



FLAME

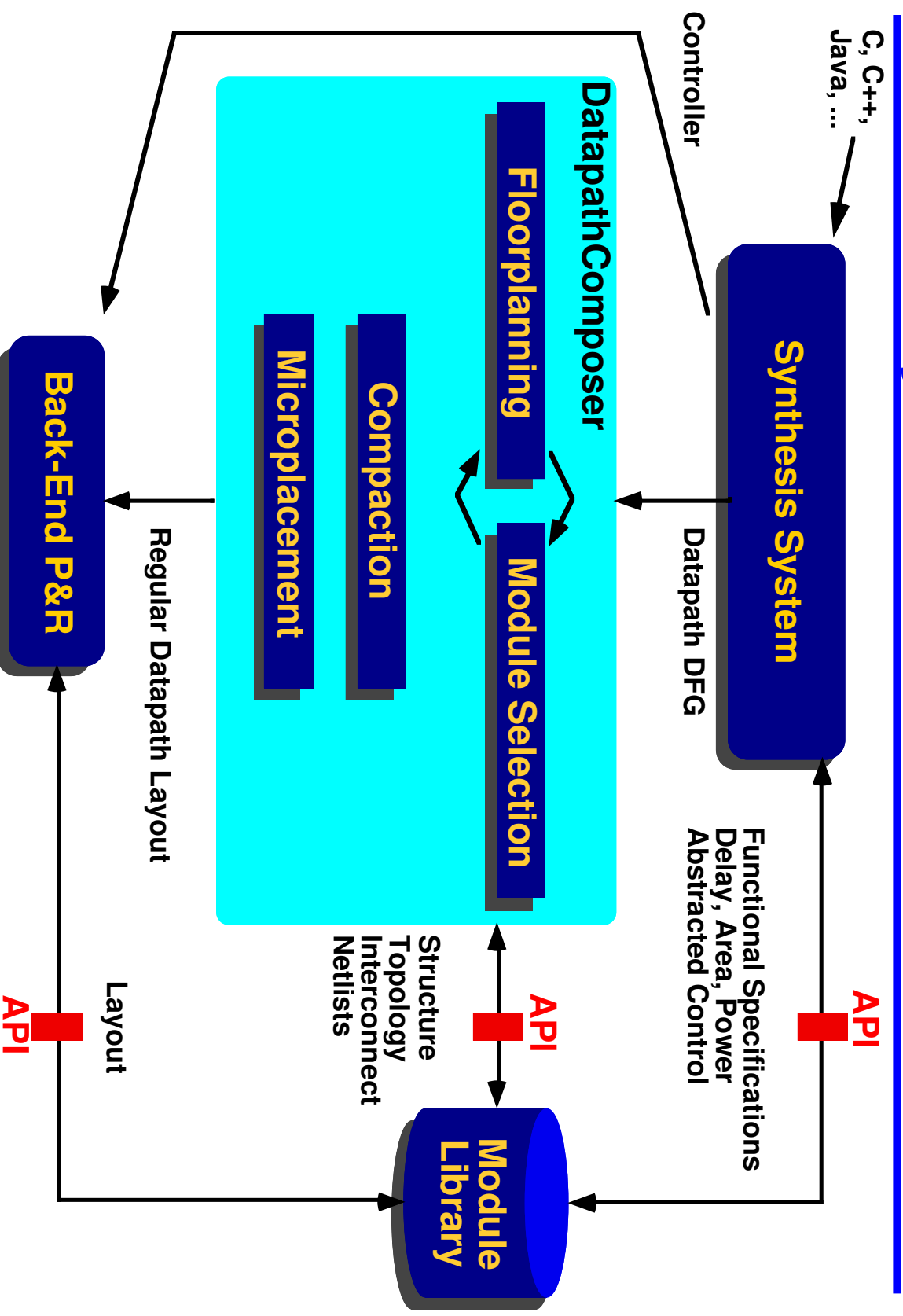
Flexible API for Module-based Environments

Andreas Koch
akoch@cs.berkeley.edu

Overview

- **System architecture**
 - ◆ **Modular**
 - ◆ **Open**
- **Flexible interfaces**
- **FLAME**
 - ◆ **Basics**
 - ◆ **Capabilities**
 - ◆ **Examples**
- **Summary**

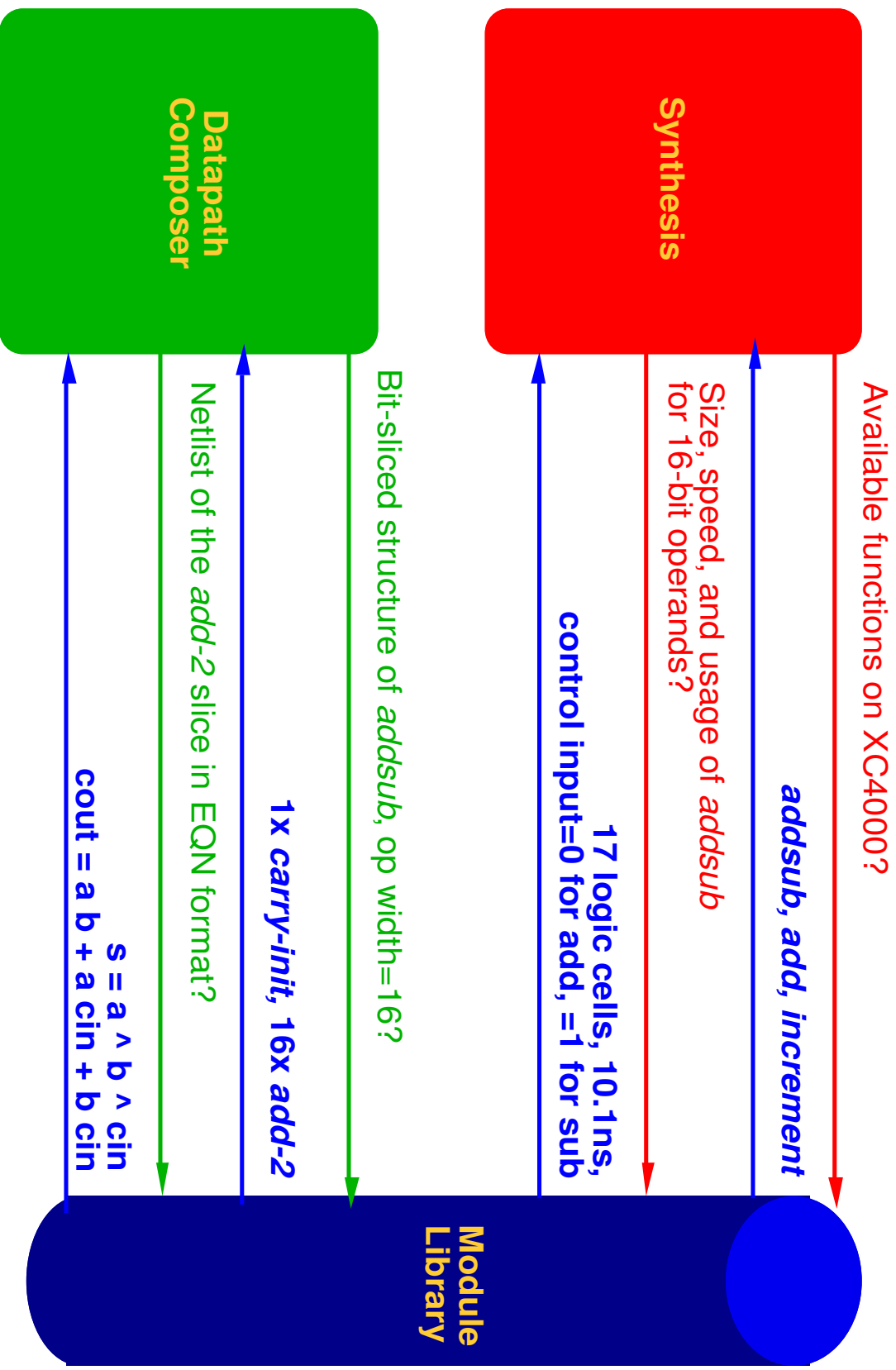
System Architecture



FLAME – Basic Concepts

- **Currently aimed at unified library access**
 - ◆ “Dynamic databook”
- **Active interface**
 - ◆ Query/Reply scheme
- **Queries constrained by**
 - ◆ Data types
 - ◆ Operand widths, ...
- **Replies use restricted views on data**
 - ◆ Functional
 - ◆ Synthesis
 - ◆ Topology, ...

Sample FLAME Scenario



FLAME Capabilities

- **Concise representations of**
 - ◆ **Module functions**
 - ◆ **Control interfaces**
 - ◆ **Timing**
 - ◆ **Area**
 - ◆ **Power**
 - ◆ **Internal composition**
 - **Regular bit-slices**
 - ◆ **Topology**
 - **Hierarchical and symmetric array**
 - **Logic and interconnect density**
 - ◆ **Embedded netlists and simulation models**

FLAME Properties

- **Flexible**
 - ◆ Extensible by introducing new parameters
 - ◆ Compatible with previous specifications
- **Easy-to-use**
 - ◆ Minimal initial requirements
 - ◆ Gradual refinement of specification
- **Portable**
 - ◆ Does not rely on specific language features
 - ◆ Single procedural entry point
 - ◆ Single data structure: associative list
- **Efficient**
 - ◆ Quick generation, parsing and manipulation
 - ◆ Direct tool integration possible

FLAME Example 1

Determine **library functions** for synthesis

```

|||||▶ (QUERY 1 1 (VIEW "functional")
      (TECHNOLOGY "Xilinx 4000"
        (DEVICE "XC4010"
          (SPEEDGRADE "-3")))))
      (REPLY 1 1 () (VIEW "functional"))|||||▶
      (TECHNOLOGY "Xilinx 4000"
        (DEVICE "XC4010"
          (SPEEDGRADE "-3"))))
      (GENERATOR "arith"
        (CELL "addsub"
          (FUNCTION ("X=A+B") ("X=A-B"))
            (INTERFACE (INPUT (("A" "B")))
              (OUTPUT (("X")))))
          (CELL "add"
            (FUNCTION ("X=A+B"))
              (INTERFACE (INPUT (("A" "B")))
                (OUTPUT (("X"))))))))

```


FLAME Example 2

Determine resource and control specifications

```
|||> (QUERY 1 2 (VIEW "synthesis")
      (TECHNOLOGY "Xilinx 4000" (DEVICE "XC4010" (SPEEDGRADE "-3")))
      (GENERATOR "arith"
      (CELL "addsub"
      (INTERFACE (WIDTH 16))))))

(REPLY 1 2 () (VIEW "synthesis") |||>
 (TECHNOLOGY "Xilinx 4000" (DEVICE "XC4010" (SPEEDGRADE "-3")))
 (GENERATOR "arith"
 (CELL "addsub"
 (INTERFACE (INPUT ((("A" "B") (WIDTH 16) (SIGNED))
                      (("ADDSUB")(CONTROL)))
                    (OUTPUT ((("X") (WIDTH 16) (SIGNED))))
 (IMPLEMENTATION "hardcarry-1"
 (FUNCTION ("X=A+B" (UCODE (LEVEL ("ADDSUB") 0)))
           ("X=A-B" (UCODE (LEVEL ("ADDSUB") 1))))
 (TIME (REQUIRED () 0 0)
        (ARRIVAL () 0 0 101))
 (AREA ("CELLS"17))))))
```

Further Work

- **Complete API specification**
- **Implement support library**
 - ◆ **Parsing**
 - ◆ **Manipulation**
 - ◆ **Generation**
- **FLAME wrapper for current generators**

Summary

- **System architecture**
 - **Need for powerful interfaces**
- **Flexible API for Module-based Environments**
 - ◆ **Properties**
 - ◆ **Capabilities**
- **Examples**
 - ◆ **Unified library access**
- **Directions for future work**