Unit -IV

Embedded system

- An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.
- An embedded system has specific requirements and performs pre-defined tasks, unlike a general-purpose personal computer.
- An embedded system is a programmed hardware device.
- Embedded systems are a combination of hardware and software which facilitates mass production and a host of applications.

Embedded system

- A processor is an important unit in the embedded system hardware. It is the heart of the embedded system.
- Embedded systems contain processing cores that are either microcontrollers or DSP
- Embedded Systems are the crucial components of the modern compacted devices with multifunction capabilities.
- They are dedicated to handle a particular task.

Embedded system

- Since the embedded system is dedicated to specific tasks, design engineers can optimize its size and cost of the product and increase the reliability and performance
- Embedded systems :
  - portable devices
    - digital watches
    - MP3 players
    - PDAs

Embedded system

– large stationary installations
  - Traffic lights
  - Factory controllers
- largely complex systems
  - hybrid vehicles
  - MRI
  - avionics

Embedded system

- Program instructions written for embedded systems are referred to as firmware
- It is stored in ROM or Flash memory chips.
- They run with limited computer hardware resources: little memory, small or non-existent keyboard or screen

Embedded system

- Embedded systems have a much wider range of processing power and cost
  - low-end 8-bit and 16-bit processors that may cost less than a dollar
  - full 32-bit microprocessors capable of operating in the 500 MIPS range that cost approximately 10 dollars
  - high-end embedded processors that cost hundreds of dollars and can execute several billions of instructions per second.

Embedded system

- Embedded systems often process information in different ways typically include deadline-driven constraints—so-called *real-time constraints*
- Embedded systems applications typically involve processing information as signals
- a signal may be
  - an image,
  - a motion picture composed of of images,
  - a control sensor measurement, and so on

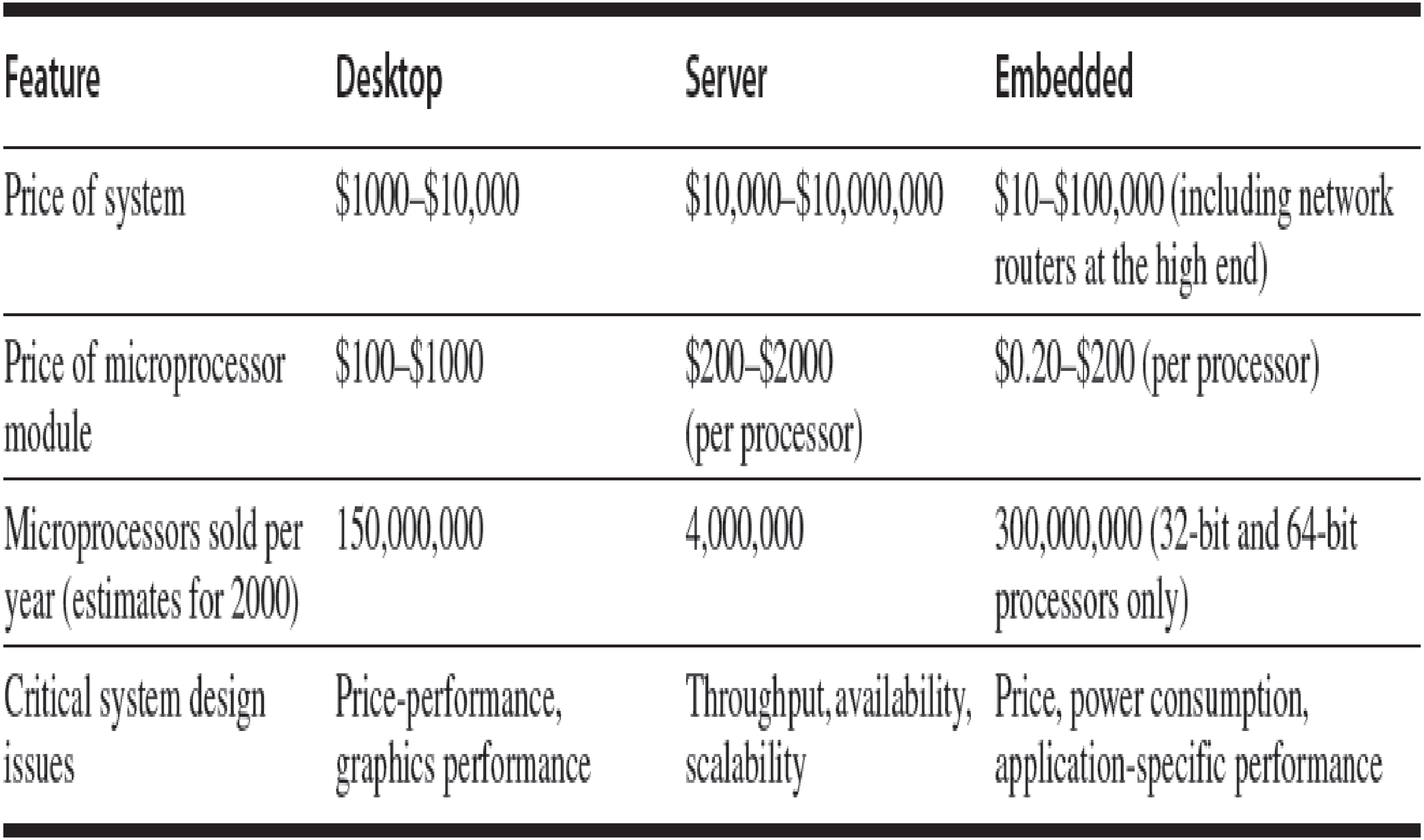
Embedded system

- Embedded problems are usually solved by one of three approaches:
  - The designer uses a combined hardware/software solution that includes some custom hardware and an embedded processor core that is integrated with the custom hardware, often on the same chip.
  - The designer uses custom software running on an off-the-shelf embedded processor.

Embedded system

  - The designer uses a digital signal processor and custom software for the processor. Digital signal processors are processors specially tailored for signal processing applications. We discuss some of the important differences between digital signal processors and general-purpose embedded processors below.

Summary of 3 Computing classes



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

Real-Time Processing

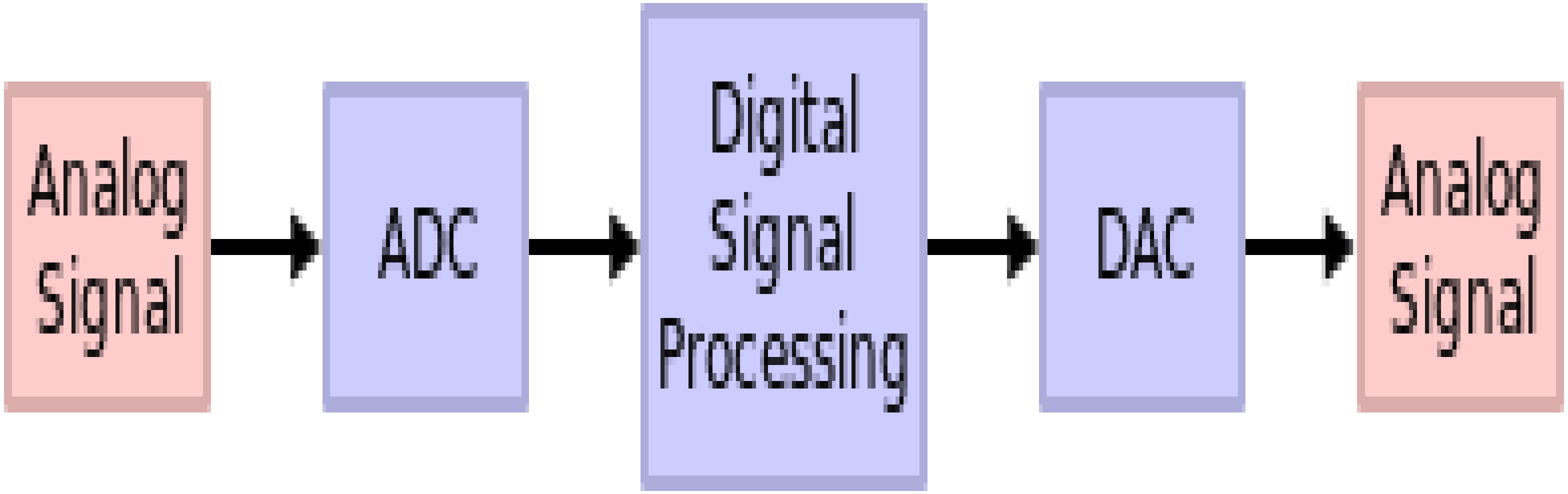
- The performance requirement in an embedded application is a real-time requirement.
- Real-time performance tends to be highly application dependent

Real-Time Processing

- The construction of a hard real-time system involves three key variables.
  - The first is the rate at which a particular task must occur.
  - Coupled to this are the hardware and software required to achieve that real-time rate.
  - Exploit such well-known architectural properties as branch behavior and access locality

Digital signal processor

- A DSP is a specialized microprocessor with an architecture optimized for the operational needs of digital signal processing



DSP Characteristics

- Digital signal processing algorithms typically require a large number of mathematical operations to be performed quickly and repeatedly on a series of data samples.
- Signals are constantly converted from analog to digital, manipulated digitally, and then converted back to analog form.

DSP Characteristics

- Many DSP applications have constraints on latency, the DSP operation must be completed within some fixed time, and deferred processing is not viable.
- Most general-purpose microprocessors and operating systems can execute DSP algorithms successfully, but are not suitable for use in portable devices such as mobile phones and PDAs because of power supply and space constraints.

DSP Characteristics

- A specialized digital signal processor, however, will tend to provide a lower-cost solution, with better performance, lower latency, and no requirements for specialized cooling or large batteries

Architectural features

- Hardware modulo addressing, allows circular buffers to be implemented.
- A memory architecture designed for streaming data
  - using DMA expecting code to be written to know about cache hierarchies and the associated delays.
- Multiple arithmetic units may require memory architectures to support several accesses per instruction cycle

Architectural features

- Separate program and data memories and sometimes concurrent access on multiple data busses
- Special SIMD operations
- Some processors use VLIW techniques
  - each instruction drives multiple arithmetic units in parallel

Architectural features

- Special arithmetic operations, such as fast multiply–accumulates (MACs).
- Many fundamental DSP algorithms, such as FIR filters or the Fast Fourier transform (FFT) depend on multiply–accumulate performance.
- Bit-reversed addressing mode useful for calculating FFTs

Architectural features

- Architectural support for executing a few instruction words in a very tight loop .
- Deliberate exclusion of a memory management unit.
- DSPs frequently use multi-tasking operating systems
- No support for virtual memory or memory protection.
- Operating systems that use virtual memory require more time for context switching among processes, which increases latency.

Architectural features

- Floating-point unit integrated directly into the datapath
- Pipelined architecture
- Highly parallel multiplier–accumulators (MAC)
- Hardware-controlled looping to reduce or eliminate the overhead required for looping operations

Architectural features

- **Memory architecture**
- DSPs often use special memory architectures that are able to fetch multiple data and/or instructions at the same time:
  - Harvard architecture
  - Modified von Neumann architecture
- Use of direct memory access
- Memory-address calculation unit

Architectural features

- **Data operations**
- Saturation arithmetic in which operations that produce overflows will accumulate at the maximum (or minimum) values that the register can hold rather than wrapping around
- Fixed-point arithmetic is often used to speed up arithmetic processing
- Single-cycle operations to increase the benefits of pipelining

Architectural features

Instruction sets

- Multiply–accumulate (MAC, including fused multiply–add, FMA) operations, which are used extensively in all kinds of matrix operations, such as convolution for filtering, dot product, or even polynomial evaluation.
- Instructions to increase parallelism: SIMD, VLIW, superscalar architecture

Architectural features

Instruction sets

- Specialized instructions for modulo addressing in ring buffers and bit-reversed addressing mode for FFT cross-referencing
- Digital signal processors sometimes use time-stationary encoding to simplify hardware and increase coding efficiency.

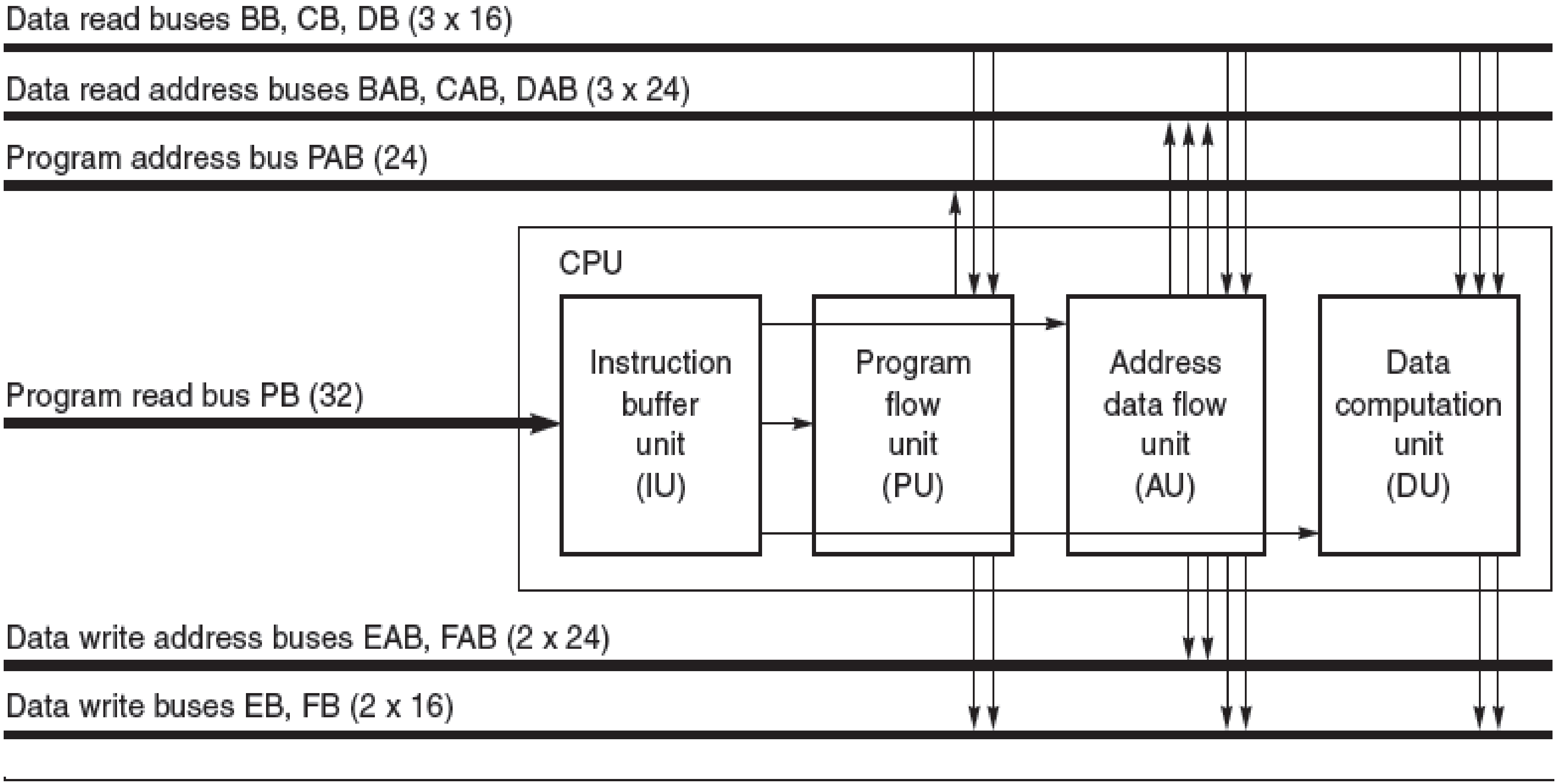
TI 320C55

- The C55 is optimized for low-power, embedded applications.
- The C55 is a seven-staged pipelined CPU
  - Fetch stage reads program data from memory into the instruction bufferqueue.
  - Decode stage decodes instructions and dispatches tasks to the other primaryfunctional units.
  - Address stage computes addresses for data accesses and branch addresses forprogram discontinuities.

TI 320C55

- The C55 pipeline stages
  - Access 1/Access 2 stages send data read addresses to memory.
  - Read stage transfers operand data on the B bus, C bus, and D bus.
  - Execute stage executes operation in the A unit and D unit and
  - performs writes on the E bus and F bus.

TI 320C55


Data read buses BB, CB, DB (3 x 16)
Data read address buses BAB, CAB, DAB (3 x 24)
Program address bus PAB (24)
CPU
Program read bus PB (32)
Instruction buffer unit (IU)
Program flow unit (PU)
Address data flow unit (AU)
Data computation unit (DU)
Data write address buses EAB, FAB (2 x 24)
Data write buses EB, FB (2 x 16)

TI 320C55

- The C55 pipeline hazard detection and will stall on RAW and WAR hazards.
- The C55 does have a 24 KB instruction cache
- It may be configured to be two-way set associative, direct-mapped, or as a “ramset.”
- This latter mode is a way to support hard realtime applications.
- In this mode, blocks in the cache cannot be replaced.

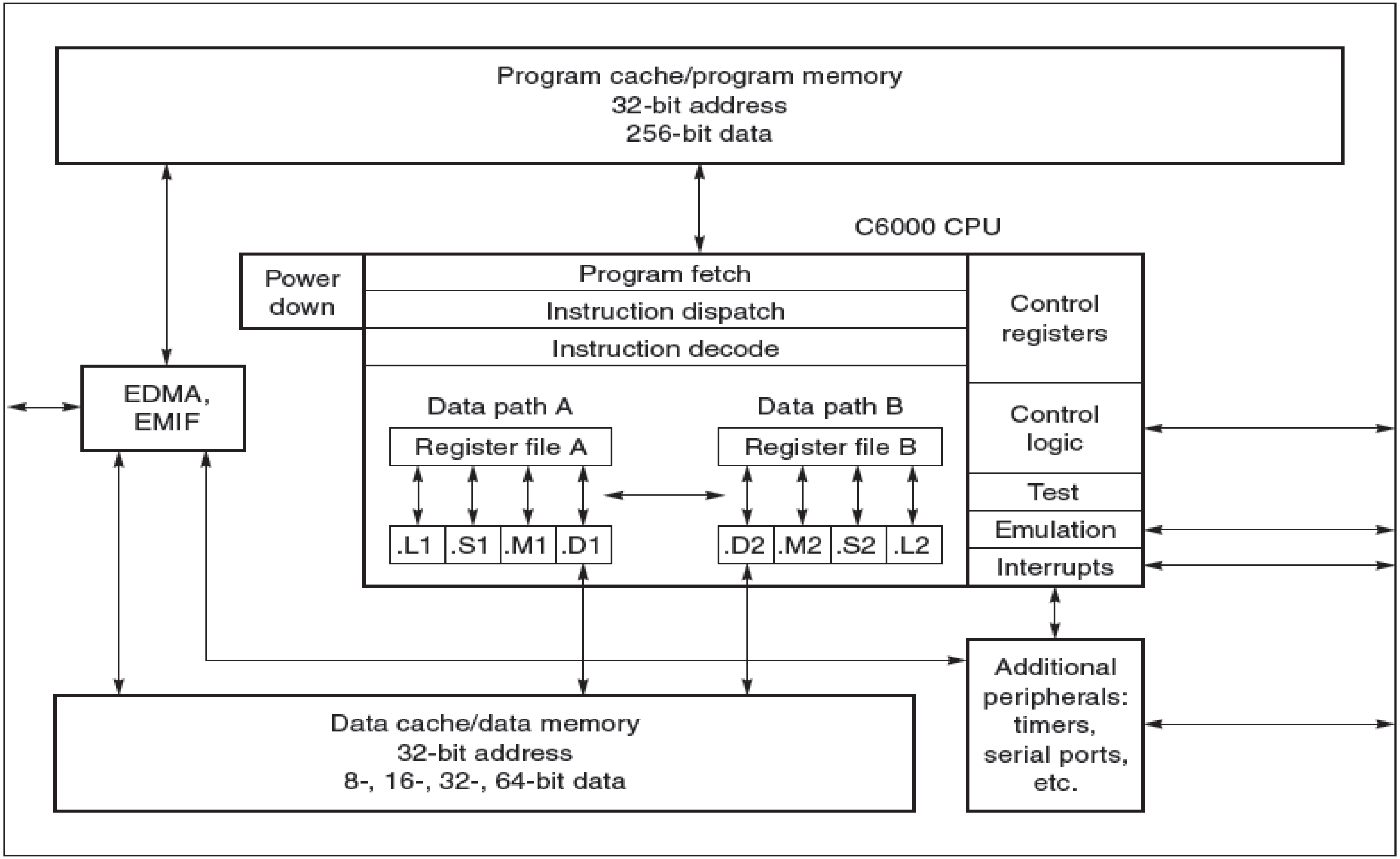
TI 320C55

- The C55 also has advanced power management.
  - It allows dynamic power management through software-programmable “idle domains.”
  - Blocks of circuitry on the device are organized into these idle domains.
  - Each domain can operate normally or can be placed in a low-power idle state.
  - A programmer accessible Idle Control Register (ICR) determines which domains will be placed in the idle state

TI 320C55

- It has six domains
  - CPU
  - DMA
  - peripherals
  - clock generator
  - instruction cache
  - external memory interface.
- When each domain is in the idle state, the functions of that particular domain are not available.

TI 320C6x


Program cache/program memory
32-bit address
256-bit data
C6000 CPU
Power down
Program fetch
Instruction dispatch
Instruction decode
Control registers
EDMA, EMIF
Data path A
Data path B
Register file A
Register file B
Control logic
Test
.L1
.S1
.M1
.D1
.D2
.M2
.S2
.L2
Emulation
Interrupts
Additional peripherals: timers, serial ports, etc.
Data cache/data memory
32-bit address
8-, 16-, 32-, 64-bit data

TI 320C6x

- High-end Texas Instruments family of processors
- C6x processors VLIW processors
- They seek to exploit the high levels of ILP
- C6x family employs different pipeline depths depending on the family member.
  - For the C64x, the pipeline has 11 stages.
- C6x family’s execution stage is divided into two parts
  - the left or “1” side and the right or “2” side.

TI 320C6x

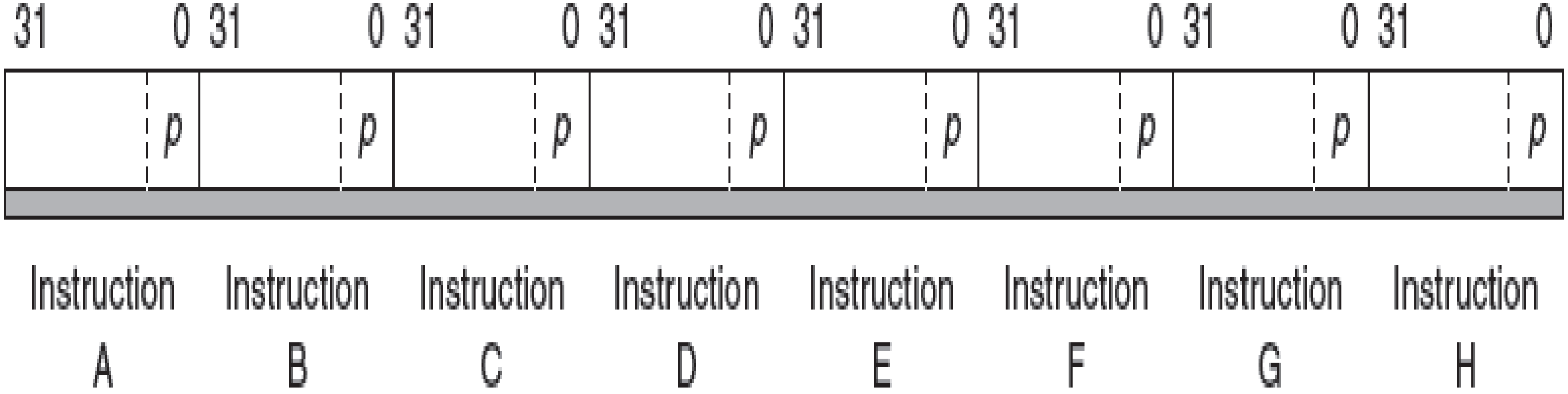
- The L1 and L2 units perform logical and arithmetic operations.
- D units in contrast perform a subset of logical and arithmetic operations but also perform memory accesses (loads and stores).
- The two M units perform multiplication and related operations (e.g., shifts).
- Finally the S units perform comparisons, branches, and some SIMD.

TI 320C6x

- Each side has its own 32-entry, 32-bit register file (the A file for the 1 side, the B file for the 2 side).
- A side may access the other side’s registers, but with a 1- cycle penalty.

TI 320C6x

- Each instruction has a p bit that specifies whether this instruction is a member of the current VLIW word or the next VLIW word



TI 320C6x

- The p bits determine whether an instruction begins a new VLIW word or not.
- If the p bit of instruction i is 1, then instruction i + 1 is to be executed in parallel with (in the same cycle as) instruction i.
- If the p bit of instruction i is 0, then instruction i + 1 is executed in the cycle after instruction *i.*

TI 320C6x

- Software pipelining is an important technique for achieving high performance in a VLIW.
- Software pipelining relies on each iteration of the loop having an identical schedule to all other iterations.
- C6x family provides a means to conditionally execute instructions using predication.

Embedded Multiprocessors

- These multiprocessors have been focused primarily on
  - high-end telecommunications
  - networking market where scalability is critical.
  - example MXP processor designed by empowerTel Networks for use in voiceover- IP systems.

Embedded Multiprocessors

- MXP processor consists of four main components:
  - An interface to serial voice streams, including support for handling jitter.
  - Support for fast packet routing and channel lookup
  - A complete Ethernet interface, including the MAC layer
  - Four MIPS 32 R4000-class processors, each with its own cache (a total of 48 KB or 12 KB per processor)

Embedded Multiprocessors

- Multiprocessing is becoming widespread in the embedded computing arena for two primary reasons.
  - First, the issues of binary software compatibility. Often software in an embedded application is written from scratch for an application (note that this is also the reason VLIW is favored over superscalar in embedded ILP).
  - Second, the applications often have natural parallelism, especially at the high end of the embedded space.