HAsim: FPGA-Based Micro-Architecture Simulator

Michael Adler
Michael Pellauer
Kermin E. Fleming*
Angshuman Parashar
Joel Emer

*MIT

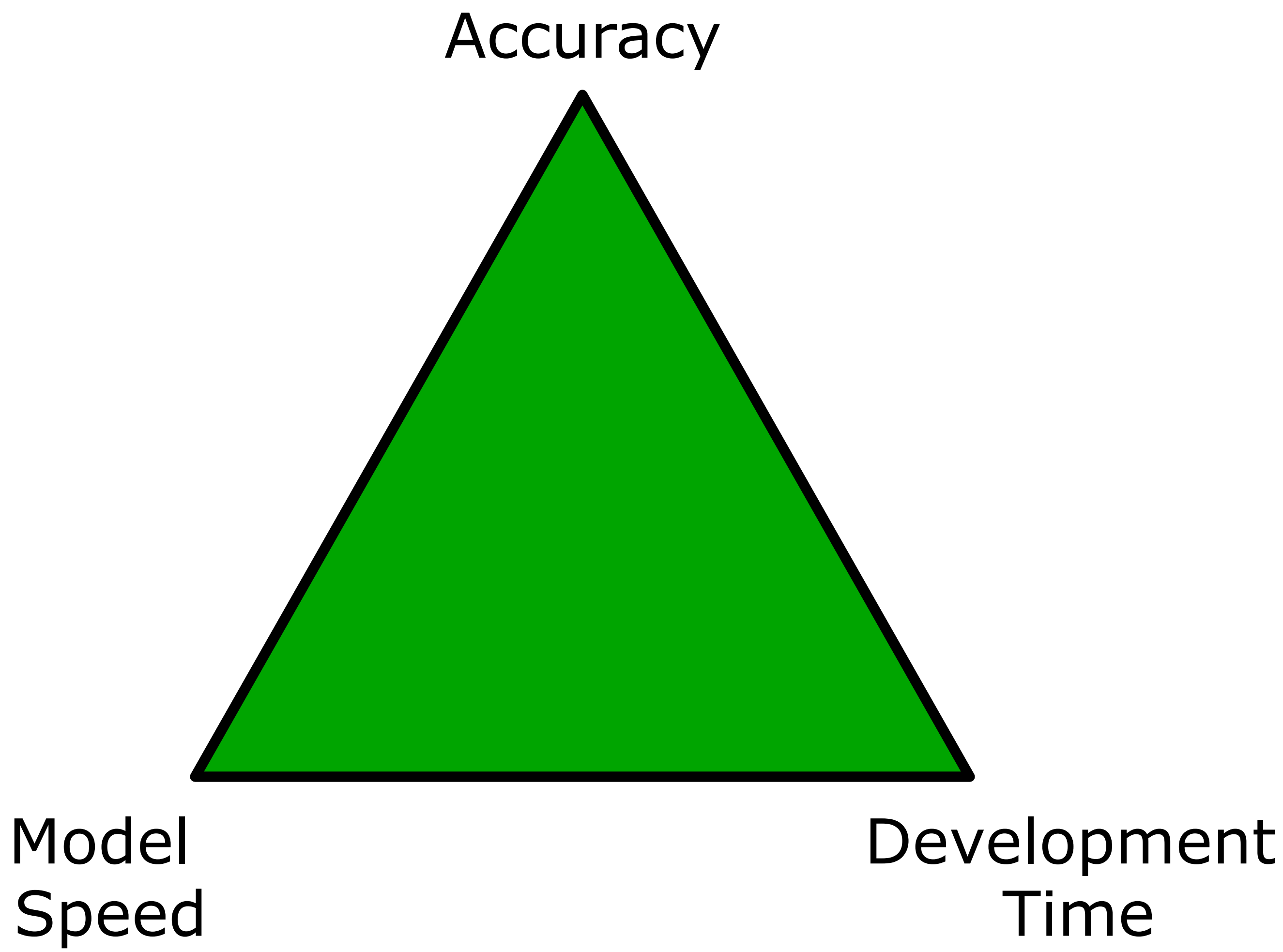
HAsim Is a Timing Model – Not RTL!

- Performance models are:
  - Highly parallel, but not easily vectorizable
  - Pipelineable
  - Full of communication channels

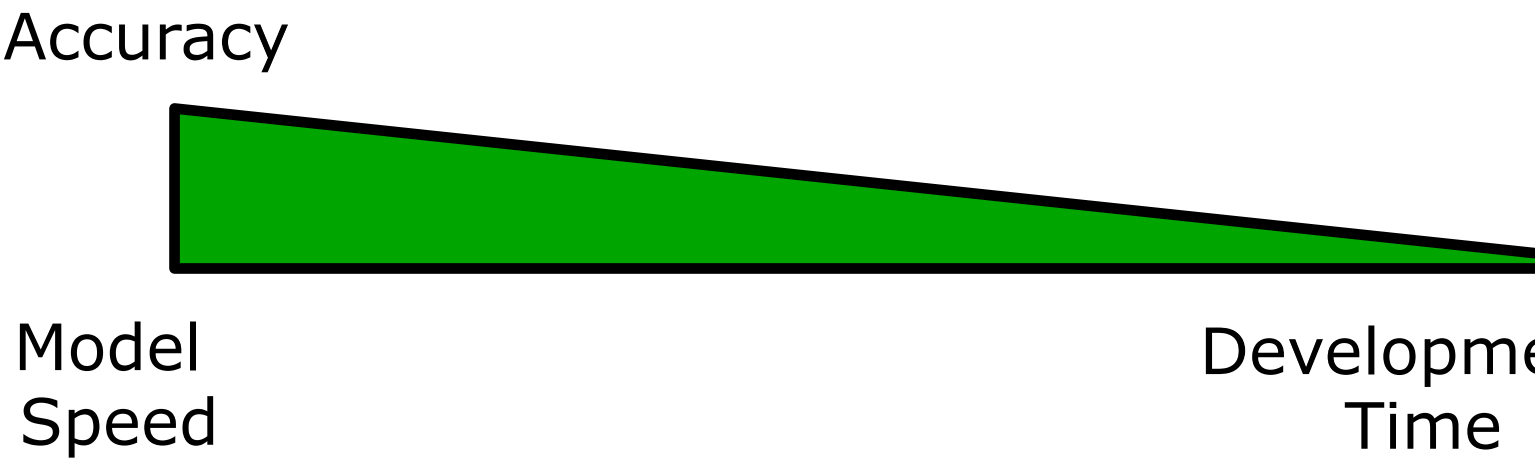
- Programmed like a software timing model
- FPGA is just a highly parallel execution engine
- FPGA cycle != Model cycle

- FPGA simulation will be faster than software if:
  - Parallelism can overcome the ~40x clock difference
  - I/O bandwidth is sufficient

Fast, Accurate or Now?



FPGA Picture is Different



Reducing Development Time: Managing Complexity

- Programming Language (Bluespec)
- Timing model infrastructure
  - Reusable functional model
  - Inter-module communication
  - Tracking simulated time
- Hybrid hardware / software models
  - GEM5 for:
    - Checkpoints
    - Loading
    - Functional memory management
    - Emulating difficult instructions

STDIO on General Purpose Machines

```
FILE *f = fopen(path, “w”);
const char *name = “Kenneth”;
fprintf(f, “%s, what is the frequency?\n”, name);
```

I/O In Hardware Description Languages (System Verilog)

```
Integer f = fopen(path, “w”);
string name = “Kenneth”;
fwrite(f, “%s, what is the frequency?\n”, name);
```

Nothing Comes from Nothing

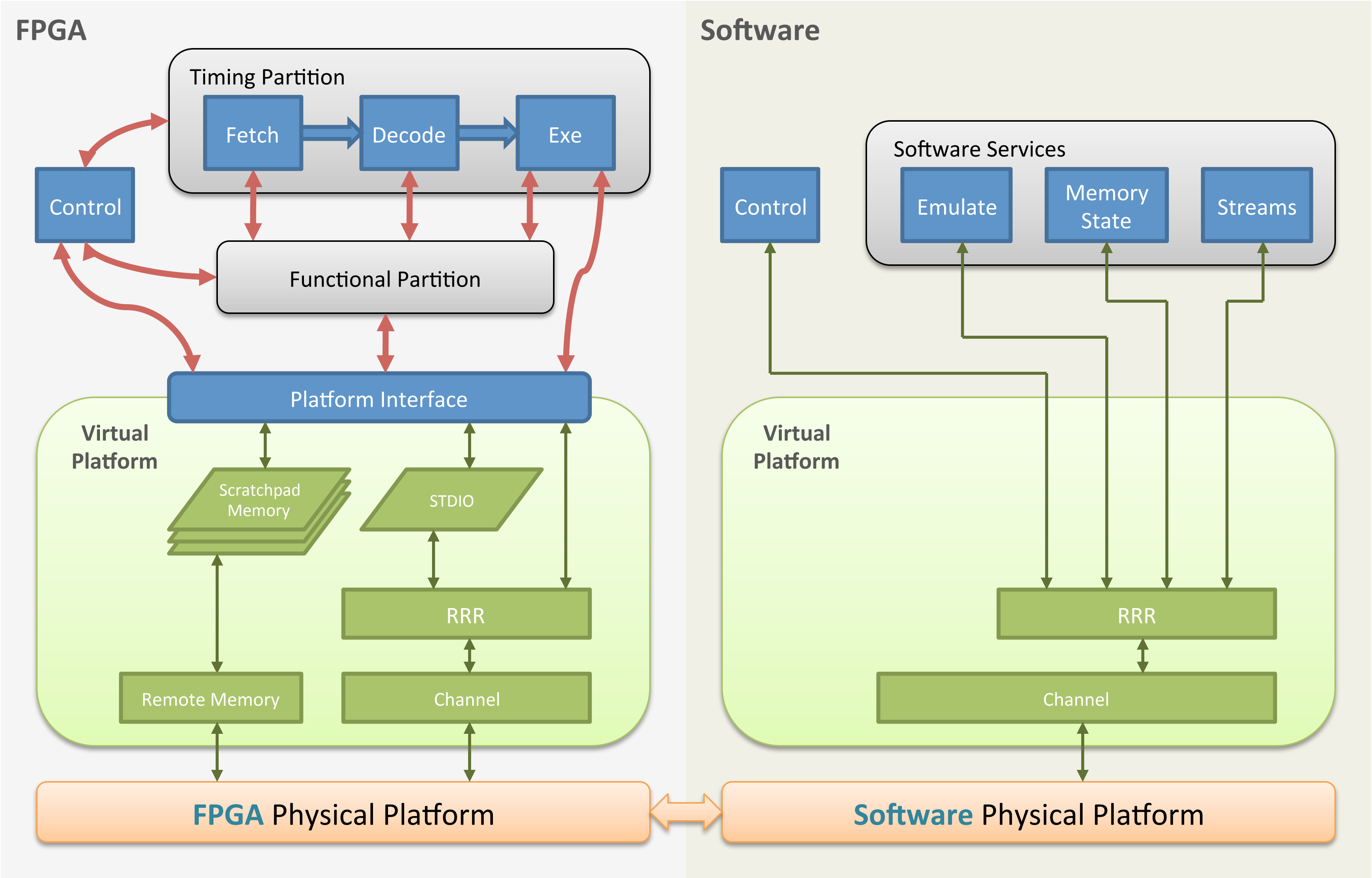
FPGAs have:

- No standard physical device
- No standard device model
- No standard system interface
- No standard API

What Makes Hardware General Purpose?

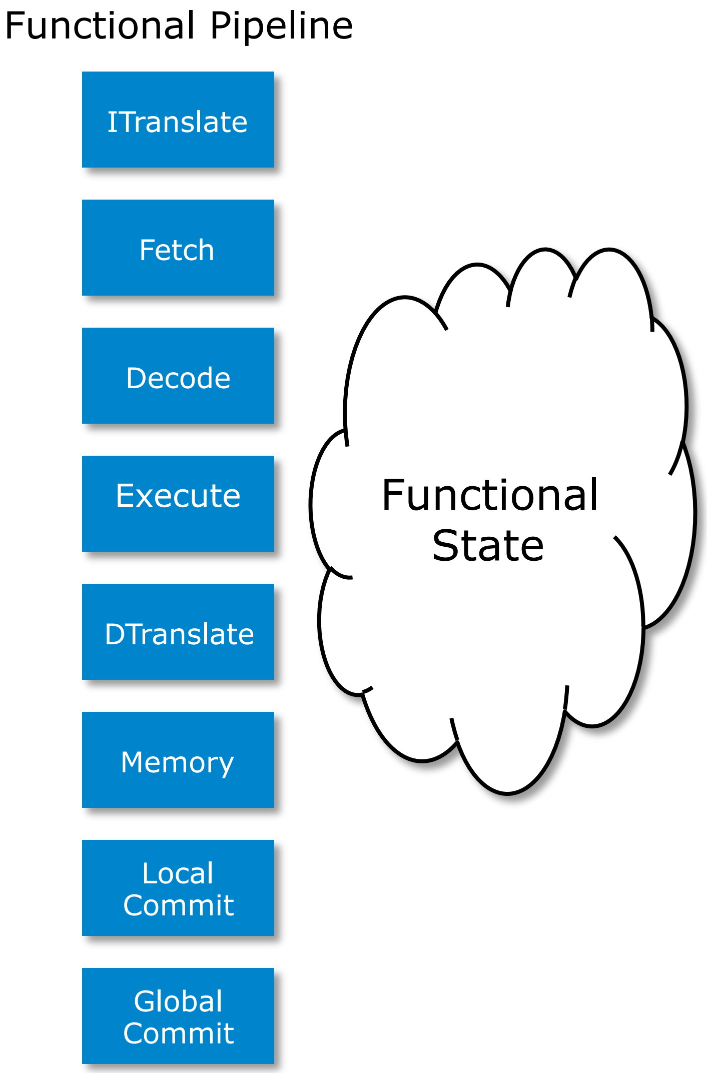
- The software
  - Compilers and library APIs make code “universal”
  - Hardware standards (ACPI, PCIe) mostly make OS development and compiler writing easier. Little impact on user programs.
  - ISA matters if you want to avoid recompiling. ISA is part of the software API, along with standard libraries.

LEAP Platform



Reducing Model Complexity: Shared Functional Model

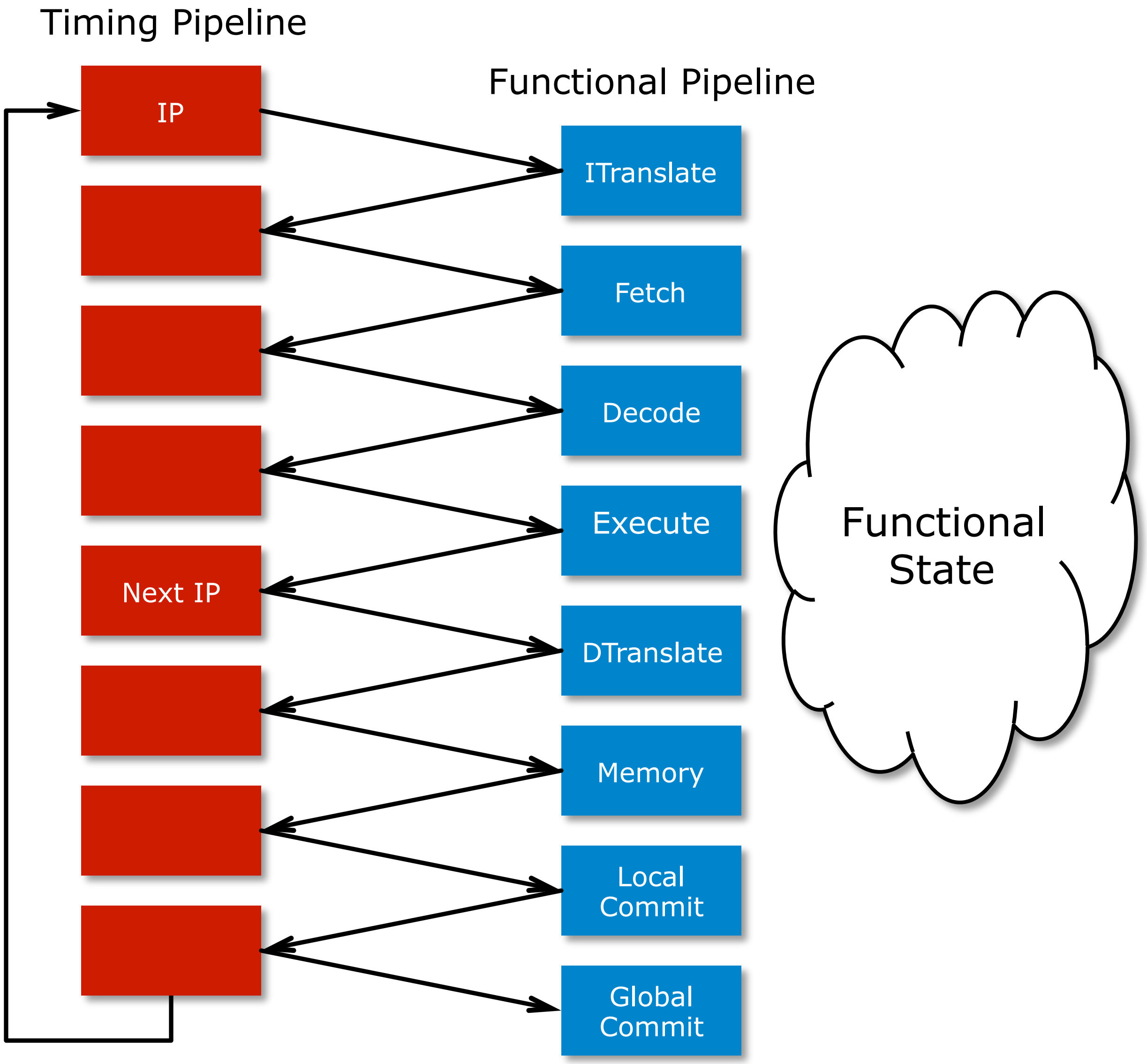
- Similar philosophy to GEM5 or Asim:
  - Single ISA functional model implementation
  - Functional machine state is completely managed
  - Timing models can be ISA-independent
- Each functional pipeline stage behaves like a request/response FIFO



ISPASS 2008 Paper:
[Quick Performance Models Quickly: Timing-Directed Simulation on FPGAs]

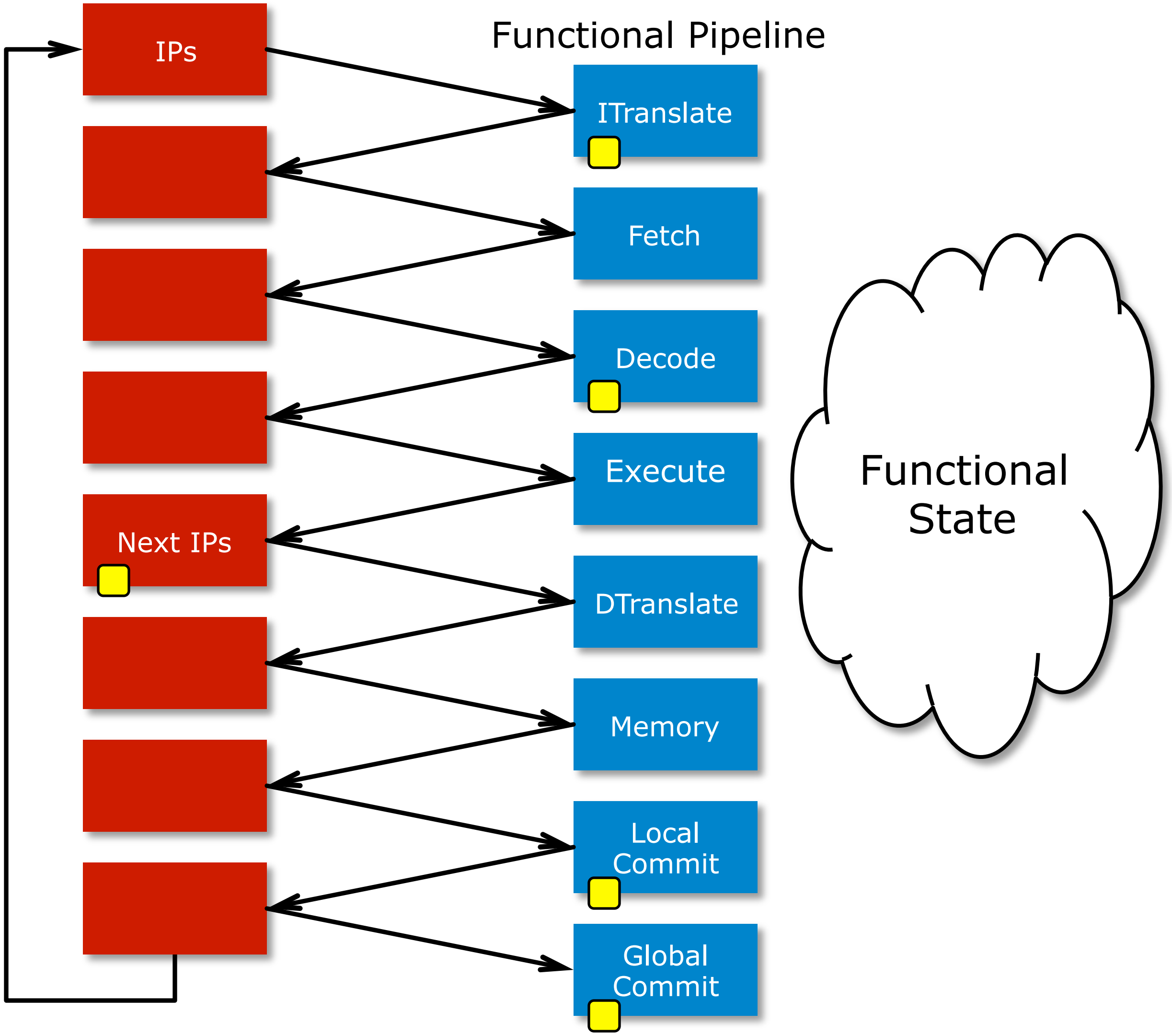
Timing Model

- Timing & functional models communicate state using tokens
- Minimal timing model:
  - Only state is IP
  - Drives a single token at a time



Pipeline Parallelism

- Model of a pipelined design naturally runs pipelined on an FPGA
- Detailed model of a pipelined design runs faster than a trivial, unpipelined model!

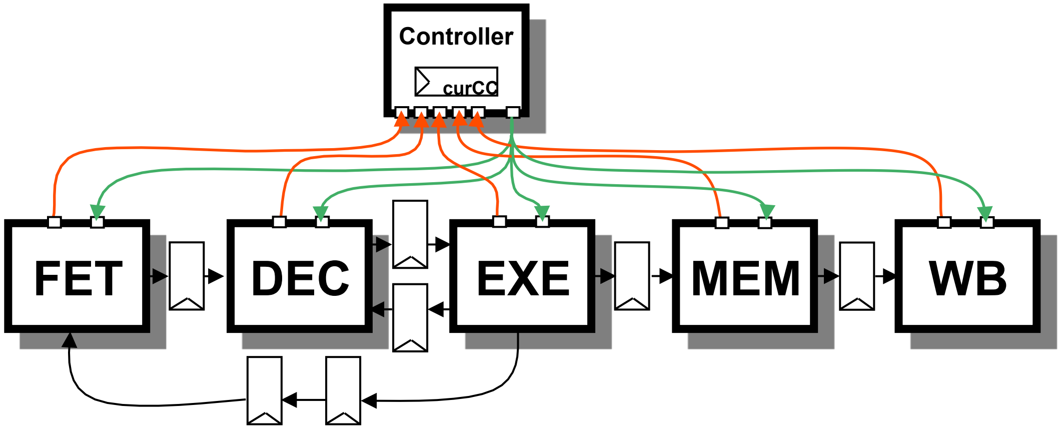


Managing Time:
A-Ports and Soft Connections

FPGA cycles != simulated cycles:

- We are building a timing model, NOT a prototype
- 1:*n* cycle mapping would force us to slow the timing clock to the longest operation, even if it is infrequent
- 1:*n* would force us either to fit an entire design on the FPGA or synchronize clock domains

Option #1: Global Controller [rejected]

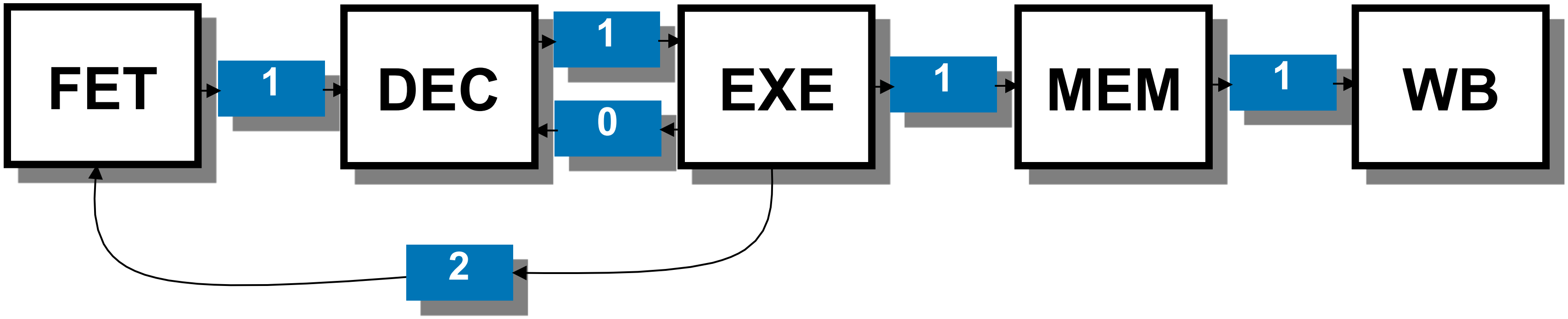


Central controller advances cycle when all modules are ready

- Improvement: slowest possible cycle no longer dictates throughput
- However:
  - Place & route becomes difficult
  - Long signal to global controller is on the critical path

Option #2: A-Ports

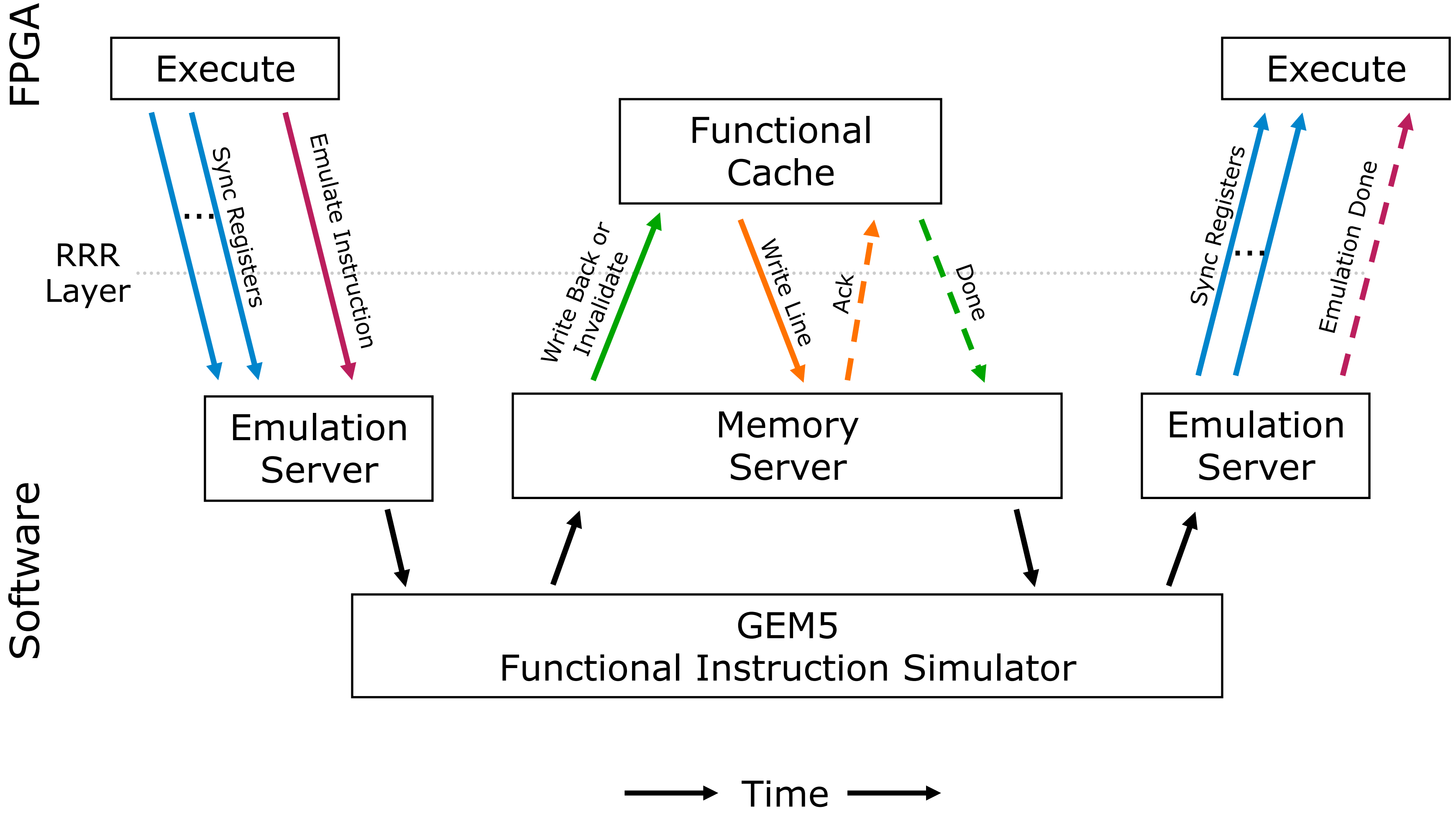
- Extension of Asim ports
- FIFO with user-specified latency and capacity
- Manage model time by guaranteeing exactly one message per cycle through every port



- Beginning of model cycle: read all input ports
- End of model cycle: write all output ports

ISFPGA 2008 Paper:
A-Ports: An Efficient Abstraction for Cycle-Accurate Performance Models on FPGAs

Hybrid Modeling:
Software Instruction Emulation



HAsim / LEAP Open Source

Redmine site with source and papers:

http://asim.csail.mit.edu/