Course Introduction

Hardware Software CoDesign

August 2011

Agenda

Course Introduction

1. Go through the Pre-requisite Embedded Systems Handout (Download from BITS).
2. Go through the Hardware Software CoDesign Handout
3. Some Historical Background

Course Introduction

Historical Background

1. The course of electronics system design changed irreversibly on November 15, 1971, when Intel introduced the first commercial microprocessor, the 4004.

2. Before that date, system design consisted of linking many hardwired blocks, some analog and some digital, with point-to-point connections.

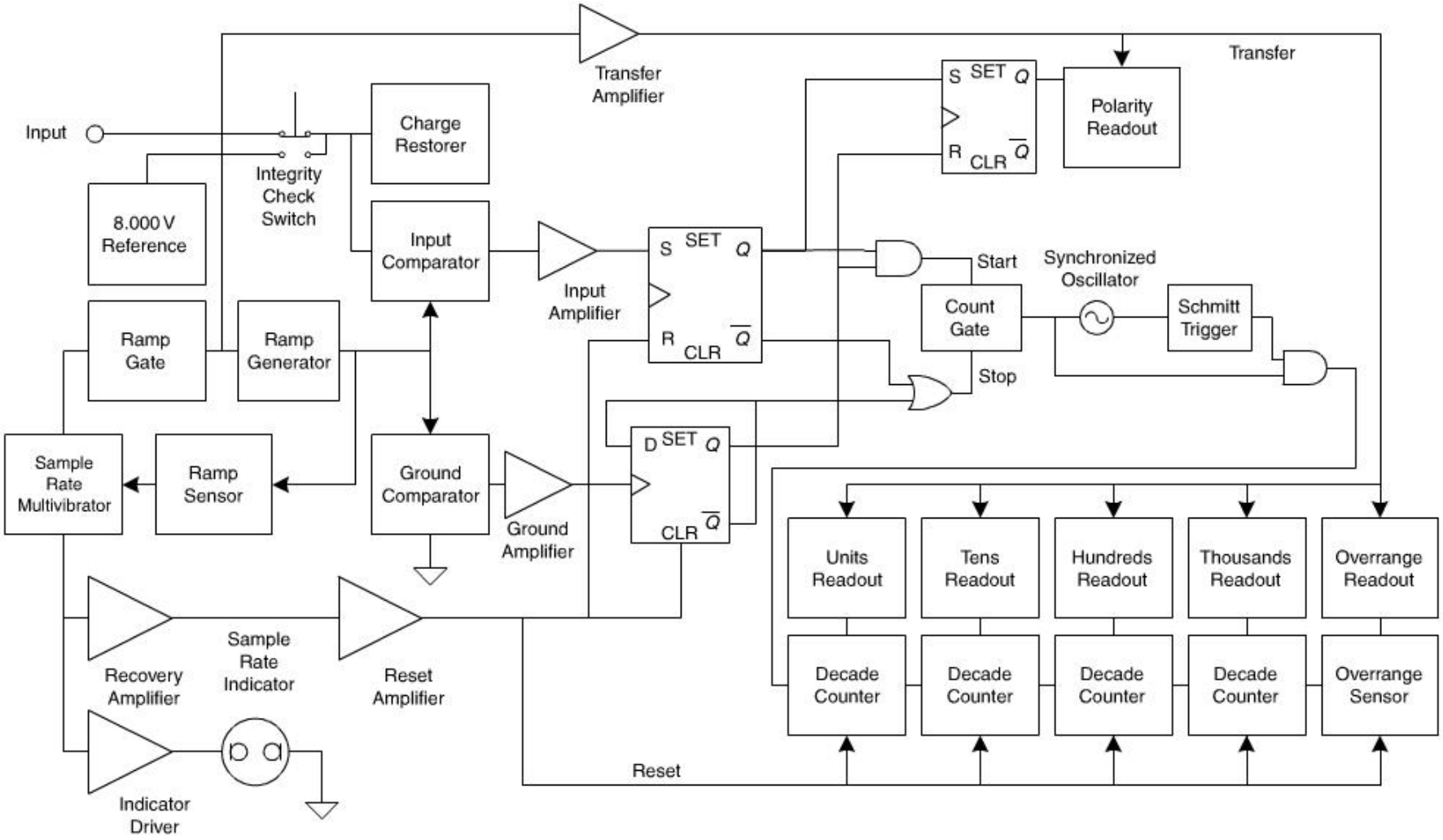
3. The change :-

   (a) The injection of sofware / firmware into the system-design

   (b) The use of buses to interconnect major system blocks

Course Introduction

Historical Background



■ **FIGURE**

A digital voltmeter block diagram adapted from the design of a HP 3440A, circa 1963.

Course Introduction

Historical Background

1. The block diagram shows mix of analog & digital elements interconnected with point-to-point connections.

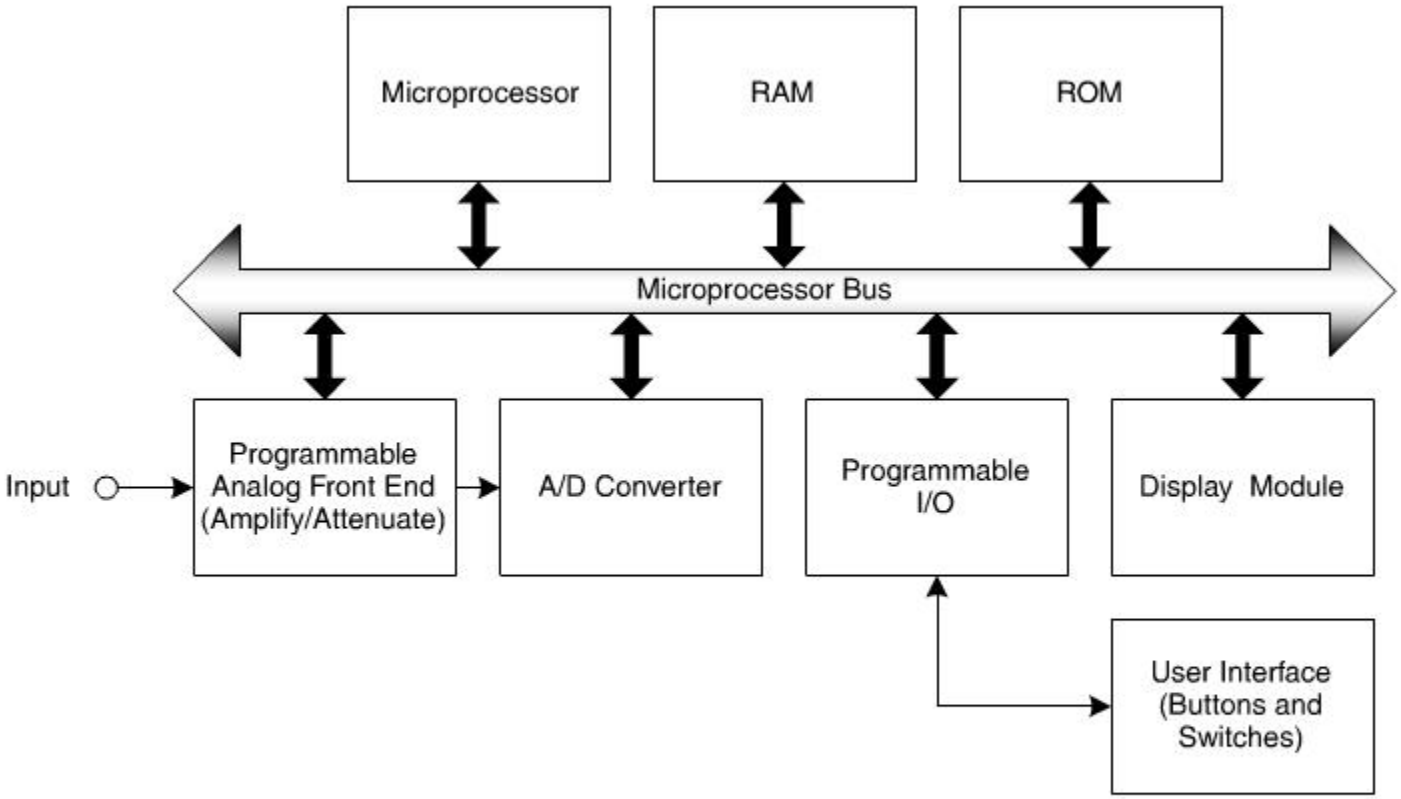
2. Even all-digital measurement counter, which stores ADC output consists of counters. Each counter drives its own numeric display and communicates to the next counter over one wire.

3. There are no busses in the design, none are needed.

4. There are no microprocessors, it would not appear for another 8 years.

Course Introduction

Historical Background → Enter the Processor



■ **FIGURE**

A version of the HP 3440A block diagram of Figure 1.1, adapted for microprocessor-based implementation. This system-design style is still quite popular.

Course Introduction

Historical Background

1. The figure illustrates how a system designer might implement a digital voltmeter like the HP3440A today.

2. A Microprocessor controls all of the major system components in the modern design implementation.

3. The processor communicates with other components over a common bus $\rightarrow$ the microprocessorś main bus.

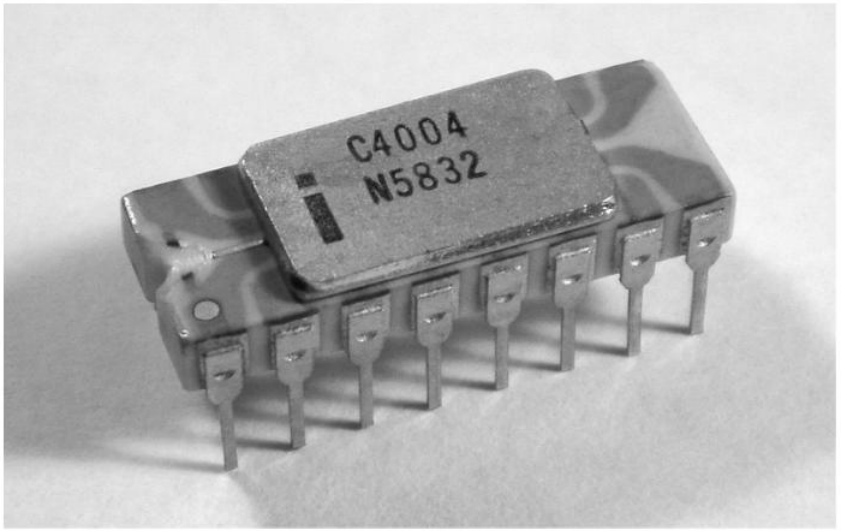
4. Now, what are the other differences :-

   - Massive amount of parallelism in early design **Vs** the one-operation-at-a-time over the microprocessor bus.
   - It was more expensive to routed multiple point-to-point connections **Vs** less expensive to route buses to move data into and out of packaged microprocessors.
   - Nothing need to be multiplexed due to full parallelism **Vs** Limitations on the frequency of operation due to multiplexing the shared bus (resource).

5. What happens NeXT :-

Course Introduction

Historical Background → Few Pins = Massive Multiplexing



■ **FIGURE 1.3**

The Intel 4004 microprocessor, introduced in 1971, was packaged in a 16-pin package that severely limited the processor's I/O bandwidth and restricted its market accordingly. Photo Courtesy of Stephen A. Emery Jr., www.ChipScapes.com.

1. The i-4004 was packaged in a 16-pin DIP.

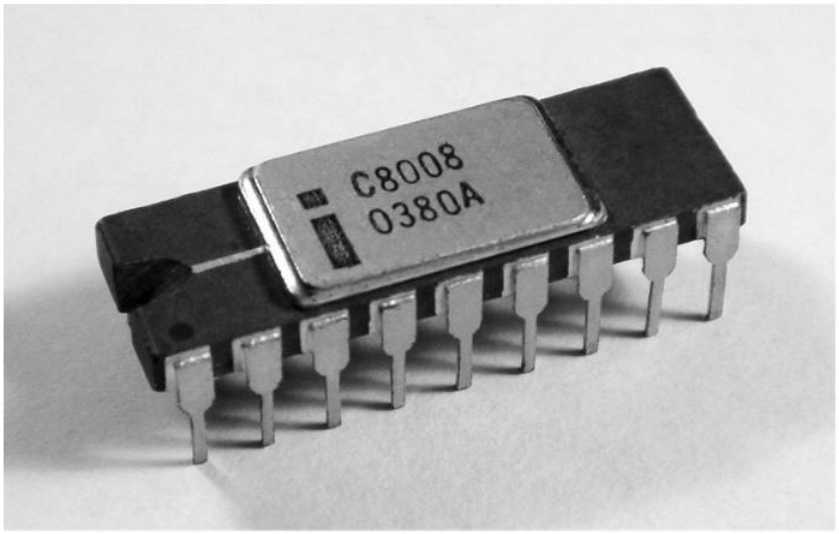
2. The 4bit bus had multiplexed access to the various components in the system. The architecture had multiplexed address and data bus.

3. It took 3 cycles to pass a 12bit address and another 2 or 4 more to read back an 8 or 16bit instruction.

4. With a maximum operating frequency of 740kHz and long, multi-cycle bus operations, the i-4004 was too slow to take on many system control tasks and was largely ignored.

Course Introduction

Historical Background → Few Pins = Massive Multiplexing



■ **FIGURE 1.4**

Intel got more bus bandwidth from the 8008 microprocessor by squeezing it into an unconventional 18-pin package. Photo Courtesy of Stephen A. Emery Jr., www.ChipScapes.com.

1. The i-8008 was introduced in April, 1972 packaged in a 18-pin DIP.

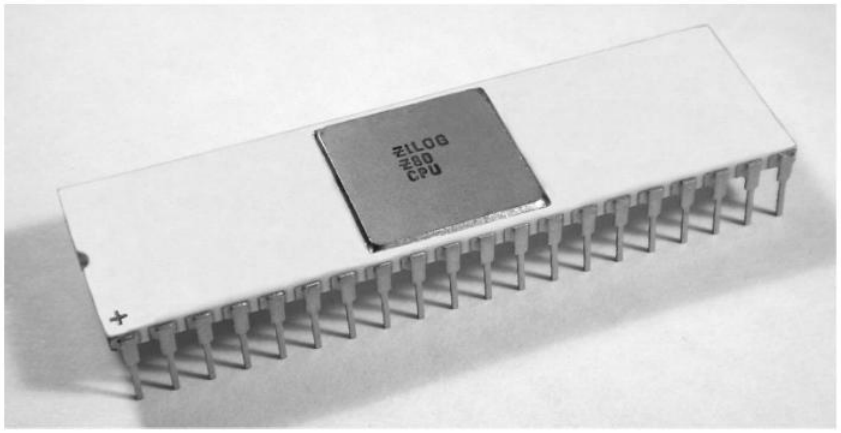
2. The 8bit bus had multiplexed access to the various components in the system. The architecture had multiplexed address and data bus.

3. It took 2 cycles to pass a 14bit address and another 1 or 2 more to read back an 8 or 16bit instruction.

4. With a maximum operating frequency of 800kHz, the i-8008 was still too slow to fire the imagination of many system designers.

Course Introduction

Historical Background → More Pins Better Bus Performance



■ **FIGURE 1.5**

Intel finally crossed the bus-bandwidth threshold into usability by packaging its third-generation 8080 microprocessor in a 40-pin package. Many competitors swiftly followed suit (shown is Zilog's 8-bit Z80 microprocessor) and the microprocessor quickly became a standard building block for system designers. Photo Courtesy of Stephen A. Emery Jr., www.ChipScapes.com.

1. The i-8080 was introduced in April, 1974 packaged in a 40-pin DIP.
2. The separate address bus was 16bits and data bus was 8bits
3. With a maximum operating frequency of 2MHz, the i-8080 finally got system designers to adopt the microprocessor as key system building block.
4. Other Microprocessor Vendors like Motorola & Zilog also introduced at about the same time in 40pin DIP (shown is the Z80 but mentioned i-8080).
5. Since then, microprocessor based design has become the nearly universal approach to system design.

Course Introduction

Historical Background → Microprocessor = Universal Building Block !

1. Economics :: Standard uP provide abilities at a modest cost.

2. Hardware is more difficult to change than Software / Firmware.

3. Hardware designer can design a uP based system and build it before the systems function is fully defined.

4. Adding the software / firmware into h/w finalizes the design and this event can occur after the h/w has been designed, prototyped, verified, tested, manufactured and even fielded. This allows for h/w and firmware to be developed concurrently which reduces the project schedule (ideally yes).

5. NOW, systems needed more tasks to be executed on processors... What happened...

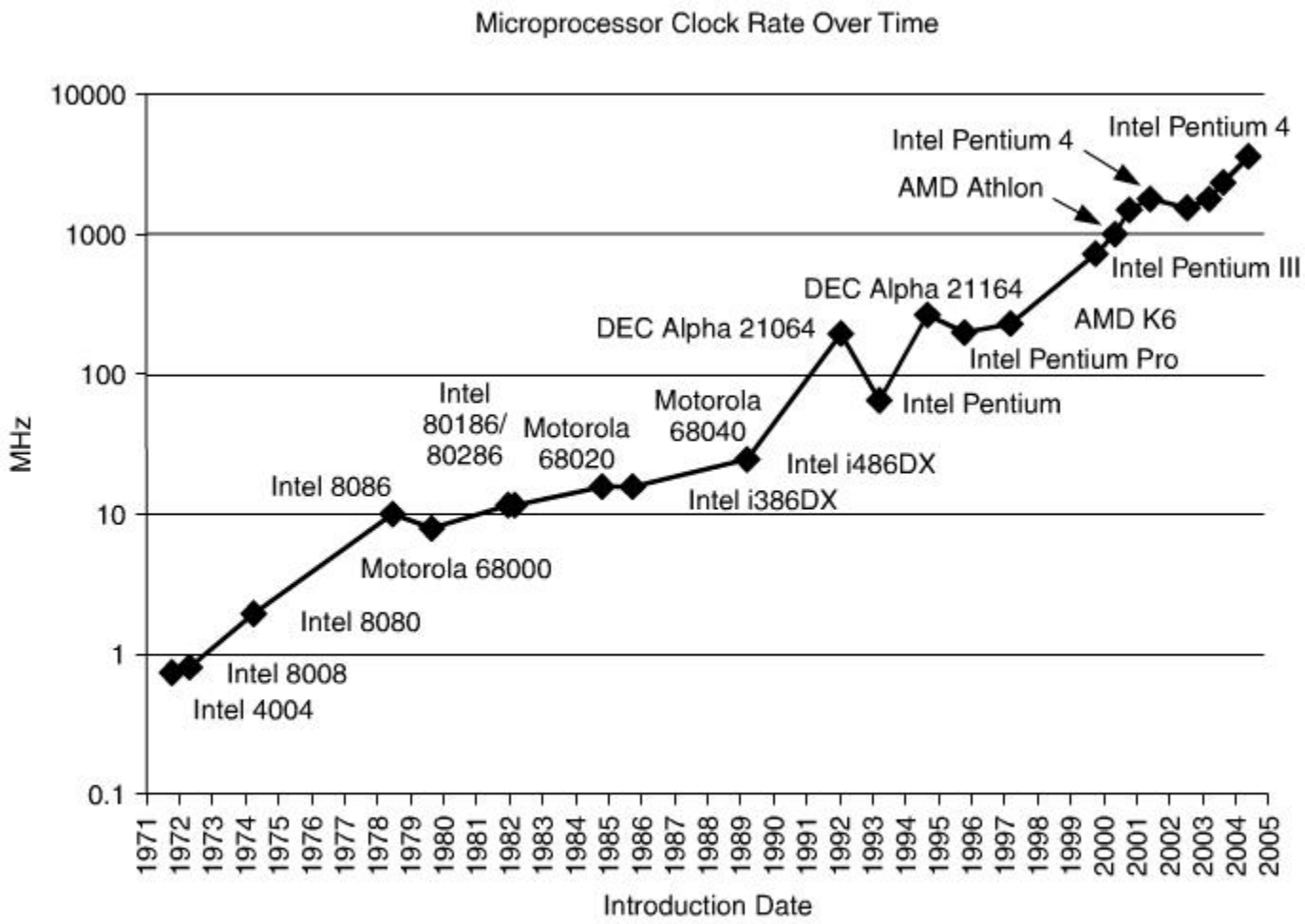
Course Introduction

Historical Background

1. NOW, systems needed more tasks to be executed on processors... What happened... uP Vendors adopted the following...
   - Increase the clock rate... rush by uP vendors to do this...
   - The uP data-word and buses widened... 16bit, 32bit, 64bit..
   - Add more buses to the processor architecture...
2. WHAT IS the fallout of all these techniques ?

Course Introduction

Historical Background



■ **FIGURE 1.6**

Microprocessor clock rates have risen dramatically over time due to the demand of system designers for ever more processor performance.

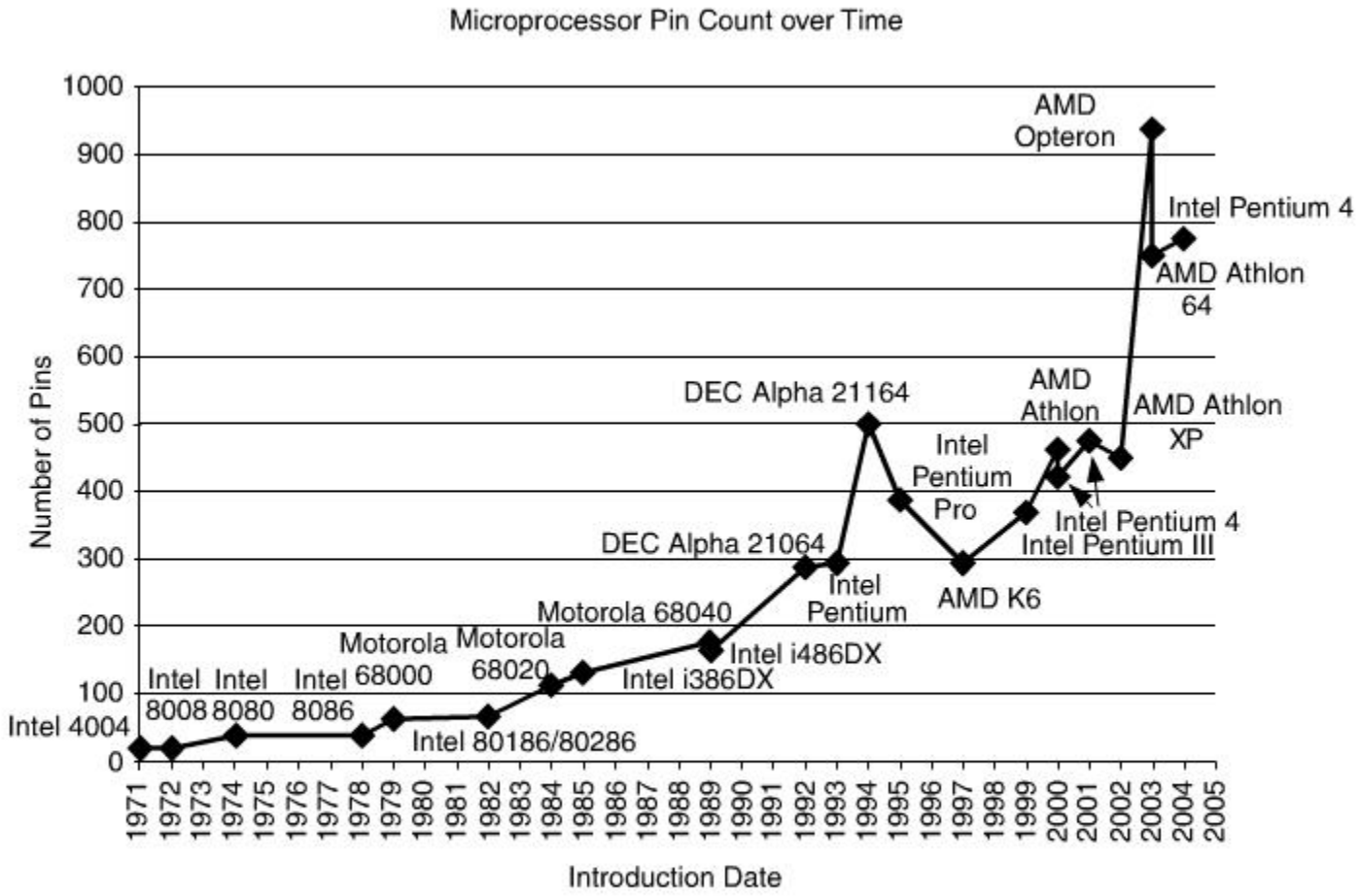
Course Introduction

Historical Background

1. NOW, systems needed more tasks to be executed on processors... What happened... uP Vendors adopted the following...
   - Increase the clock rate... rush by uP vendors to do this...
   - The uP data-word and buses widened... 16bit, 32bit, 64bit..
   - Add more buses to the processor architecture...
2. WHAT IS the fallout of all these techniques ?
   - Device pin size increase.
   - Device power dissipation increase.

Course Introduction

Historical Background

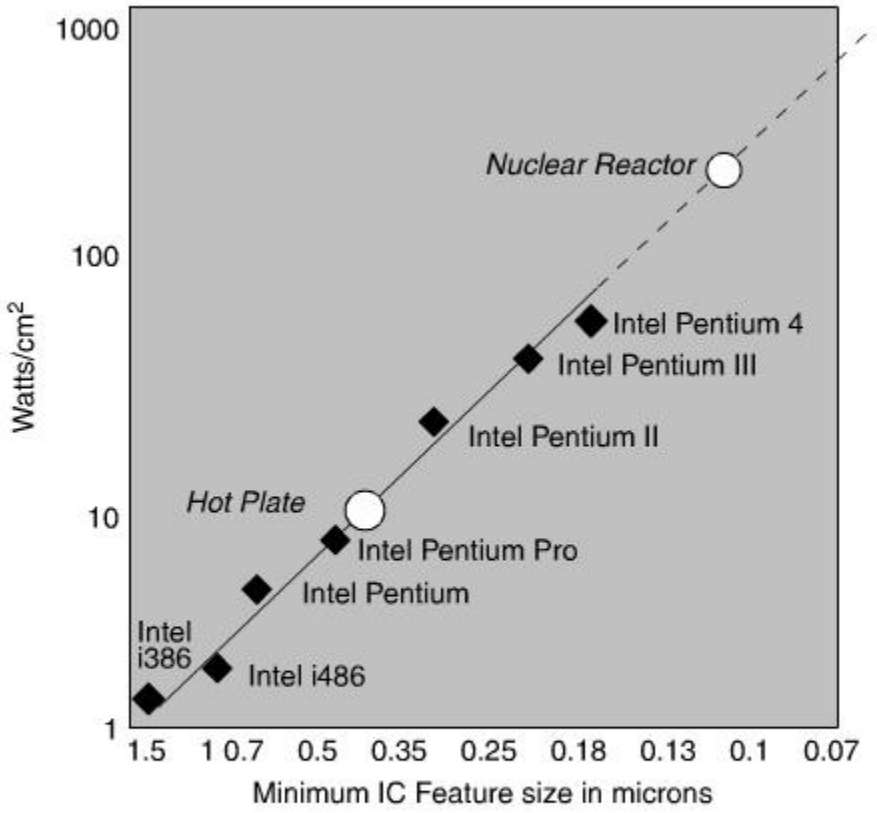


■ **FIGURE 1.7**

Microprocessor pin counts have also risen dramatically over time due to the demand of system designers for ever more processor performance.

Course Introduction

Historical Background



■ **FIGURE 1.8**

Packaged microprocessor power density has risen exponentially for decades. (*Source*: F. Pollack, keynote speech, "New microarchitecture challenges in the coming generations of CMOS process technologies," MICRO-32, Haifa, Israel, 1999.)

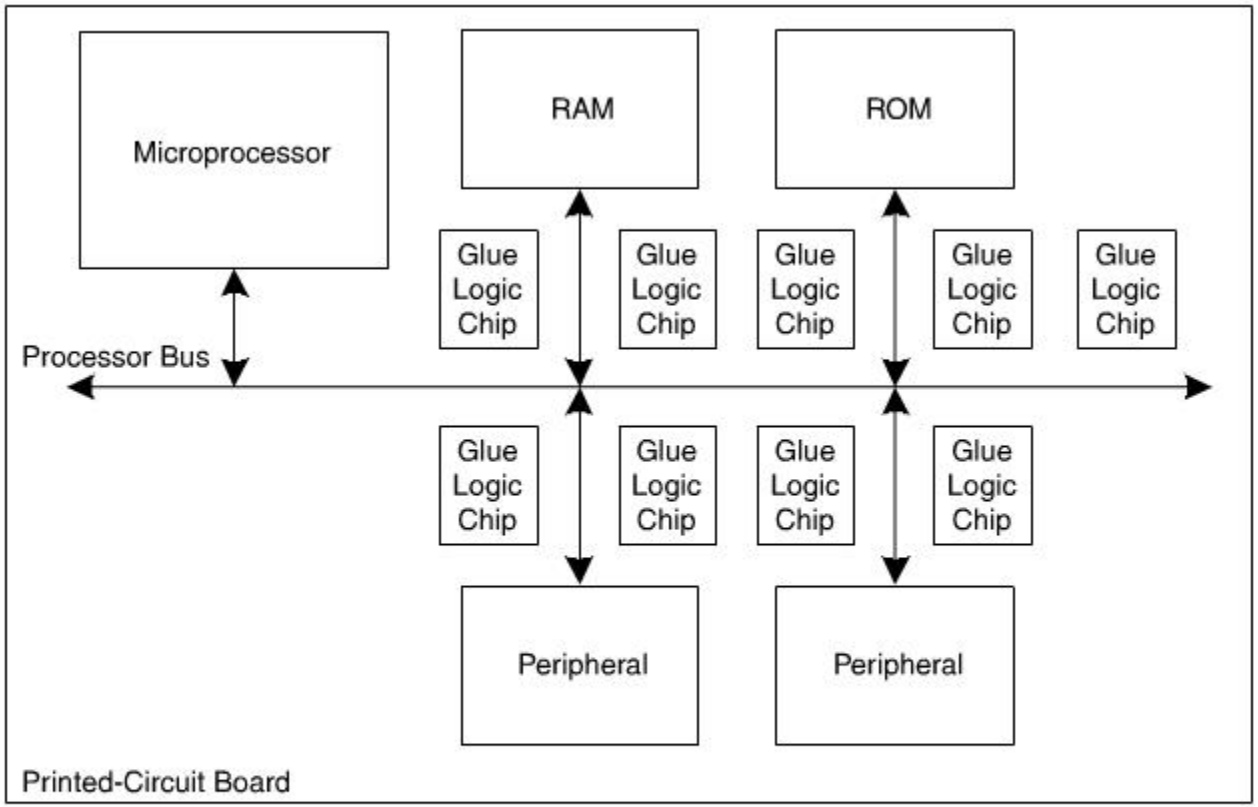
Course Introduction

Historical Background

1. APART from the technological challenges, system design evolved...
   - First with uPs and components like RAM, ROM, Glue, Peripherals on a system PCB.
   - Then with uPs and system components on a single die (SoC), with fewer components on system PCB.
   - The Glue logic evolved from simple 74LS devices, to Glue ASICs (PLAs), to FPGAs

Course Introduction

Historical Background



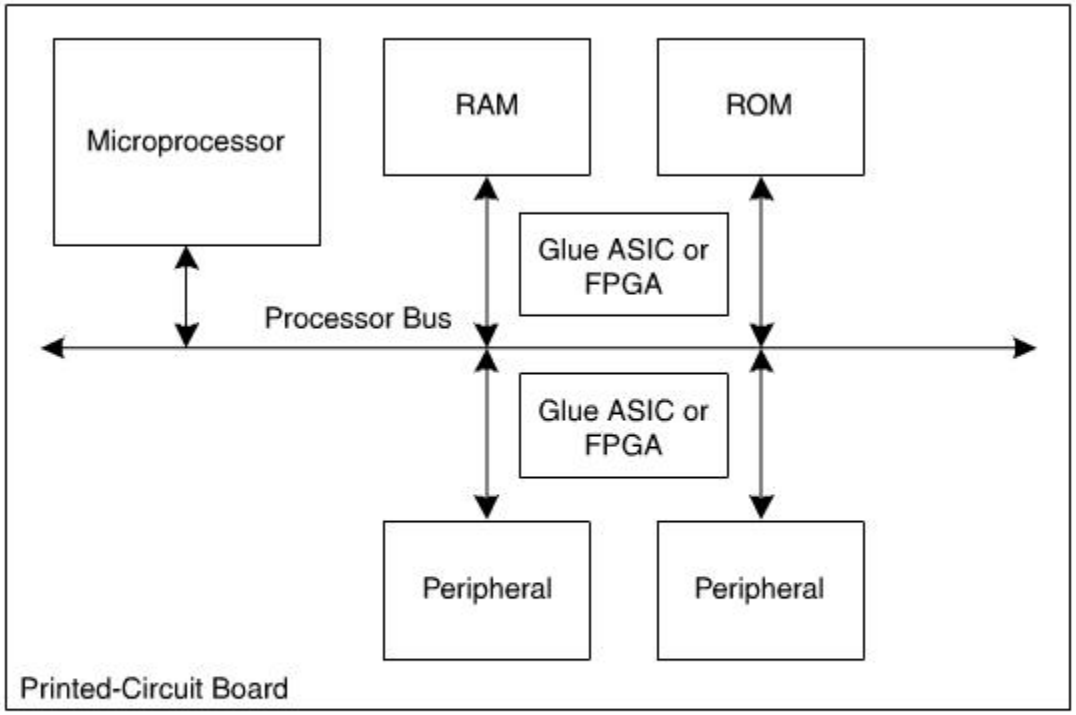
■ **FIGURE 1.11**

By 1985, microprocessor-based system design using standard LSI parts and printed-circuit boards was common.

1. First with uPs and components like RAM, ROM, Glue, Peripherals on a system PCB.

Course Introduction

Historical Background



■ **FIGURE 1.12**

By 1990, system design was still mostly done at the board level but system designers started to use ASICs and FPGAs to consolidate glue logic into one or two chips.

1. Then with uPs and system components on a single die (SoC), with fewer components on system PCB.

2. The Glue logic evolved from simple 74LS devices, to Glue ASICs (PLAs), to FPGAs

Course Introduction

Historical Background

1. By 1995, ASIC capabilities advanced enough to include a processor core. Thus began the SoC design era.

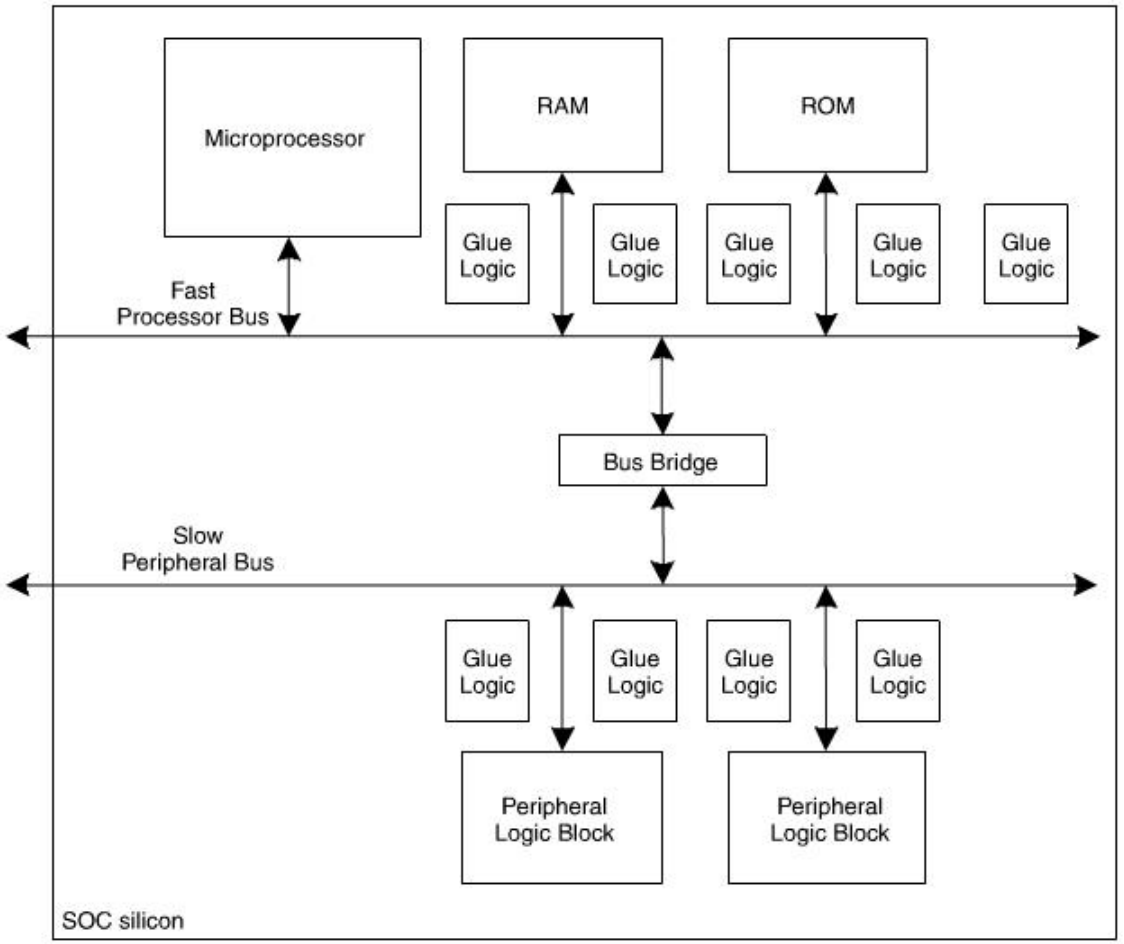
2. However, the SoC block diagrams looked similar to the system block diagrams.

3. By 2000, technology allowed processor cores to be instantiated in SoCś with increasingly high clock rates.

4. This led to the de-coupling of fast processor subsystems from the slower processor subsystems (slower peripherals). This means there were on-chip bus hierarchies like fast buses and slow buses separated by a bus bridge.

Course Introduction

Historical Background

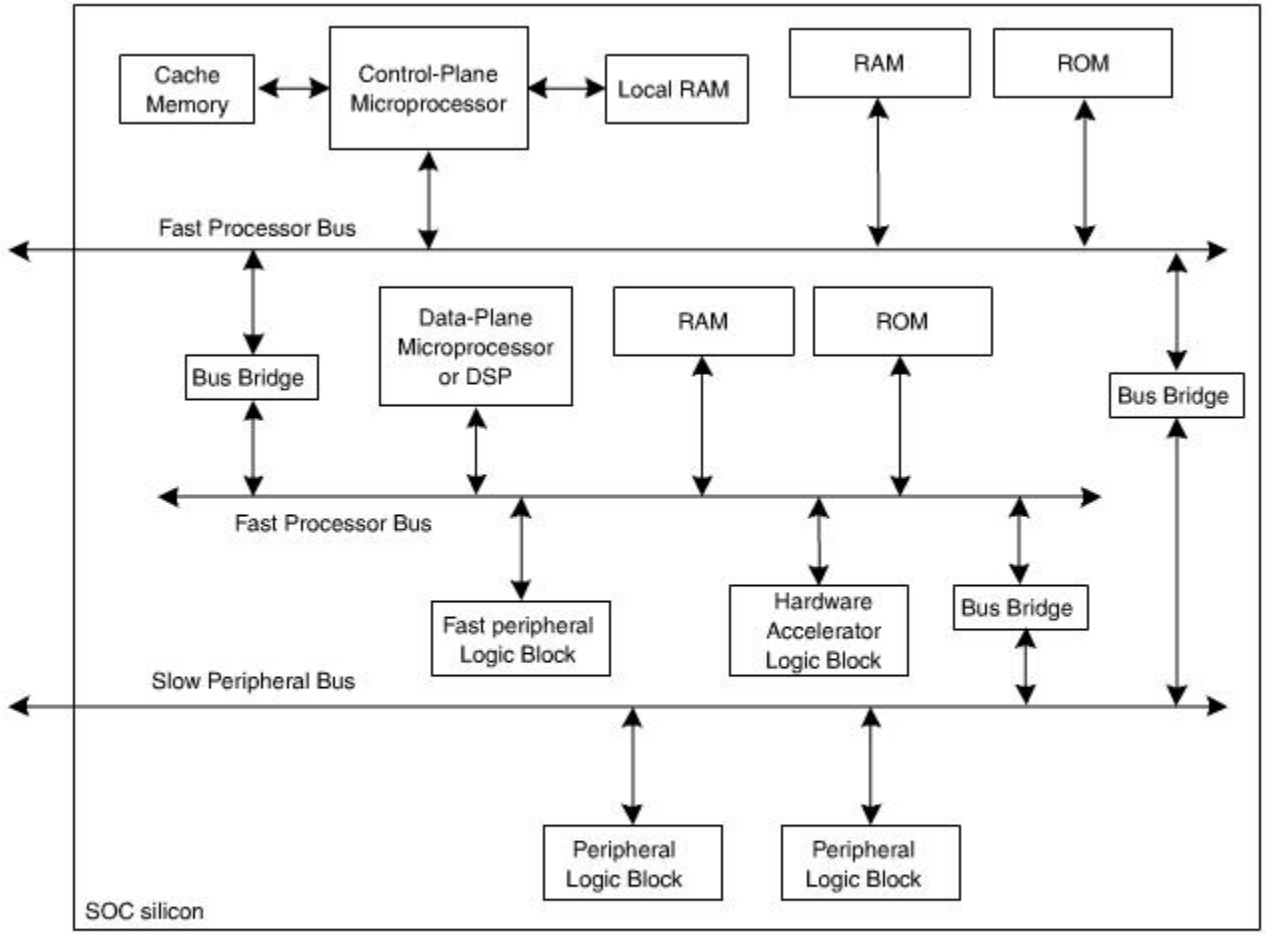


■ **FIGURE 1.14**

By the year 2000, system designers were splitting the on-chip bus into a small, fast processor-memory bus and a slower peripheral bus. A hierarchical bus topology allowed chip designers to greatly reduce the capacitance of the high-speed bus by making it physically smaller. Even so, the SOC's logical block diagram continued to strongly resemble single-processor, board-level systems designed 15 years earlier.

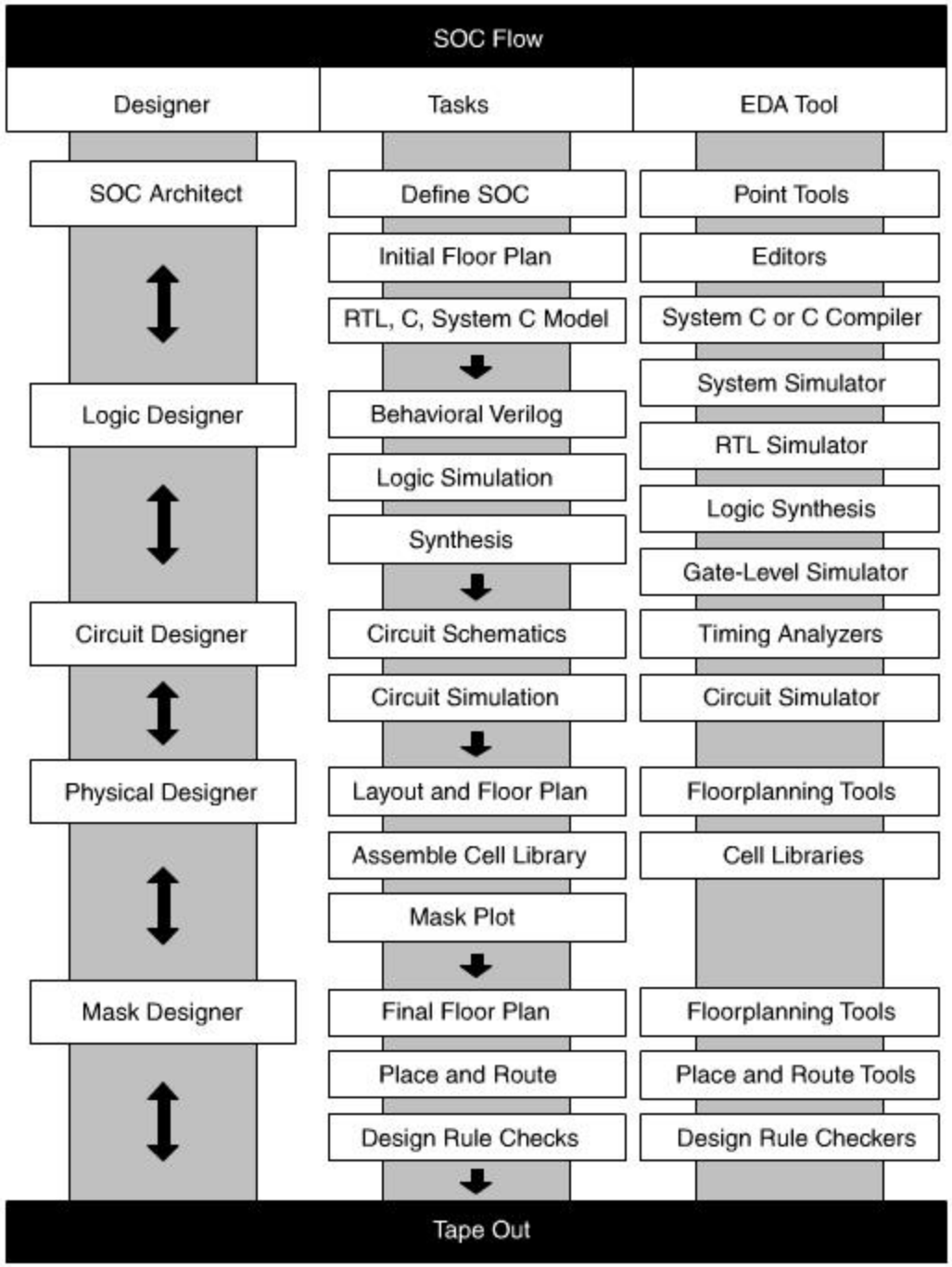
Course Introduction

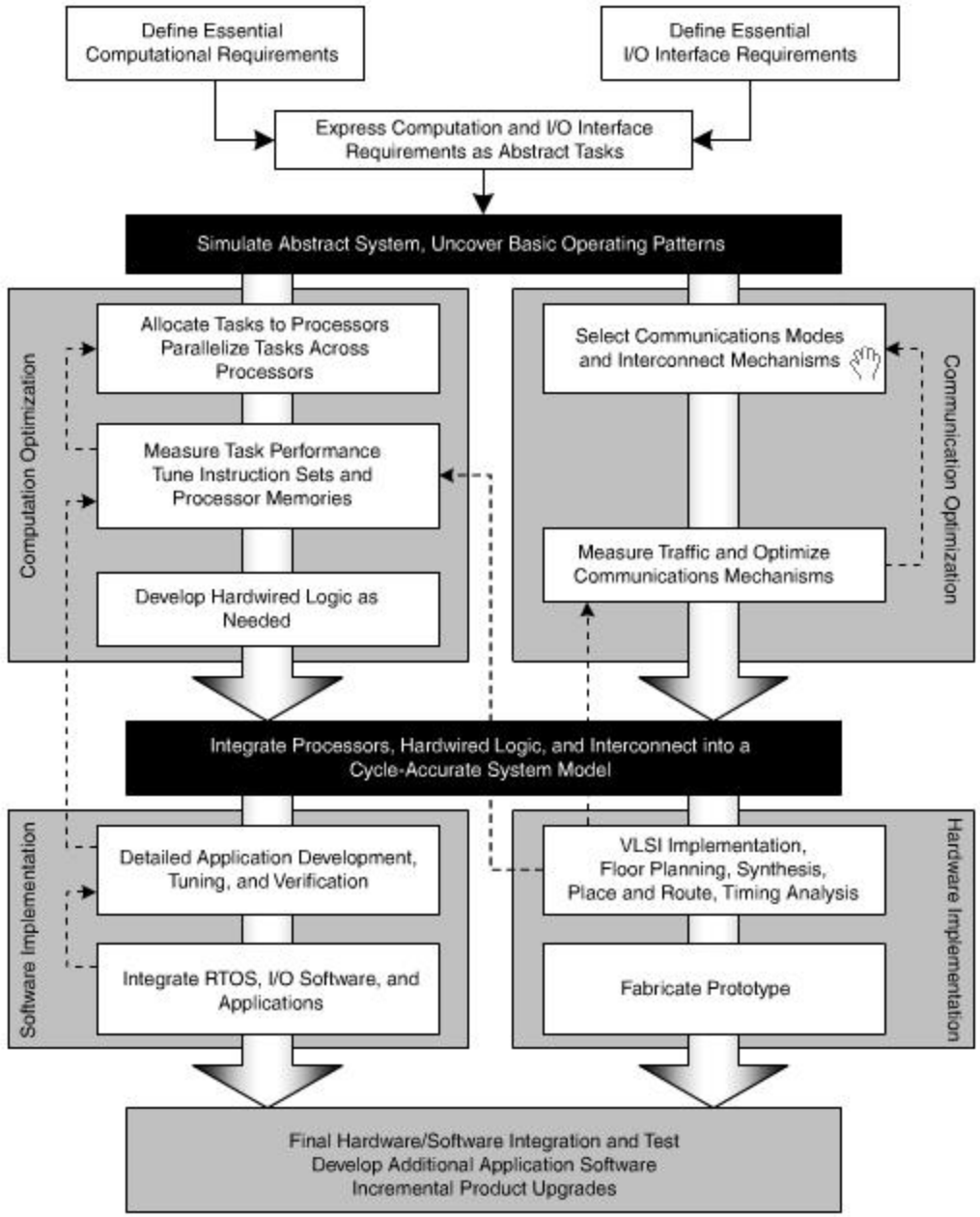
Historical Background → Present Day SoCś



■ **FIGURE 1.15**

Present-day SOC design has started to employ multiple processors instead of escalating clock rates to achieve processing goals.


SOC Flow
Designer
Tasks
EDA Tool
SOC Architect
Define SOC
Point Tools
Initial Floor Plan
Editors
RTL, C, System C Model
System C or C Compiler
System Simulator
Logic Designer
Behavioral Verilog
RTL Simulator
Logic Simulation
Logic Synthesis
Synthesis
Gate-Level Simulator
Circuit Designer
Circuit Schematics
Timing Analyzers
Circuit Simulation
Circuit Simulator
Physical Designer
Layout and Floor Plan
Floorplanning Tools
Assemble Cell Library
Cell Libraries
Mask Plot
Mask Designer
Final Floor Plan
Floorplanning Tools
Place and Route
Place and Route Tools
Design Rule Checks
Design Rule Checkers
Tape Out


Define Essential Computational Requirements
Define Essential I/O Interface Requirements
Express Computation and I/O Interface Requirements as Abstract Tasks
Simulate Abstract System, Uncover Basic Operating Patterns
Computation Optimization
Allocate Tasks to Processors Parallelize Tasks Across Processors
Measure Task Performance Tune Instruction Sets and Processor Memories
Develop Hardwired Logic as Needed
Communication Optimization
Select Communications Modes and Interconnect Mechanisms
Measure Traffic and Optimize Communications Mechanisms
Integrate Processors, Hardwired Logic, and Interconnect into a Cycle-Accurate System Model
Software Implementation
Detailed Application Development, Tuning, and Verification
Integrate RTOS, I/O Software, and Applications
Hardware Implementation
VLSI Implementation, Floor Planning, Synthesis, Place and Route, Timing Analysis
Fabricate Prototype
Final Hardware/Software Integration and Test
Develop Additional Application Software
Incremental Product Upgrades

Course Introduction – Class Example

An Example → Simple 4bit Multiplier

Example of a 4bit Multiplier								
Multiplier		12d			1	1	0	0
Multiplicand		5d			0	1	0	1
					1	1	0	0
				0	0	0	0	.
			1	1	0	0	.	.
		0	0	0	0	.	.	.
Result	60d	0	1	1	1	1	0	0

1. Assume for now that you only have an 8085 processor at 1MHz. How would you code it.

2. Assume you have a 8bit multiplier hardware block, attached to the 8085 processor. How would you code it.

3. What would you do if you do have h/w flexibility, but not as much as a 8bit multiplier block. AND s/w flexibility that it's ok if it is not a single cycle. In other words, its ok to have a multiplier throughput of say 500kHz or 256kHz (two flavors).

4. Assume external h/w also works at 1MHz max.

Course Introduction – Class Example

DATA TRANSFER GROUP	
MOV	Move
MVI	Move Immediate
LDA	Load Accumulator Directly from Memory
STA	Store Accumulator Directly in Memory
LHLD	Load H & L Registers Directly from Memory
SHLD	Store H & L Registers Directly in Memory
An X in the name indicates a register pair (16bits)	
LXI	Load Register Pair with Immediate data
LDAX	Load Accumulator from Address in Register Pair
STAX	Store Accumulator in Address in Register Pair
XCHG	Exchange H & L with D & E
XTHL	Exchange Top of Stack with H & L
Logical Group	
ANA	Logical AND with Accumulator
ANI	Logical AND with Accumulator Using Immediate Dat
ORA	Logical OR with Accumulator
OR	Logical OR with Accumulator Using Immediate Data
XRA	Exclusive Logical OR with Accumulator
XRI	Exclusive OR Using Immediate Data
CMP	Compare
CPI	Compare Using Immediate Data
RLC	Rotate Accumulator Left
RRC	Rotate Accumulator Right
RAL	Rotate Left Through Carry
RAR	Rotate Right Through Carry

Arithmetic Group	
ADD	Add to Accumulator
ADI	Add Immediate Data to Accumulator
ADC	Add to Accumulator Using Carry Flag
ACI	Add Immediate data to Accumulator Using Carry
SUB	Subtract from Accumulator
SUI	Subtract Immediate Data from Accumulator
SBB	Subtract from Accumulator Using Borrow (Carry) Flag
INR	Increment Specified Byte by One
DCR	Decrement Specified Byte by One
INX	Increment Register Pair by One
DCX	Decrement Register Pair by One
DAD	Double Register Add; Add Content of Register
	Pair to H & L Register Pair
Branch Group	
JMP	Jump
CALL	Call
RET	Return
Conditions for Branching	
NZ	Not Ze 0)
Z	Zero (Z = 1)
NC	No Car 0)
C	Carry (C = 1)
PO	Parity = 0)
PE	Parity = 1)
P	Plus (S = 0)
M	Minus (S = 1)

A B C D E H L	Registers (8-bit)
BC DE HL	Register pairs (16-bit)
PC	Program Counter register (16-bit)
PSW	Processor Status Word (8-bit)
SP	Stack Pointer register (16-bit)

Course Introduction – Class Example

An Example → Simple 4bit Multiplier → HINTS

1. Assume for now that you only have an 8085 processor at 1MHz. How would you code it.
   - Think of left shift multiplicand, conditional add

2. Assume you have a 8bit multiplier hardware block, attached to the 8085 processor. How would you code it.
   - Think of sending operands to hardware block, read results.
   - Cost of Area at the gain of higher throughput.

3. What would you do if you do have h/w flexibility, but not as much as a 8bit multiplier block. AND s/w flexibility that it's ok if it is not a single cycle. In other words, its ok to have a multiplier throughput of say 500kHz or 256kHz (two flavors).
   - Think of sending operands to hardware block, read results.
   - What will reduce Area at the expense of throughput.
   - What is the system requirements, What is the sweet spot.

4. Assume external h/w also works at 1MHz max..

5. Throw power into the equation on a more complex example.

Course Introduction

Acknowledgements

1. Designing SoCs with Configured Cores: Unleashing the Tensilica Xtensa and Diamond Cores :: Steve Leibson :: Ch 1

2. Principles of CMOS VLSI Design - A System Perspective :: N. H. E. Weste and K. Eshranghian :: Multipliers

3. Foils on multiplier implementations :: EE5324 - VLSI Design II :: Kia Bazargan (U. Minnesota)