OUTLINE

I - INTRODUCTION

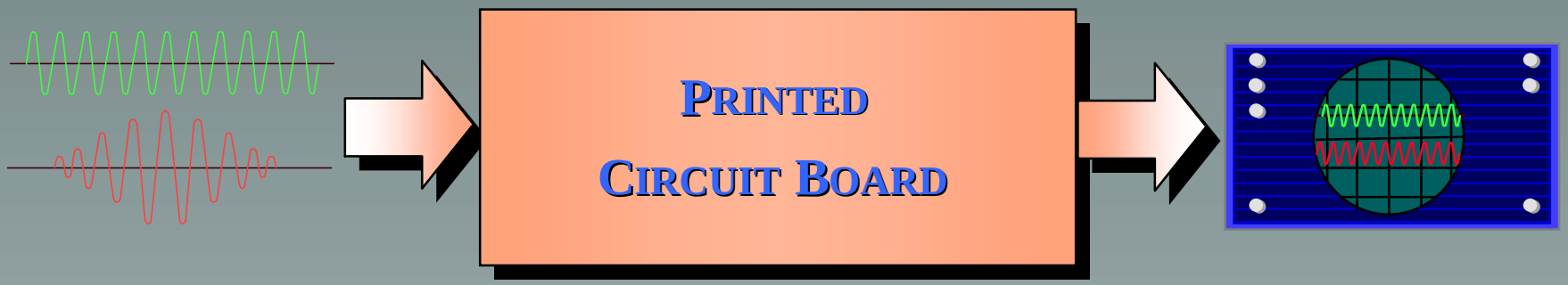
II - DESIGN METHODOLOGY: AN OVERVIEW

III - ABSTRACTION LEVELS IN ALLIANCE

IV - VHDL: A HARDWARE DESCRIPTION LANGUAGE

Why an HDL ? (1)

- Hardware Solutions Limits



Why an HDL ? (2)

- ✗ Increasing Complexity
- ✗ Increasing Cost in Time & Investment
- ✗ Increasing Knowledge Requirement

A Software Solution is Needed

Why an HDL ? (3)

✗ Programming Language not Suited

A Special Purpose Language : HDL

Why VHDL ? (1)

Circuit Manufacturers Fully Satisfied with their Proprietary HDLs...

Why VHDL ? (2)

Problems for system manufacturers

- ✗ Different vendors ➡ different incompatible HDLs
- ✗ Impossible to verify a whole mixed-system

Why VHDL ? (3)

- ✗ Vendor dependency
- ✗ Design documentation exchange

A Standard HDL from the System Manufacturer's Point of View: V H D L

VHDL

Very High Speed Integrated Circuits (VHSIC)

Hardware

Description

Language

History

- 1981: an Extensive Public Review (DOD)
- 1983: a Request for Proposal

  (Intermetrics, IBM, and Texas Instruments)
- 1986: VHDL in the Public Domain
- 1987: a Standard Language VHDL'87 (IEEE-1076)
- 1992: a New Standard VHDL'92

Advantages & Drawbacks

Standard  Open language

- ✔ Vendor independent
- ✔ User definable
- ✔ Wide capabilities

- ✘ Complex tools
- ✘ Slow tools

Abstraction Levels (1)

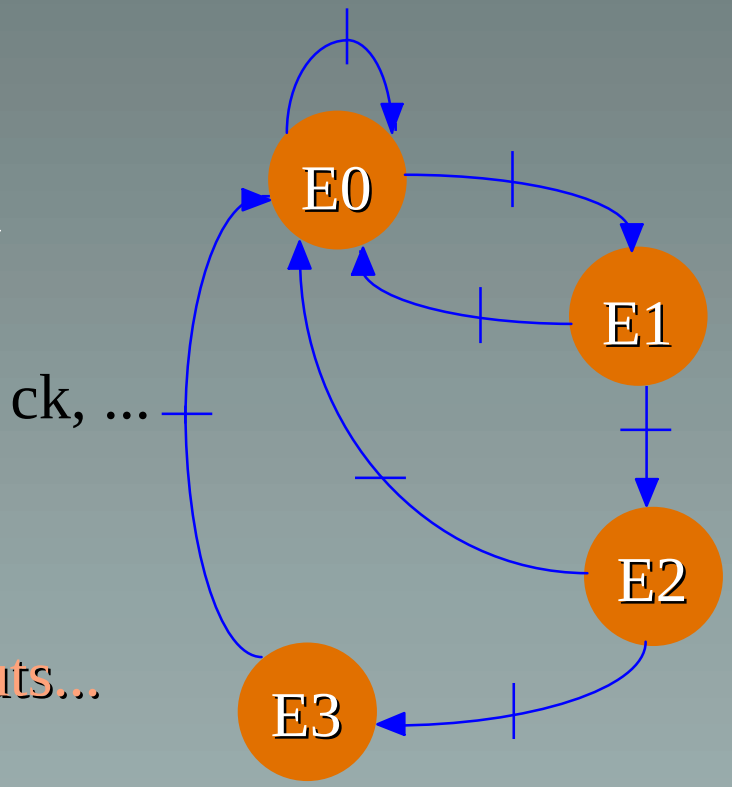
Algorithmic Level

- Very High Abstraction Level
- Functional Interpretation of a Discrete System
- No Implementation Details
- Sequential Program-Like Description
- Programmer's Point of View

Abstraction Levels (2)

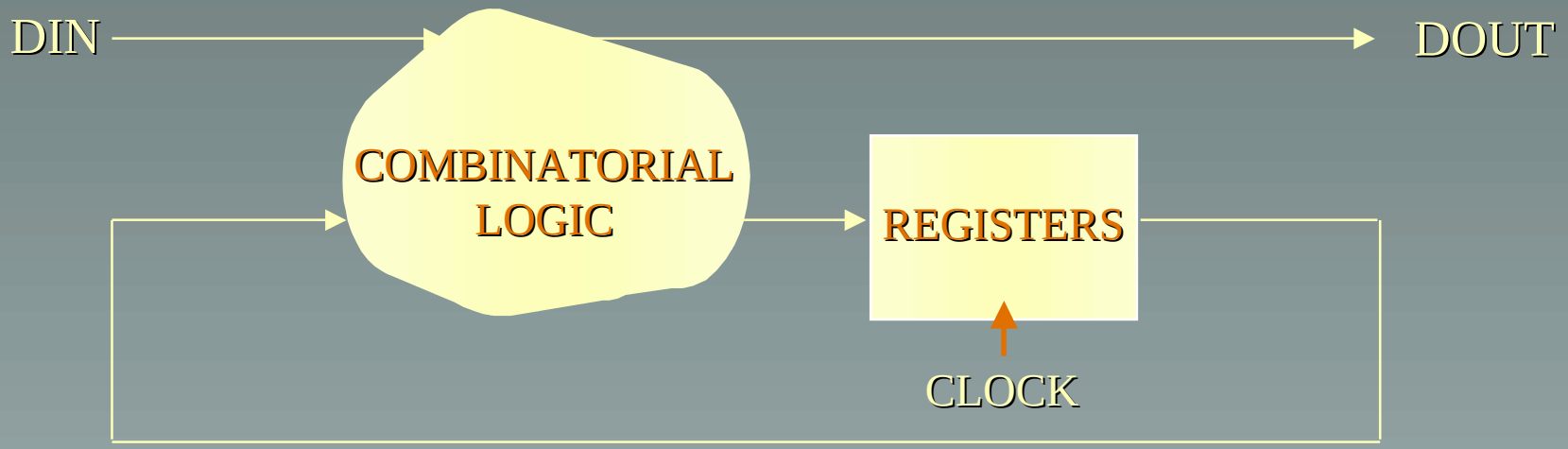
Finite State Machine Level

- Controller Part of a Digital Design
- Internal States
- State Changes Driven by:
  - Status Information
  - Clock and other External Inputs...



Abstraction Levels (3)

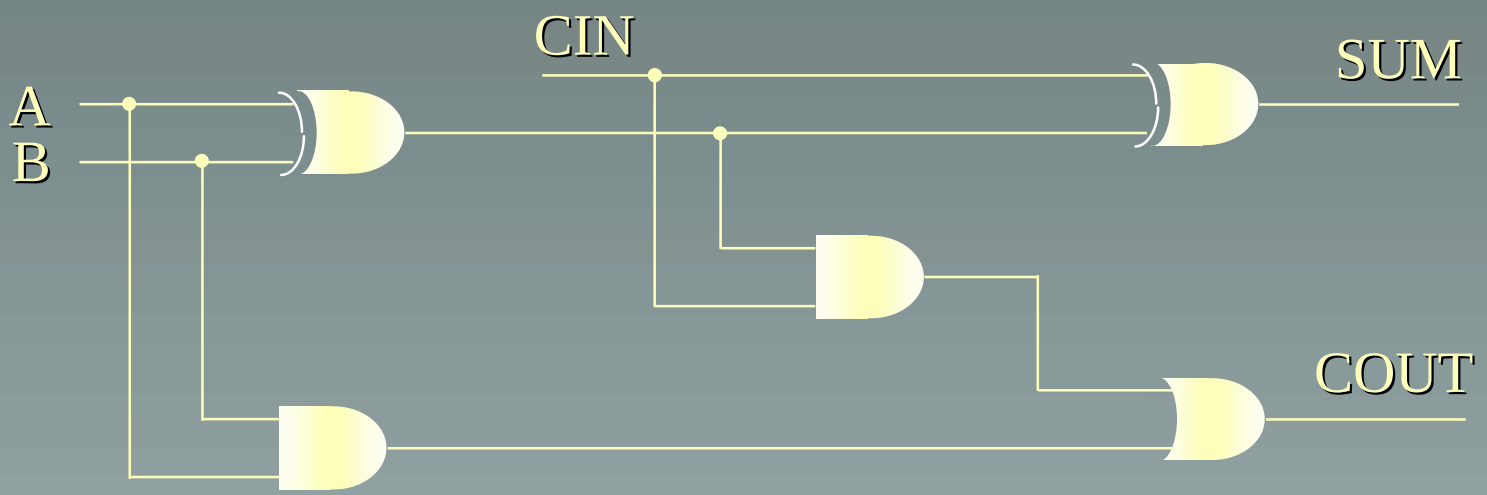
Register Transfer Level



- Registers Connected by Combinatorial Logic
- Very Close to the Hardware

Abstraction Levels (4)

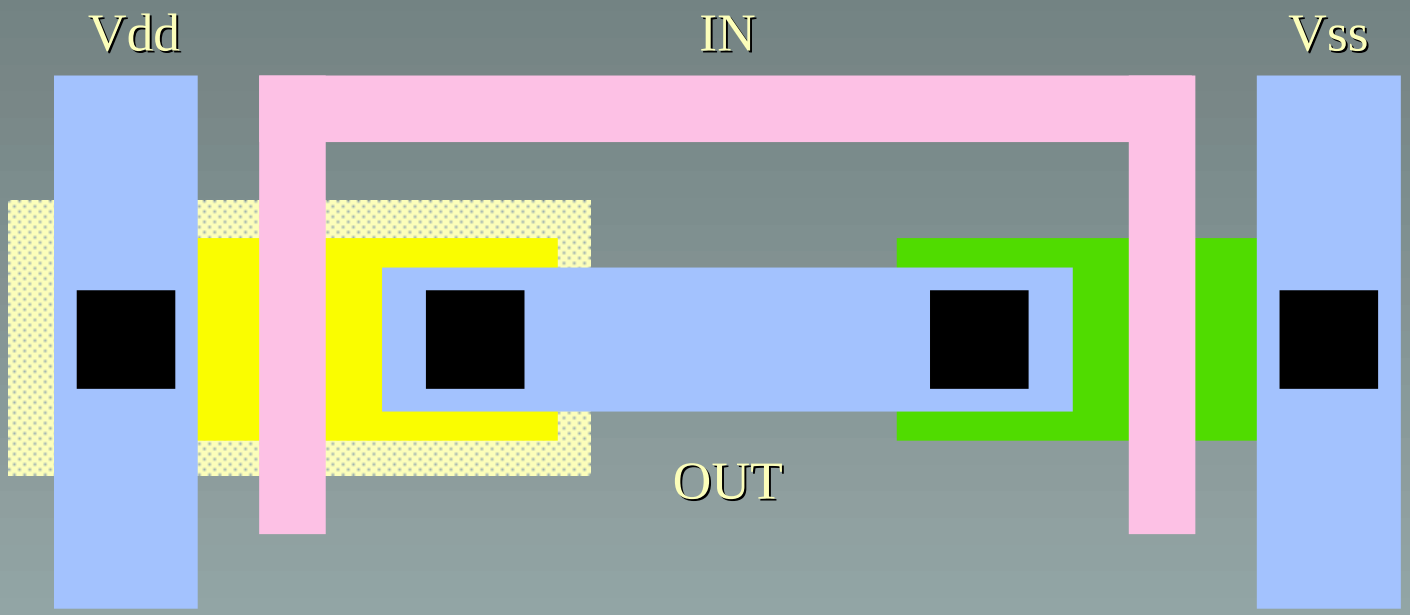
Gate Level



- A Gate Net-List Describing Instantiation of Models

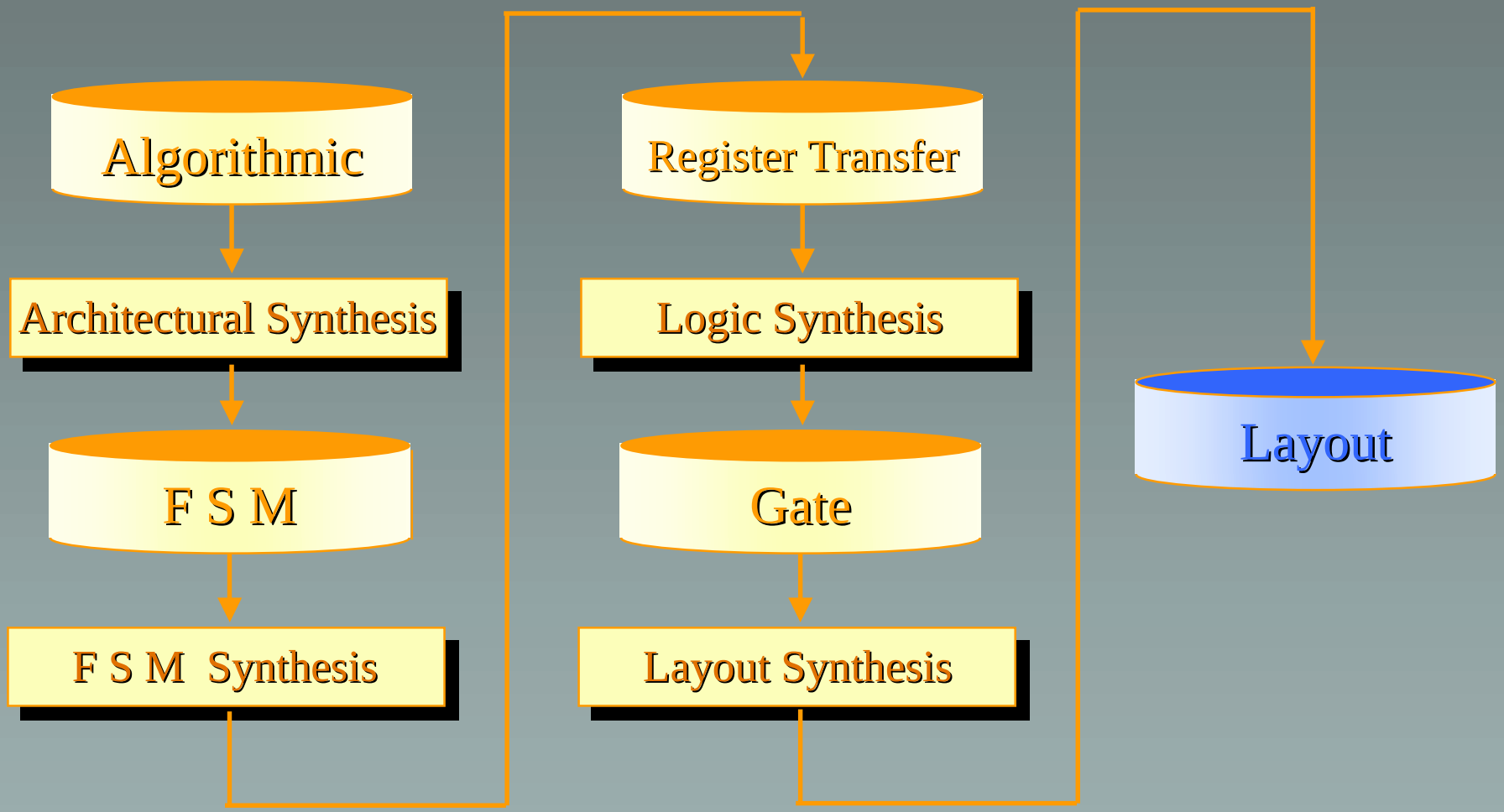
Abstraction Levels (5)

Layout Level

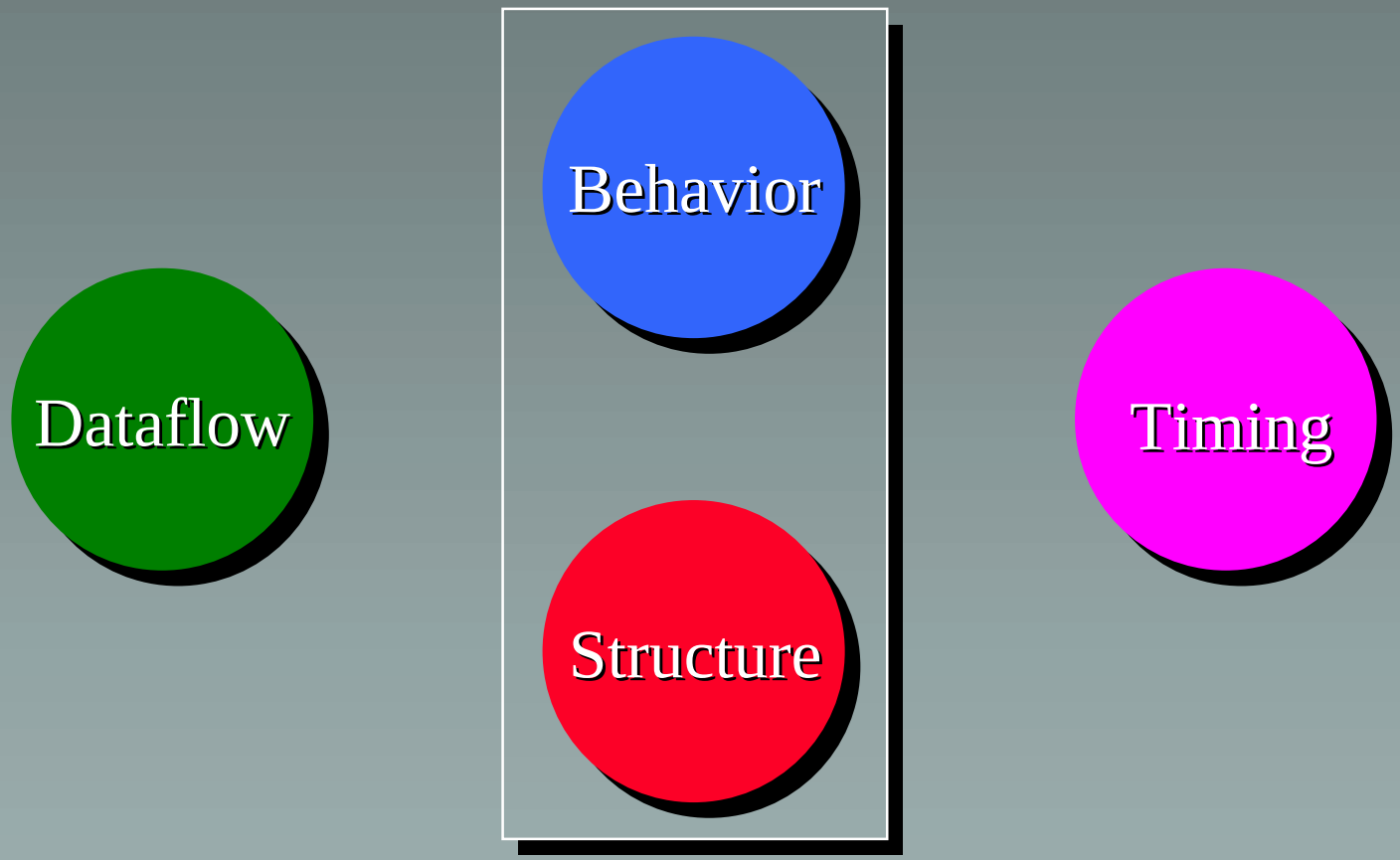


- A Set of Segments and Layers

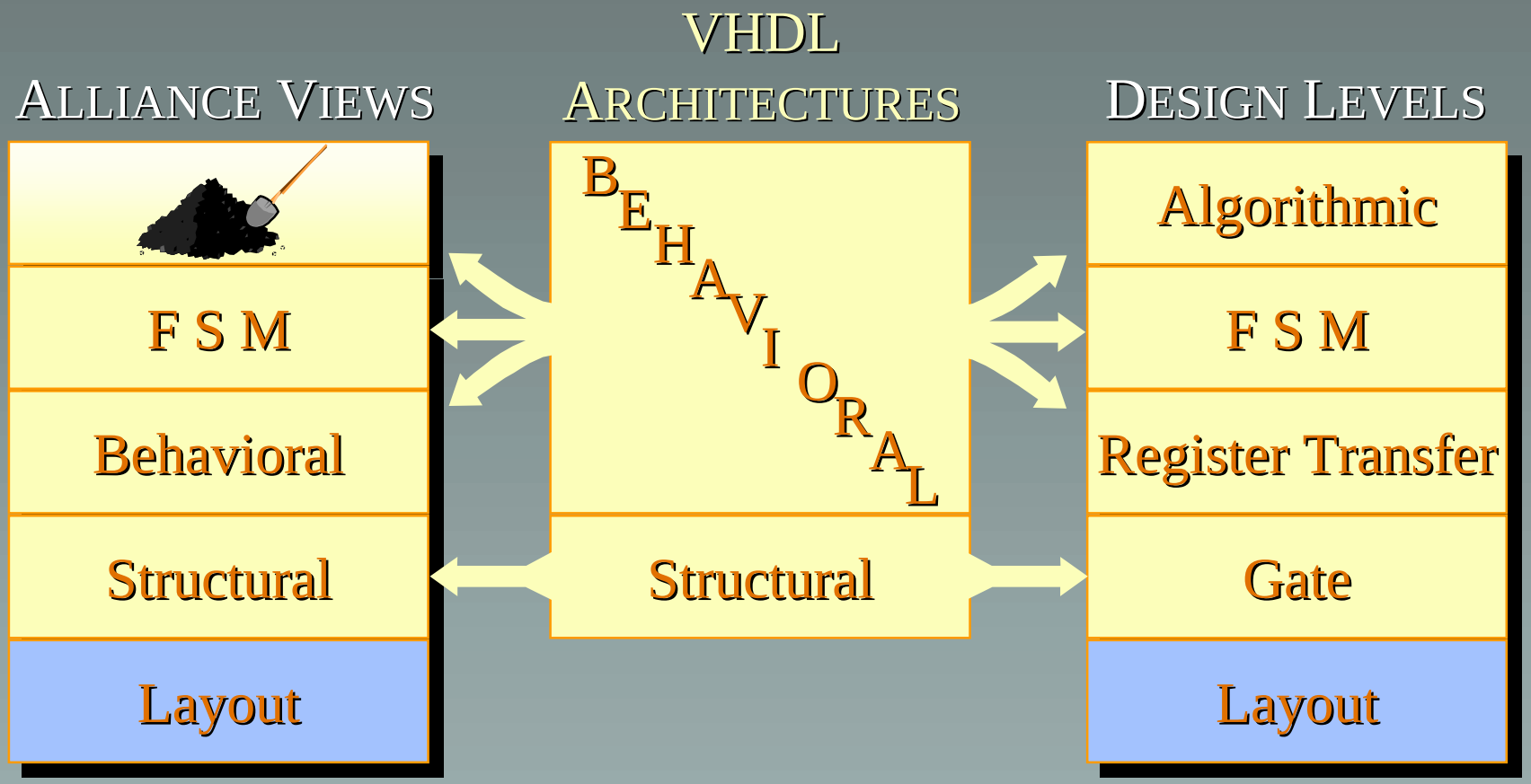
Synthesis Flow



VHDL Main Features



VHDL Architectures



A Dataflow Language (1)

CONTROLFLOW ≠ DATAFLOW

EX: C language assignment

X = A & B;

X is computed out of A and B ONLY each time this assignment is executed

EX: VHDL signal assignment

X <= A and B;

A PERMANENT link is created between A, B, and X

X is computed out of A and B WHENEVER A or B change

A Dataflow Language (2)

CONTROLFLOW ≠ **DATAFLOW**

EX: C language assignment

```
X = A & B;
------
X = C & D;
```

✔ YES

EX: VHDL assignment

```
X <= A and B;
------
X <= C and D;
```

✘ NO