AHIR: A Hardware Intermediate Representation for Hardware Synthesis from High-level Programs

Sameer D. Sahasrabuddhe
Hakim Raja
Kavi Arya
Madhav P. Desai

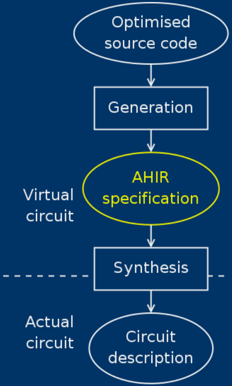
IIT Bombay

8th January, 2007

Motivation

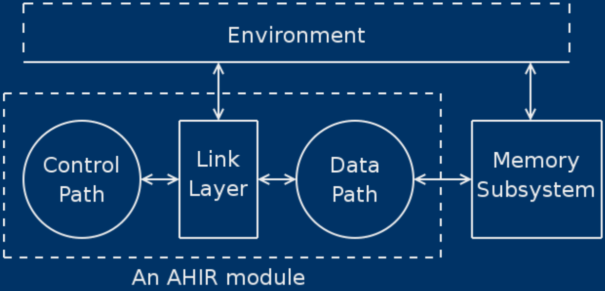
Getting programmers to create hardware

- Construct large hardware systems from programs in a high-level language
- Generate hardware specifications in an intermediate representation (IR)
  1. A verifiable path from high-level programs to IR specifications
  2. Transformations that maintain equivalence with the original program
  3. Minimum constraints that must be satisfied by an implementation to guarantee equivalence

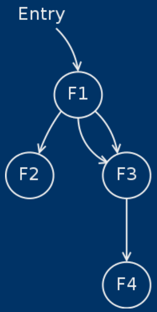


Outline

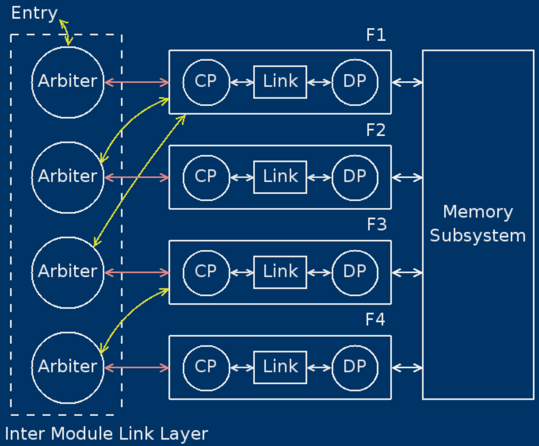
AHIR
A decoupled view of hardware



Translating call-graphs to AHIR modules

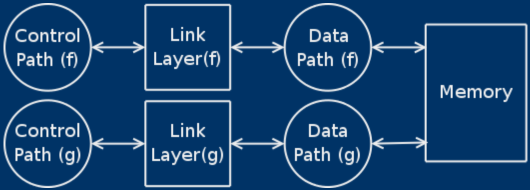


Call-graph

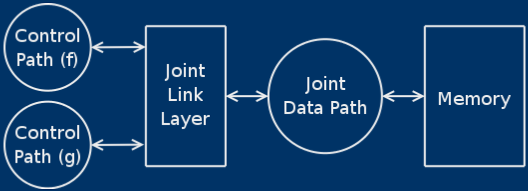


AHIR modules

Opportunity: Sharing resources across modules

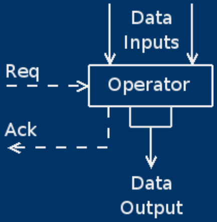


A "virtual circuit" in AHIR



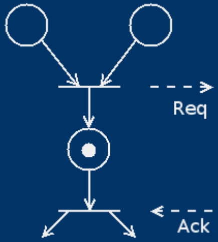
Actual Implementation

Data-path



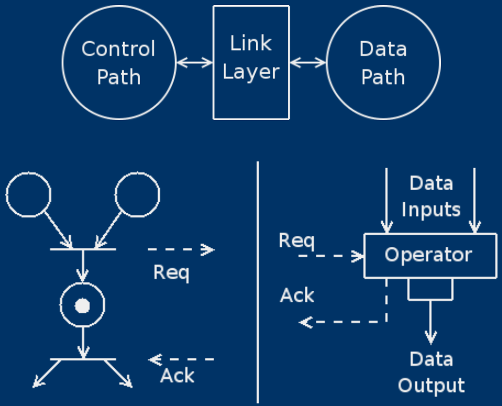
- The data-path is a graph of data-path elements and wires, derived from the original program.
- An element begins execution on receiving a `Req` and signals completion with an `Ack`.
- Every output of an element is stored in a register.
- Each register drives a wire that fans out to multiple users; each wire can have only one driver.
- Elements: operators, multiplexers, condition decoders and memory access.

Control-path

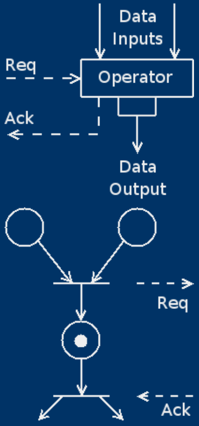


- The control-path is the sequence of operations derived from the original program.
- A petri-net with two kinds of transitions - input and output.
- Output transitions generate requests for the data-path.
- Input transitions respond to acknowledge and other signals received from the data-path.

Interaction using symbols



Timing constraints



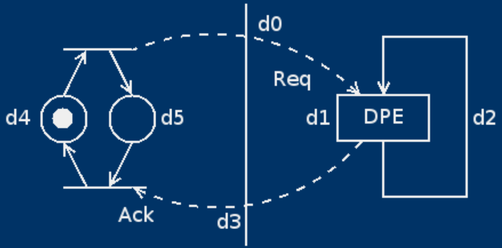
Data Path

- Values at the data-inputs are assumed to be correct and stable when a *Req* arrives.
- An output is assumed to be correct and stable when the corresponding *Ack* is emitted.

Control Path

- An output transition can fire as soon as it is enabled by sufficient tokens.
- An enabled input transition can fire only on the arrival of the symbol it is waiting for.
- It is an *error* if a symbol arrives while the corresponding input transition is not enabled.

Constraints on path delays



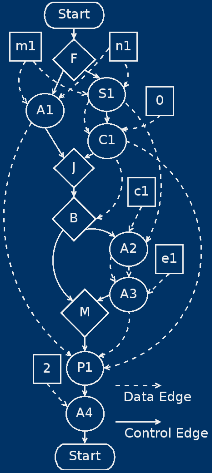
$$d_5 \leq d_0 + d_1 + d_3$$

$$d_2 \leq d_3 + d_4 + d_0$$

Compiler flow
Control Data Flow Graphs (CDFG)

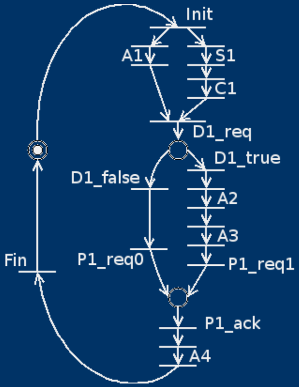
```
A1: d = m + n
S1: b = m - n
C1: if (b > 0)
{
A2: a = b + c
A3: d = e + a
}
A4: x = d + 2
```

Sample program in C

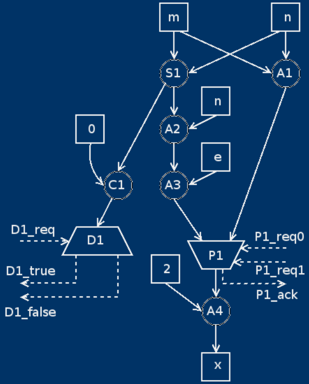


Control Data Flow Graph

Control-path and data-path in AHIR

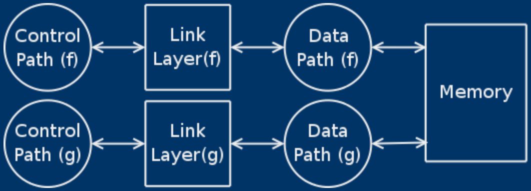


Control Path

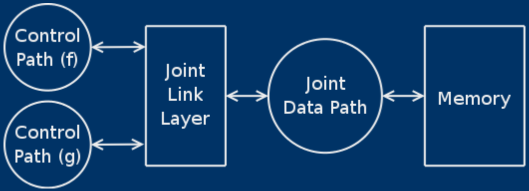


Data Path

Implementing AHIR specifications



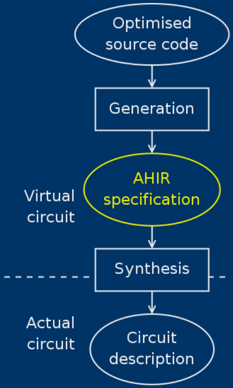
A “virtual circuit” in AHIR



Actual Implementation

Current implementation

- XML format for AHIR
- An AHIR backend for the LLVM compiler framework
- A SystemC simulator for AHIR
- A VHDL generator for AHIR



Thank You.