ESS'96, Genoa, Italy

Design and Simulation of Asynchronous Micropipeline Circuits in Rainbow

Howard Barringer   Donal Fellows   Graham Gough
Pete Jinks   Brian Marsden   Alan Williams

Department of Computer Science,
University of Manchester

October 1996

(supported by the EPSRC, UK)

THE UNIVERSITY
of MANCHESTER

Contents

- Background
- Rainbow Design Framework
- Simulation
- Hierarchy
- Current Work
- 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)

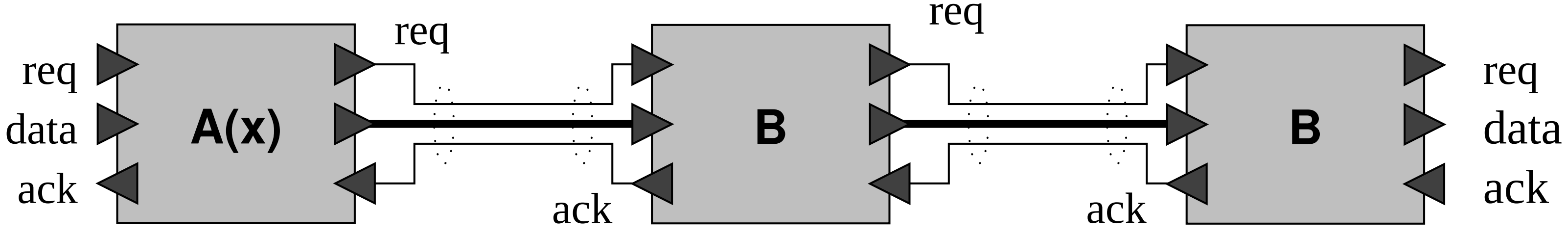
Problems facing asynchronous design:

- lack of specialised design tools
- design analysis

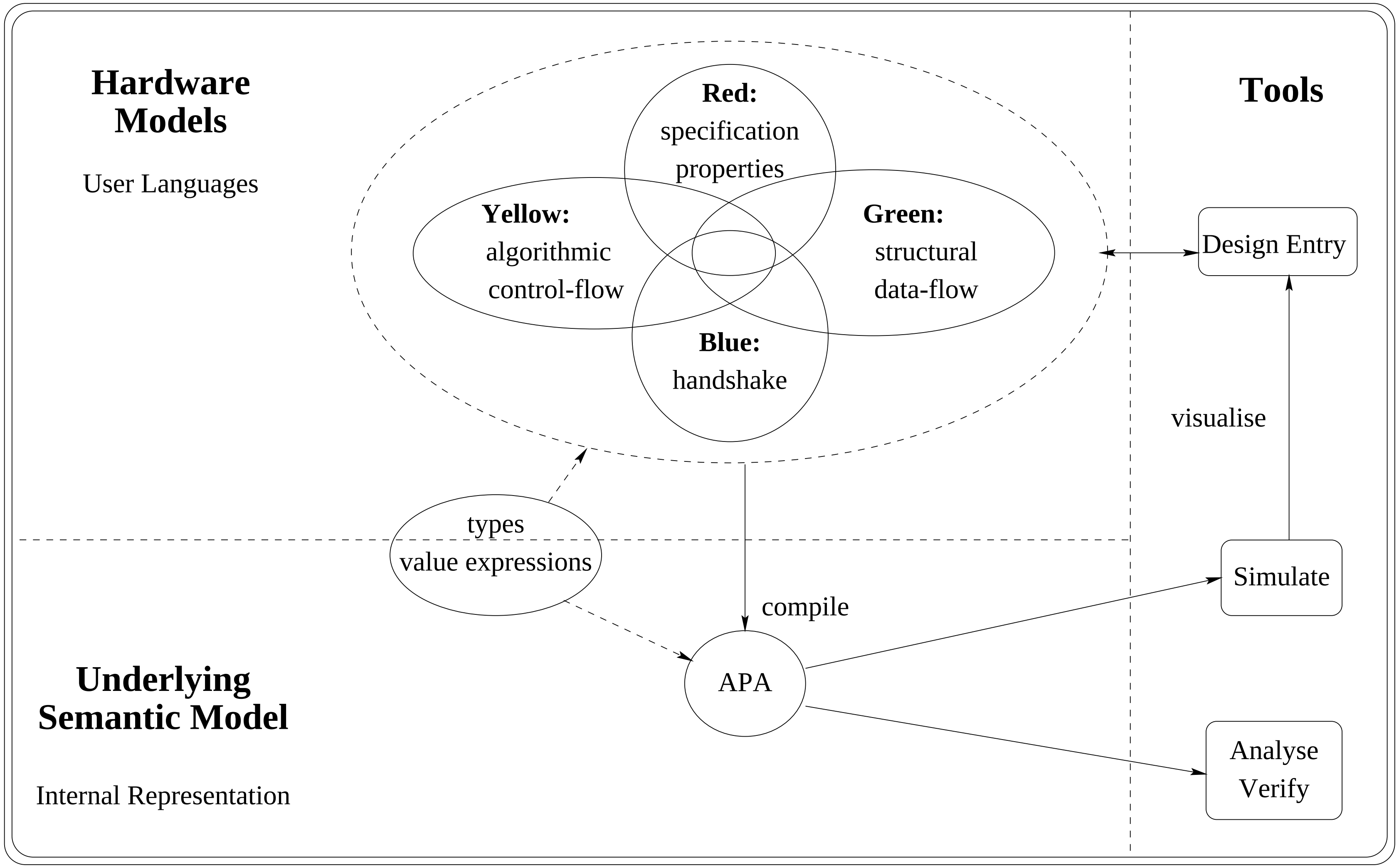
Require:

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

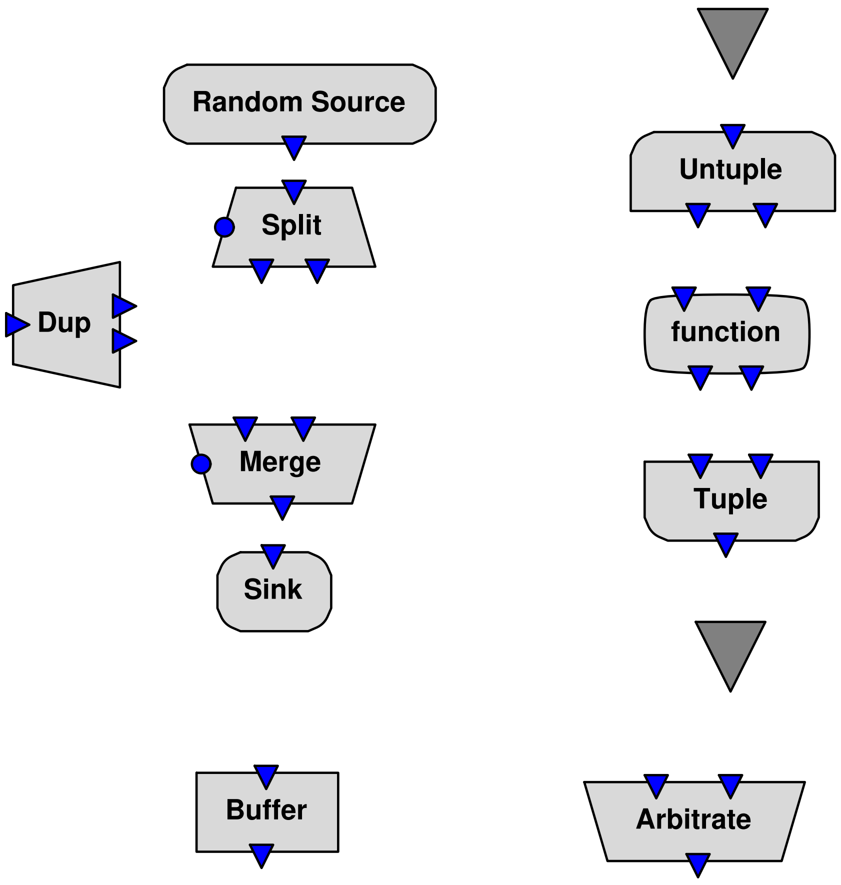
Micropipelines

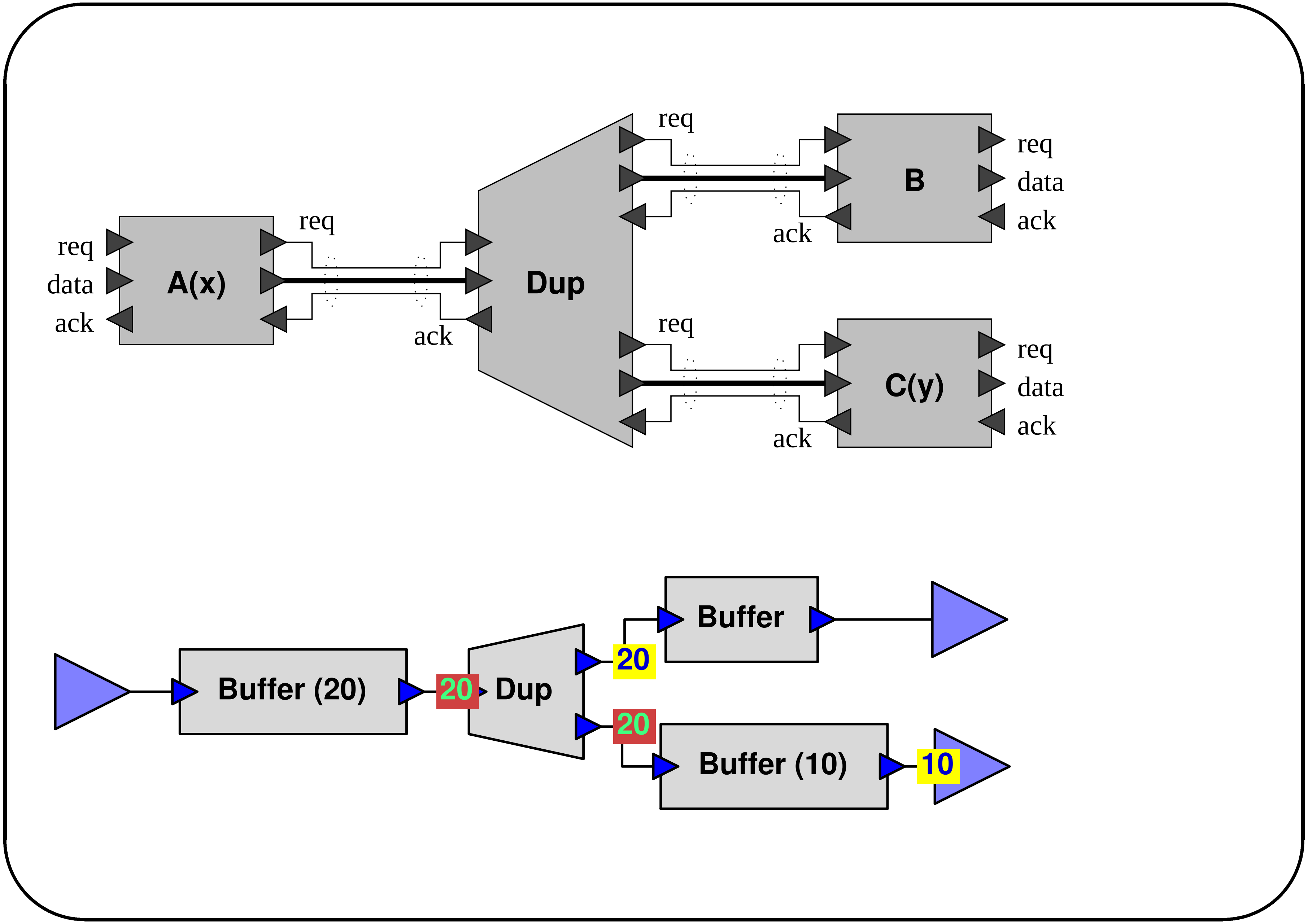


Rainbow Design Framework

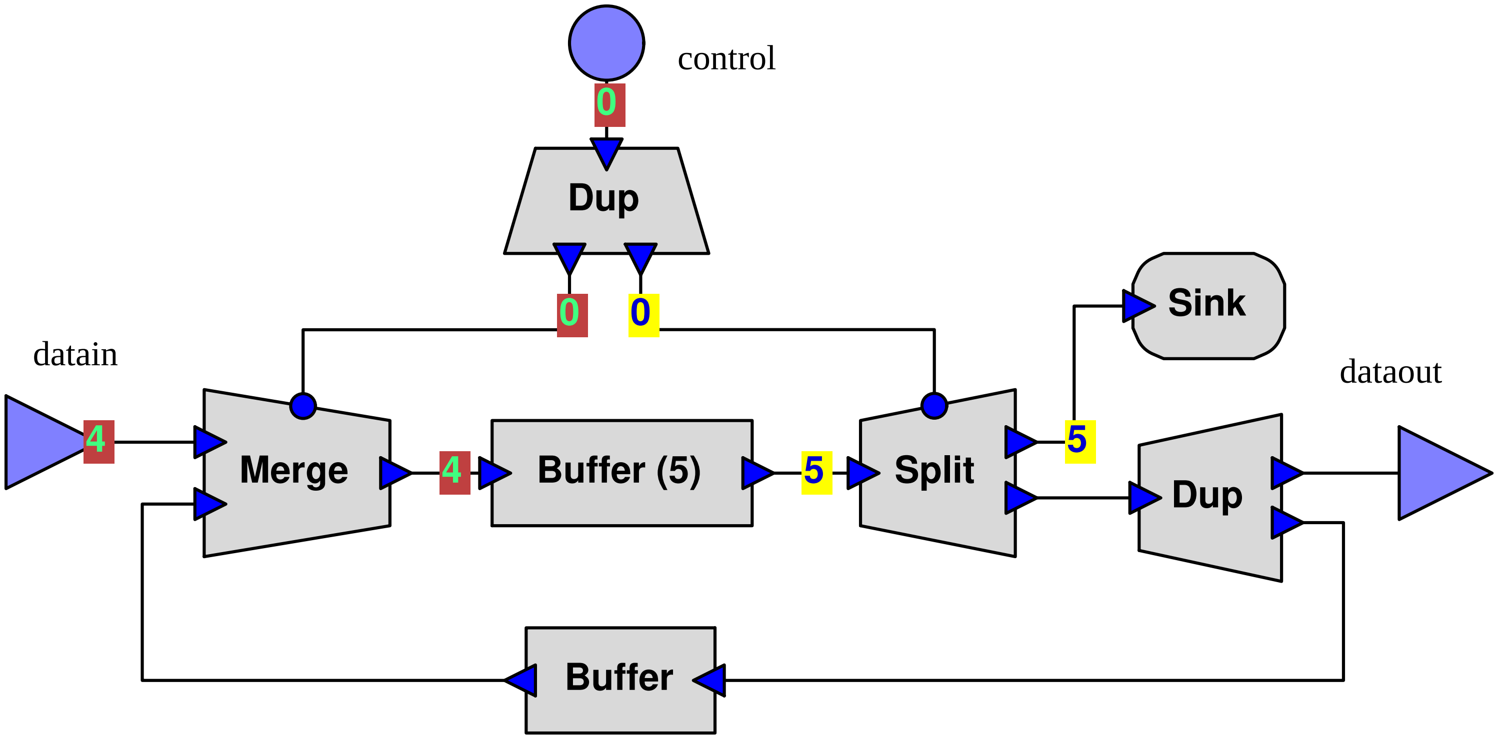


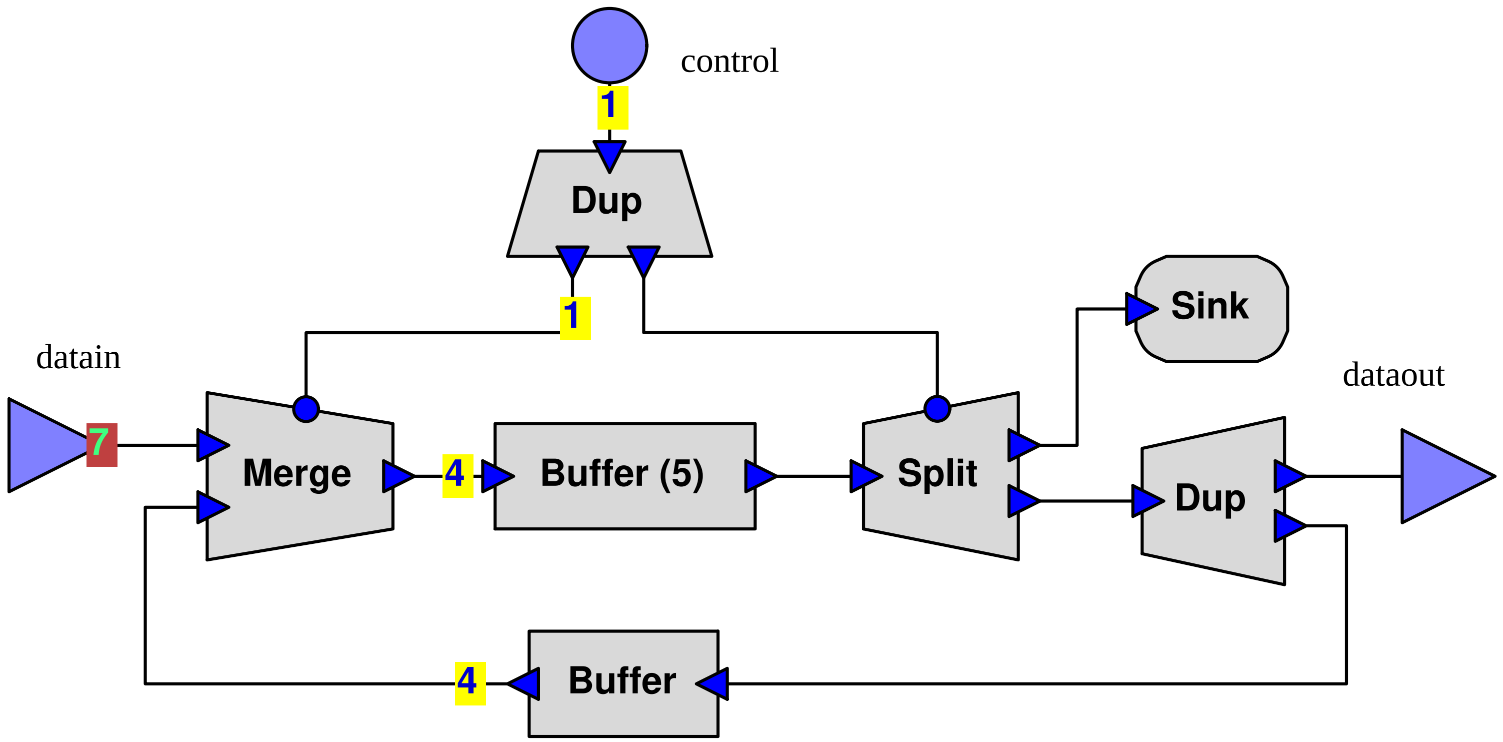
- Application-specific
- User-oriented language design
  - AMULET
- Abstract representation of bundled data
- Mixed-view modelling
- Hierarchical design
- Underlying formal semantics


Random Source
Split
Dup
Merge
Sink
Buffer
Untuple
function
Tuple
Arbitrate

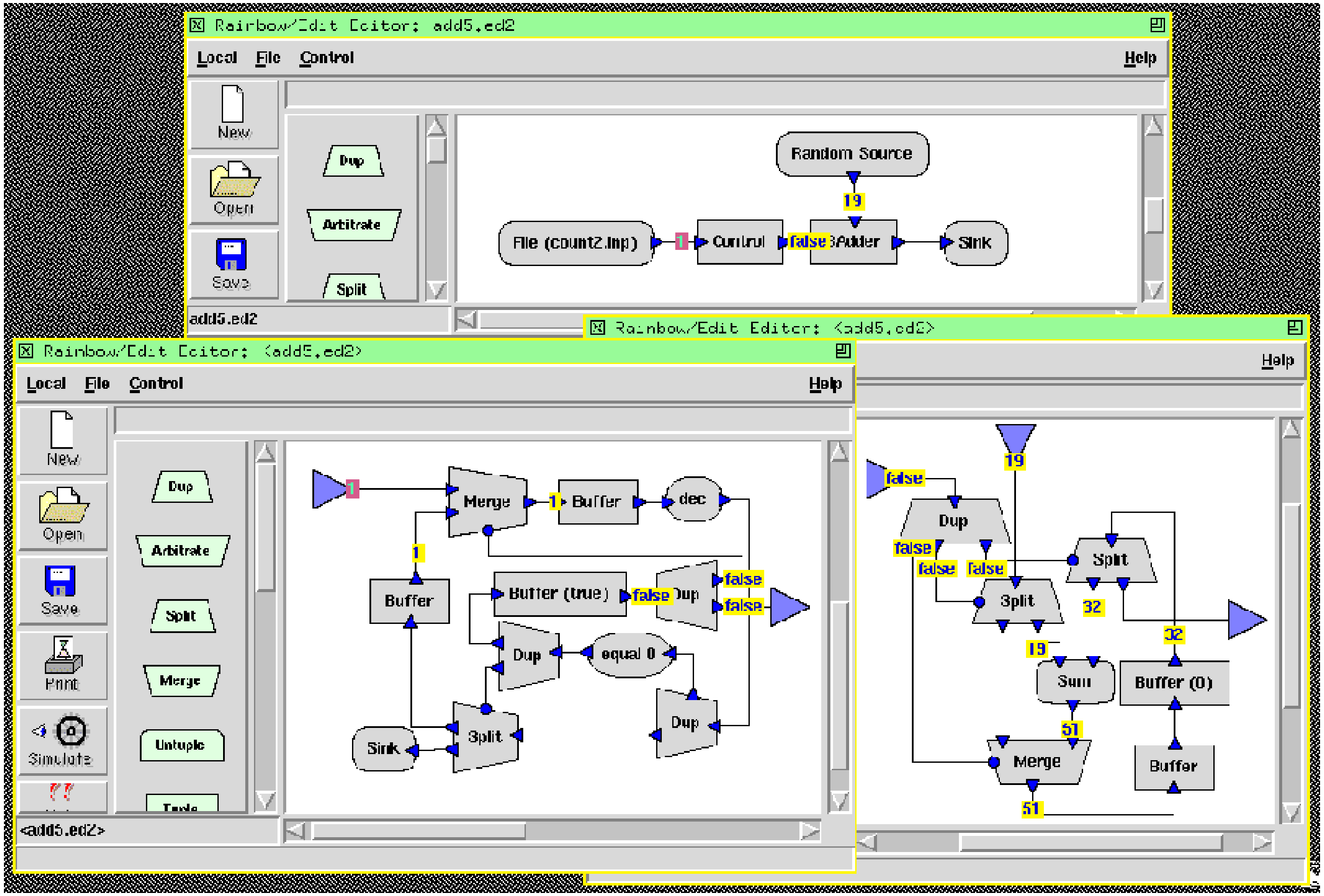

req
data
ack
A(x)
req
ack
Dup
req
ack
B
req
data
ack
req
ack
C(y)
req
data
ack
Buffer (20)
20
Dup
20
Buffer
20
Buffer (10)
10

Simulation

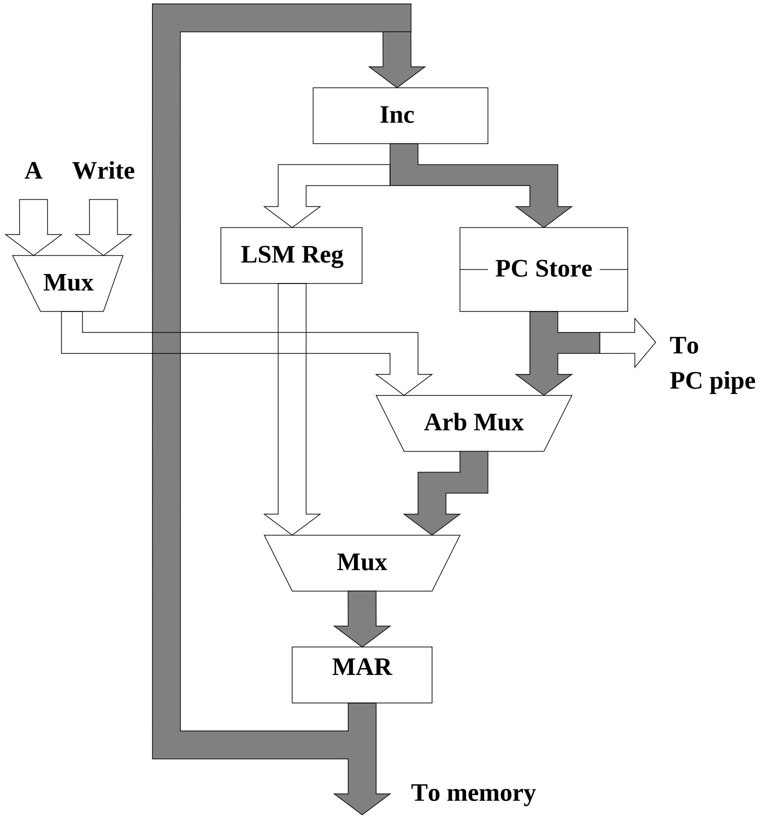

control
0
Dup
0
0
Sink
datain
dataout
4
Merge
4
Buffer (5)
5
Split
5
Dup
Buffer

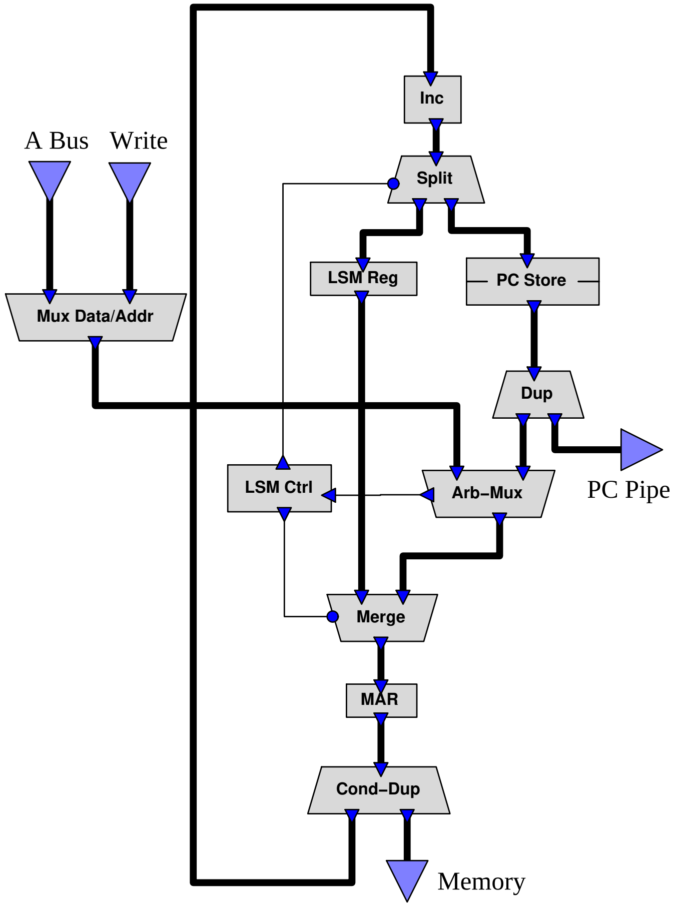

control
1
Dup
1
datain
7
Merge
4
Buffer (5)
Split
Sink
dataout
Dup
4
Buffer

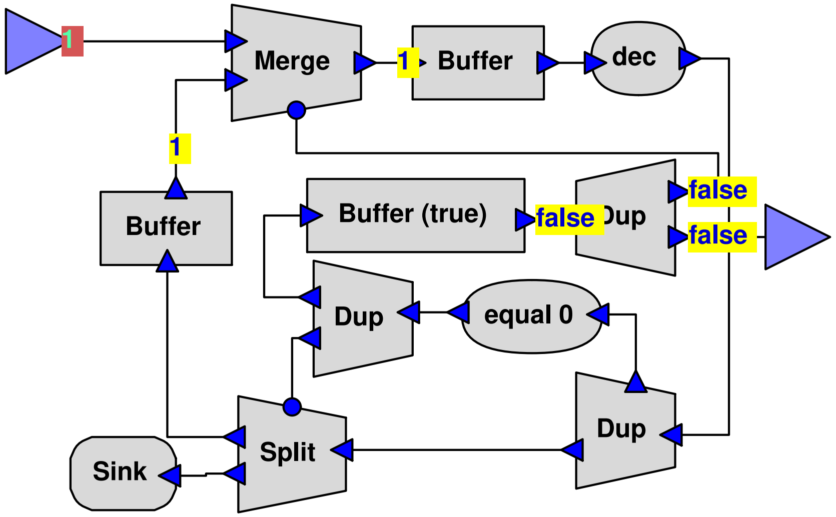
Hierarchy


Rainbow/Edit Editor: add5.ed2
Local File Control
Help
New
Open
Save
Dup
Arbitrate
Split
Random Source
File (count2.inp)
Control
Adder
Sink
add5.ed2
Rainbow/Edit Editor: <add5.ed2>
Merge
Buffer
dec
Buffer (true)
Dup
equal 0
Print
Simulate
Untuple
Sum
Buffer (0)
<add5.ed2>

Current Work








```
yellow control (input size: integer;
                output ctrl: boolean) = {
   reg count: integer end
   call ctrl!true;
   loop true ->
     {accept size?x do count:=x end;
      loop (count > 0) ->
        {call ctrl!false;
         count := count - 1}
      end;
      call ctrl!true}
   end
} end
```

Summary

- micropipeline design methodology
- Rainbow Design Framework
- abstract modelling
- design hierarchy
- 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`