NFMW 1996, Ilkley

A Formally Based Framework for Supporting Design and Analysis of Asynchronous Hardware Systems

Howard Barringer Donal Fellows Graham Gough
Pete Jinks Alan Williams

Department of Computer Science,
University of Manchester

September 1996

THE UNIVERSITY
of MANCHESTER

Contents

- Background
- Rainbow Design Framework
- Example: Simple Processor
- Formal Semantics
- Summary

Background

Hardware design:

- enhanced performance, functionality
- speed-to-market

Need:

- fast prototyping
- simulation
- early design analysis

Asynchronous design:

- increased speed
- reduced power
- easier design (compositional)

AMULET microprocessor (Steve Furber, Manchester)

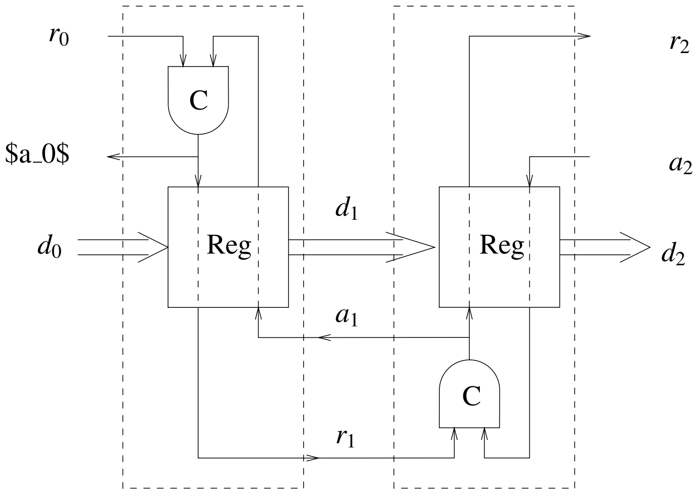
Problems facing asynchronous design:

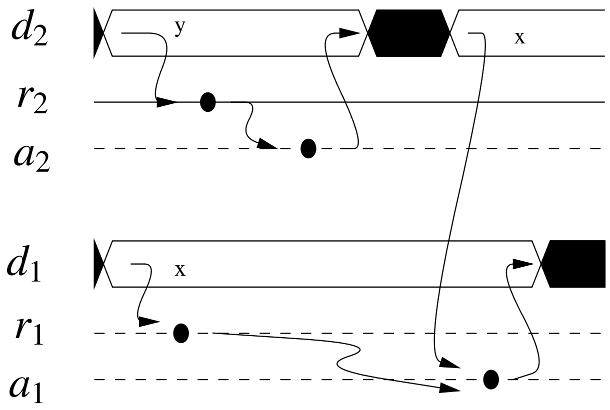
- lack of specialised design tools
- design analysis

Require:

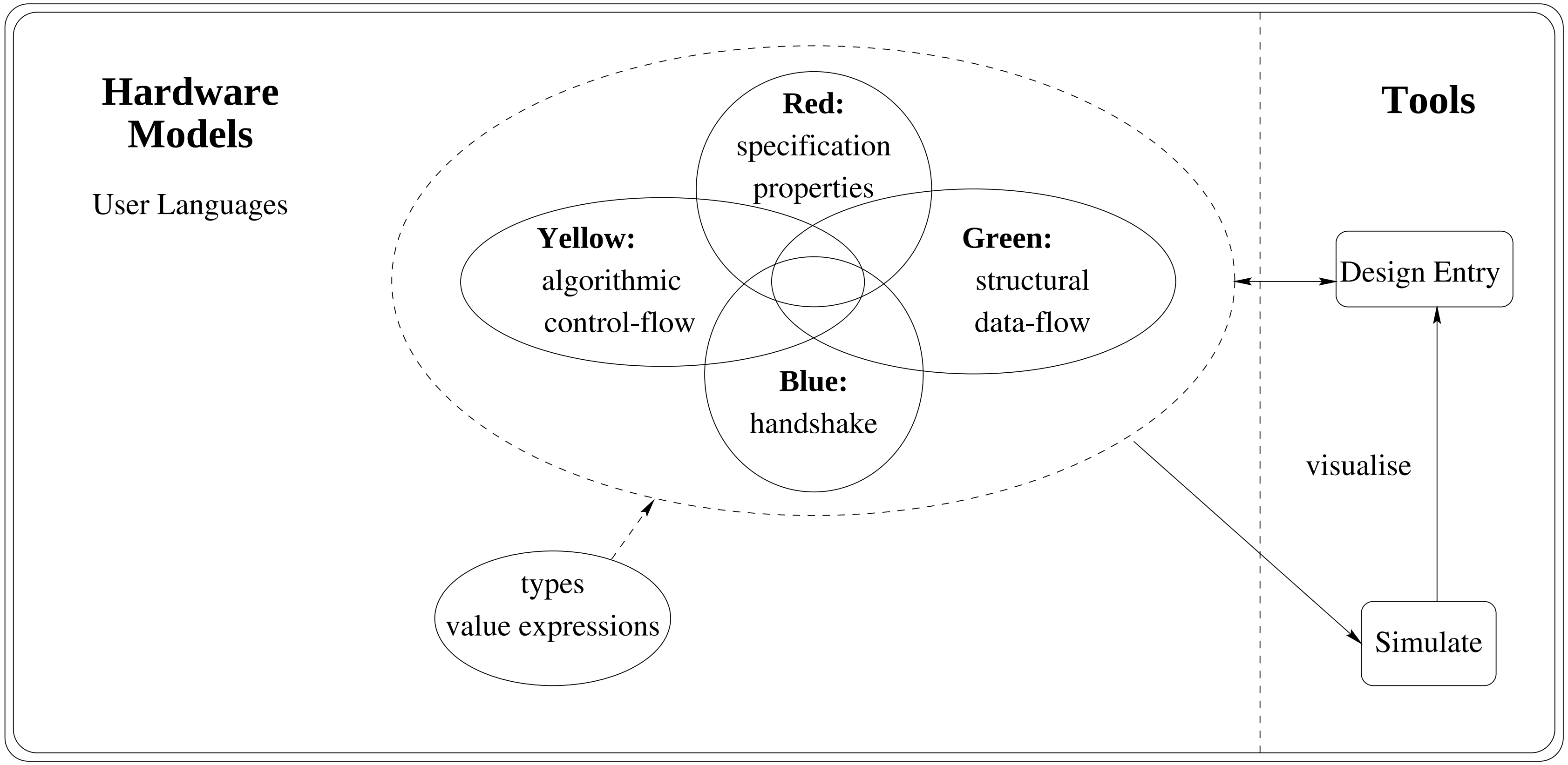
- design methodology
- suitable design representation
- analysis tools
  - deadlock detection
  - high level power, speed estimation
  - functional verification

Micropipelines

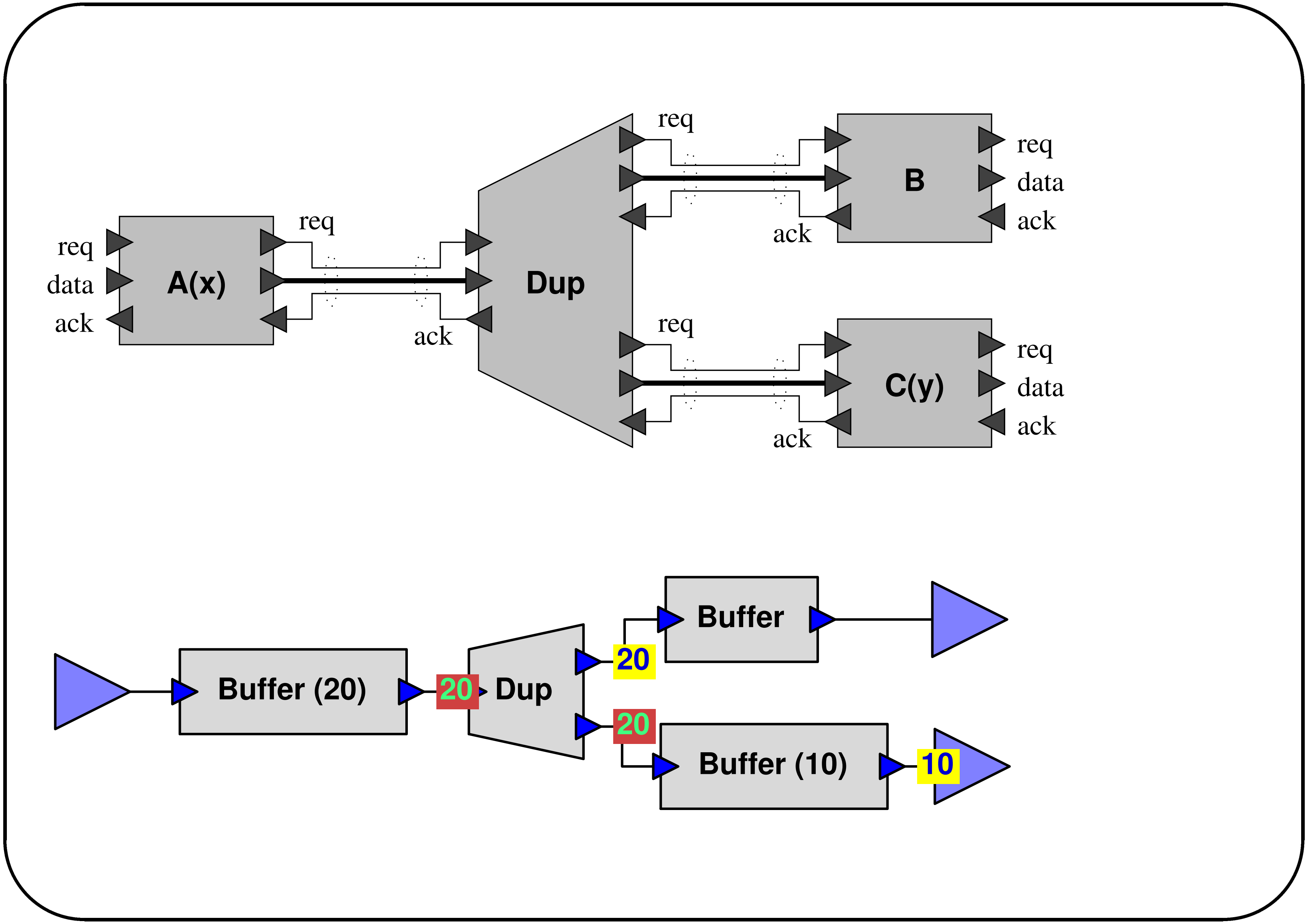




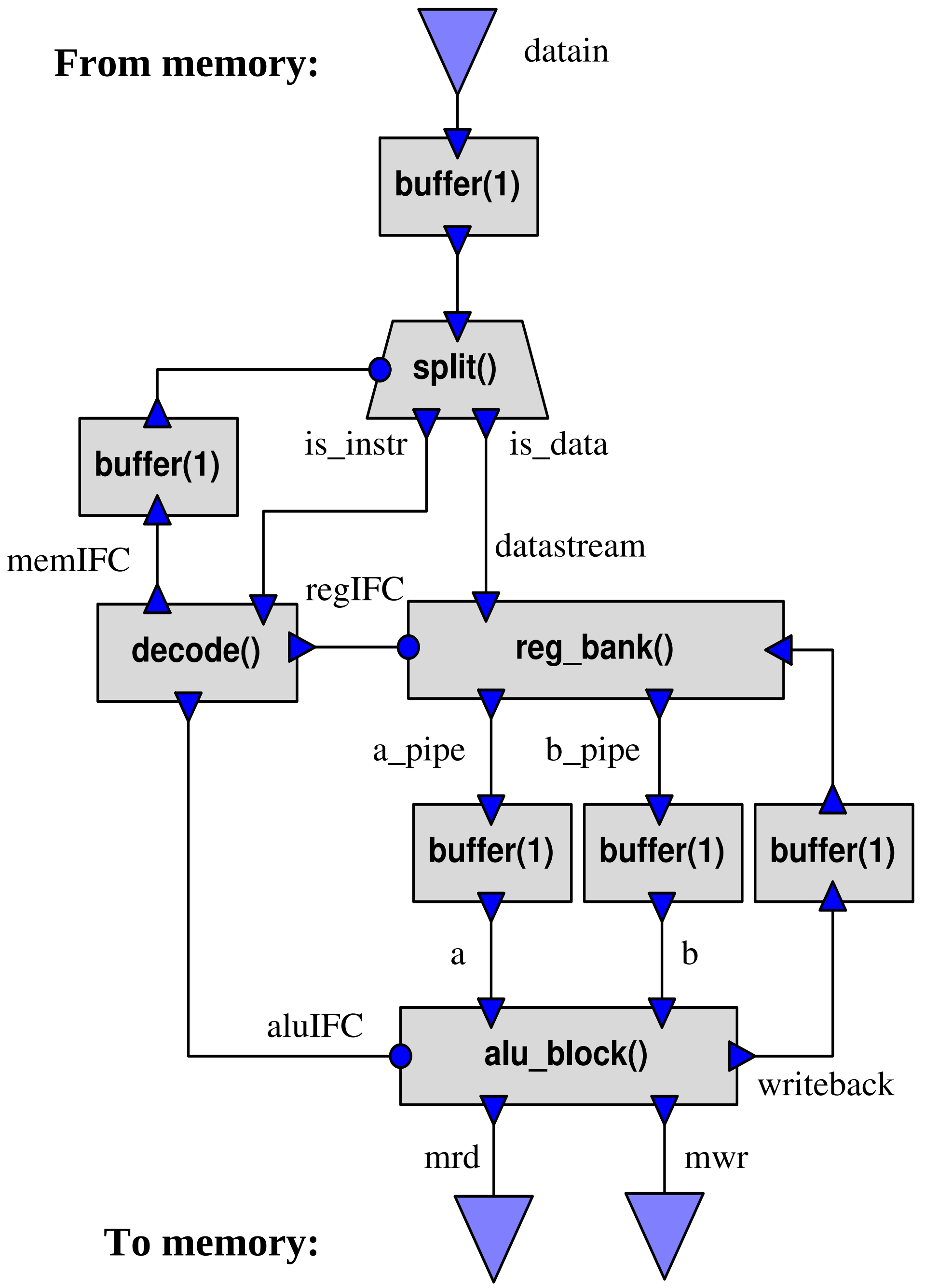
Rainbow Design Framework



- User-oriented language design (AMULET)
- Abstract representation of bundled data
- Mixed-view modelling
- Hierarchical design
- Application-specific


req
data
ack
A(x)
req
ack
Dup
req
ack
B
req
data
ack
req
ack
C(y)
req
data
ack
Buffer (20)
20
Dup
20
Buffer
20
Buffer (10)
10

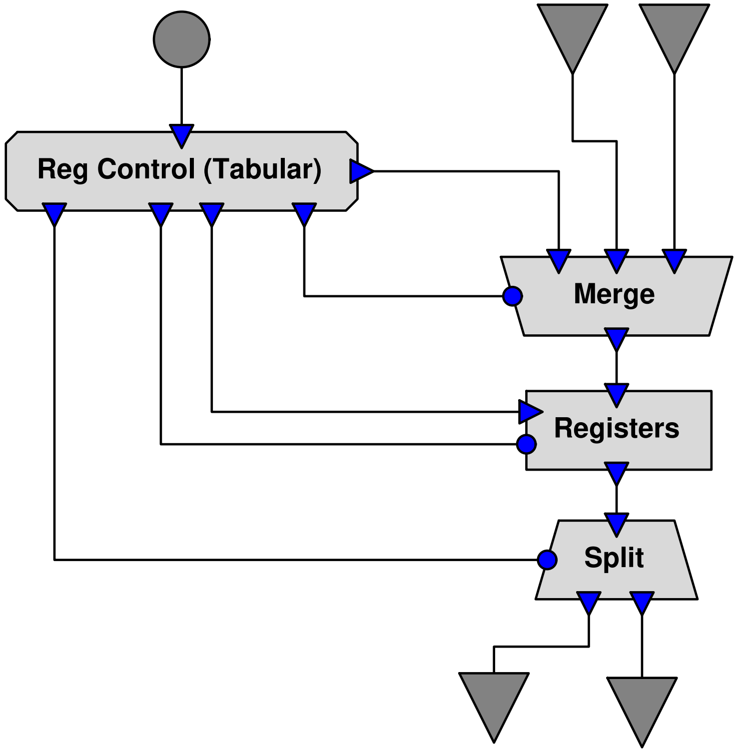
Example: Simple Processor



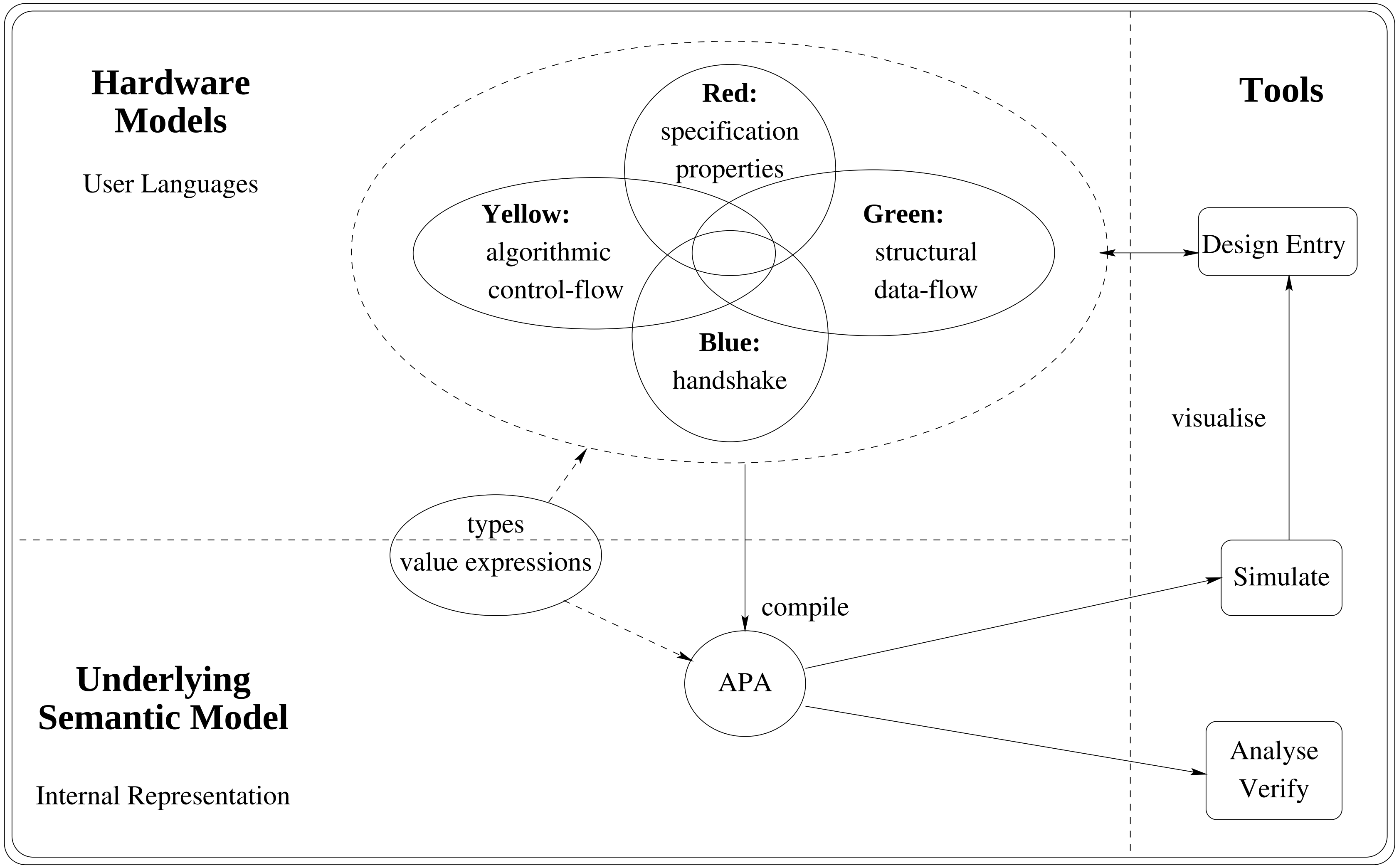
Green Table — ALU:

```
table alu_block (input  aluIFC: aluctrl;
                 input  a,b: val;
                 output mrd: memread;
                 output mwr: memwrite;
                 output writeback: val) = {
    FETCHINC, a,   -  =>  a,    -,      a+1
 or FROMMEM,  a,   b  =>  a+b,  -,      -
 or TOMEM,    a,   b  =>  -,    (a,b),  -

 or ADD,      a,   b  =>  -,    -,      a+b
 or SUB,      a,   b  =>  -,    -,      a-b
 or AND,      a,   b  =>  -,    -,      a/\b
 or CMP,      a,   b  =>  -,    -,      a>b

 or MOV,      a,   -  =>  -,    -,      a
 or NOT,      a,   -  =>  -,    -,      not a
}.
```

- Yellow Decoder: `fetch(); execute(); store()`
- Green sub-network: register bank (textual or schematic)



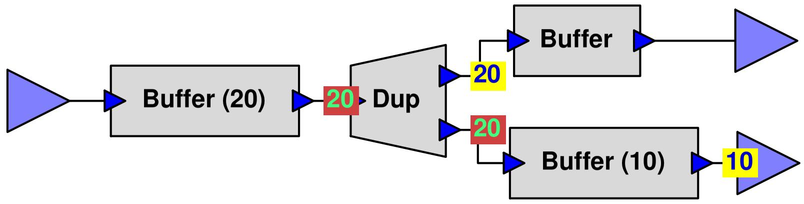
Formal Semantics



APA

- developed from EPA (ELLA Process Algebra)
- complex actions
- bundled dataflow blocks as basic constructs
- Ada-like rendezvous communication primitive
- Rainbow sub-languages map to APA
- transition rule define APA semantics

Example:



$$\dfrac{\mathcal{B}[20] \xrightarrow{[o!20]} \mathcal{B}[20] \qquad \textsf{Dup} \xrightarrow{[i?20]\ /\ o_1!20} \textsf{Dup}_{o_2!20}}{\mathcal{B}[20]\ >>\ \textsf{Dup} \xrightarrow{o_1!20} \mathcal{B}[20]\ >>\ \textsf{Dup}_{o_2!20}}\ \boxed{\textsf{Pipe}}$$

$$\mathcal{B}[\,]\mid\mathcal{B}[10] \xrightarrow{i_1?20} \mathcal{B}[20]\mid\mathcal{B}[10]$$

$$\mathcal{B}[20]\ >>\ \textsf{Dup}\ >>\ \mathcal{B}[\,]\mid\mathcal{B}[10]$$
$$\xrightarrow{\varepsilon} \mathcal{B}[20]\ >>\ \textsf{Dup}_{o_2!20}\ >>\ \mathcal{B}[20]\mid\mathcal{B}[10]$$

Summary

- micropipeline design methodology
- rainbow Design framework with interworking
- abstract modelling
- application-specific methods
- prototype tool implementation
  - Visual Green editor/simulator
  - Rainbow expression compiler/evaluator
  - embedded into extensible design environment
- formal semantics
- analysis

URL:

`http://www.cs.man.ac.uk/fmethods/projects/AHV-PROJECT/ahv-project.html`