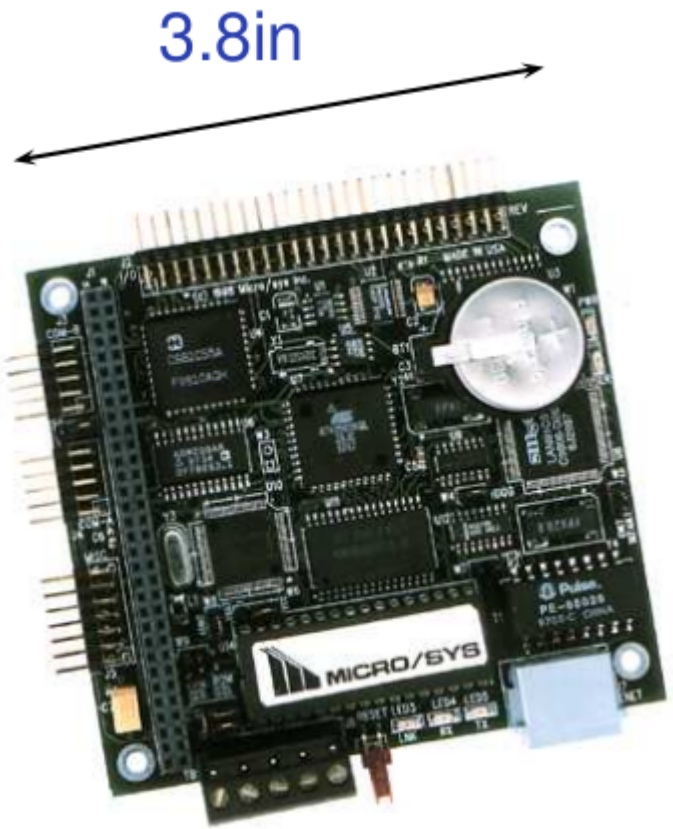
Embedded PC

- Embedded PC can be very different from a desktop PC. It can be hidden from user, it might not have user interface, a display or keyboard, or it might have a different user dialog unit that we are not used to.
- PC provides user with many development tools and desktop PC's all advantages. The advantages of today's PCs can be listed as follows:
  - PCs are ubiquitous.
  - PCs make it easy to create prototypes.
  - It can be easier to develop a project on a PC
  - PCs offer low cost hardware.
  - Low cost, high quality development tools are available for PC.
  - A wide variety of PC-compatible products are available.
  - The PC architecture continues to offer increasing performance.
  - PCs offer a huge range of display and input options.

PC/104 (1)

PC/104 is a standard for PC-compatible modules (circuit boards) that can be stacked together to create an embedded computer system. These types of systems are often found in factories, laboratories, and machinery to provide programmable control of a complex system.

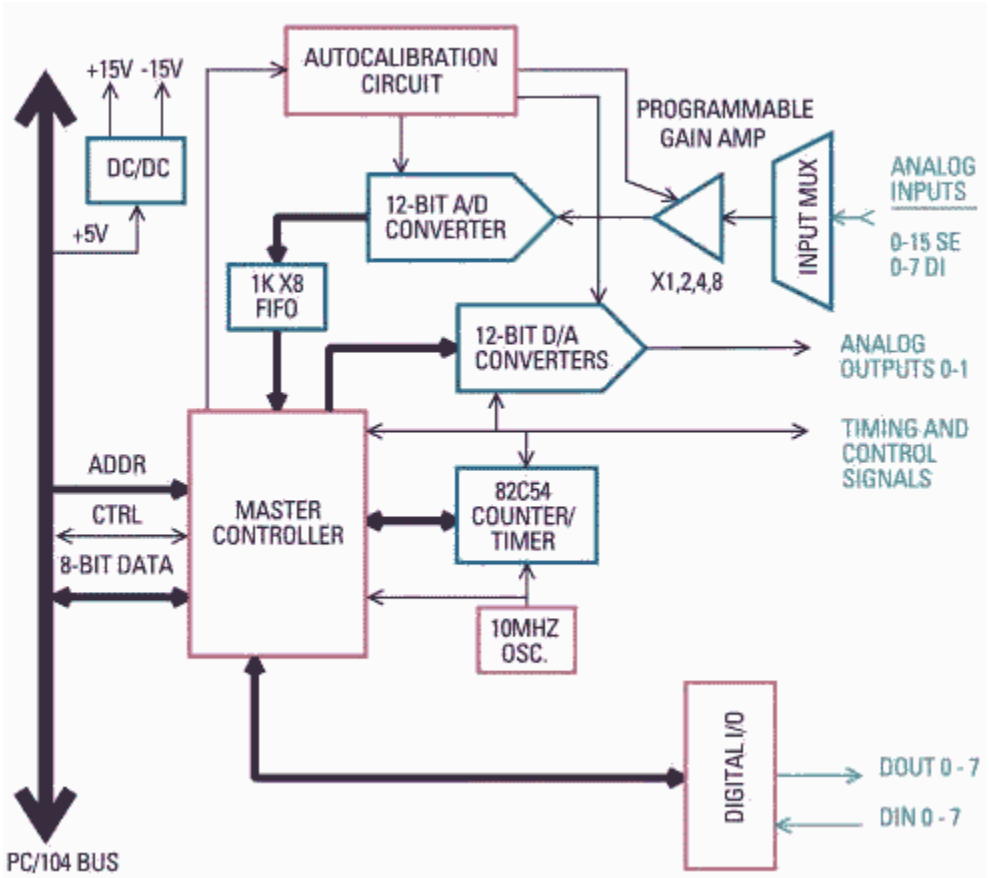
Form Factor	Release Year	Bus Communication	Current Version
PC/104	1992	ISA (AT and XT)	2.5
PC/104-*Plus*	1997	ISA and PCI	2.0
PCI-104	2003	PCI	1.0



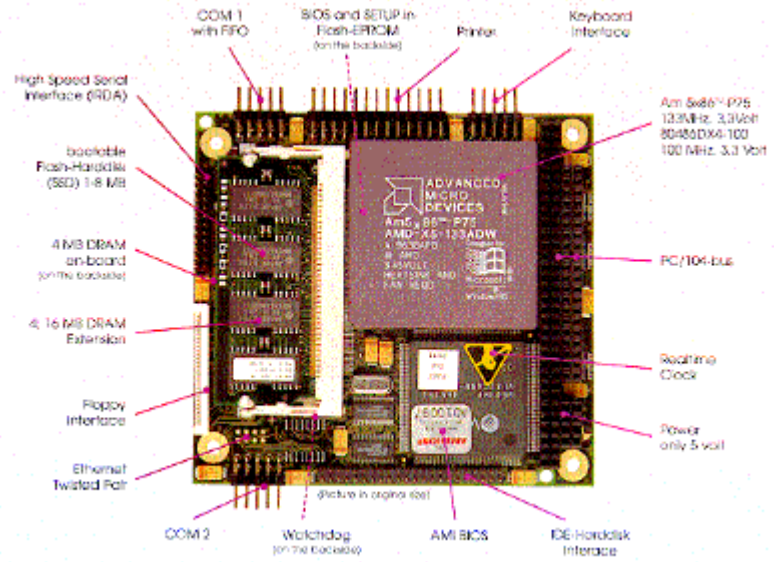
For more information: www.pc104.org

PC104 (2)

enclosure



Block diagram of a PC104 analog module



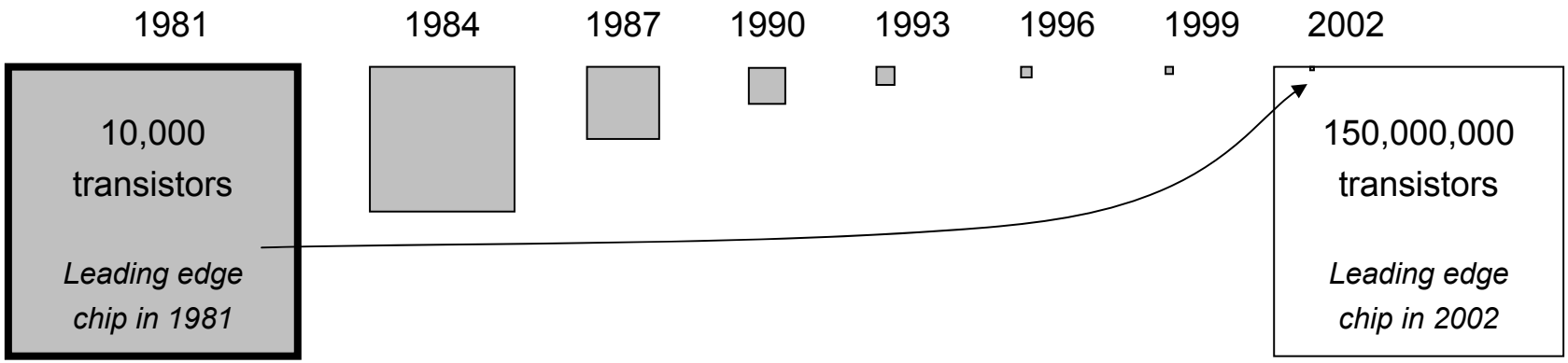
PC104 mainboard

stack

Moore's Law

- Predicted in 1965 by Intel co-founder Gordon Moore

IC transistor capacity has doubled roughly every 18 months for the past several decades



- A 2002 chip can hold about 15,000 1981 chips inside itself

Types of Embedded Systems

- General Computing

  Applications similar to desktop computing, but in an embedded package.
  Video games, set-top boxes, wearable computers.
  Automatic tellers

- Control Systems

  Closed-loop feedback control of real-time system.
  Vehicle engines, chemical processes, nuclear power, flight control.

- Signal Processing

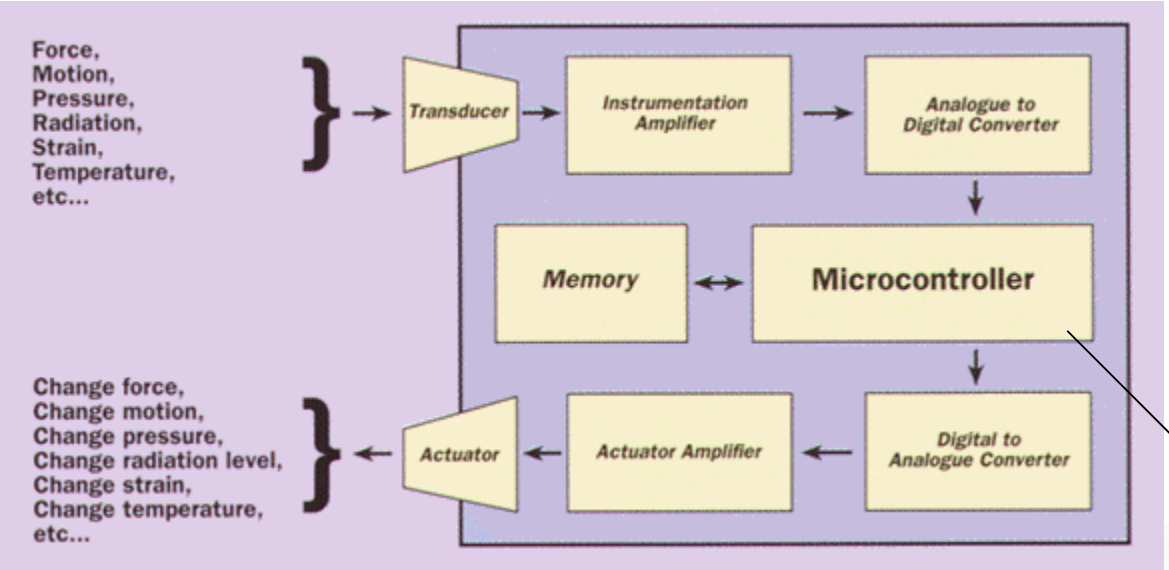
  Computations involving large data streams.
  Radar, Sonar, video compression.

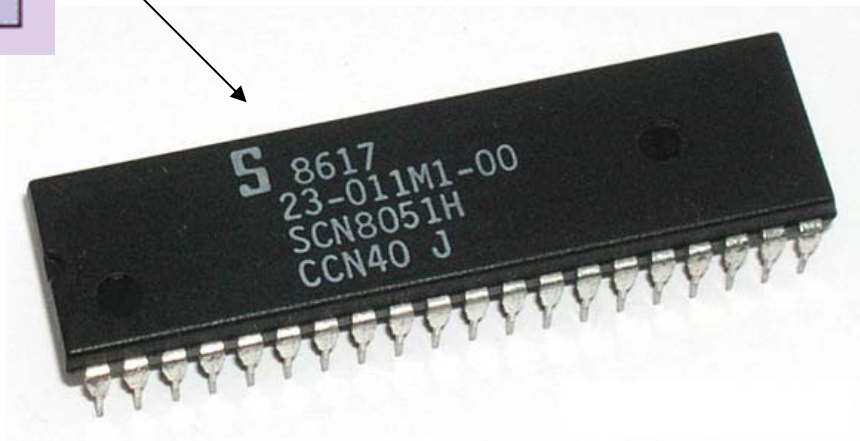
- Communication & Networking

  Switching and information transmission
  Telephone system, Internet

Design with µ-Processors/Controllers

Right combination of :
- a piece of microprocessor based hardware
- a suitable software to undertake your specific task





- Many choices:
  - 8 bit: MCS-48, MCS-5, PIC, picoblaze
  - 16 bit: MCS-96, PIC
  - 32 bit: AVR32, microblaze

Layers of a Computer System

High Level Sum := Sum + 1

Assembly MOV BX,SUM INC (BX)

Machine 1101010100001100 0010001101110101
1111100011001101

Register Transfer Fetch Instruction, Increment PC, Load ALU with SUM ...

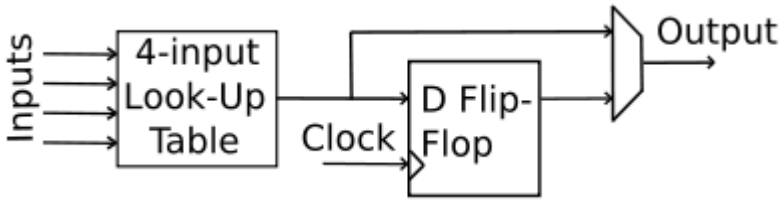
Gate

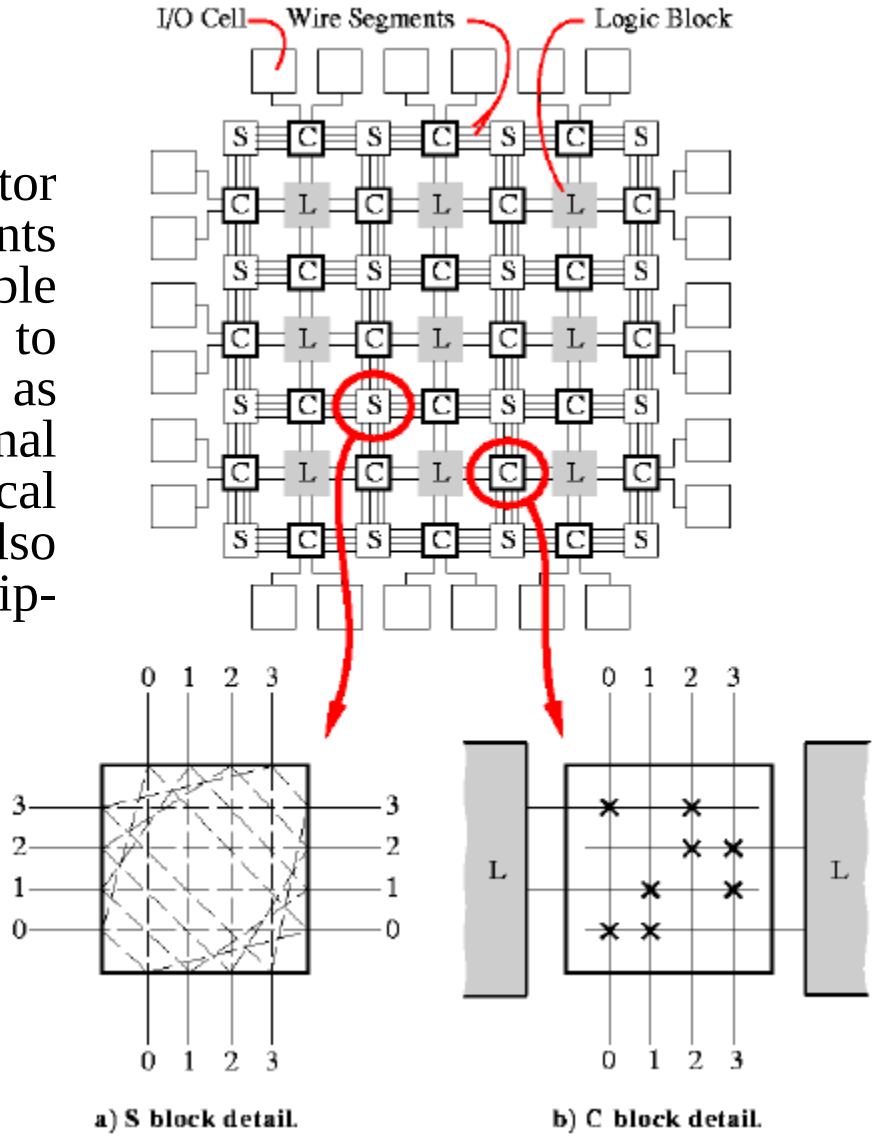
Circuit

FPGA Design

A field-programmable gate array is a semiconductor device containing programmable logic components called "logic blocks", and programmable interconnects. Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinational functions such as decoders or simple mathematical functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

A classic FPGA logic block consists of a 4-input lookup table (LUT), and a flip-flop:





Soft Processors

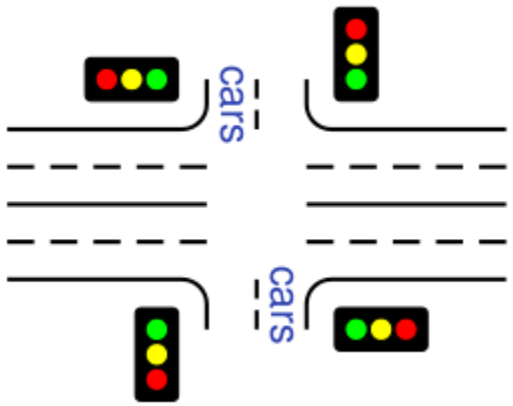
A soft microprocessor (also called softcore microprocessor or a soft processor) is a microprocessor core that can be wholly implemented using logic synthesis. It can be implemented via different semiconductor devices containing programmable logic (e.g., FPGA, CPLD).

Notable soft microprocessors include:

- MicroBlaze
- Nios II

Processor	Developer	Open Source	Bus Support	Notes	Project Home
MicroBlaze	Xilinx	no	OPB, FSL, LMB		Xilinx MicroBlaze
PicoBlaze	Xilinx	no			Xilinx PicoBlaze
Nios, Nios II	Altera	no			Altera Nios II
Cortex-M1	Arm	no			[1]
Mico32	Lattice	yes			LatticeMico32
AEMB	Shawn Tan	yes	Wishbone	MicroBlaze EDK 3.2 compatible Verilog core	AEMB
OpenFire	Virginia Tech CCM Lab	yes	OPB, FSL	Binary compatible with the MicroBlaze	VT OpenFire
PacoBlaze	Pablo Bleyer	yes		Compatible with the PicoBlaze processors	PacoBlaze

Example: Basic Traffic Light Controller



Traffic light controller controls a traffic light at the intersection of a busy highway and a farm road. Normally, the highway light is green but if a sensor detects a car on the farm cars road, the highway light turns yellow then red. The farm road light then turns green until there are no cars or after a long timeout. Then, the farm road light turns yellow then red, and the highway light returns to green. The inputs to the machine are the car sensor, a short timeout signal, and a long timeout signal. The outputs are a timer start signal and the colors of the highway and farm road lights.

How to design this controller?