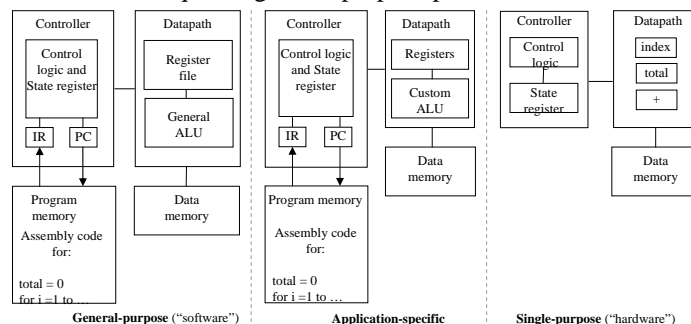


Three key embedded system technologies

- 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

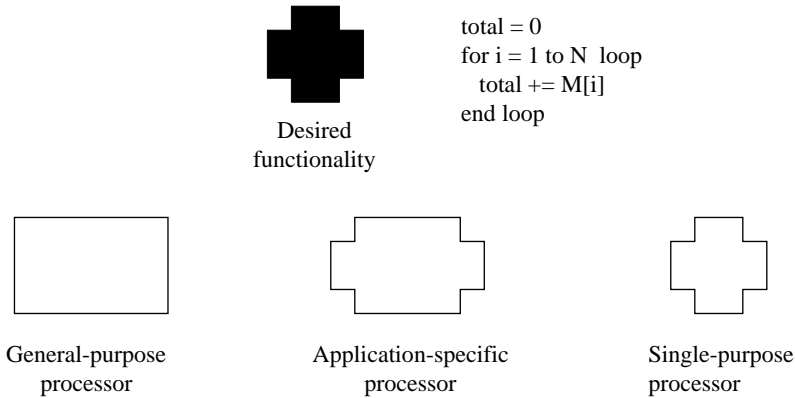
Processor technology

- 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



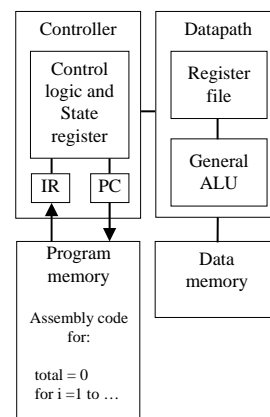
Processor technology

- Processors vary in their customization for the problem at hand



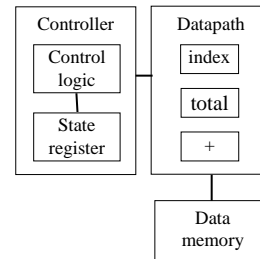
General-purpose processors

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



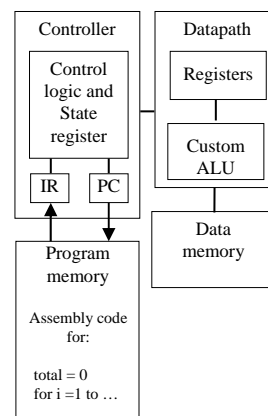
Single-purpose processors

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



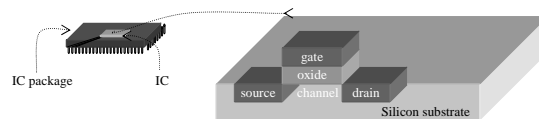
Application-specific processors

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



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)

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

Semi-custom

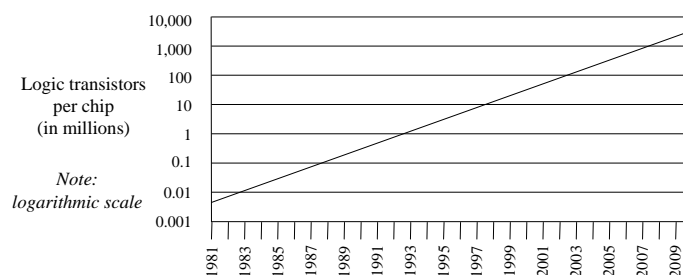
- Lower layers are fully or partially built
 - Designers are left with routing of wires and maybe placing some blocks
- Benefits
 - Good performance, good size, less NRE cost than a full-custom implementation (perhaps \$10k to \$100k)
- Drawbacks
 - Still require weeks to months to develop

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

Moore's law

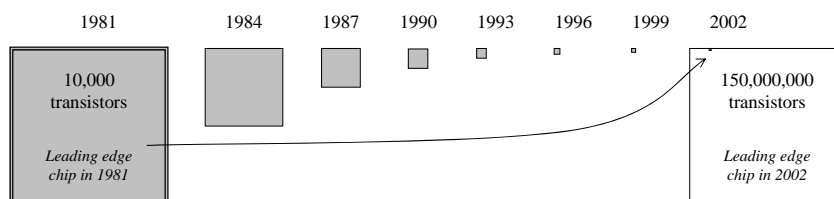
- The most important trend in embedded systems
 - Predicted in 1965 by Intel co-founder Gordon Moore
- IC transistor capacity has doubled roughly every 18 months
for the past several decades**



Moore's law

- Wow
 - This growth rate is hard to imagine, most people underestimate
 - How many ancestors do you have from 20 generations ago
 - i.e., roughly how many people alive in the 1500's did it take to make you?
 - 2^{20} = more than 1 million people
 - (*This underestimation is the key to pyramid schemes!*)

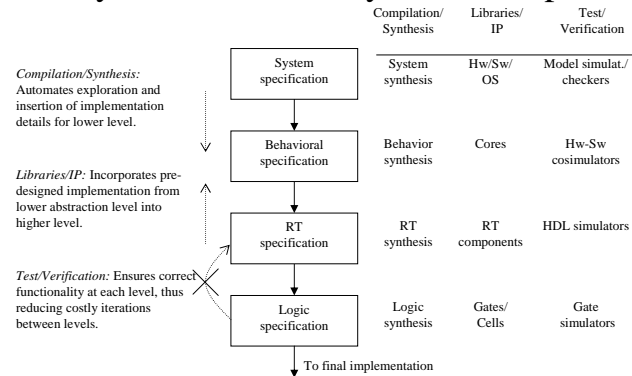
Graphical illustration of Moore's law



- Something that doubles frequently grows more quickly than most people realize!
 - A 2002 chip can hold about 15,000 1981 chips inside itself

Design Technology

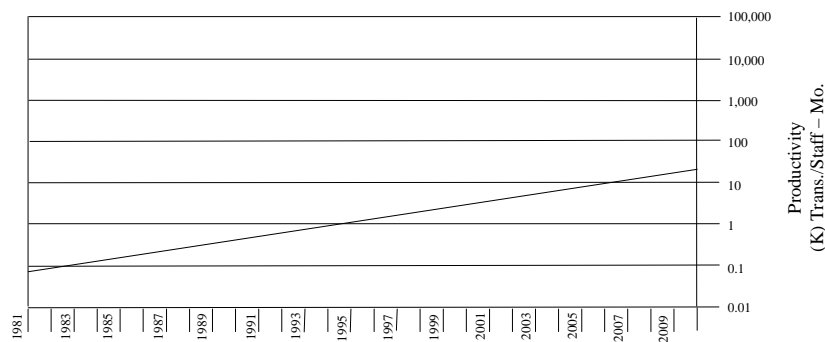
- The manner in which we convert our concept of desired system functionality into an implementation



Embedded Systems Design: A Unified
Hardware/Software Introduction, (c) 2000 Vahid/Givargis

15

Design productivity exponential increase



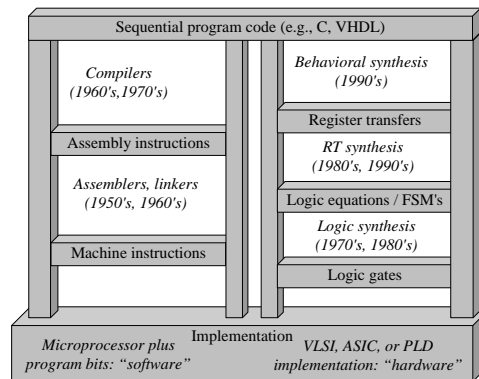
- Exponential increase over the past few decades

Embedded Systems Design: A Unified
Hardware/Software Introduction, (c) 2000 Vahid/Givargis

16

The co-design ladder

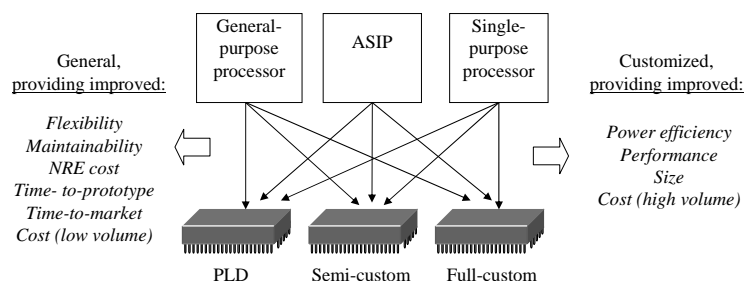
- In the past:
 - Hardware and software design technologies were very different
 - Recent maturation of synthesis enables a unified view of hardware and software
- Hardware/software “codesign”



The choice of hardware versus software for a particular function is simply a tradeoff among various design metrics, like performance, power, size, NRE cost, and especially flexibility; there is no fundamental difference between what hardware or software can implement.

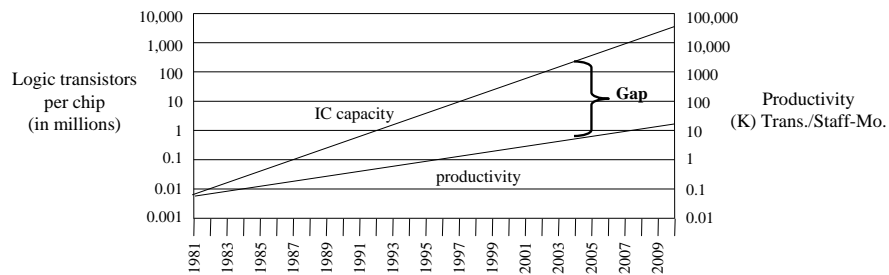
Independence of processor and IC technologies

- Basic tradeoff
 - General vs. custom
 - With respect to processor technology or IC technology
 - The two technologies are independent



Design productivity gap

- While designer productivity has grown at an impressive rate over the past decades, the rate of improvement has not kept pace with chip capacity

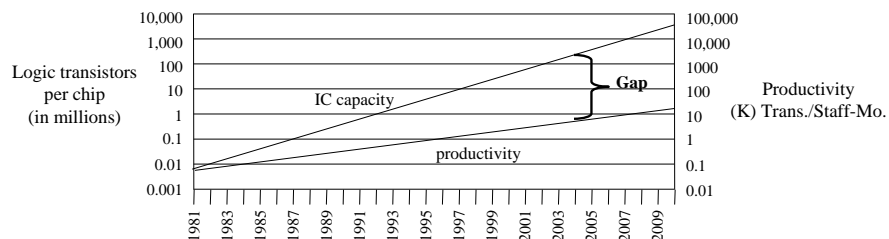


Embedded Systems Design: A Unified
Hardware/Software Introduction, (c) 2000 Vahid/Givargis

19

Design productivity gap

- 1981 leading edge chip required 100 designer months
 - 10,000 transistors / 100 transistors/month
- 2002 leading edge chip requires 30,000 designer months
 - 150,000,000 / 5000 transistors/month
- Designer cost increase from \$1M to \$300M



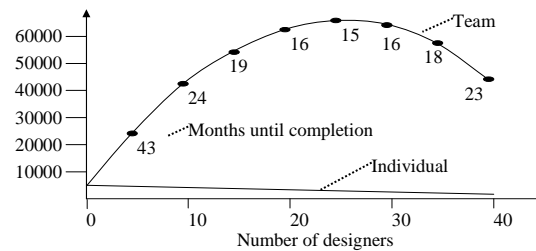
Embedded Systems Design: A Unified
Hardware/Software Introduction, (c) 2000 Vahid/Givargis

20

The mythical man-month

- The situation is even worse than the productivity gap indicates
- In theory, adding designers to team reduces project completion time
- In reality, productivity per designer decreases due to complexities of team management and communication
- In the software community, known as “the mythical man-month” (Brooks 1975)
- At some point, can actually lengthen project completion time! (“Too many cooks”)

- 1M transistors, 1 designer=5000 trans/month
- Each additional designer reduces for 100 trans/month
- So 2 designers produce 4900 trans/month each



*Embedded Systems Design: A Unified
Hardware/Software Introduction, (c) 2000 Vahid/Givargis*

21

Summary

- Embedded systems are everywhere
- Key challenge: optimization of design metrics
 - Design metrics compete with one another
- A unified view of hardware and software is necessary to improve productivity
- Three key technologies
 - Processor: general-purpose, application-specific, single-purpose
 - IC: Full-custom, semi-custom, PLD
 - Design: Compilation/synthesis, libraries/IP, test/verification

*Embedded Systems Design: A Unified
Hardware/Software Introduction, (c) 2000 Vahid/Givargis*

22