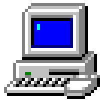
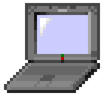
B649 Class Presentation - Embedded Systems

Abhijeet K
Prashant S

Overview

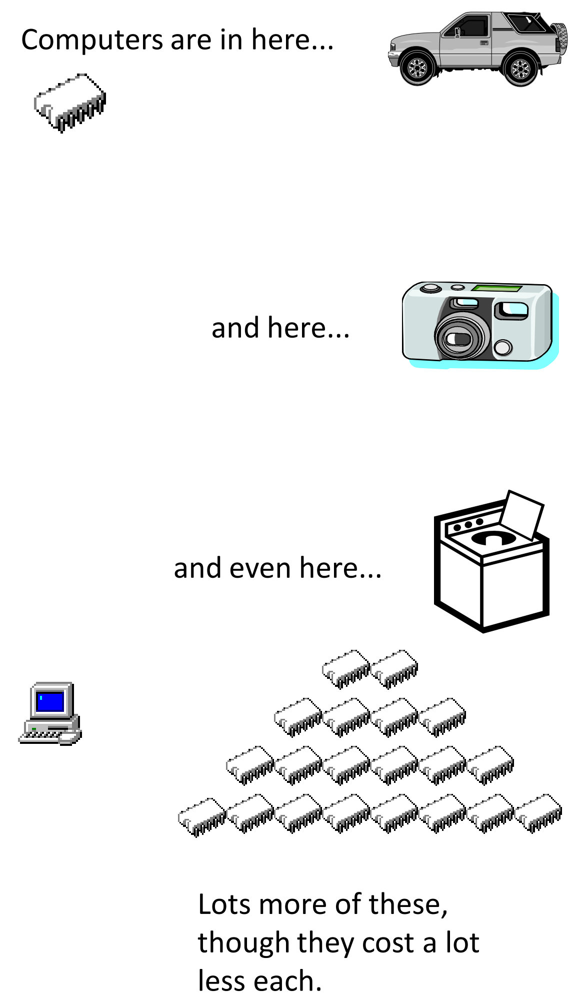
- Embedded systems overview
  - What are they?
- Design challenge – optimizing design metrics
- Technologies
  - Processor technologies
  - IC technologies
  - Design technologies
- Case Studies

Embedded Systems Overview

- Computing systems are everywhere
- Most of us think of “desktop” computers
  - PC’s 
  - Laptops 
  - Servers
  - Mainframes
- But there’s another type of computing system
  - Far more common...

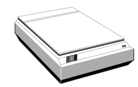
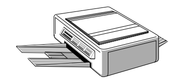
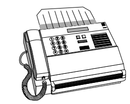
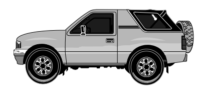
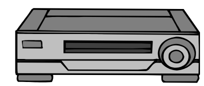
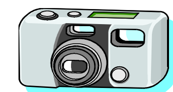
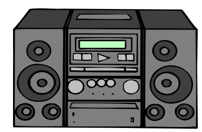
Embedded Systems Overview

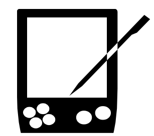
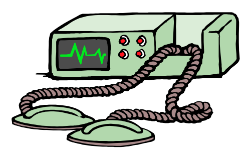
- Embedded computing systems
  - Computing systems embedded within electronic devices
  - Hard to define. Nearly any computing system other than a desktop computer
  - Billions of units produced yearly, versus millions of desktop units
  - Perhaps 50 per household and per automobile

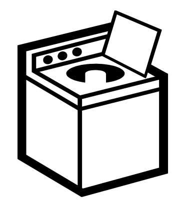


A “short list” of Embedded Systems

- Microwaves
- Washing machines
- Printers
- Networking devices
- Automobiles
- Cell phones
- PDAs
- Mp3 players
- Video game consoles
- TVs
- Children's toys



And the list goes on and on

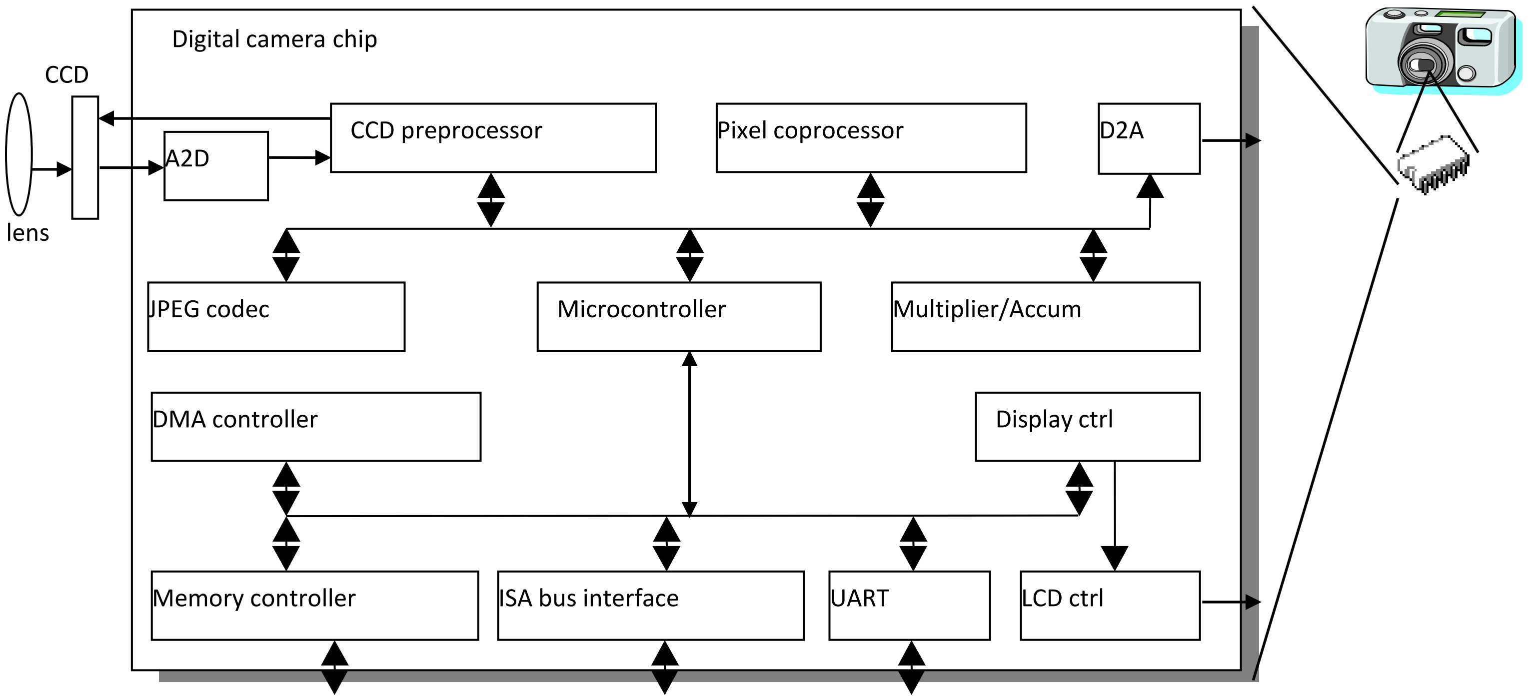
Cost Comparison

Feature	Desktop	Server	Embedded
Price of system	\$1000–\$10,000	\$10,000–\$10,000,000	\$10–\$100,000 (including network routers at the high end)
Price of microprocessor module	\$100–\$1000	\$200–\$2000 (per processor)	\$0.20–\$200 (per processor)
Microprocessors sold per year (estimates for 2000)	150,000,000	4,000,000	300,000,000 (32-bit and 64-bit processors only)
Critical system design issues	Price-performance, graphics performance	Throughput, availability, scalability	Price, power consumption, application-specific performance

Characteristics of Embedded Systems

- Single-functioned
  - Executes a single program, repeatedly
- Tightly-constrained
  - Low cost, low power, small, fast, etc.
- Reactive and real-time
  - Continually reacts to changes in the system's environment
  - Must compute certain results in real-time without delay

An Embedded System Example -- A Digital Camera



- Single-functioned -- always a digital camera
- Tightly-constrained -- Low cost, low power, small, fast
- Reactive and real-time -- only to a small extent

Design Challenge – Optimizing Design Metrics

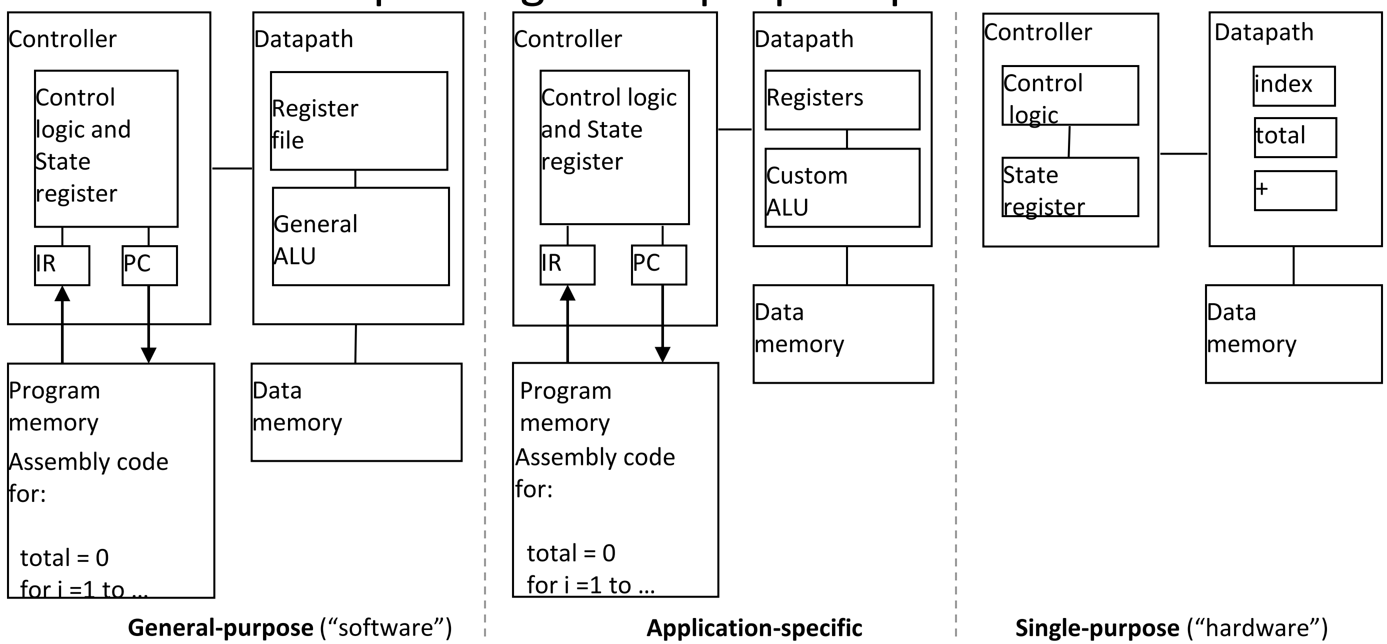
- Common metrics
  - Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
  - NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
  - Size: the physical space required by the system
  - Performance: the execution time or throughput of the system
  - Power: the amount of power consumed by the system
  - Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost

Three Key Embedded System Technologies

- Technology
  - A manner of accomplishing a task, especially using technical processes, methods, or knowledge
- Three key technologies for embedded systems
  - Processor technology
  - IC technology
  - Design technology

Processor Technology

- The architecture of the computation engine used to implement a system's desired functionality
- Processor does not have to be programmable
  - "Processor" *not* equal to general-purpose processor

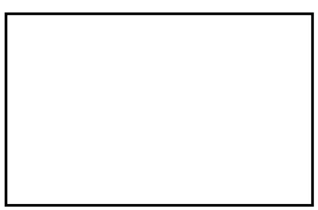


Processor Technology

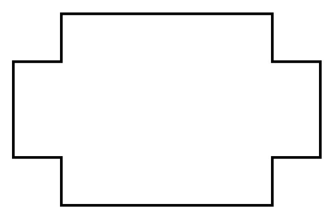
- Processors vary in their customization for the problem at hand

Desired functionality

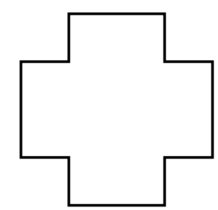
```
total = 0
for i = 1 to N  loop
   total += M[i]
end loop
```



General-purpose processor



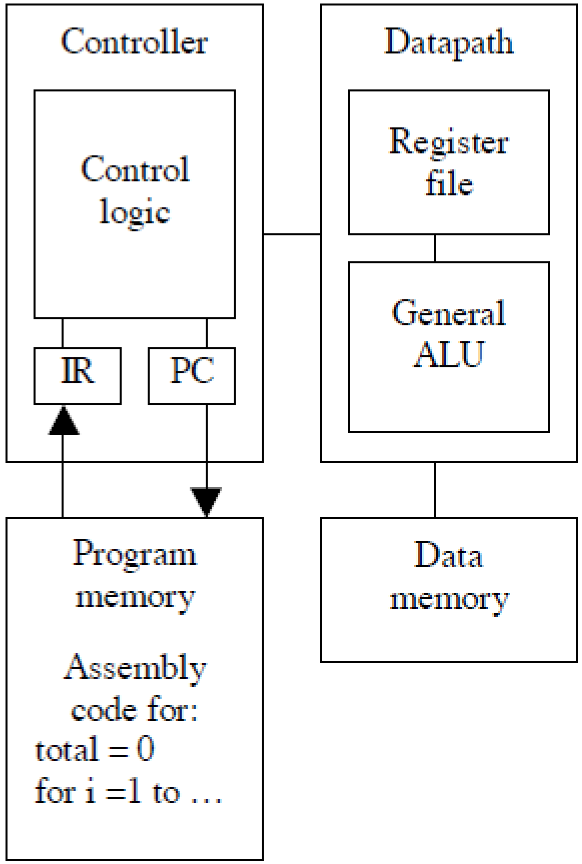
Application-specific processor



Single-purpose processor

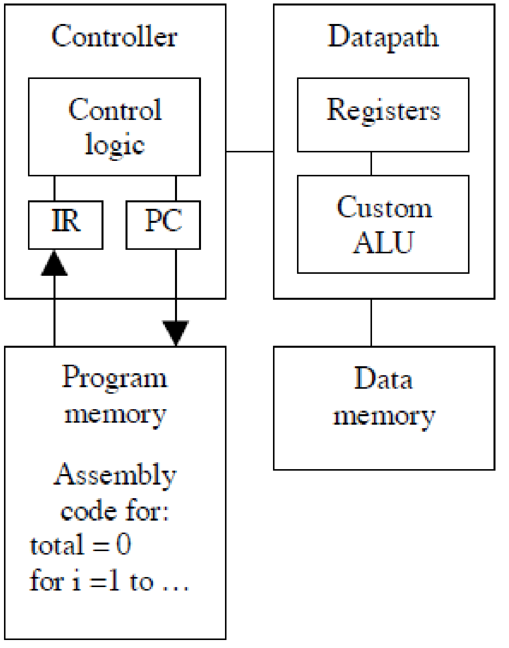
General-Purpose Processors

- Programmable device used in a variety of applications
  - Also known as “microprocessor”
- Features
  - Program memory
  - General data path with large register file and general ALU
- User benefits
  - Low time-to-market and NRE costs
  - High flexibility
- “Pentium” the most well-known, but ther are hundreds of others



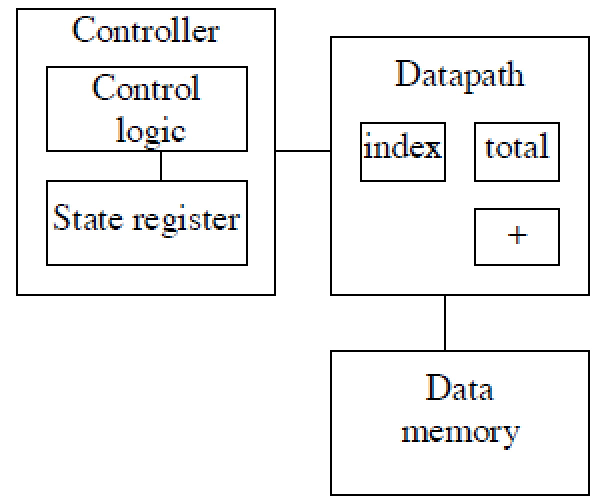
Application-Specific Processors

- Programmable processor optimized for a particular class of applications having common characteristics
  - Compromise between general-purpose and single-purpose processors
- Features
  - Program memory
  - Optimized data path
  - Special functional units
- Benefits
  - Some flexibility, good performance, size and power



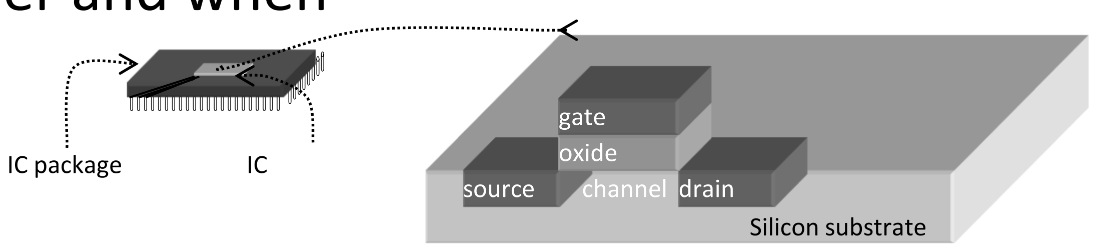
Single-Purpose Processors

- Digital circuit designed to execute exactly one program
  - Also called as coprocessor, accelerator or peripheral
- Features
  - Contains only the components needed to execute a single program
  - No program memory
- Benefits
  - Fast
  - Low power
  - Small size



IC Technology

- The manner in which a digital (gate-level) implementation is mapped onto an IC
  - IC: Integrated circuit, or “chip”
  - IC technologies differ in their customization to a design
  - IC’s consist of numerous layers (perhaps 10 or more)
    - IC technologies differ with respect to who builds each layer and when



IC Technology

- Three types of IC technologies
  - Full-custom/VLSI
  - Semi-custom ASIC (gate array and standard cell)
  - PLD (Programmable Logic Device)

IC Technology: Full-Custom/VLSI

- All layers are optimized for an embedded system's particular digital implementation
  - Placing transistors
  - Sizing transistors
  - Routing wires
- Benefits
  - Excellent performance, small size, low power
- Drawbacks
  - High NRE cost (e.g., $300k), long time-to-market

IC Technology: Semi-Custom

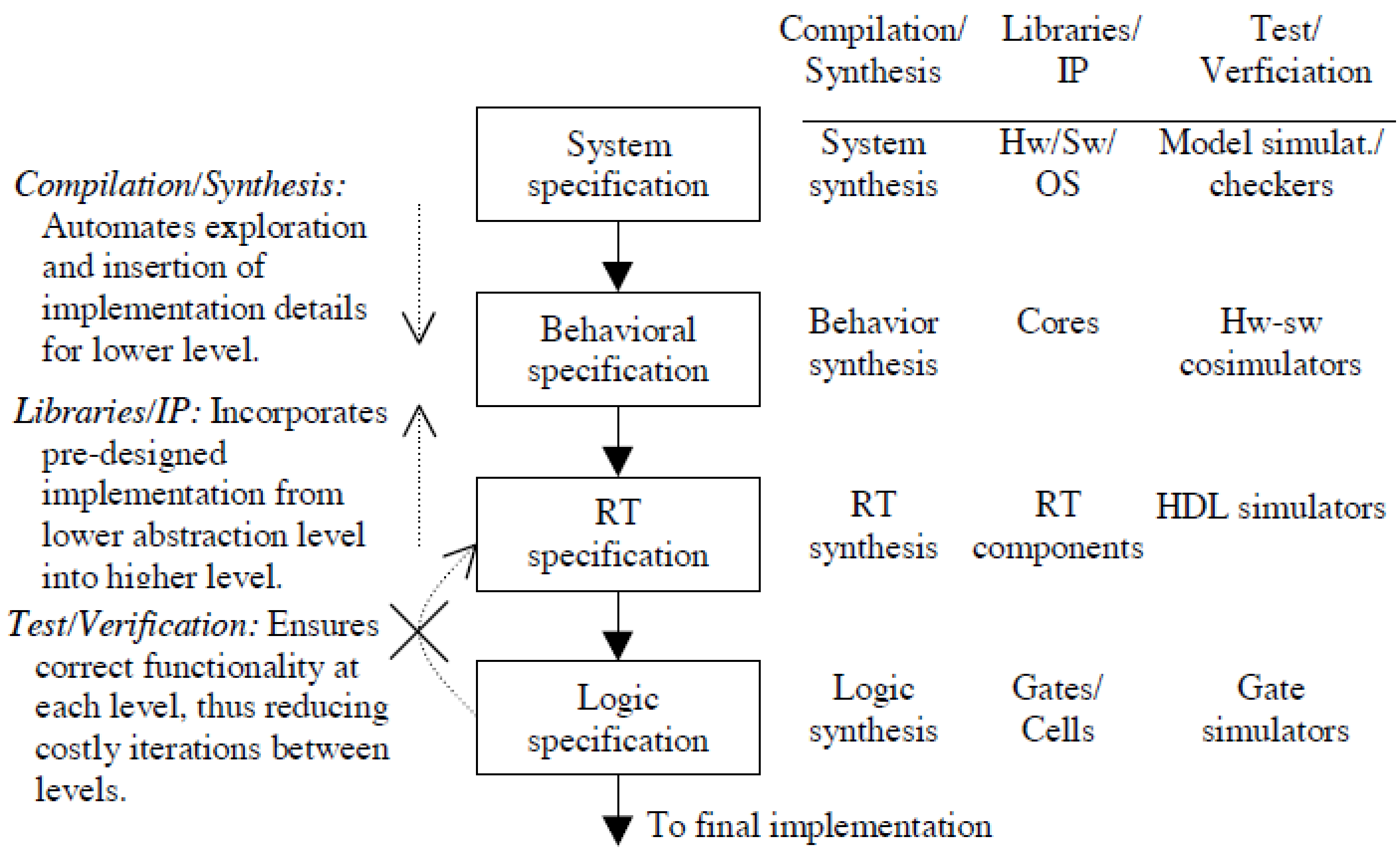
- Lower layers are fully or partially built
  - Designers are left with routing of wires and maybe placing some blocks
- Benefits
  - Good performance, good size, less NRE cost than a full-custom implementation (perhaps $10k to $100k)
- Drawbacks
  - Still require weeks to months to develop

IC Technology: PLD (Programmable Logic Device)

- All layers already exist
  - Designers can purchase an IC
  - Connections on the IC are either created or destroyed to implement desired functionality
  - Field-Programmable Gate Array (FPGA) very popular
- Benefits
  - Low NRE costs, almost instant IC availability
- Drawbacks
  - Bigger, expensive (perhaps $30 per unit), power hungry, slower

Design Technology

- The way in which we convert our concept of desired system functionality into an implementation



Real Time Systems

- A real-time system is a system that performs its functions and responds to external, asynchronous events within some specified time period.
- There are hard and soft real-time systems.
- What is Hard real-time System?
- What is Soft real-time System?

Real Time Systems

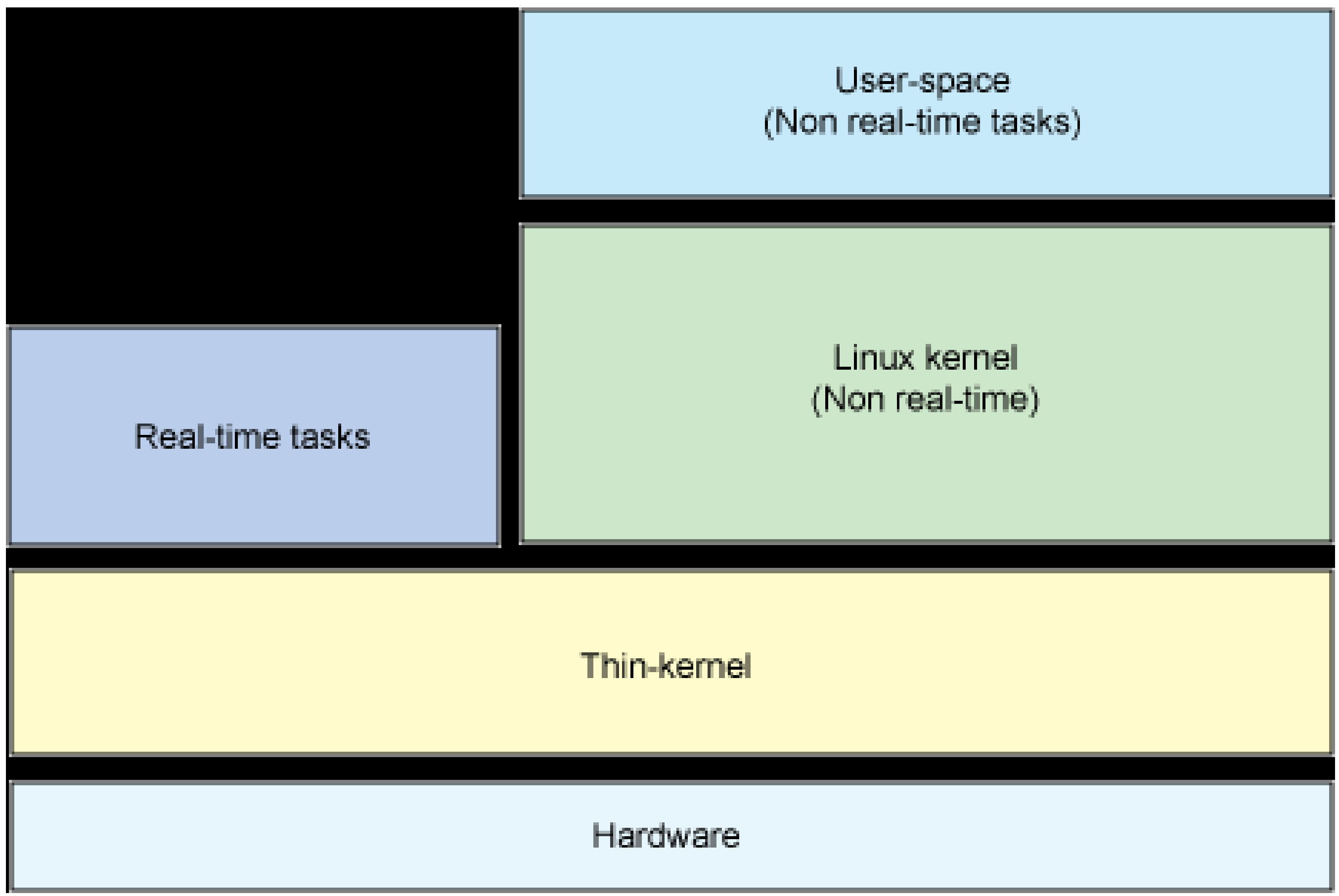
- Few years ago, real-time applications were simple and usually placed on dedicated, customized and isolated hardware.
- Real-time applications today are getting more and more powerful and yet complicated.
- telescopes connected to the Internet, cell phones generating graphic displays, routers and telephone switches.

Real Time Systems

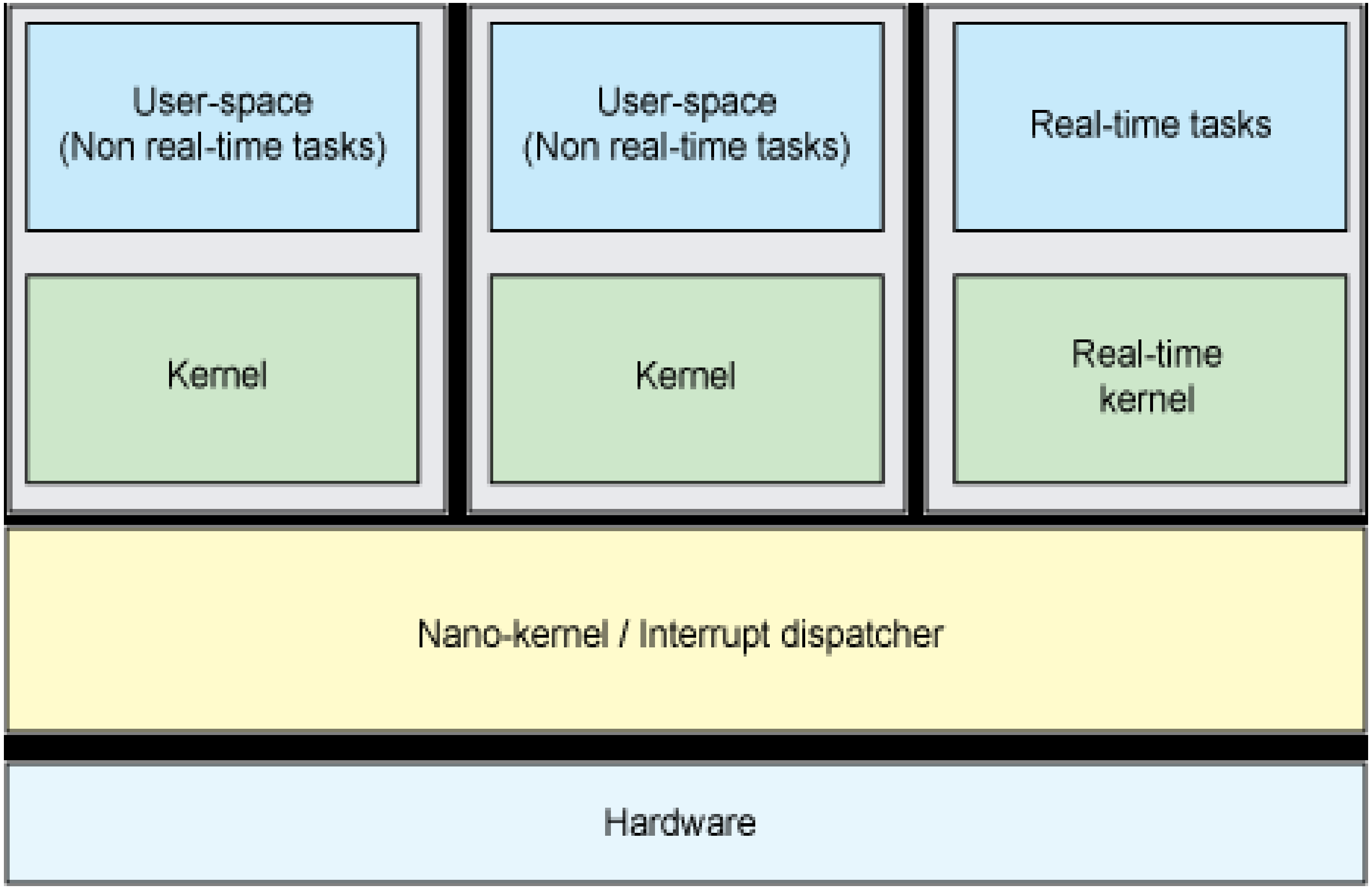
- A good real-time operating system is required to be embedded into those application facilities.
- Why RTOS is needed, cant we use time sharing OS like UNIX, Linux?
- (RTOS) is an operating system capable of guaranteeing timing requirements of the processes under its control.
- For a RTOS, correct timing is the key feature. Throughput is of secondary concern.

Different Approaches to implement Real Time Operating System

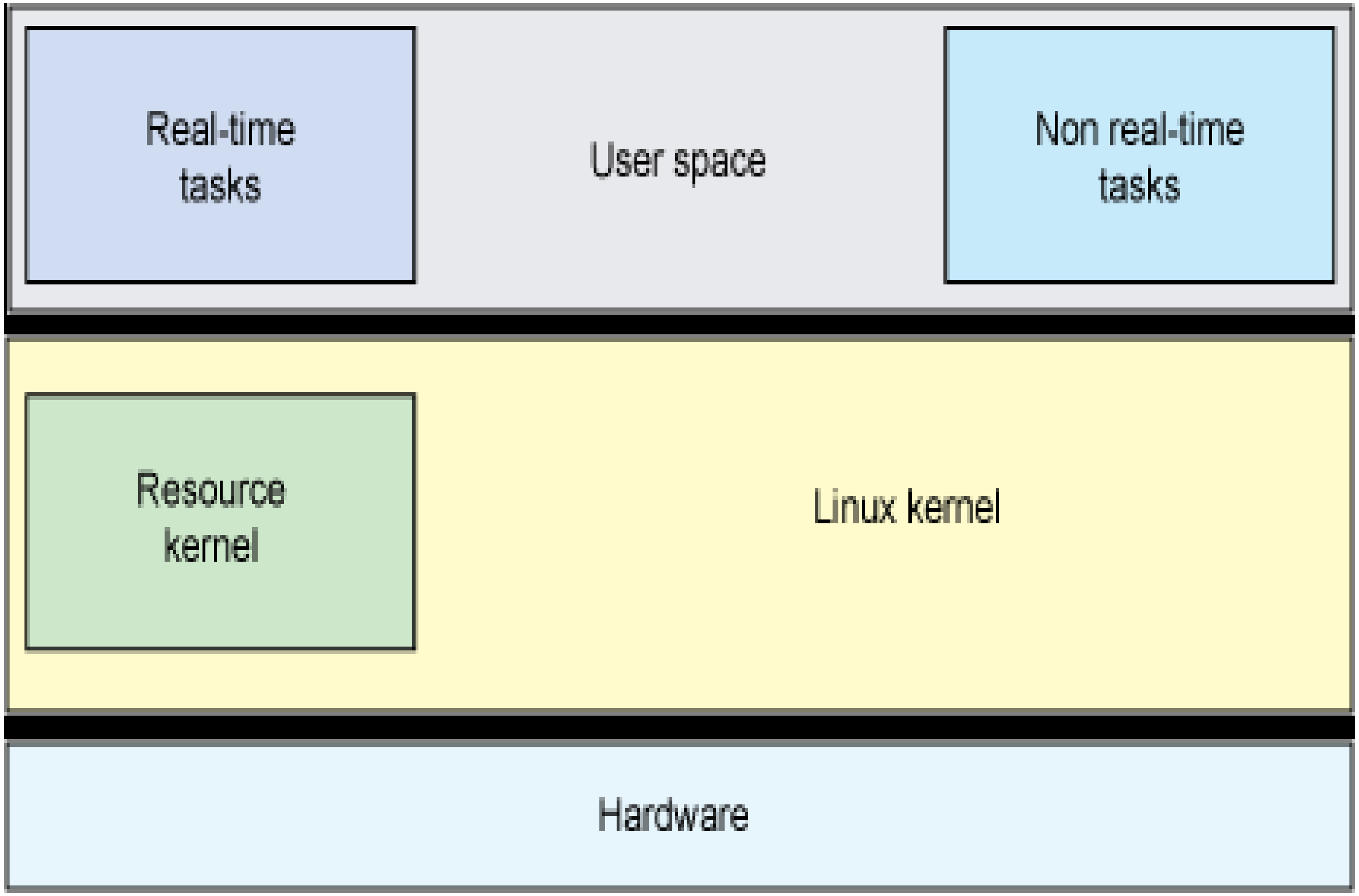
Thin Kernel Approach



Nano Kernel Approach



Resource-kernel Approach



Real Time Systems

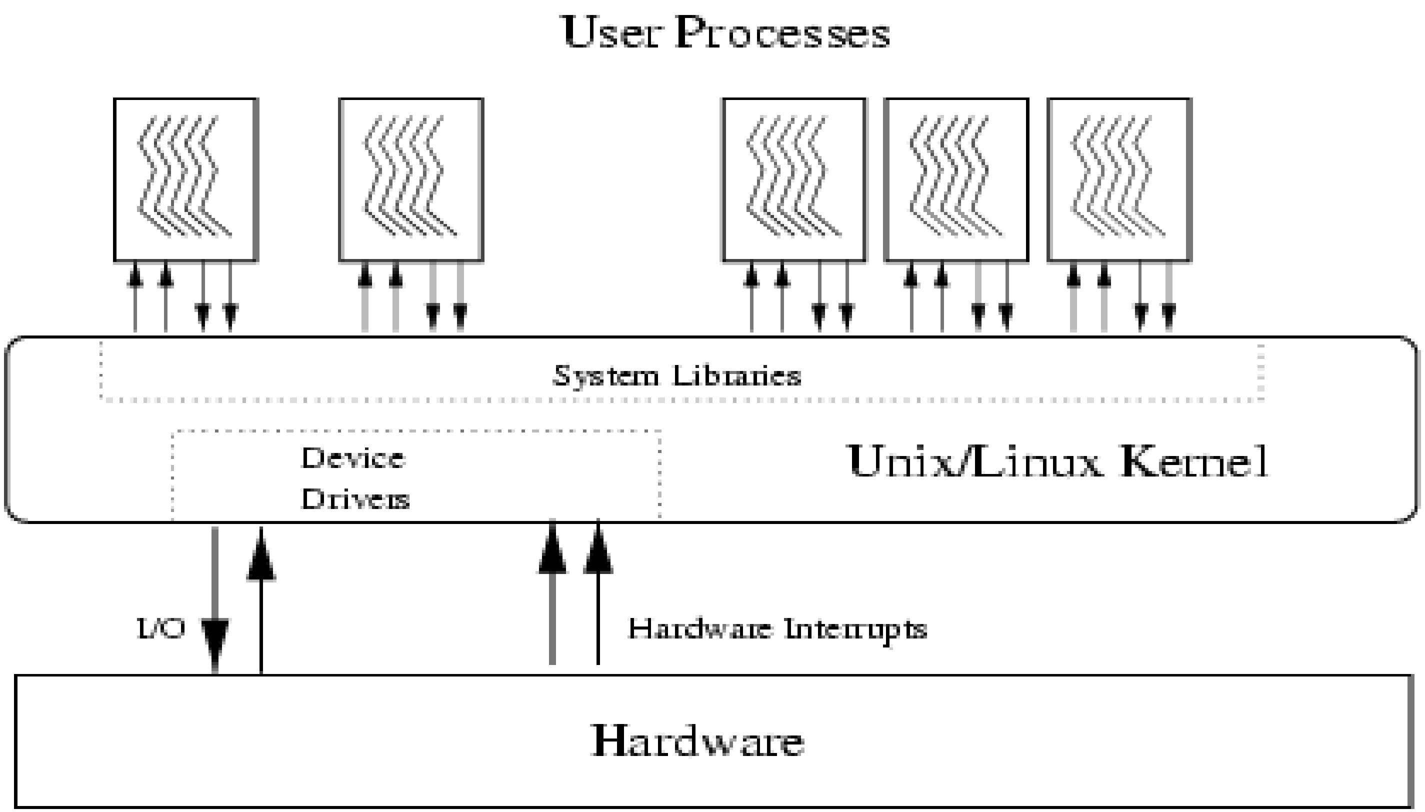


Figure 1.1: Detail of the bare Linux kernel

Real Time Systems

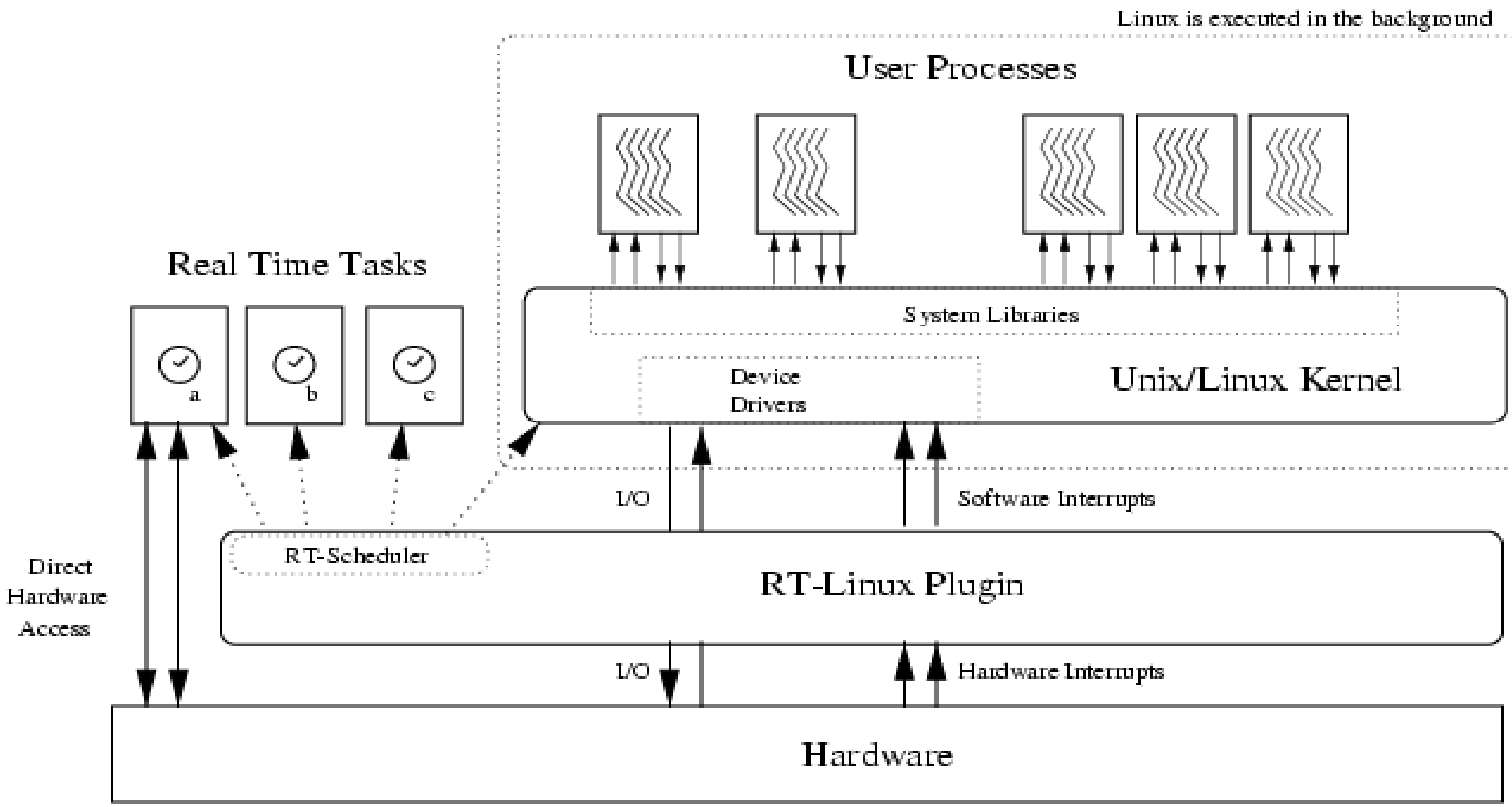


Figure 1.2: Detail of the RTLinux kernel

Signal Processing and Embedded Applications: The Digital Signal Processor

Digital Signal Processor

- Special-purpose processor optimized for executing DSP algorithms
- Most of these algorithms perform the same operation: a multiply-accumulate operation
- Discrete Fourier Transform

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \text{ where } W_N^{kn} = e^{j\frac{2\pi kn}{N}} = \cos\left(2\pi\frac{kn}{N}\right) + j\sin\left(2\pi\frac{kn}{N}\right)$$

- Discrete Cosine Transform

Digital Signal Processor

- Common observation

  Either transform has its core as the SUM OF A PRODUCT

  A = A + B * C

- Digital Signal Processors feature special purpose hardware to *multiply-accumulate (MAC)*

Digital Signal Processor

- Fixed-Point Arithmetic

Example Here are three simple 16-bit patterns:

0100 0000 0000 0000

0000 1000 0000 0000

0100 1000 0000 1000

*i*th digit to the left of the binary point 2^{14}, 2^{11}, and $(2^{14} + 2^{11} + 2^{3})$

*i*th digit to the right of the binary point 2^{-1}, 2^{-4}, and $(2^{-1} + 2^{-4} + 2^{-12})$

Digital Signal Processor

- Fixed Point Arithmetic

  - Low-cost arithmetic as exponent not sent in same word

  - Exponent sent in separate variable

  - *Blocked Floating Point* : exponent variable is shared by set of fixed-point variables

Digital Signal Processor

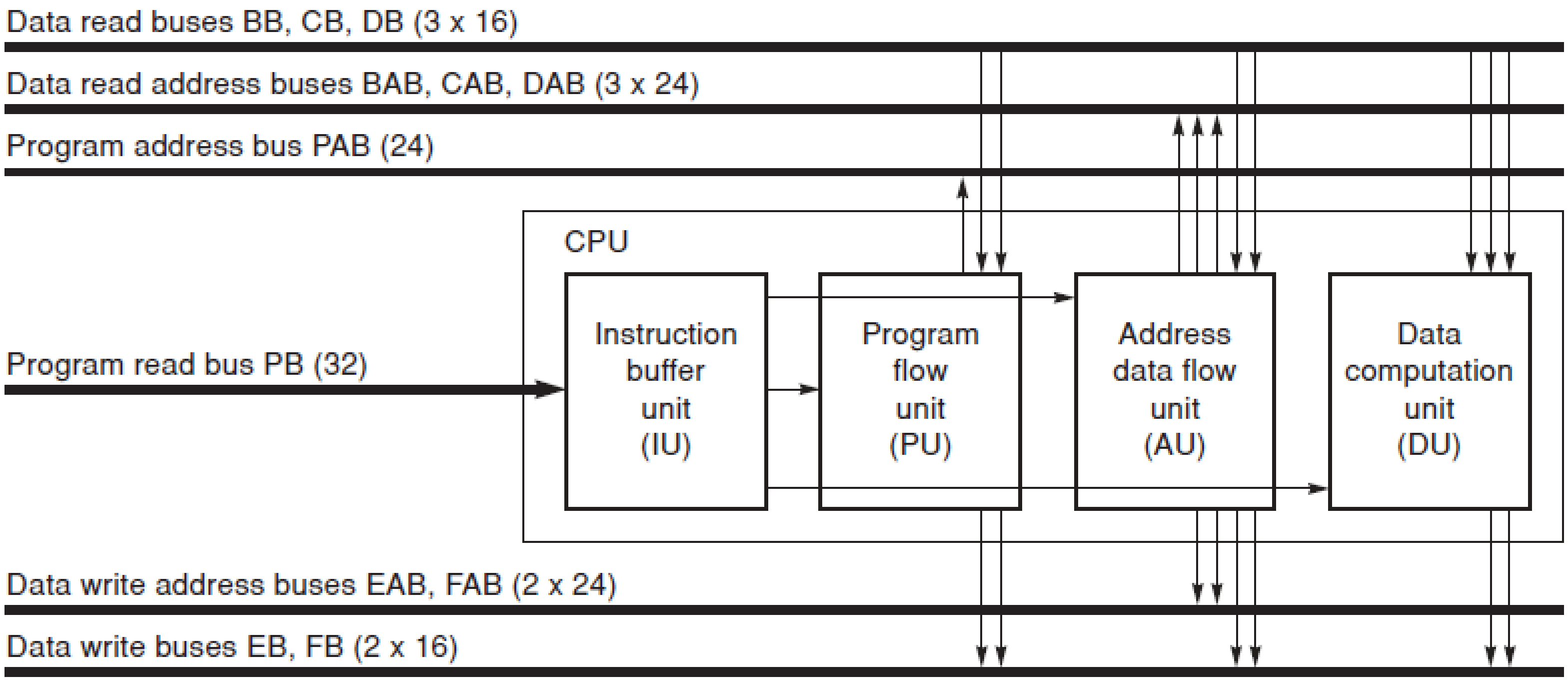
- Accumulator register width more than data register width

Generation	Year	Example DSP	Data width	Accumulator width
1	1982	TI TMS32010	16 bits	32 bits
2	1987	Motorola DSP56001	24 bits	56 bits
3	1995	Motorola DSP56301	24 bits	56 bits
4	1998	TI TMS320C6201	16 bits	40 bits

Figure D.2 Four generations of DSPs, their data width, and the width of the registers that reduces round-off error.

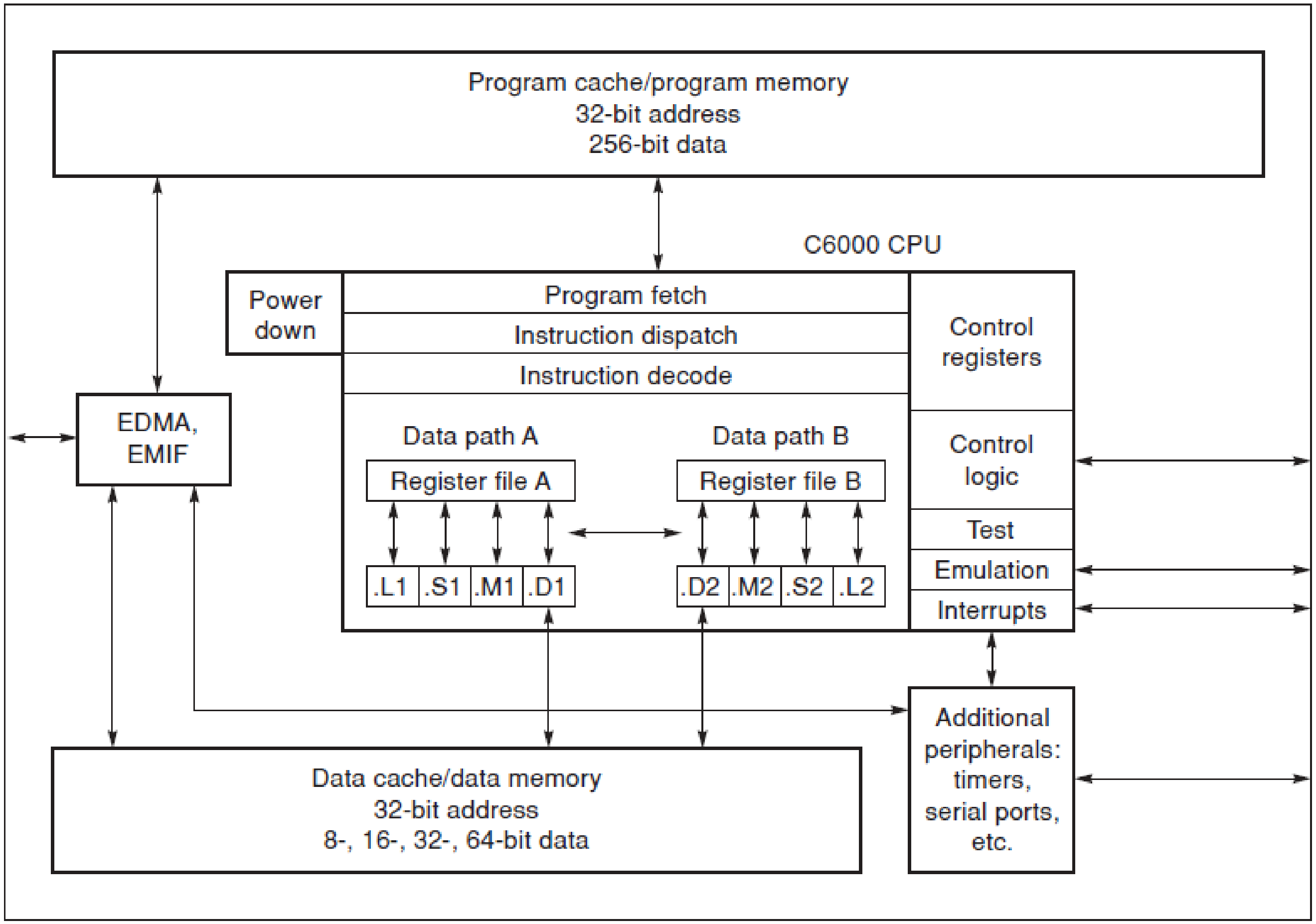
- Accelerate communication algorithms

DSP TI 320C55



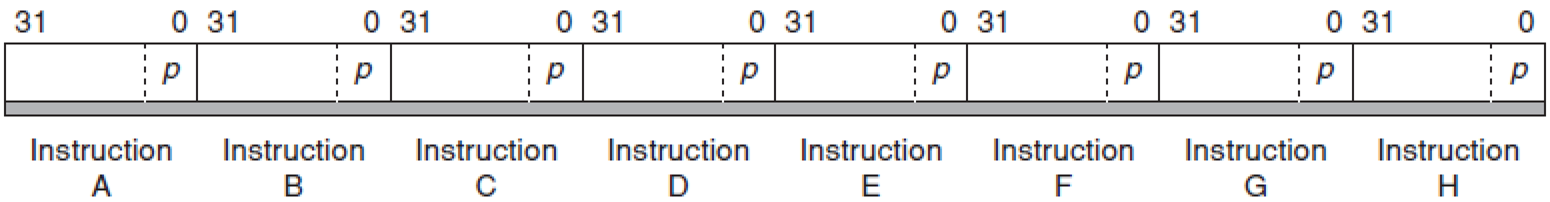
Architecture of the TMS320C55 DSP.

DSP – TI 320C6x



Architecture of the TMS320C64x family of DSPs

DSP – TI 320C6x



Instruction packet of the TMS320C6x family of DSPs.

Media Extensions

- Precision require for multimedia operations is less
- Size of data items is smaller
- Multiple operations can be performed in single cycle: Single-Instruction Multiple-Data (SIMD) or vector instructions

Embedded Benchmarks

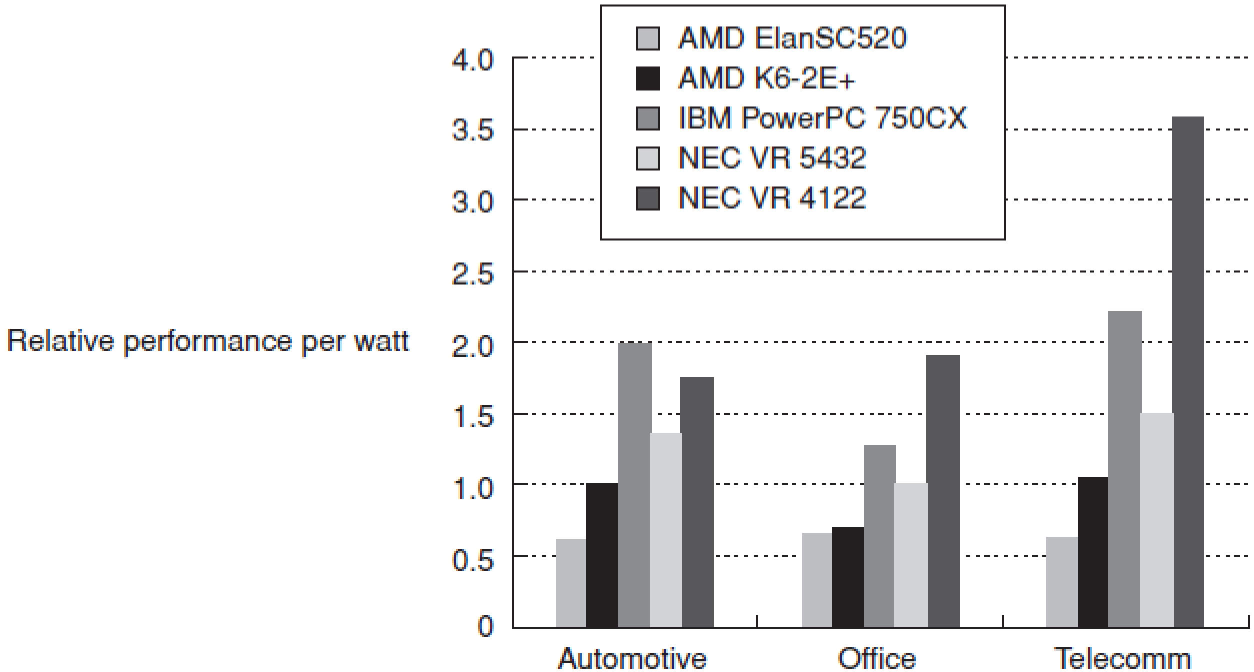
- Till Couple of years back: Dhrystone performance – Benchmark criticized; dropped by desktop systems 20 years back
- EDN Embedded Microprocessor Benchmark Consortium EEMBC (pronounced "embassy")
- EEMBC benchmarks can only be used to partially access performance

Embedded Benchmarks

Benchmark type ("subcommittee")	Number of kernels	Example benchmarks
Automotive/industrial	16	6 microbenchmarks (arithmetic operations, pointer chasing, memory performance, matrix arithmetic, table lookup, bit manipulation), 5 automobile control benchmarks, and 5 filter or FFT benchmarks
Consumer	5	5 multimedia benchmarks (JPEG compress/decompress, filtering, and RGB conversions)
Telecommunications	5	Filtering and DSP benchmarks (autocorrelation, FFT, decoder, encoder)
Digital entertainment	12	MP3 decode, MPEG-2 and MPEG-4 encode and decode (each of which are applied to five different data sets), MPEG Encode Floating Point, 4 benchmark tests for common cryptographic standards and algorithms (AES, DES, RSA, and Huffman decoding for data decompression), and enhanced JPEG and color-space conversion tests
Networking version 2	6	IP Packet Check (borrowed from the RFC1812 standard), IP Reassembly, IP Network Address Translator (NAT), Route Lookup, OSPF, Quality of Service (QOS), and TCP
Office automation version 2	6	Ghostscript, text parsing, image rotation, dithering, bezier

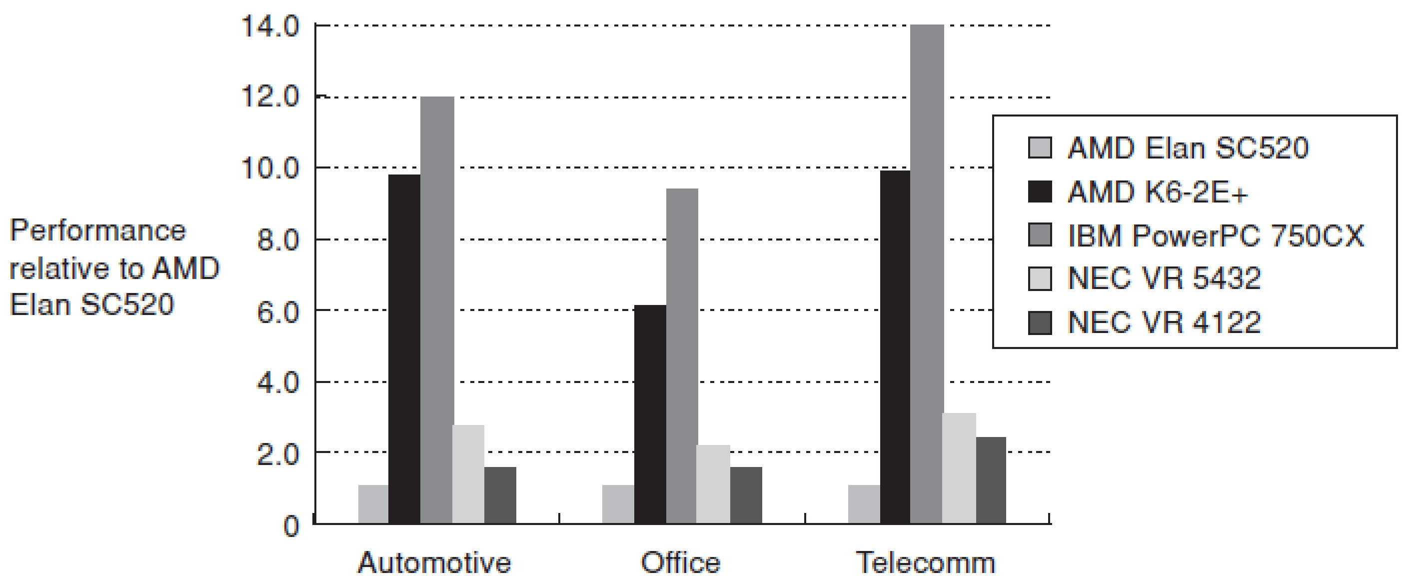
The EEMBC benchmark suite, consisting of 50 kernels in six different classes

Power Consumption and Efficiency as the Metric



Relative performance per watt for the five embedded processors.

Power Consumption and Efficiency as the Metric



Raw performance for the five embedded processors

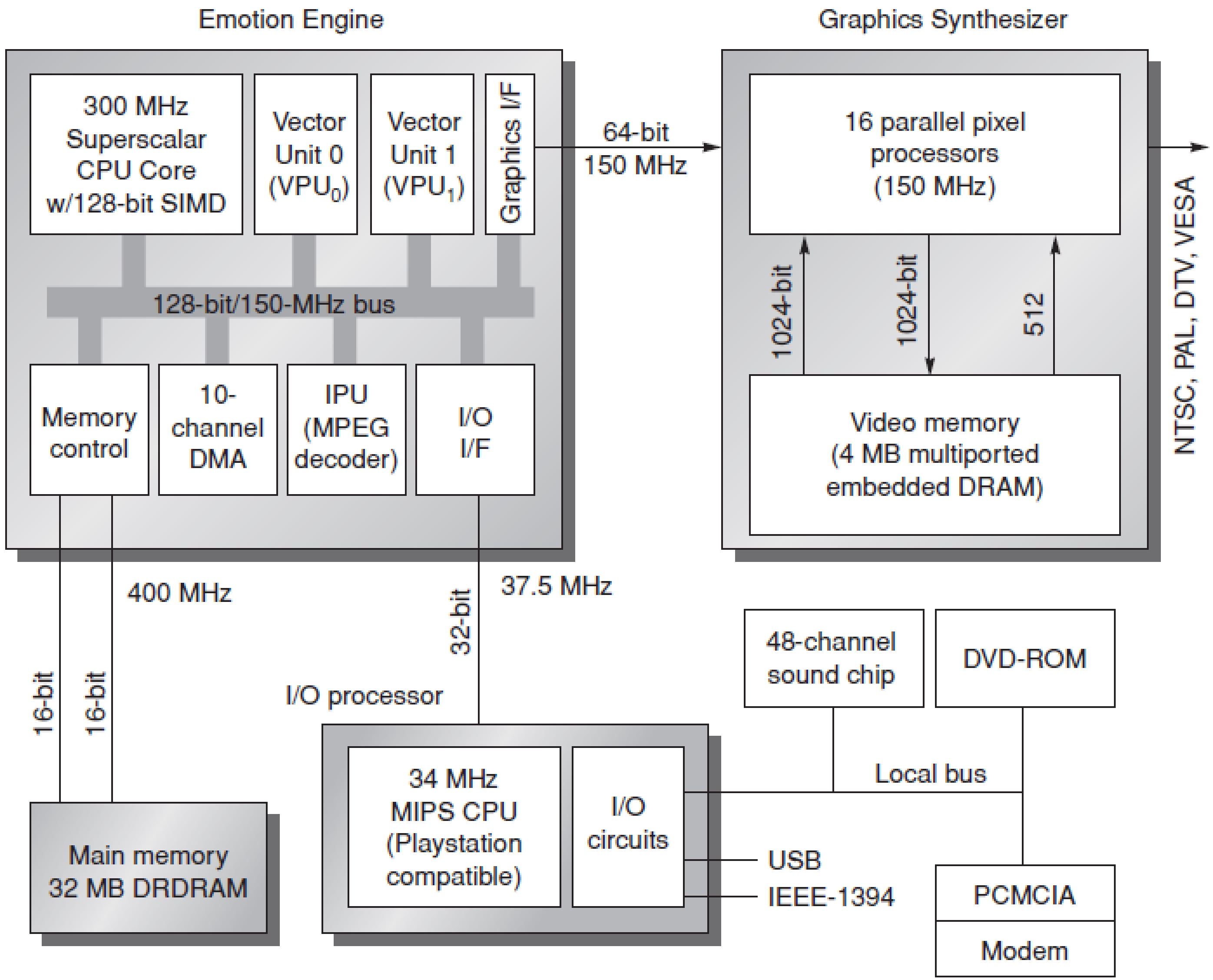
Embedded Multiprocessors

- Embedded Multiprocessors = Several General-Purpose Processors
- Useful where scalability is critical
- Example: MXP processor for use in voice-over-IP systems

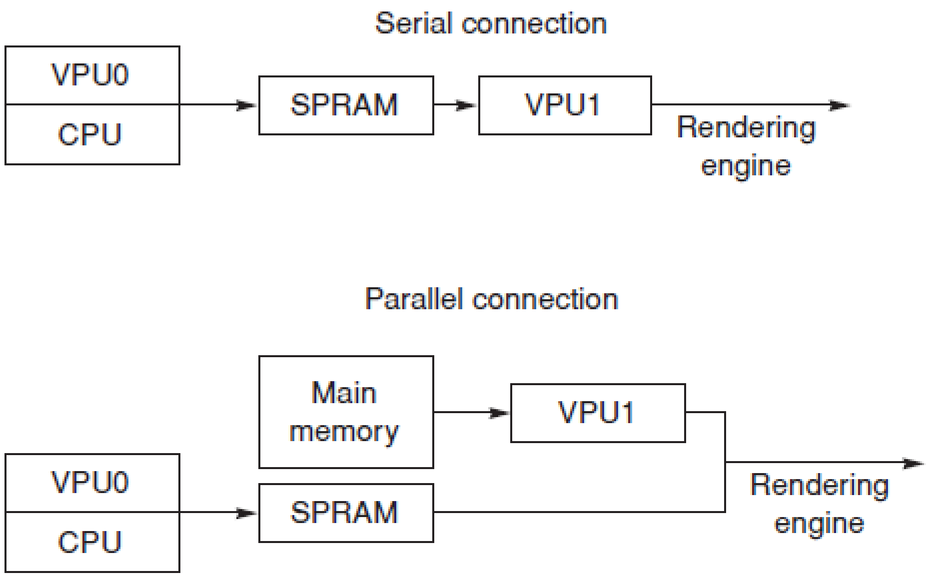
Embedded Multiprocessors

- Multiprocessing has wide spread application in embedded computing:
  - Software is written from scratch or significantly modified
  - Applications have natural parallelism

Case Study – The Emotion Engine of Sony Playstation 2

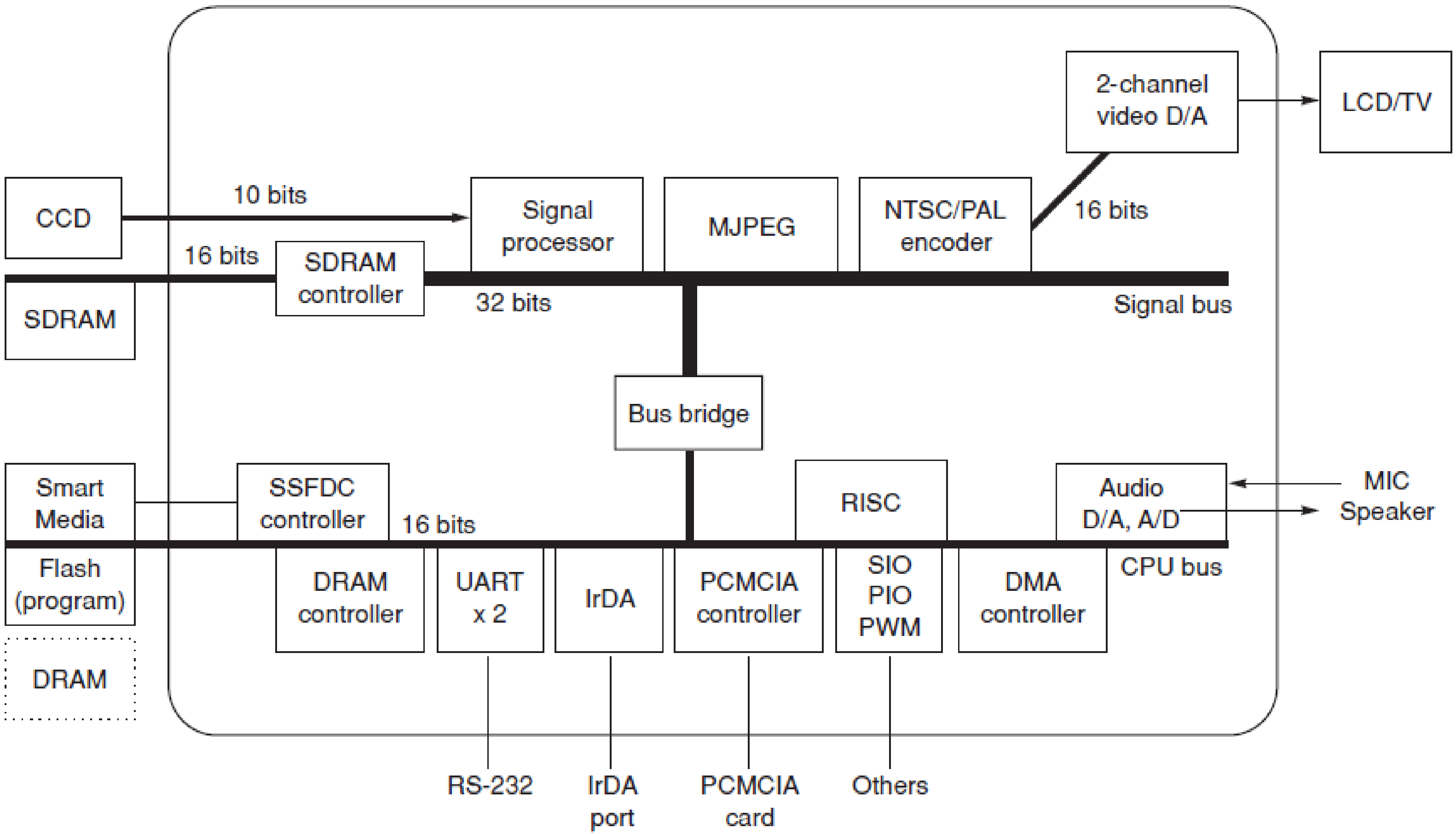


Block diagram of the Sony Playstation 2.



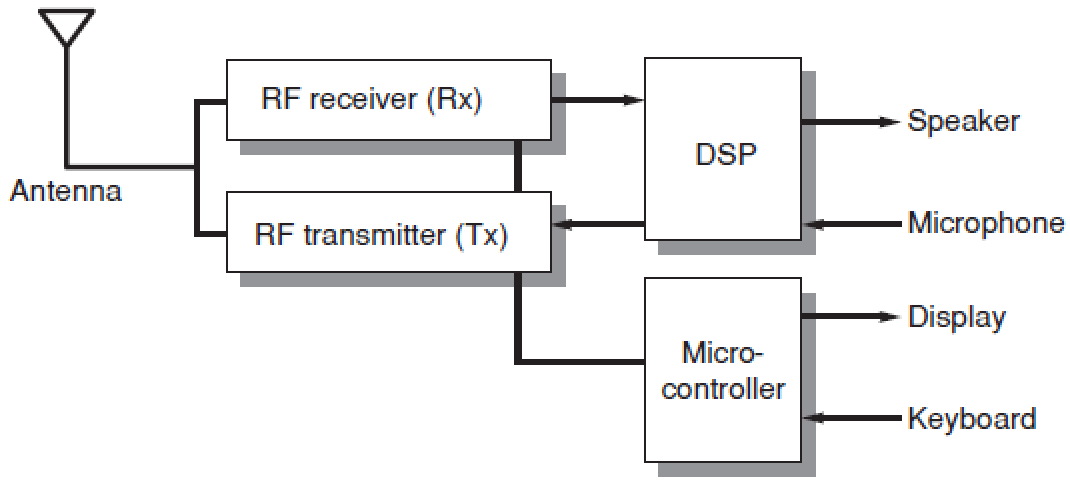
Two modes of using Emotion Engine organization

Case Study – Sanyo VPC – SX 500 Digital Camera

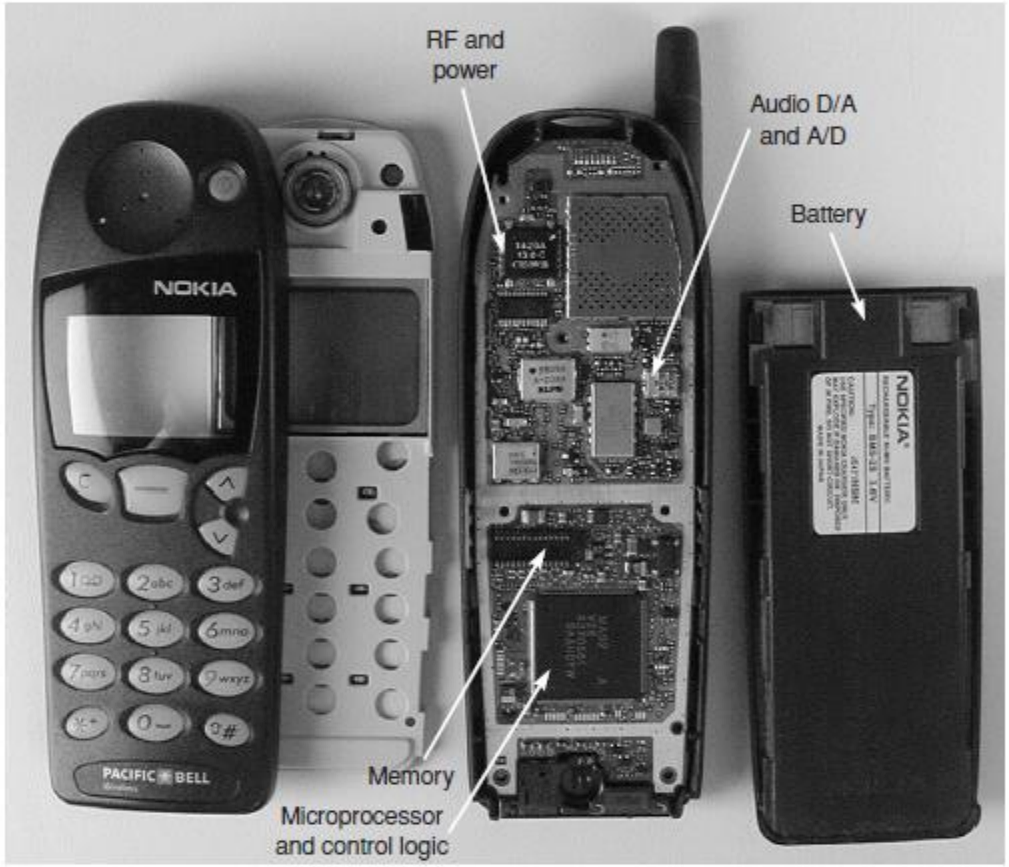


The system on a chip (SOC) found in Sanyo digital cameras.

Case Study – The Cell Phone



Block diagram of a cell phone



Circuit board from a Nokia cell phone

Concluding Remarks

- Architectural Decisions for general-purpose applications are less desirable in embedded applications
- Due to chip area, cost, power, and real-time constraints
- Programming model for these systems places more demand on programmer and compiler to extract parallelism

References

- Computer Architecture – A Quantitative approach, 4th edition By John L. Hennessy and David A. Patterson – Appendix D – Embedded Systems
- Embedded System design – Unified Hardware/Software Approach By Frank Vahid and Tony Givargis
- http://www.ibm.com/developerworks/linux/library/l-real-time-linux/

Thank You