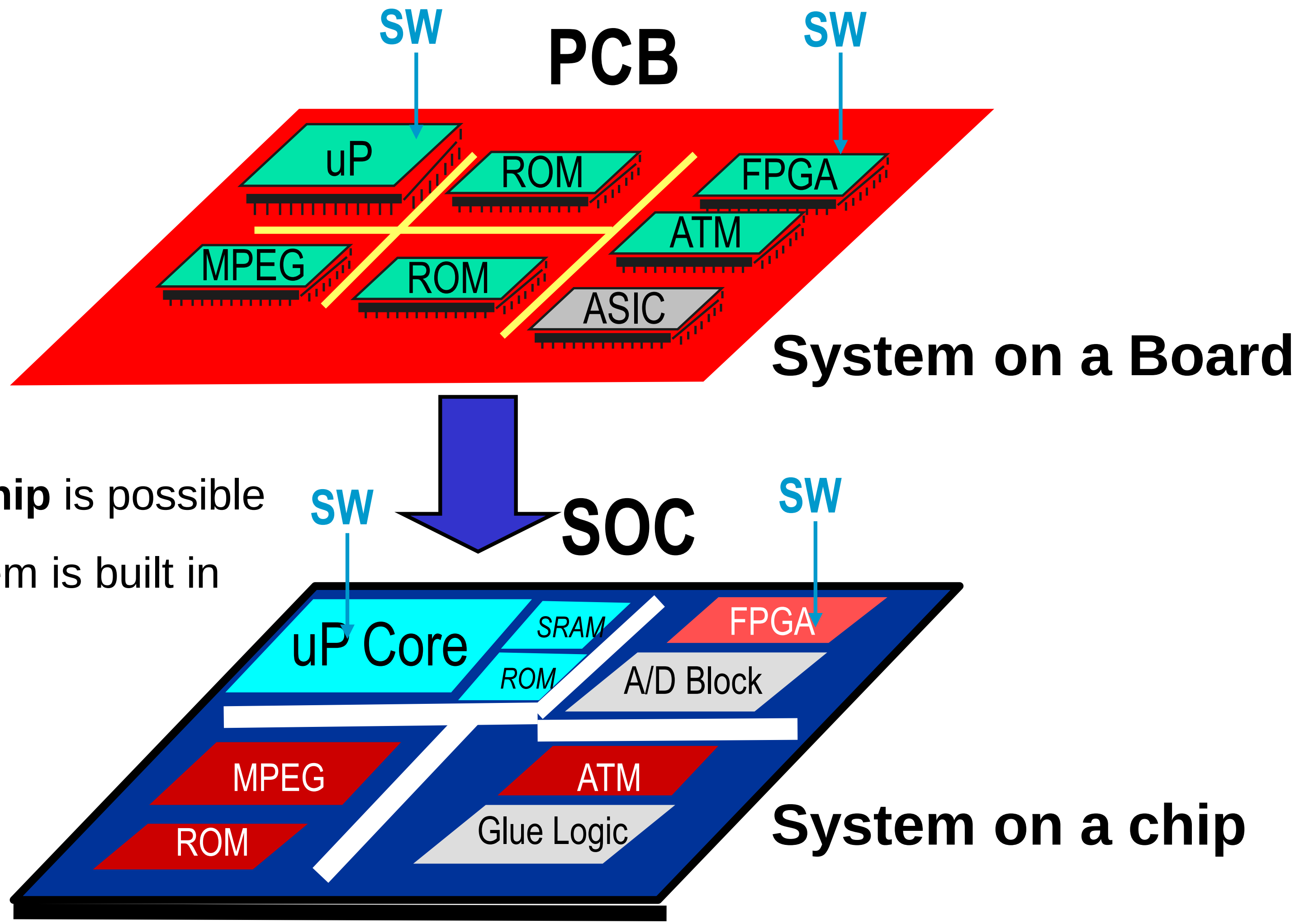
數位IC設計

System on a Chip (SoC)

Introduction

Chip: tens of millions of transistors or more

Design shifts from ASIC/board to system

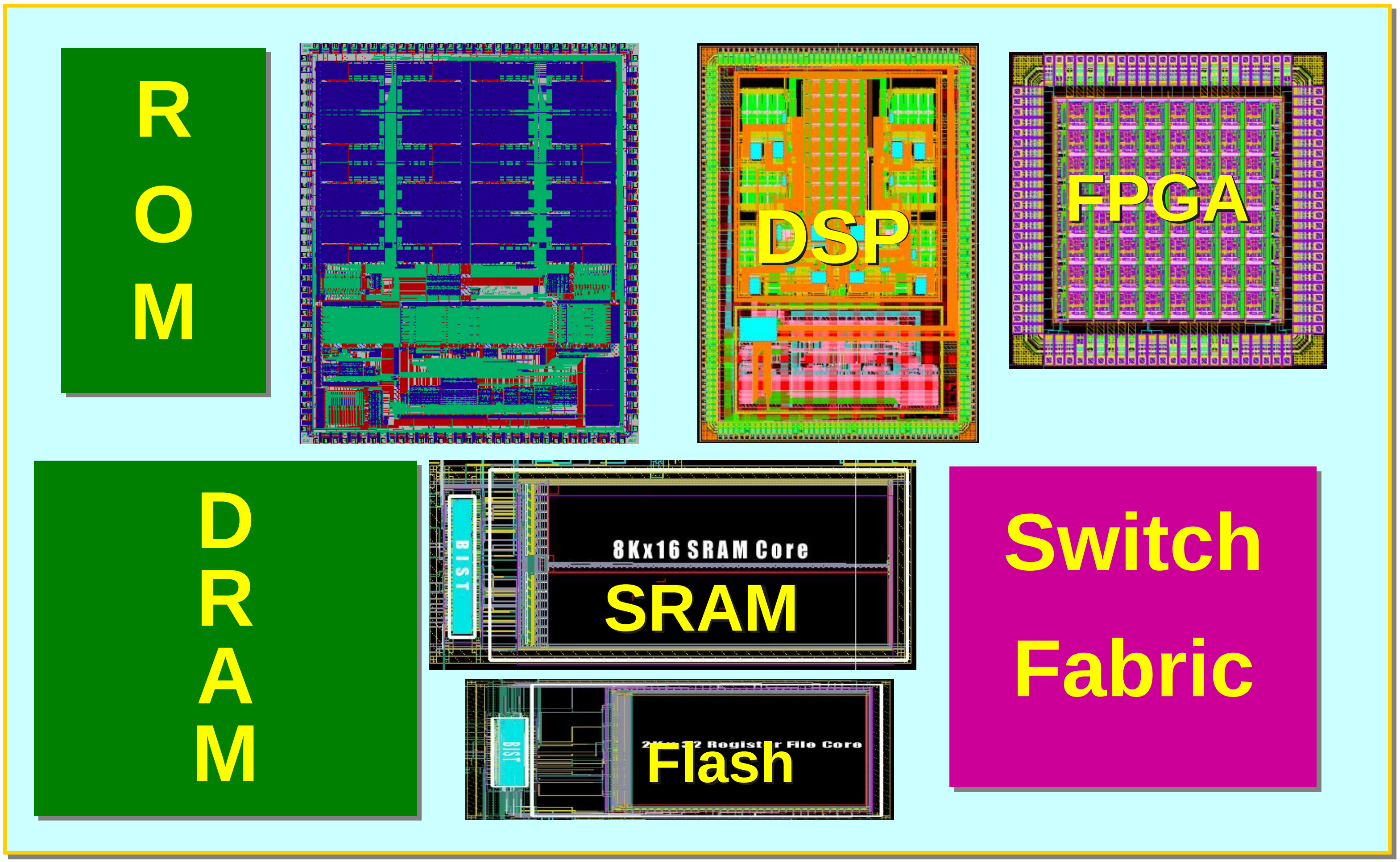


System-on-a chip is possible

(the whole system is built in

a single chip)

SoC Example



SoC Design

Common problems for SoC:

- Time-to-market pressures
- Quality of results, in performance, area and power
- Verification and testing becomes difficult
- Development team consists of experts of different areas
- SOC includes embedded processor cores and significant software (a key part)
- Mixed hardware and software, logic and memory, digital and analog circuits

⇒ Old design methodologies must change

⇒ Block-based design that emphasizes design reuse is necessary

- the use of pre-designed and pre-verified modules

- cores (a well designed circuit) become very attractive and important

IP (Silicon Intellectual Properties, 矽智產), Macro, Core, Block

Silicon Intellectual Properties (1/2)

- Intellectual Property
- Pre-design and pre-verified blocks or circuits
- Competitive in functionality, size, power, or performance
- Well characterized (Silicon verified)
- Integratable (documents, executable models, confirmed to standards)
- Test strategy
- Examples:
  - Processor: MIPS R-3000
  - Memory: single-port SRAM, dual-port SRAM, ROM
  - Communication: Pager decoder (POCSAG & FLEX)
  - Others: GPS decoder

Silicon Intellectual Properties (2/2)

- Soft IP or Soft Macro (“code”)
  - HDL description
  - Flexible, i.e., can be changed to fit into an application
  - Technology independent: may be re-synthesized across processes
  - Example: Processor cores
- Firm IP or Firm Macro (“code+structure”)
  - Gate-level netlist to be placed and routed
  - Library dependent
  - Example: GPS decoder & Pager decoder
- Hard IP or Hard Macro (“physical”)
  - Ready for “drop in”
  - Tuned finely based on block size
  - Include layout and timing (technology dependent)
  - Example: SRAM

A Possible SoC Design

A possible SoC Design:

processor :

8-bit 8051 to a 64-bit RISC...

memory :

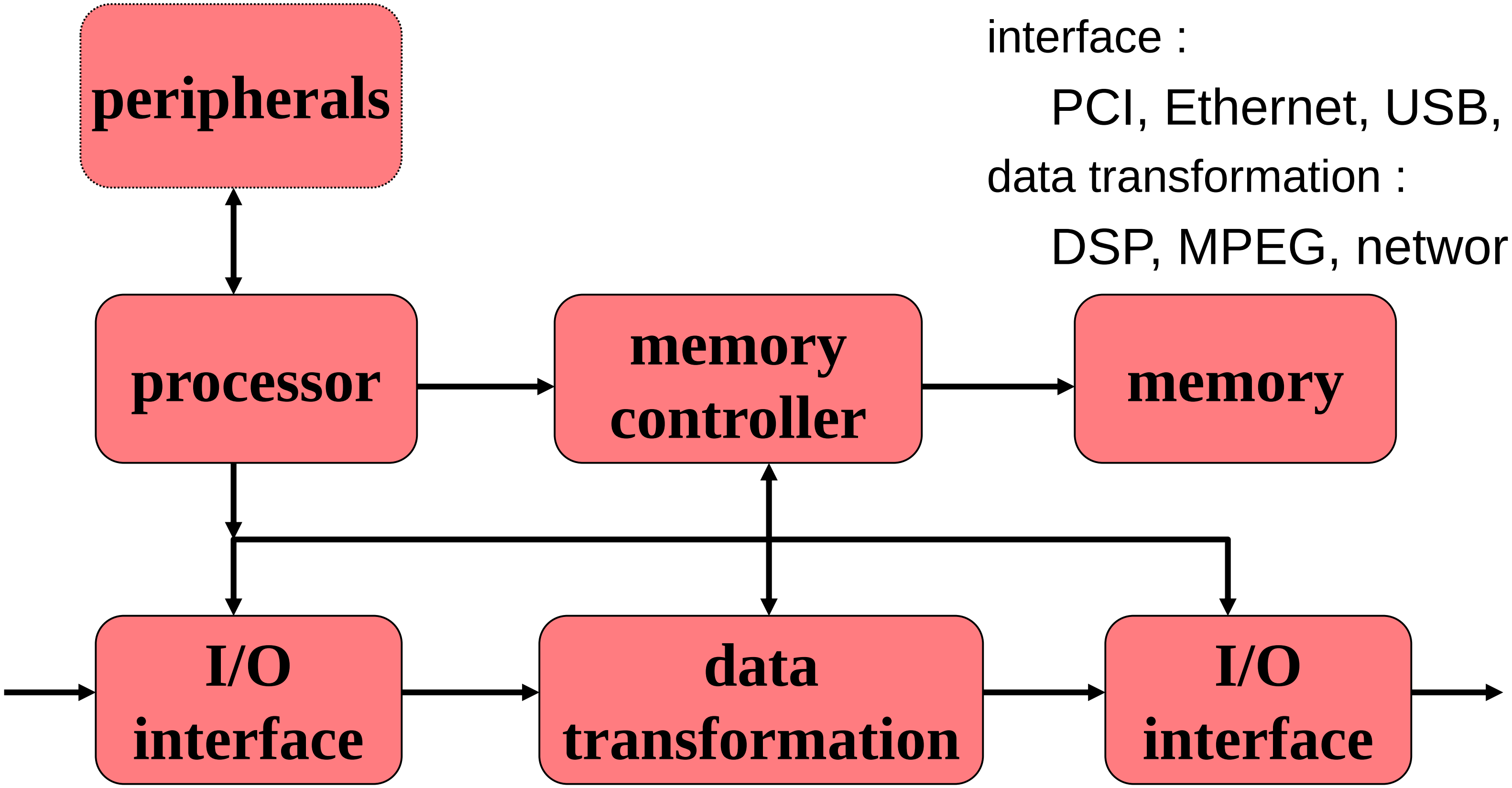
SRAM, DRAM...

interface :

PCI, Ethernet, USB, A/D, D/A...

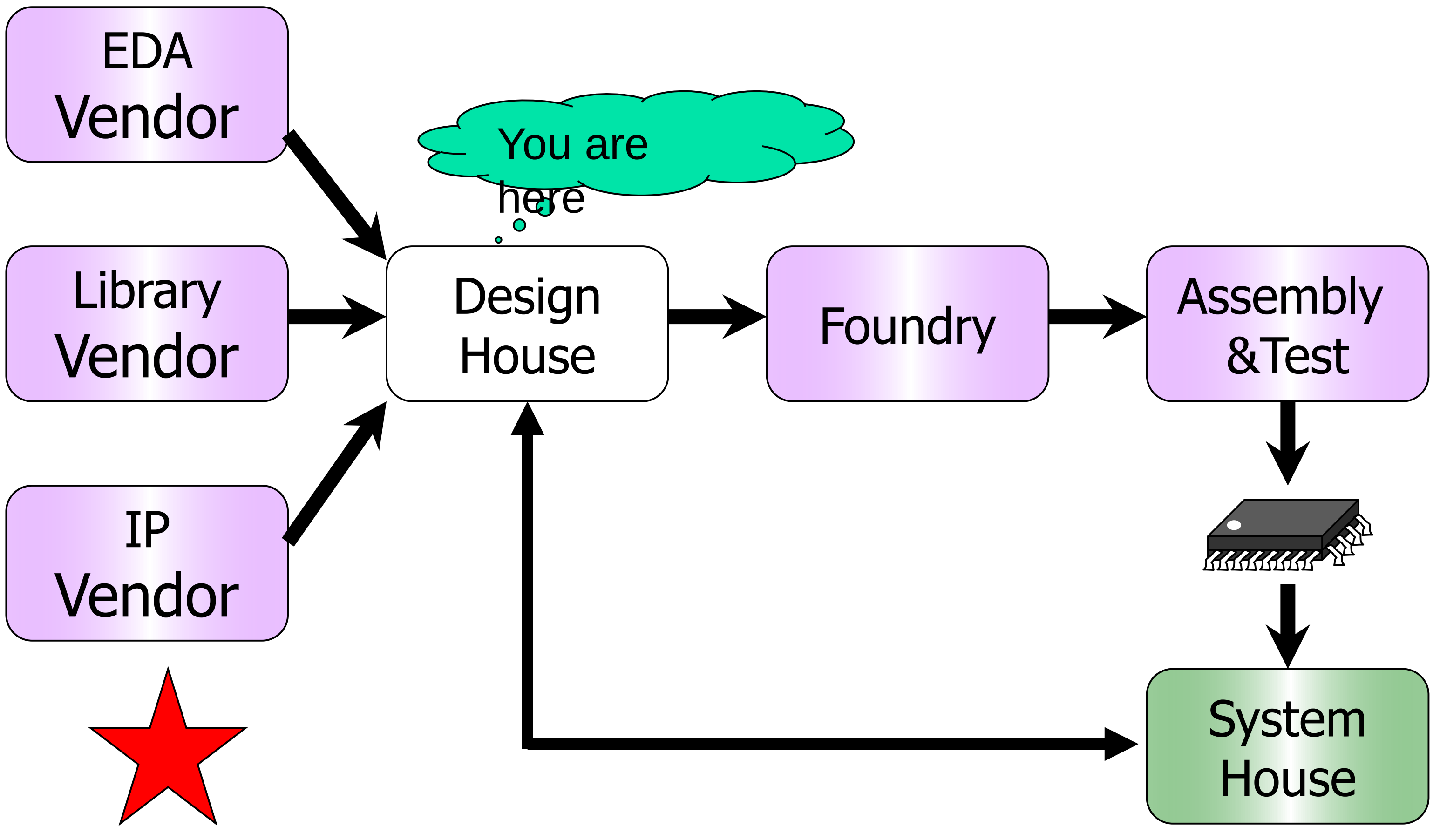
data transformation :

DSP, MPEG, network router...

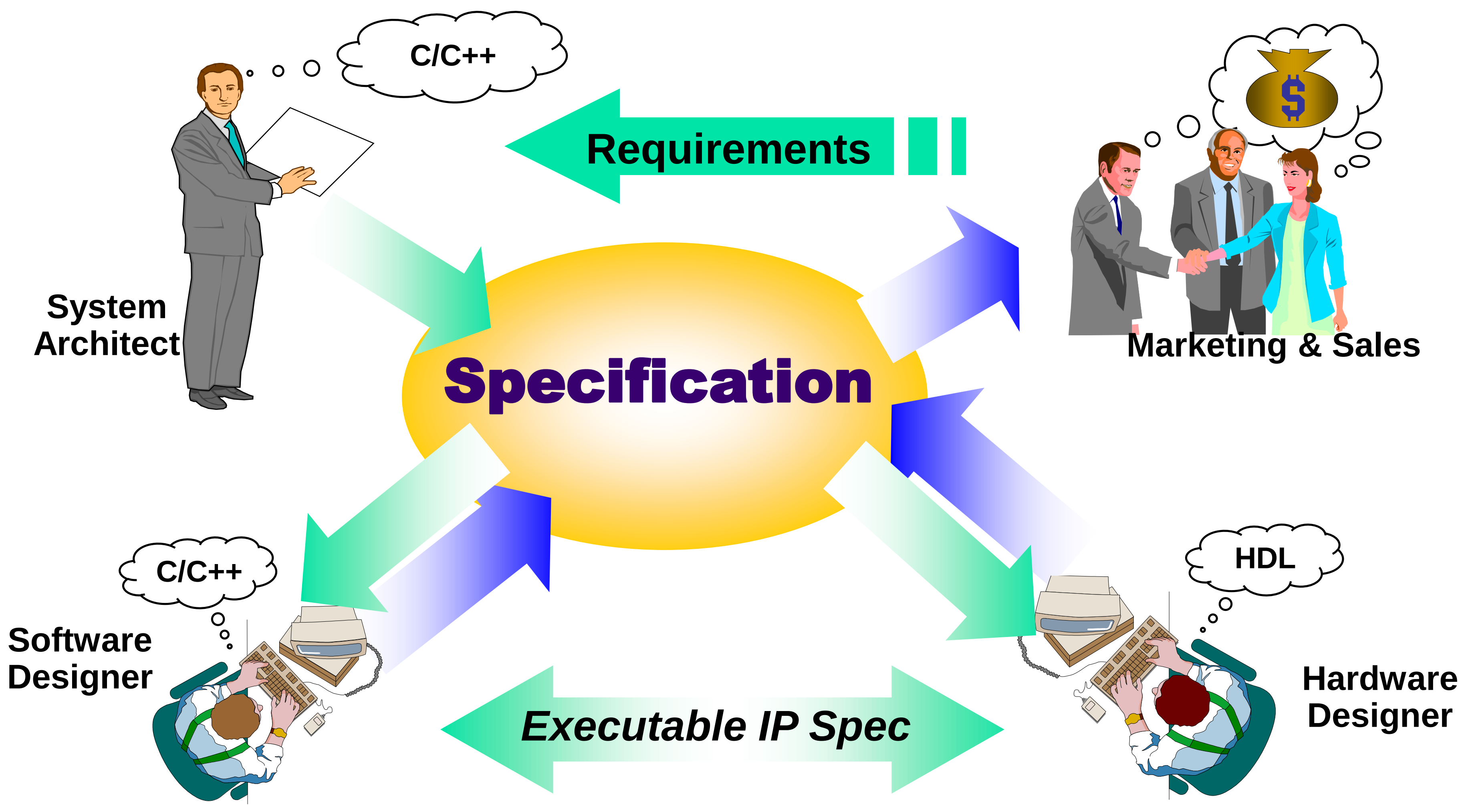


Each component might be a IP

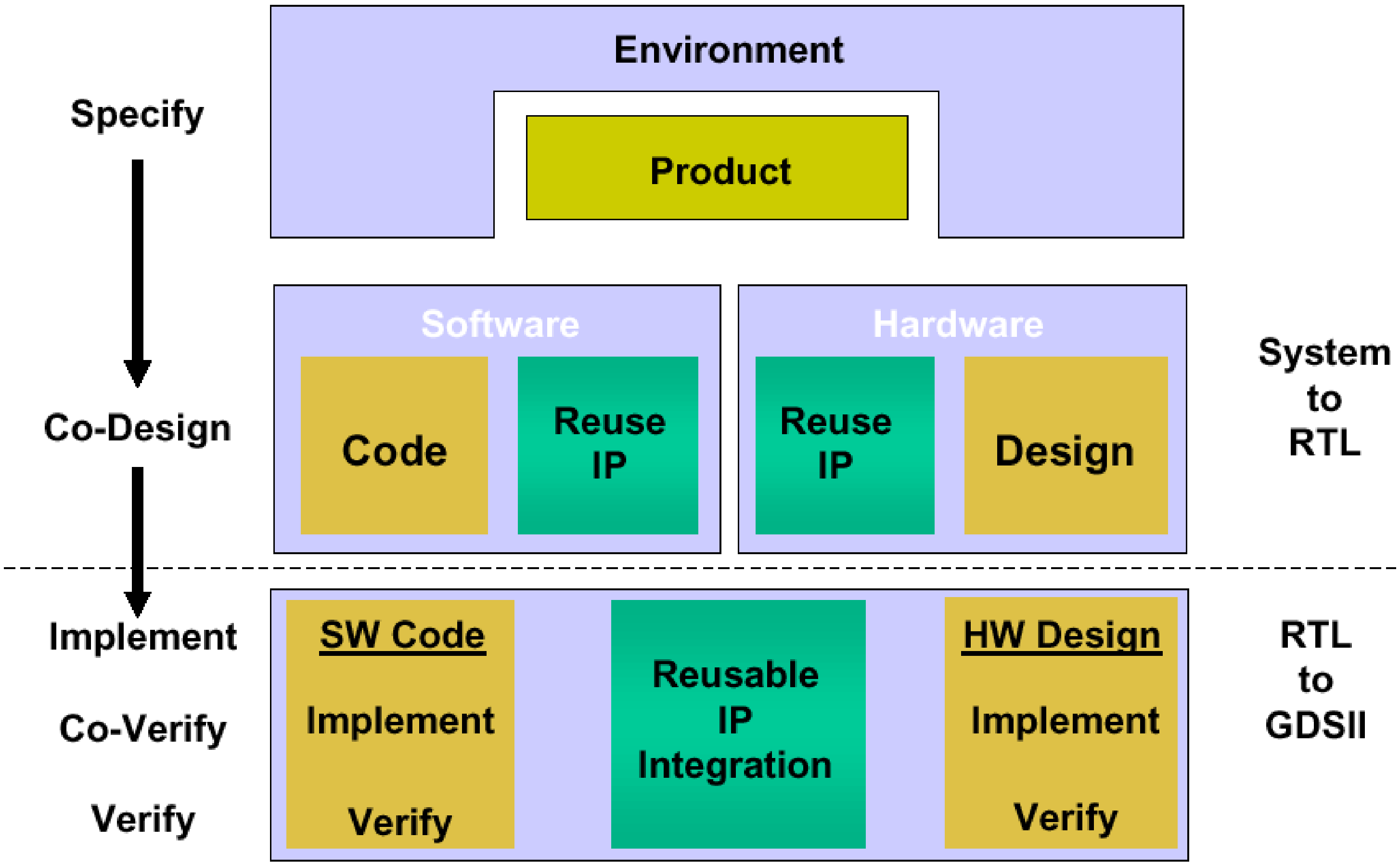
Design Flow of SoC (1/3)



Design Flow of SoC (2/3)



Design Flow of SoC (3/3)



"H/W and S/W development concurrently : functionality, timing, physical design, and verification"

Specification

Specification Problem

1. The first part of the design process
2. It is very difficult to develop a complete and clear spec. quickly for SOC design
3. Clear and early documenting is very important

Specification Requirements

Hardware:

1. Functionality
2. Timing
3. Performance
4. Interface to SW
5. Physical design issues such as area and power

Software:

1. Functionality
2. Timing
3. Performance
4. Interface to HW
5. SW structure and kernel

Key Points for Reusable Design

Design for use:

1. Good code
2. Good documentation and thorough commenting
3. Robust scripts
4. Well-designed verification environment

Design for reuse:

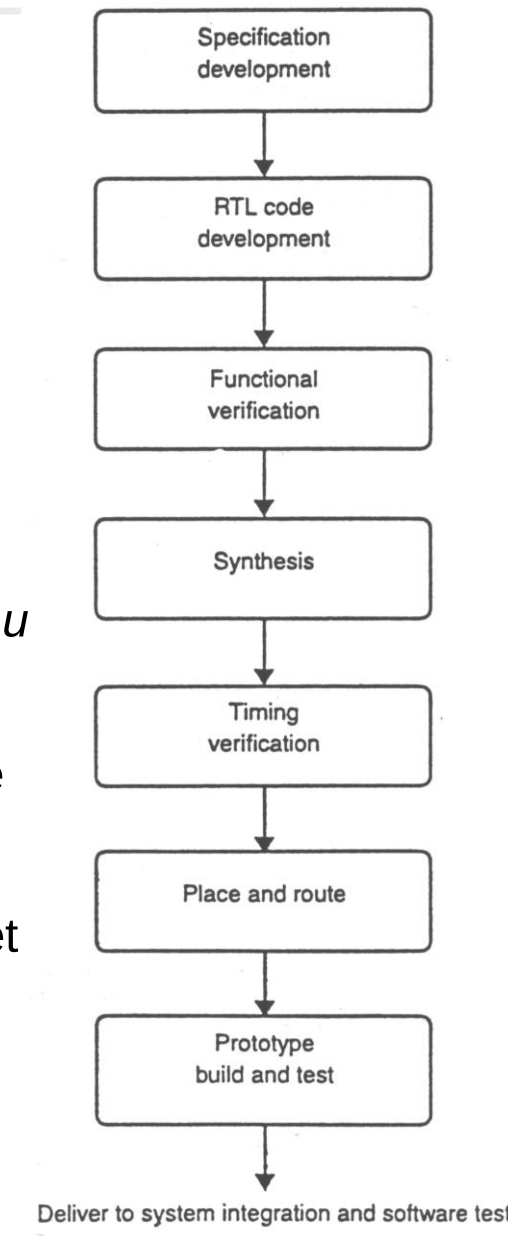
1. Robust design
2. Designed to solve a general problem
3. Designed for use in multiple technologies
   (soft IPs: for different libraries; hard IPs: for different technology)
4. Designed for simulation with different simulators (HDLs)
5. Verified independently
6. Fully documented in terms of appropriate applications
   and restrictions

System Design Flow

Waterfall vs. Spiral

Waterfall: the project transition from phase to phase in a step function, never returning to the activities of the previous phase

1. The whole process is done by different design teams
2. Work well in the design up to 100k gates and down to .5*u*
3. Work bad for large, deep submicron designs
4. Large systems must develop the hardware and software concurrently to ensure correct system functionality
5. Physical design issues must be considered early to meet the performance goals



Spiral Design Flow (1/2)

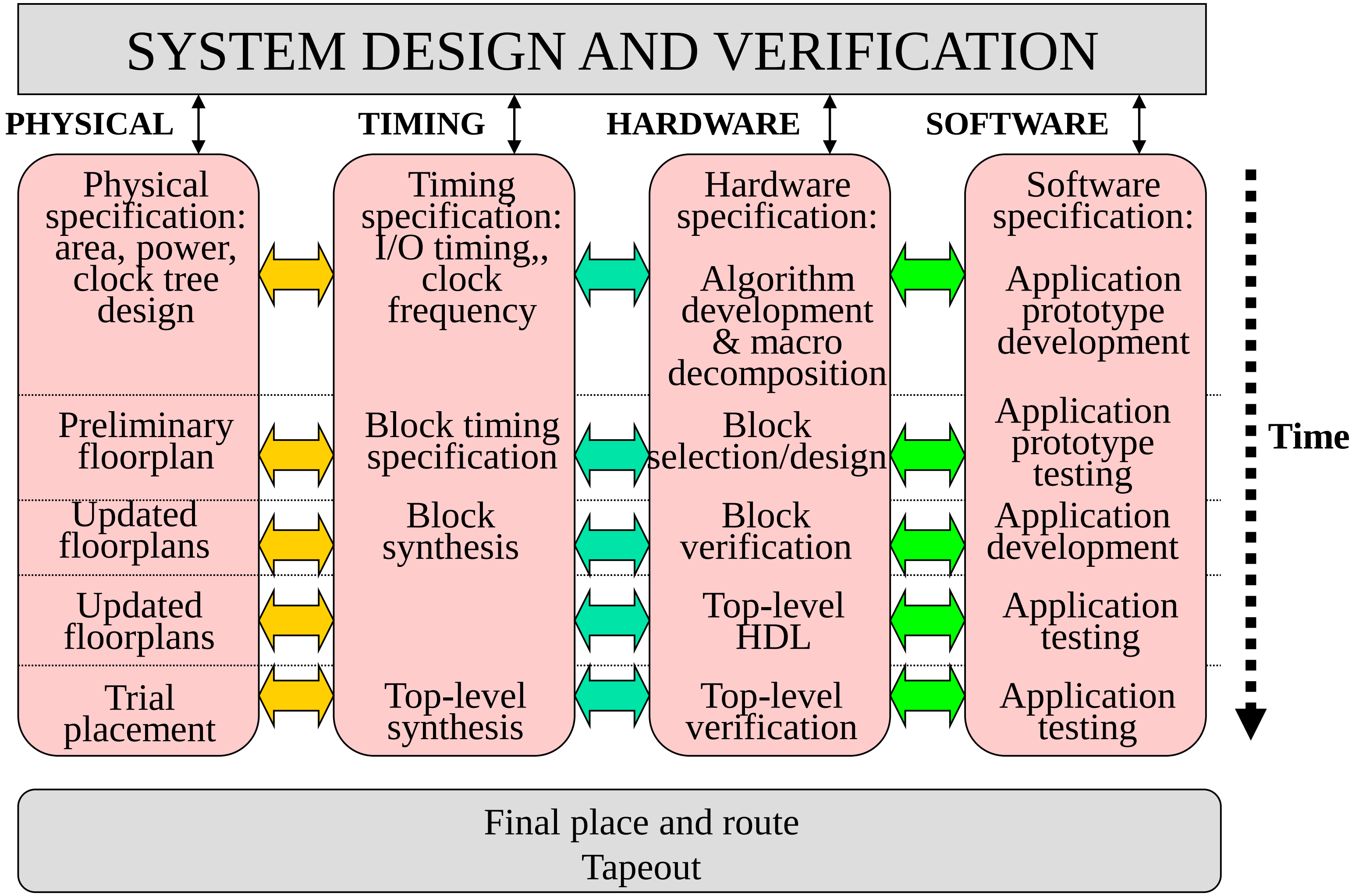
Spiral: work on multiple aspects of the design simultaneously,

incrementally improving in each area

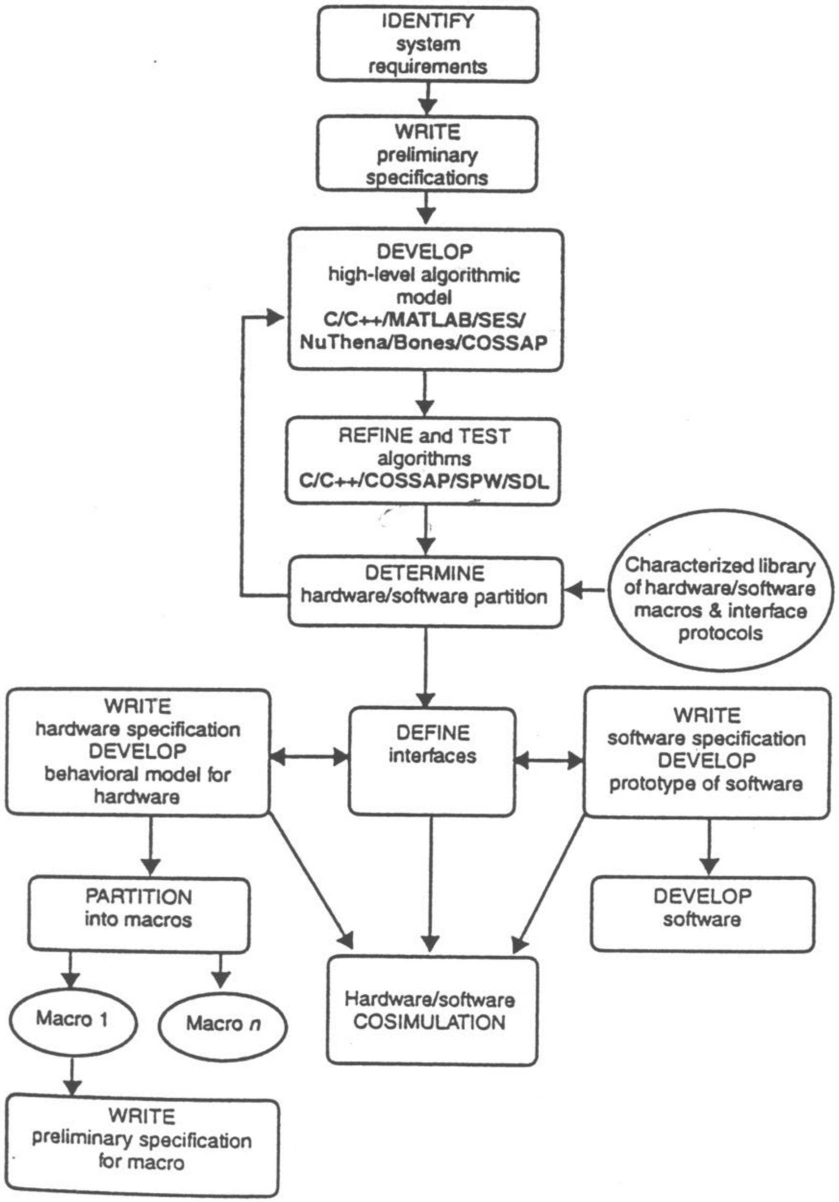
1. Parallel and concurrent development of hardware and software
2. Parallel verification and synthesis of modules
3. Floorplaning and routing in the synthesis process.
4. Modules developed only if a predefined hard or soft macro is not available.
5. Planned iteration throughout

To implement SOC, spiral development model is adopted. In other words, a mixture of top-down and bottom-up methodologies is used

Spiral Design Flow (2/2)



System Design Process



Most contents referred to “Reuse methodology manual for system-on-a-chip designs,” Kluwer Academic, *Michael Keating.*