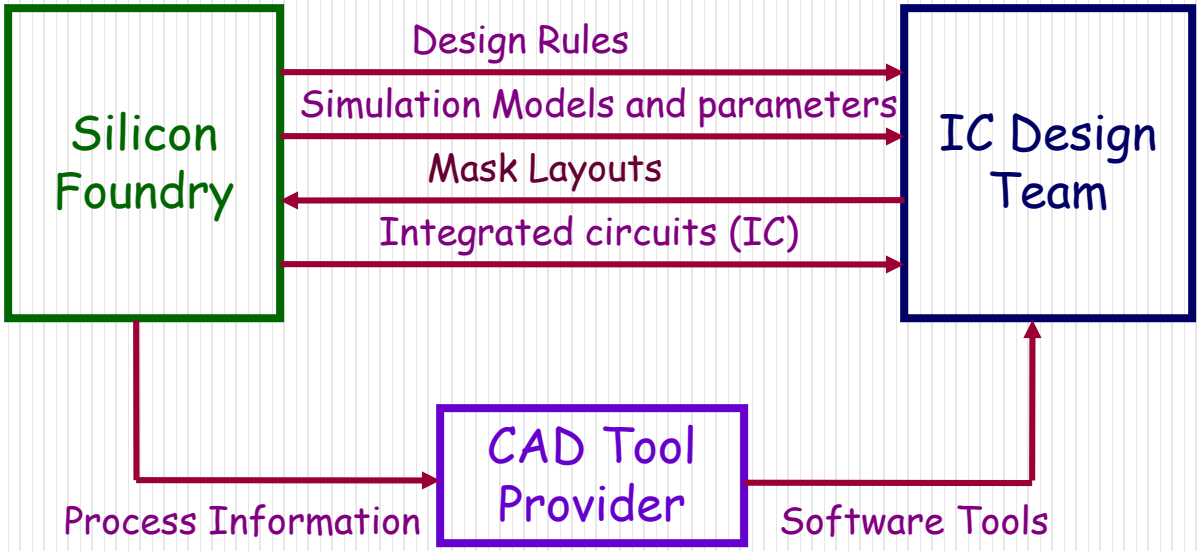
CSEN 1040 Advanced ES

Architectural Choices

Prof. Ayman Elnaggar

VLSI Design Methodology



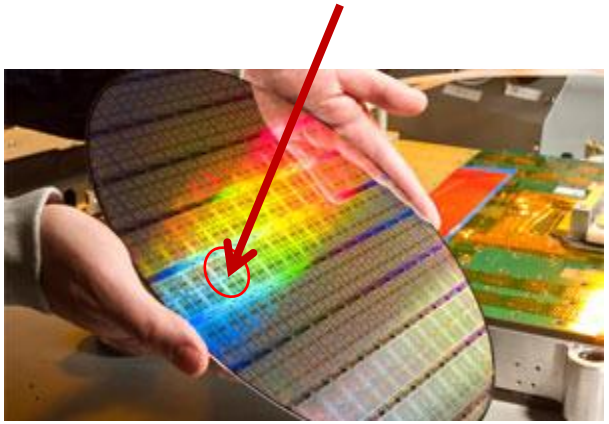
Relationship between a silicon foundry, an IC design team and a CAD tool provider

How Does an IC Fabrication Lab look like?

IC Manufacturing

- ICs are typically created on larger circular sheets of silicon called "wafers"
- A wafer is typically 100-300 mm in diameter, and about 4 mm thick
- A large silicon circuit is about 1 cm long
- In order to fabricate a number of chips, the wafer passes through thousands of steps including: thermal, chemical, cleaning, … etc
- Each IC is tested before and after "packaging"

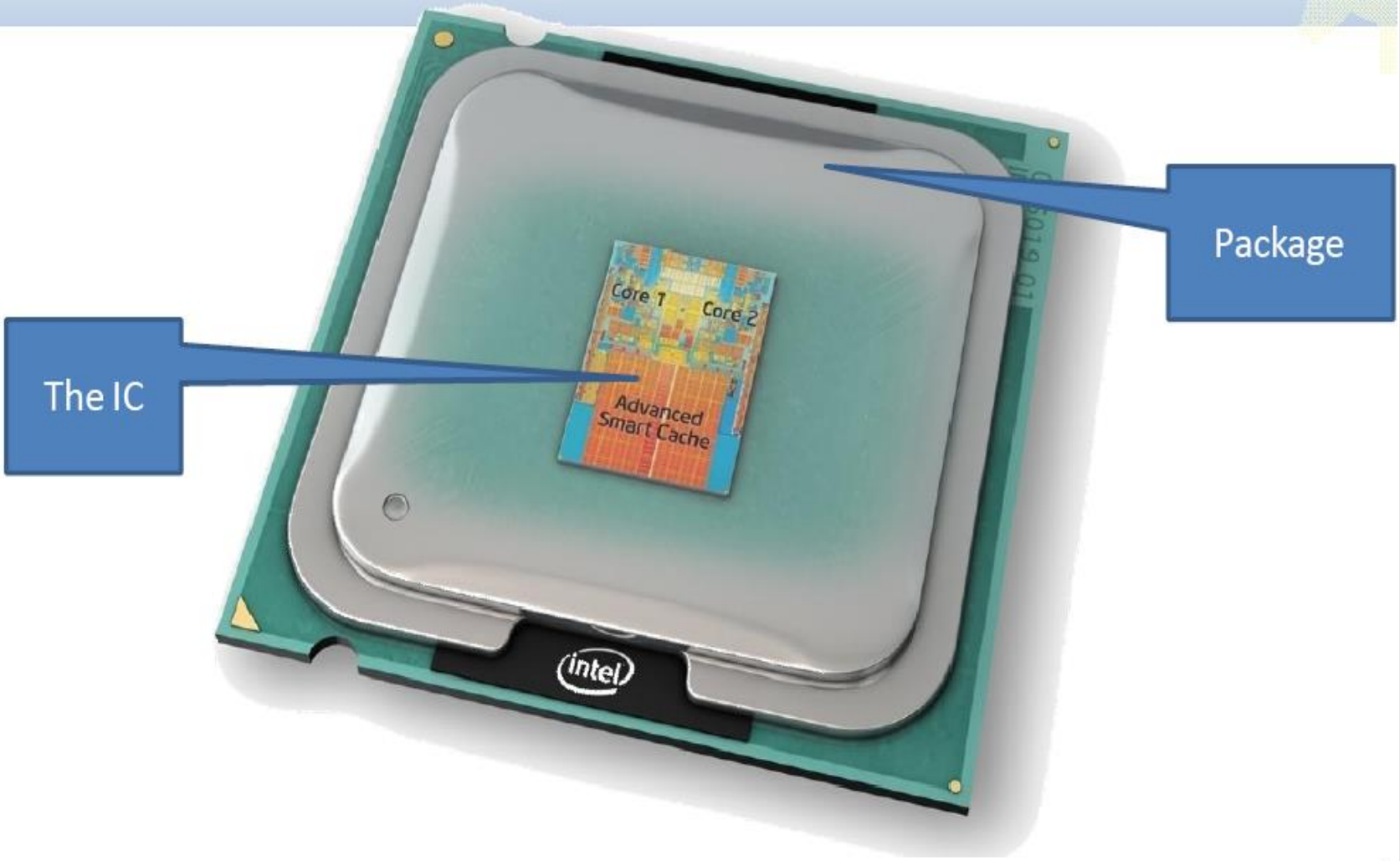
A Single "Die" or "Chip"



A 300mm IBM Power-7 (8-Core) Server-Class Processor Wafer

What does an IC look like?

Intel dual core

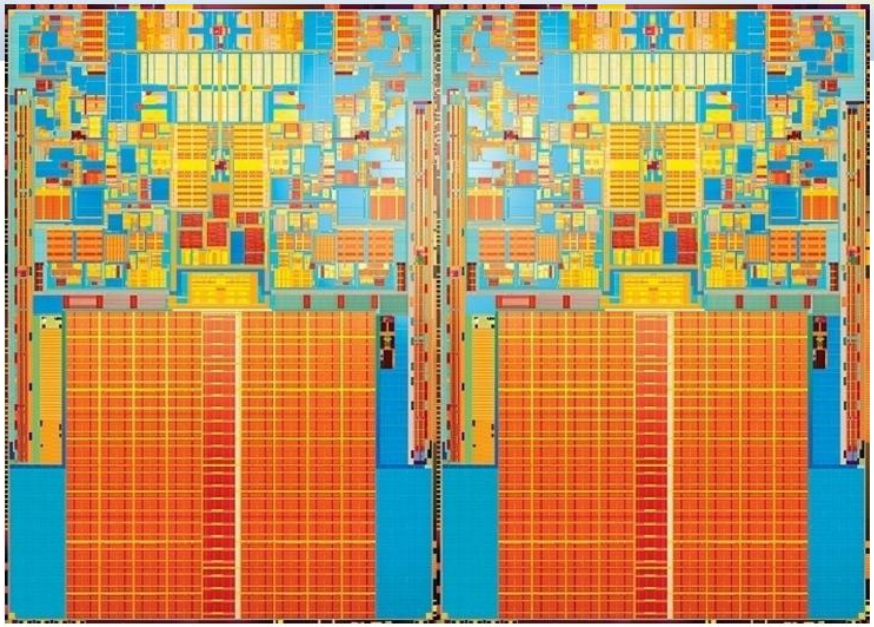


What does an IC look like?

45 nm, quad-core

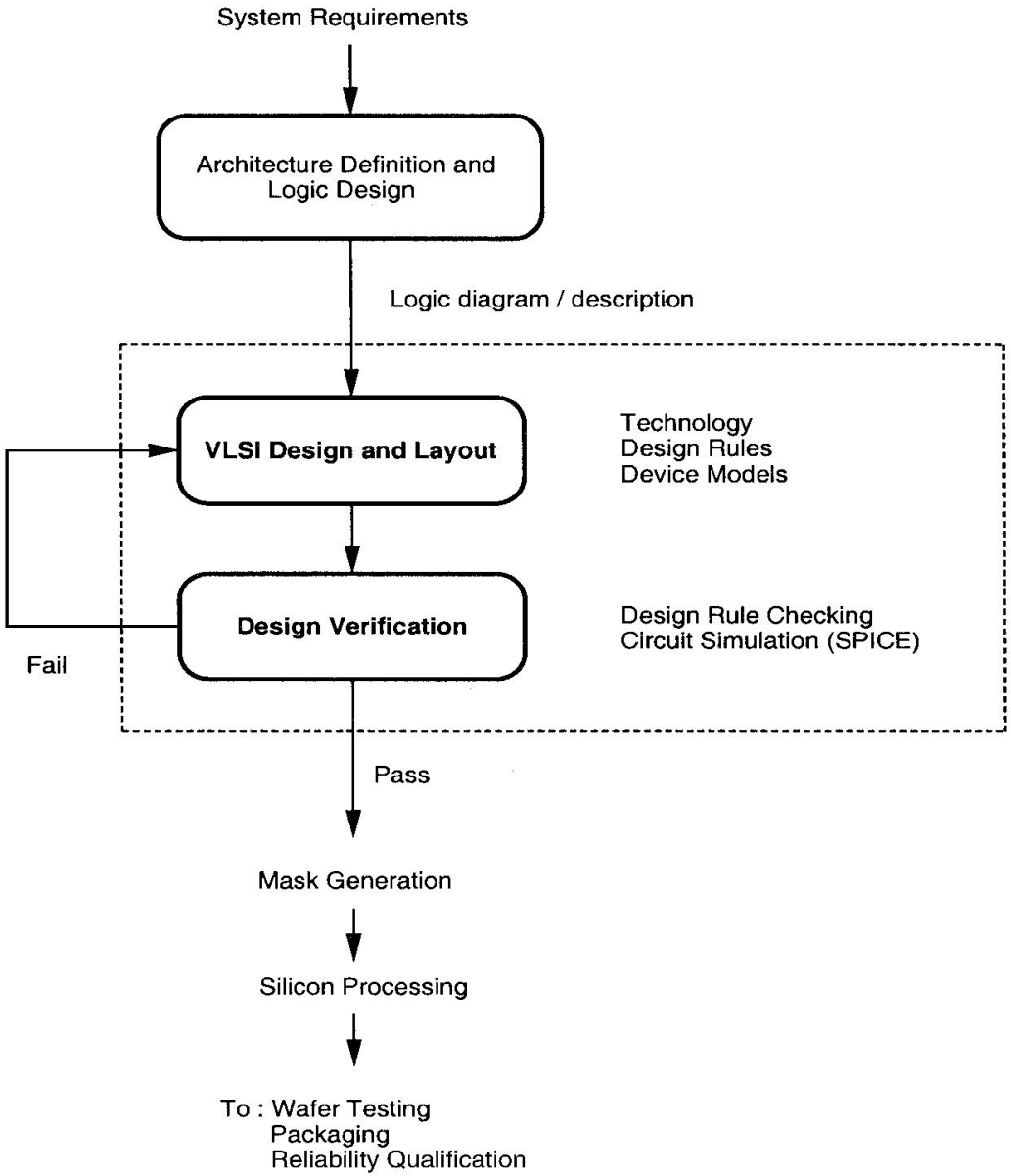
Note the symmetry

Two dual-cores integrated



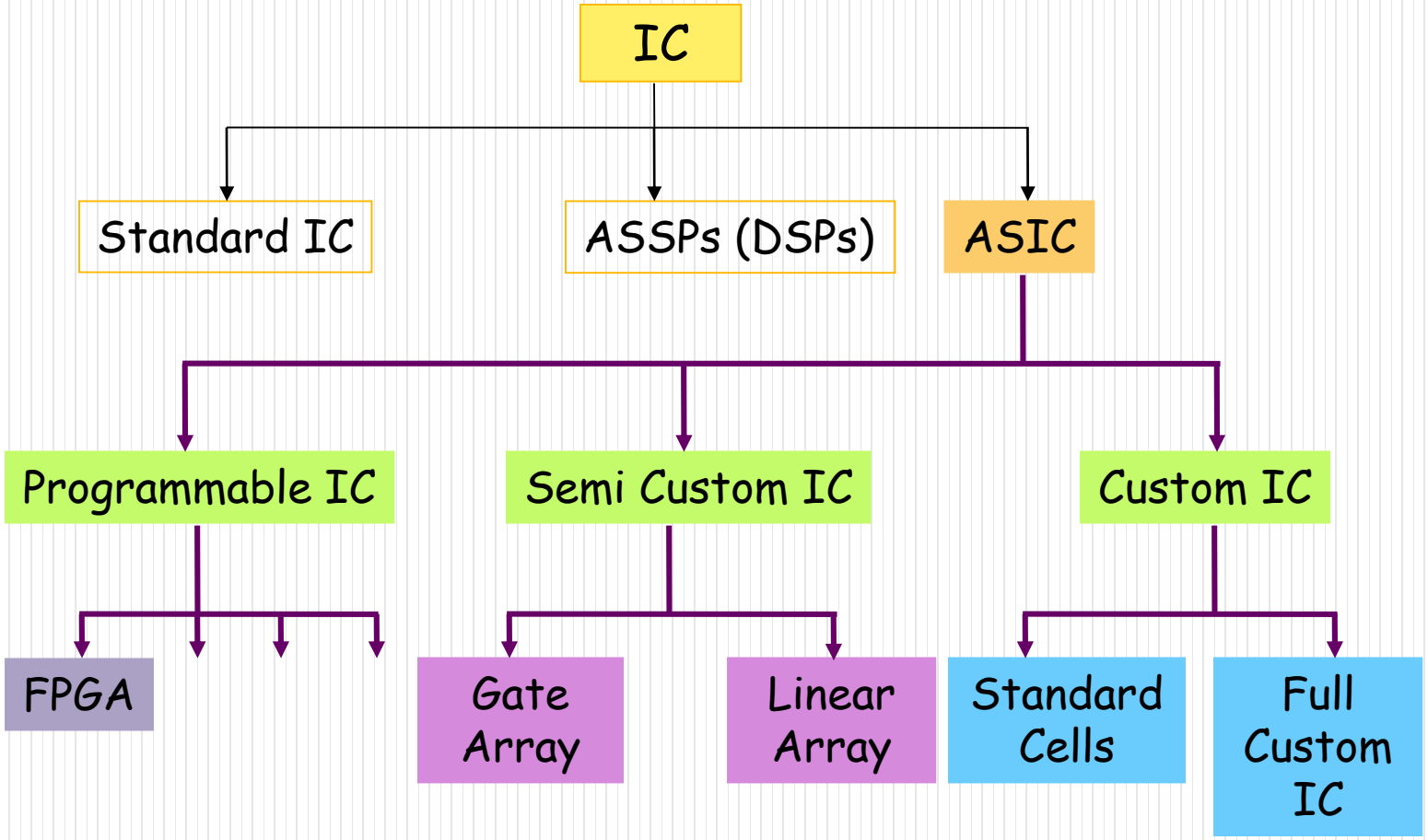
Pinless Chip!!

Bottom view showing the gold contacts



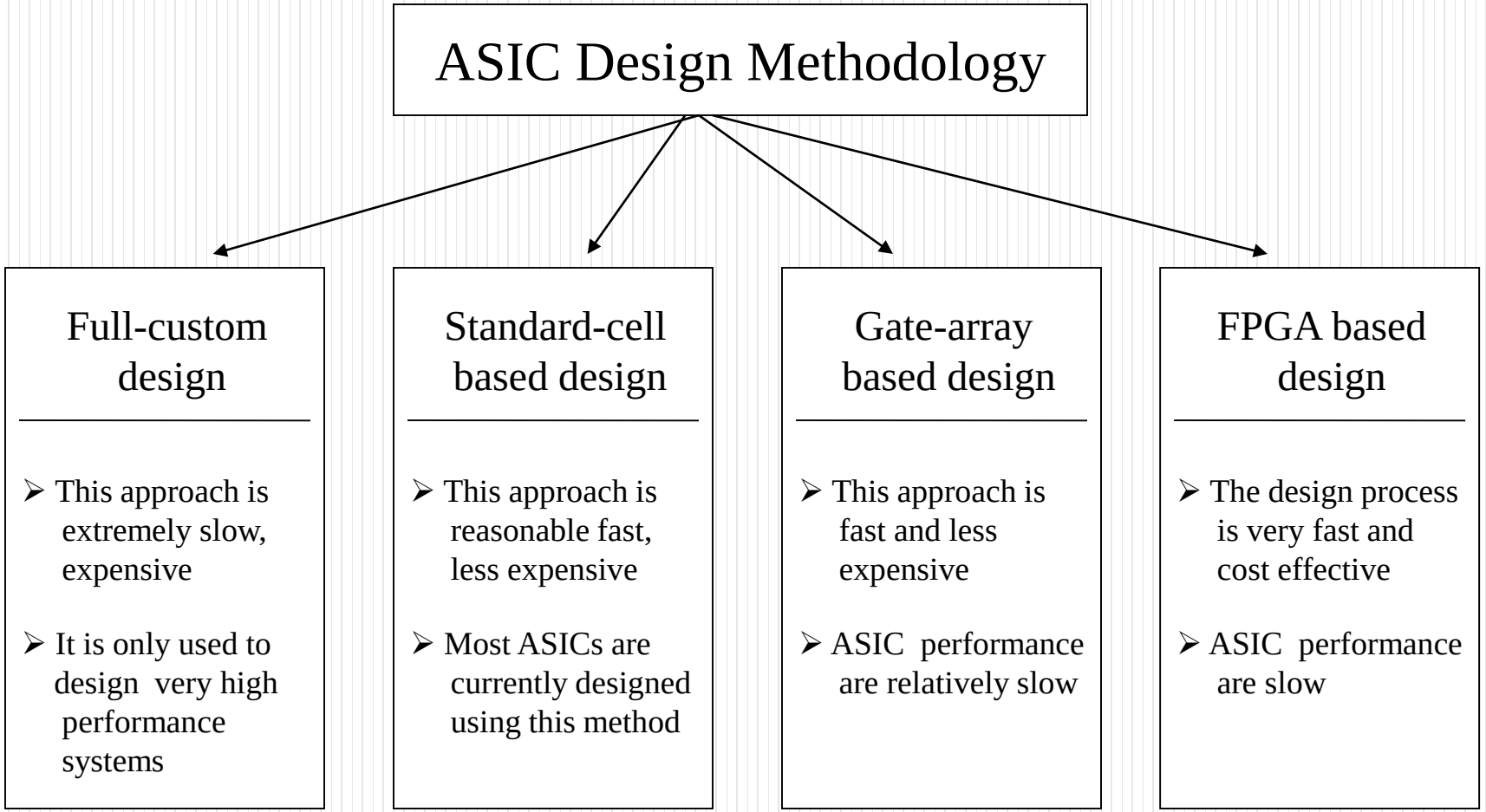
The flow of circuit design procedures.

Application Specific Integrated Circuits (ASICs)



ASSPs : *Application Specific Standard Products*

ASIC Design Methodologies

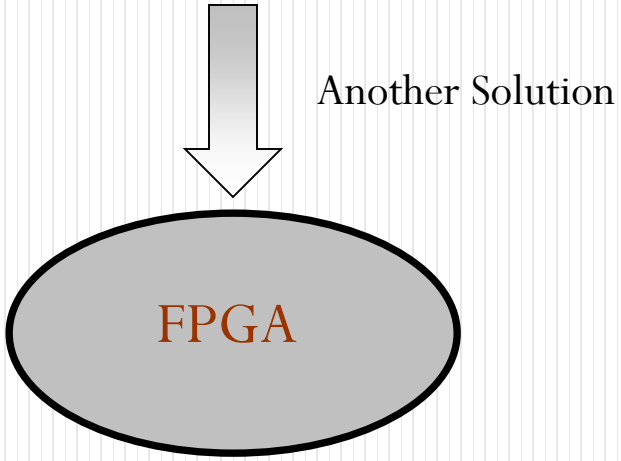


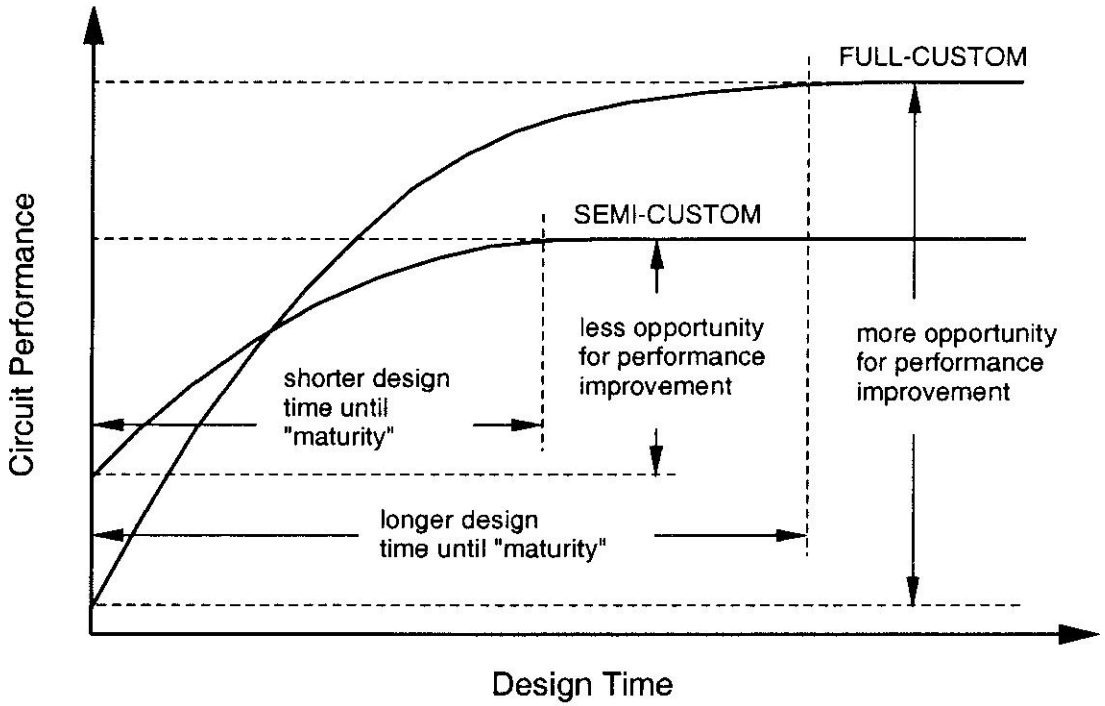
Custom ICs-Benefit

- Improve cost/performance
- Reduce power consumption
- Mixed Analog and Digital Designs
- Design optimization through IC manufacturing process
- Development Tools support HDL and Schematic design approach

Custom ICs -Drawbacks

- Large development cost
- Inflexible design
- Deployed systems can not be upgraded
- Mistakes in product development are costly
- Updates requires a redesign
- Complex and expensive development tools





Impact of different VLSI design styles upon the design cycle time and the achievable circuit performance.

Why Programmable Logic?

- Facts:
  - It is most economical to produce an IC in large volumes
  - Many designs required only small volumes of ICs
- Need an IC that can be:
  - Produced in large volumes
  - Handle many designs required in small volumes
- A programmable logic part can be:
  - made in large volumes
  - programmed to implement large numbers of different low-volume designs

Programmable Logic - Additional Advantages

- Many programmable logic devices are *field- programmable*, i. e., can be programmed outside of the manufacturing environment
- Most programmable logic devices are *erasable* and *reprogrammable*.
  - Allows "updating" a device or correction of errors
  - Allows reuse the device for a different design - the ultimate in re-usability!
  - Ideal for course laboratories
- Programmable logic devices can be used to prototype design that will be implemented for sale in regular ICs.
  - Complete Intel Pentium designs were actually prototype with specialized systems based on large numbers of VLSI programmable devices!

Programming Technologies

- Programming technologies are used to:
  - Control connections
  - Build lookup tables
  - Control transistor switching
- The technologies
  - Control connections
    - Mask programming
    - Fuse
    - Antifuse
    - Single-bit storage element

Programming Technologies

- The technologies (continued)
  - Build lookup tables
    - Storage elements (as in a memory)
  - Transistor Switching Control
    - Stored charge on a floating transistor gate
      - Erasable
      - Electrically erasable
      - Flash (as in Flash Memory)
    - Storage elements (as in a memory)

Technology Characteristics

- Permanent - Cannot be erased and reprogrammed
  - Mask programming
  - Fuse
  - Antifuse
- Reprogrammable
  - Volatile - Programming lost if chip power lost
    - Single-bit storage element
  - Non-Volatile
    - Erasable
    - Electrically erasable
    - Flash (as in Flash Memory)

Technology Characteristics

- Reprogrammable (Cont.)
  - Build lookup tables
    - Storage elements (as in a memory)
  - Transistor Switching Control
    - Stored charge on a floating transistor gate
      - Erasable
      - Electrically erasable
      - Flash (as in Flash Memory)
    - Storage elements (as in a memory)

Technology Breakthrough Impacts Windows-Compatible Hardware Market

Field Programmable Gate Arrays (FPGAs) break through price/capability barriers

- 1 million gates drop from $200+ to <$10
- 90nm will offer another >2X improvement
- 32 bit RISC processors "for free"

Embedded "soft" 32-bit processors debut allowing complete System-on-Chip designs

Previously relegated to high-margin/low-volume applications

Why Should We Care?

- Cost-effective FPGAs Enable New Windows-Compatible Products
- Greater product differentiation
- Functionality and performance never available at this price point
- Shorter development cycles = faster time to market
- Improved product flexibility = longer market life
- Reduced part inventory
- More product variants

New Low-Cost Technologies

FPGA Families

Altera: Cyclone

Xilinx: Spartan 3

QuickLogic: QuickMIPS, Eclipse II

Actel: ProASIC Plus

Embedded Processors

Altera: Nios

Xilinx: MicroBlaze

QuickLogic: MIPS

Actel: 8051

Full-custom ICs Versus FPGA Comparison

- Tooling cost
- Non-recurring engineering costs (NRE)
- Time to market
- Product risk
- Product flexibility
- Inventory

Non-Recurring Engineering (NRE)

- Full-Custom Designs
  - NRE is very high
  - Each subsequent re-spin costs another NRE
  - For new 90nm technology NRE >$1M
  - High-risk methodology requiring massive volume to recoup costs
- FPGA Designs
  - No NRE charges
  - Some cost-reduction available by ASIC conversion with minimal (<$100K) NRE
- Cost of FPGA device is offset by NRE in all but the highest volume applications

Time To Market

- Full-Custom Designs
  - Typical design cycle 12-18 months, minimum 9 months
  - Additional re-spins add 8-10 weeks each
- FPGA Designs
  - Typical design cycle 4 months
  - Re-spins not an issue

Extremely Fast Time-To-Prototype

- Software-based prototype in days
  - System assembly on FPGA in minutes with processor, memory, bus, interfaces, peripherals
  - C/C++ based application (with RTOS if needed) running in minutes using actual hardware
- Iterative development/refinement flow
  - Performance-critical routines easily migrated to "hardware" implementations
  - Software development in parallel with working hardware
  - Embedded "virtual instrumentation" offers in-circuit debugging
  - Allows evolutionary design style

Benefits Of FPGA-Based Design

- Improved product flexibility
  - Changes hardware/software up to (and even after) deployment
  - Inventoried parts can be re-deployed in multiple applications
  - More product variants on single platform
  - Upgrade/enhance in the field
- Reduce inventory
  - Single part for multiple variations and versions of product

What's Needed For FPGA-Based Product Development?

- Complete toolsets provided by FPGA vendors
- Robust libraries of pre-tested IP components
  - Processor cores (8,16,32 bit configurable)
  - Peripherals (USB, PCI, I/O, DSP, ethernet, Memory…)
- Ready-to-use development prototyping boards for a variety of application types
- Development environments are PC/Windows-based, no UNIX workstations required

DSP Processors
Architecture and Examples

What is Real-Time Digital Signal Processing?

Time-constrained Operation or Transformation performed on *digital* signals within a required period of time to maintain synchronization with occurring events.

- Example:
  - Processor clocked at 120 MHz and can perform 120MIPS
    - Sampling rate = 48KHz (Digital Audio Tape - DAT) number of instructions per sample = $(120 \times 10^6)/(48 \times 10^3) = 2500$.
    - Sampling rate = 8KHz (voice-band, telephony) number of instructions per sample = 15000.
    - Sampling rate = 75MHz (CIF 360x288 Video at 30 frames per second) number of instructions per sample = 1.6.

Real-Time Digital Signal Processing

- Constraints:
  - real-time DSP applications limited to cases where the required sampling rate is sufficiently lower than the processor's instruction rate
- Challenge:
  - Produce working code.
  - Produce sufficiently compact code to execute in real-time.
  - A sufficient number of instructions need to be performed between sample periods.

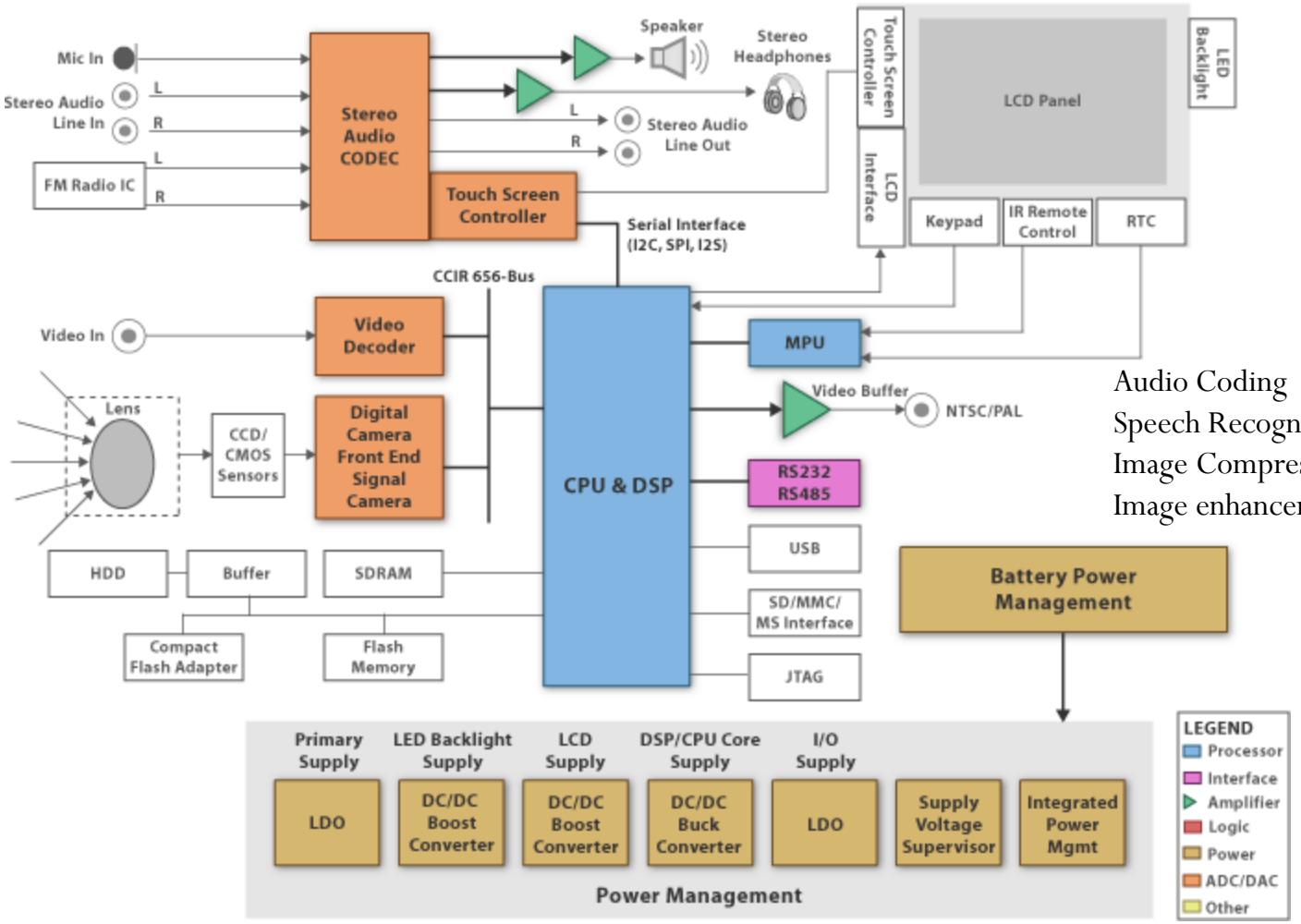
What are DSP Processors?

- Microprocessor specifically designed to perform fast DSP operations (e.g., Fast Fourier Transforms, inner products, Multiply & Accumulate)

Why Go Digital?

- Programmability
  - One hardware can perform several tasks.
  - Upgradeability and flexibility.
- Repeatability
  - Identical performance from unit to unit.
  - No drift in performance due to temperature or aging.
- Immune to noise
- Offers higher performance : CD players versus phonographic turntable

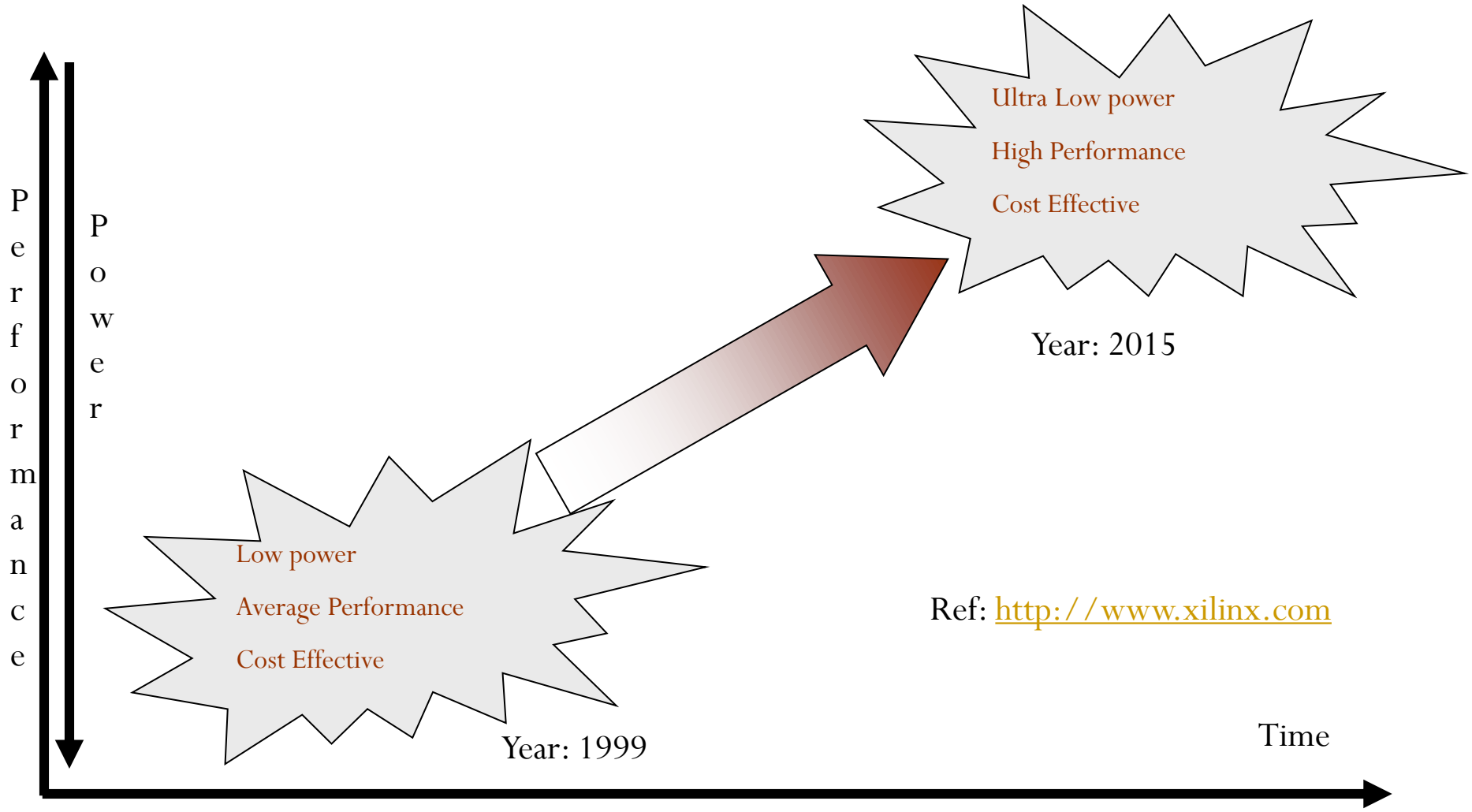
TYPICAL PORTABLE MEDIA DEVICES



Audio Coding
Speech Recognition
Image Compression
Image enhancement

Web Link: http://focus.ti.com/vf/docs/blockdiagram.tsp?blockDiagramId=6046&appId=267

Portable Applications – Need High Performance Processors



Ref: http://www.xilinx.com

What is Special about Signal Processing Applications?

- Large number of samples being continuously fed to the system (samples or blocks).
- Repetitive Operations:
  - The same operation being applied to different set of samples
  - Parallel processing
- Vector and Matrix Operations
- Real time operations

Example: Digital Filtering

- The two most common real-time digital filters are:
  - Finite Impulse Filter (FIR)
  - Infinite Impulse Filter (IIR)
- The basic FIR Filter equation is

$$y[n] = \sum h[k].x[n-k]$$

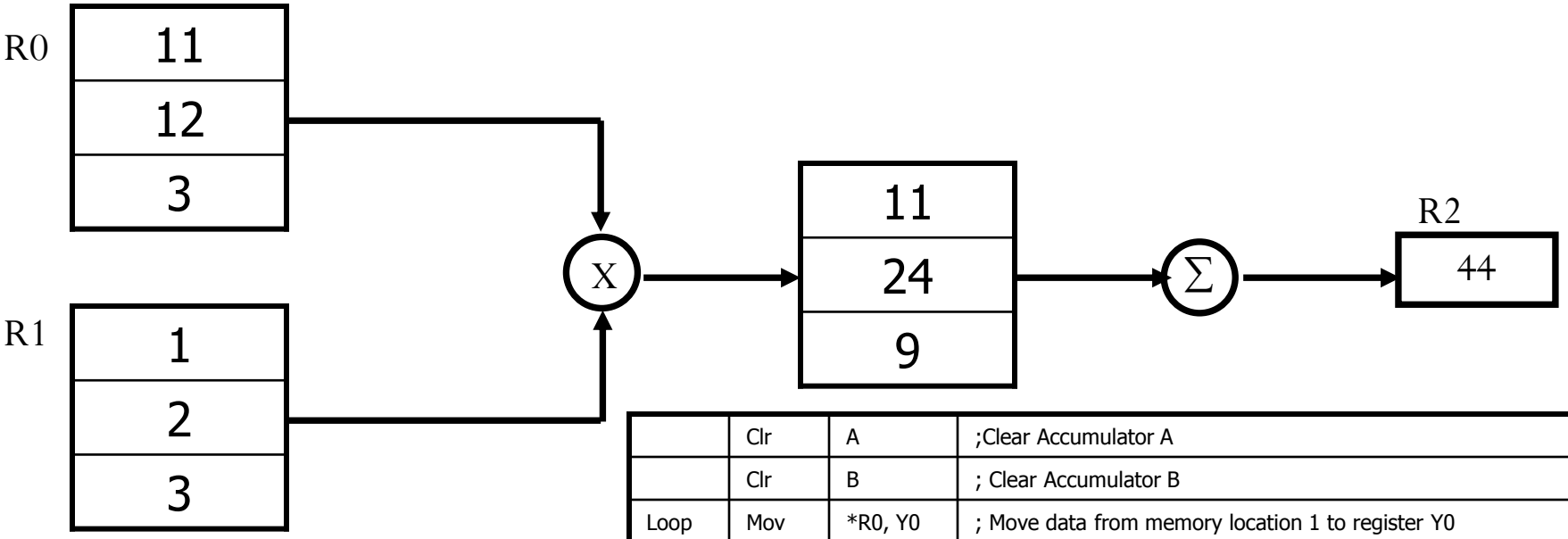
where h[k] is an array of constants

In C language

```
y[n]=0;
For (n=0; n<N;n++)
{
For (k = 0;k<N;k++)
//inner loop
                y[n] = y[n] + h[k]*x[n-k];}
```

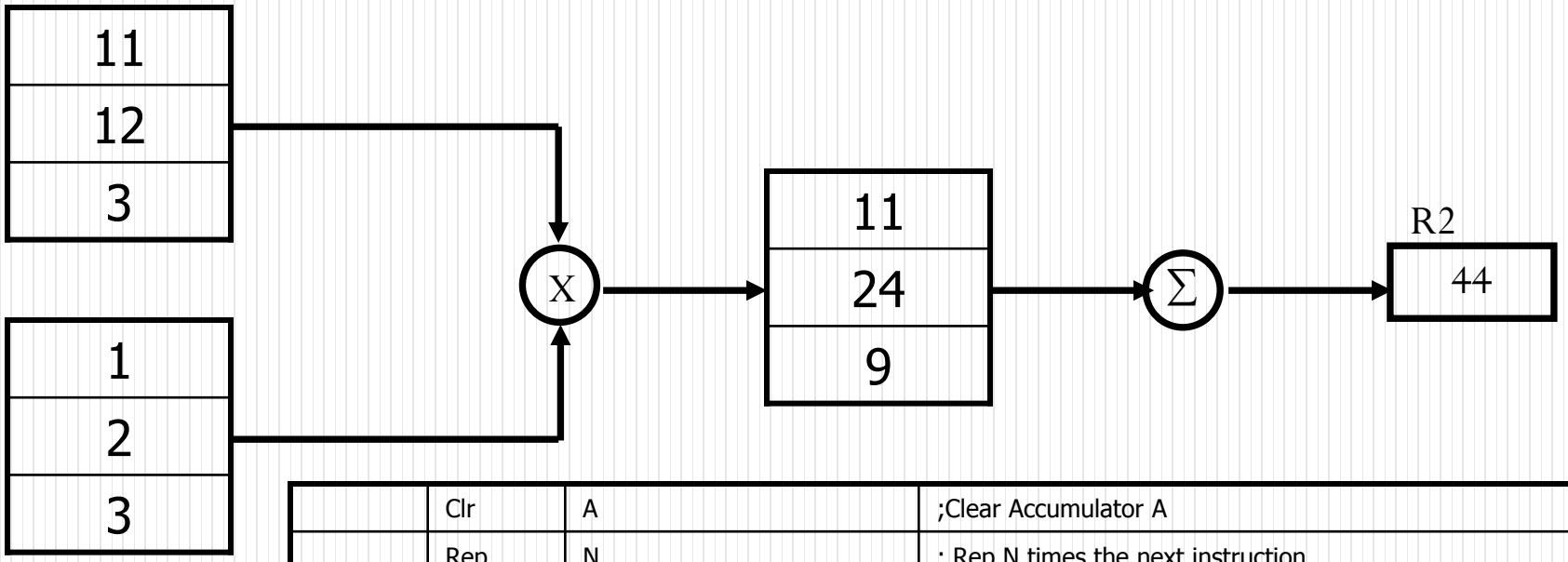
Only Multiply and Accumulate (MAC) is needed!

Using General Purpose Processor (GPP)



	Clr	A	;Clear Accumulator A
	Clr	B	; Clear Accumulator B
Loop	Mov	*R0, Y0	; Move data from memory location 1 to register Y0
	Mov	*R1,X0	; Move data from memory location 2 to register X0
	Mpy	X0,Y0,A	;X0*Y0 ->A
	Add	A,B	;A + B -> B
	Inc	R0	;R0 + 1 -> R0
	Inc	R1	;R1 + 1 -> R1
	Dec	N	;Dec N (initially equals to 3)
	Tst	N	;Test for the value
	Jnz	Loop	;Different than zero loop again
	Mov	B,*R2	;Move result to memory

Using DSP



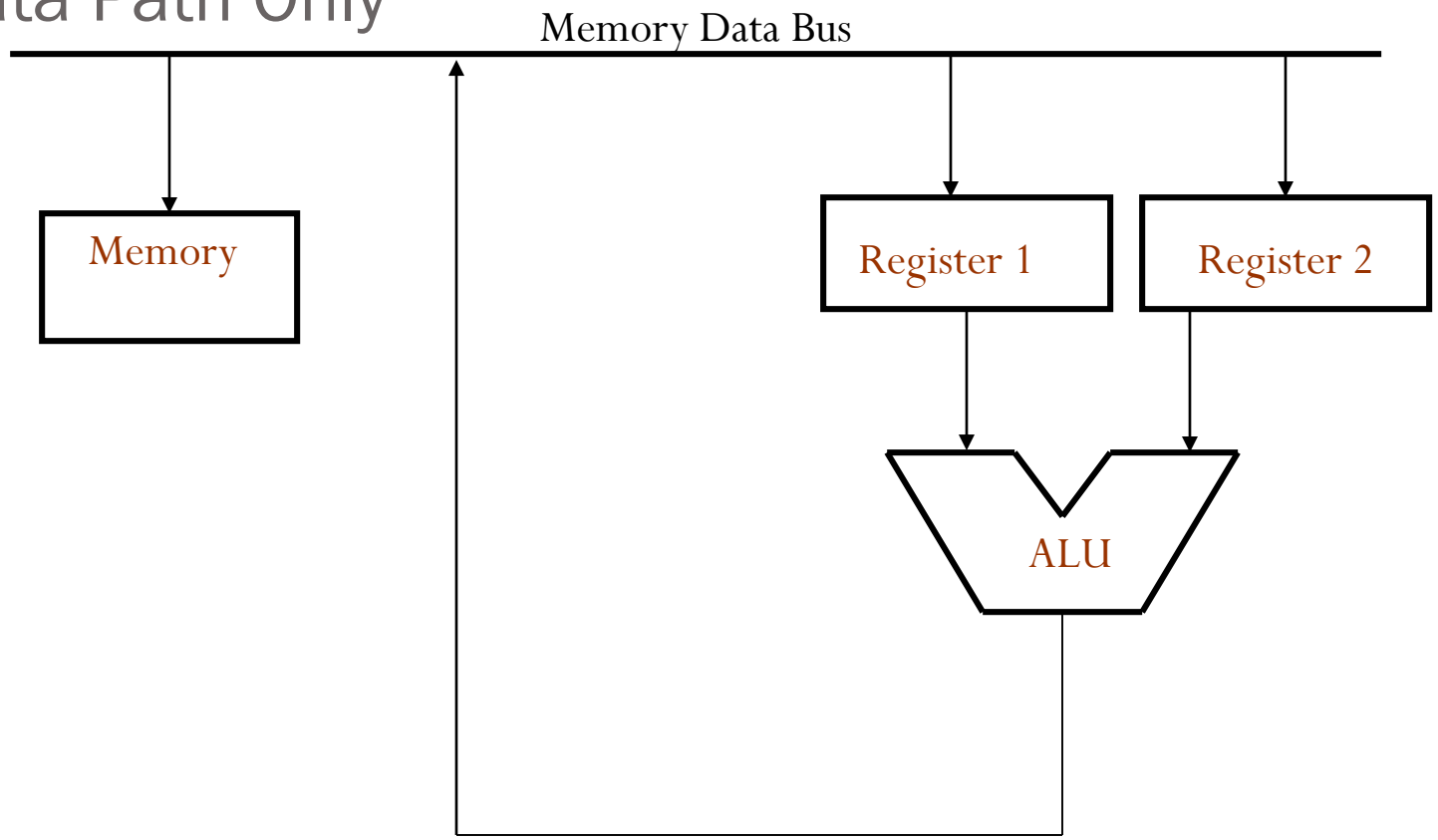
	Clr	A	;Clear Accumulator A
	Rep	N	; Rep N times the next instruction
	MAC	*(R0)+, *(R1)+, A	; Fetch the two memory locations pointed by R0 and R1, multiply them together and add the result to A, the final result is stored back in A
	Mov	A, *R2	; Move result to memory

GPP Drawbacks

- More instructions/task
- Common Memory for data and program
  - Limited bus/memory bandwidth

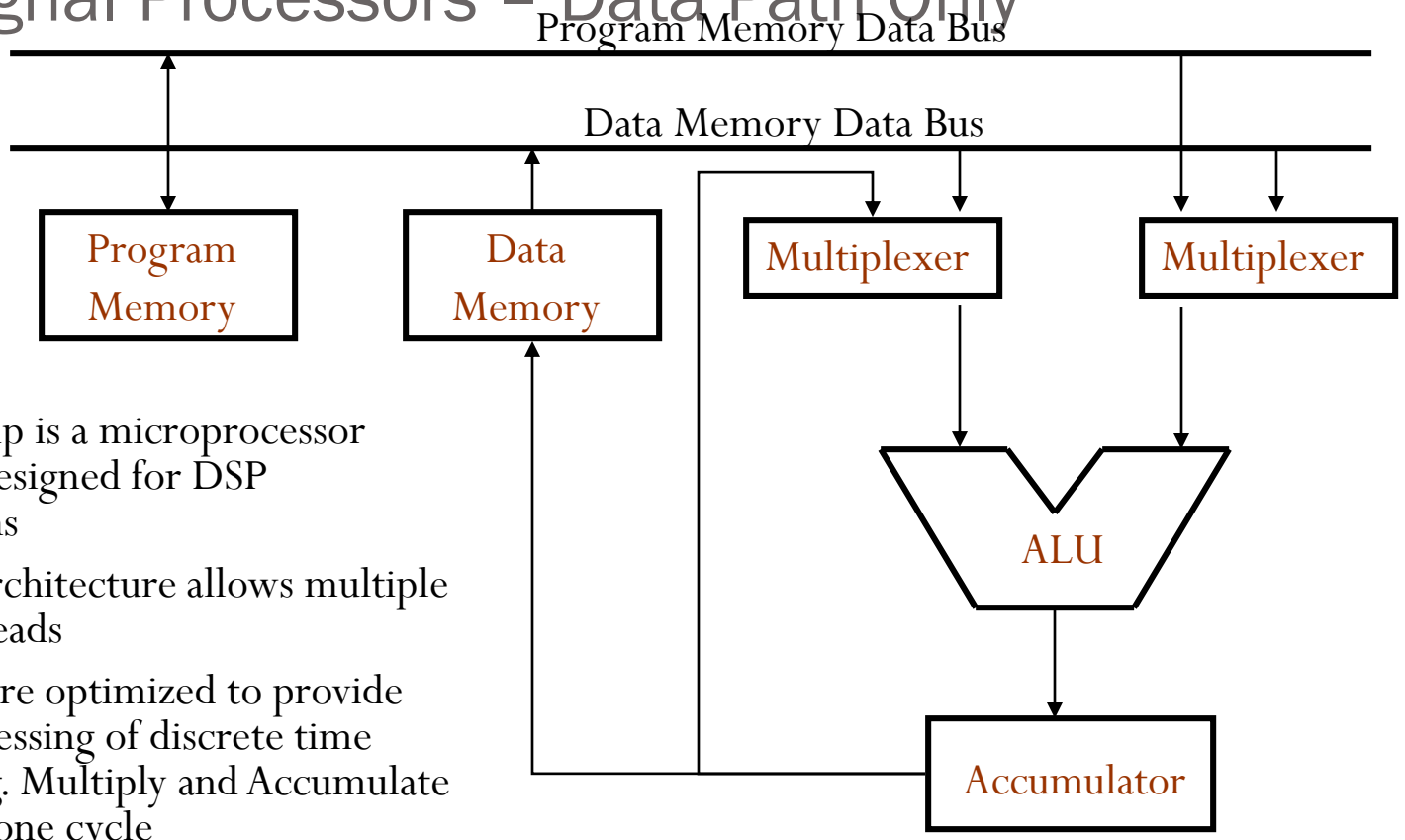
Solution : DSP Architectures

GPP – Data Path Only



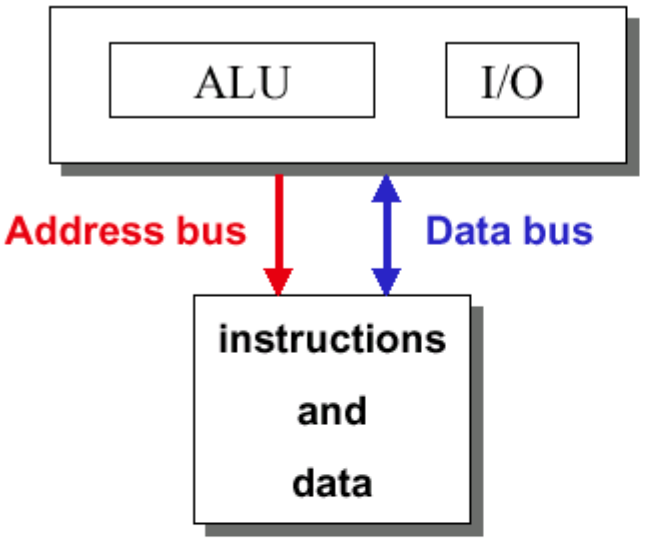
Same memory for program and data

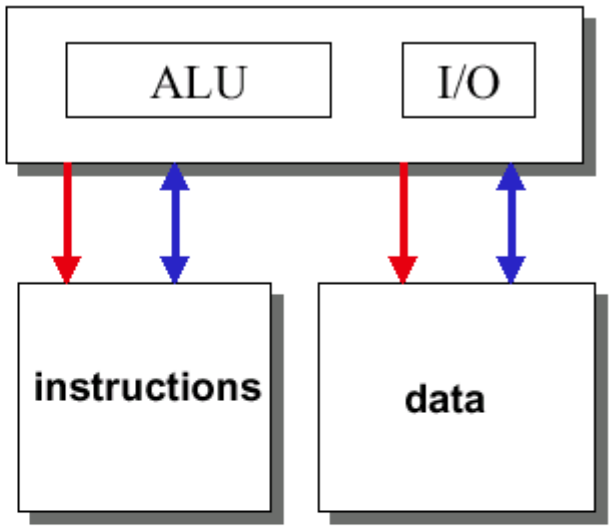
Digital Signal Processors – Data Path Only



- A DSP Chip is a microprocessor specially designed for DSP applications
- Harvard architecture allows multiple memory reads
- Architecture optimized to provide rapid processing of discrete time signals, e.g. Multiply and Accumulate (MAC) in one cycle

Memory structures





Von Neuman architecture **Area efficient but requires higher bus bandwidth because instructions and data must compete for memory.**

Harvard architecture **was coined to describe machines with separate memories. Speed efficient: Increased parallelism.**

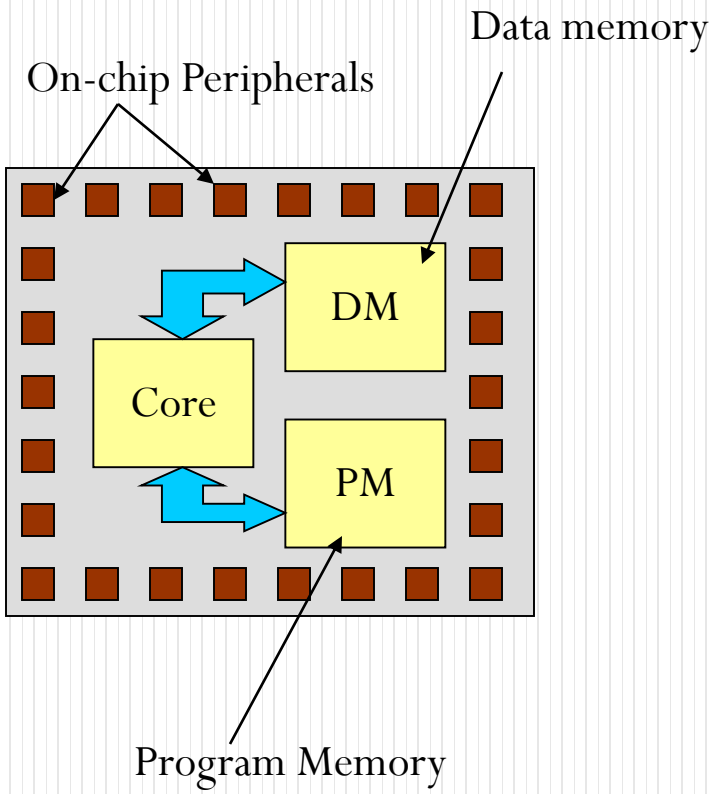
DSP versus GPP

- Multiple parallel units
  - multiply accumulate (possibly several units)
  - address calculation in parallel to processing
  - barrel shifter
- Memory Access
  - special ALU for address calculation
  - Bit reversed addressing
  - circular addressing
- Automatic loops
  - Software looping: writing assembly code to perform branching
  - Hardware looping: dedicated hardware loop counter register
- Hardware support for managing arithmetic computation (in GPP it needs multiple cycles)
  - Shifters
  - Guard bits
  - Saturation

Preventing Overflow!!

Digital Signal Processor (DSP) - Overview

- DSP Core includes:
  - Address buses
  - Data buses
  - Data arithmetic logic unit (ALU)
  - Address generation unit (AGU)
  - Program controller
  - Bit-manipulation unit
  - Enhanced debugging module
- Peripherals on chip
  - Timer
  - serial link
  - communication links
    - DSP to DSP
    - Ethernet
    - ATM
  - host ports
  - input/output pins
- Adaptation for FFT
  - bit reverse addressing
- Special instructions
  - Parallel move support
  - Loop instructions; special hardware instructions (e.g., FIR)



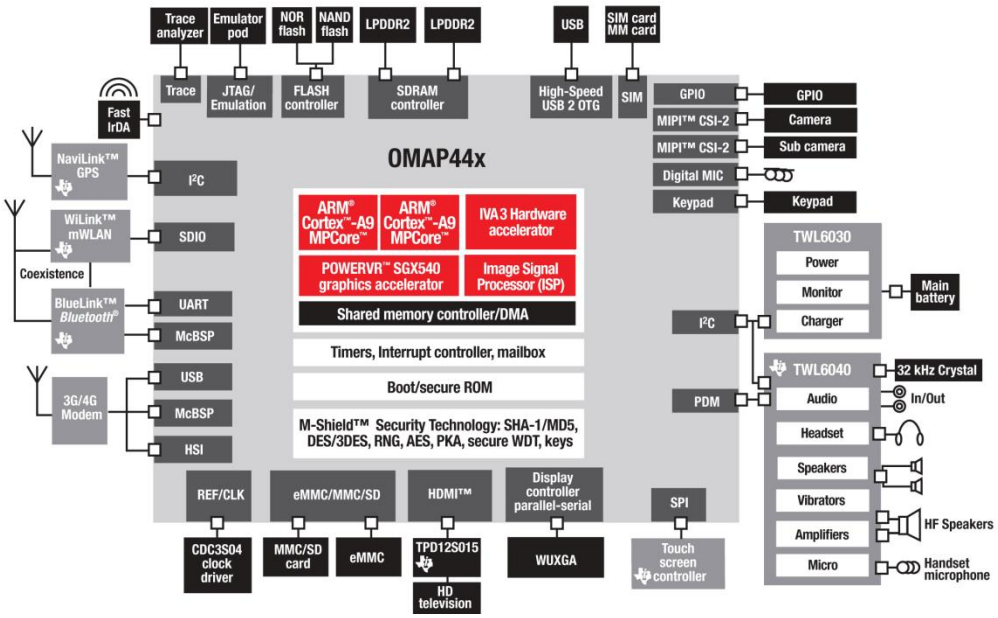
Enhancing DSP Architectures

- More parallelism
  - Increase the number of operations that can be performed in each instruction
    - Adding More Executing units (e.g., Multipliers)
  - Increase the number of instructions that can be issued and executed in every cycle
- Highly specialized hardware in core (such as ARM processors)
- Co-processors (such as Image Signal Processor (ISP))
- Multi-Core DSPs

Example: TI OMAP Chip

Support Features:

- Integrates a TMS320C55x™ DSP core with an ARM GPP on a Single Chip
- OMAP 4 (released December 2010) Architecture includes a dedicated
  - 32-bit Dual-channel LPDDR2 memory technology
  - Uses dual-core ARM Cortex-A9s
  - Uses a PowerVR SGX540 integrated 3D graphics accelerator
  - IVA3 multimedia hardware accelerator with a programmable DSP that enables 1080p Full HD and multi-standard video encode/decode



The Galaxy Nexus, example of a smartphone with an OMAP 4460 SoC

- Comparison: DSP, FPGA, ASIC

"*A tradeoff between microcontroller, DSP, FPGA and ASIC technologies*", EE Times 2009.

"DSP versus FPGA", Electronics Weekly 2012.

"Are FPGA ready to replace ASIC", Design Automation Conference (DAC'12)2012 .

Processor Selection Criteria

- Wide range of different platforms are available, which one to select?
- It depends about the application: what is the most important criteria?
  - Speed.
  - Memory bandwidth.
  - Cost.
  - Ease of use of development tools.
  - Packaging options.
  - On-chip integration.
  - Power consumption.

Processor Selection Criteria

- Use of available benchmarks:
  - BDTI kernel benchmarks.
  - BDTI application benchmarks.
- Use a hierarchical approach to pick a processor
  - List your requirements.
  - Start with critical criteria; and prioritize the remaining ones.
  - Trade-offs may be required.

Mini Assignment

- Search the Internet for the state-of-the-art processors of the platforms we covered; ASIC, FPGA, and DSP in terms of:
- Architecture
- Performance
- Power
- Cost
- Market share
- Leading manufactures