IODINE: A Tool to Automatically Infer Dynamic Invariants for Hardware Designs

Design Automation Conference 2005
16th June, 2005




Sudheendra Hangal
Sun Microsystems
Bangalore

Naveen Chandra
Sun Microsystems
Bangalore

Sridhar Narayanan
P. A. Semi Inc
CA

Sandeep Chakravorty
Sun Microsystems
Bangalore

Motivation for Dynamic Analysis

We simulate designs extensively.

Simulation runs contain <u>lots</u> of information.

How can this information be used ?

1) Understanding behavior of the design

2) Automatically extracting design properties

3) Understanding impact of design differences

...

Property Checking Challenges

- Potential payoff of formal property checking is huge
  ... but in practice:
  - hard to write design properties
  - hard to write input constraints
  - Expert knowledge is scarce
  - Design documentation is missing (or wrong!)
  - How do you know when all properties are written ?
  - Only proof that code works is that it passes a test suite
- Formally verifying protocols in the abstract is insufficient
  - Formal verification is applied to what % of million-line Verilog programs ?

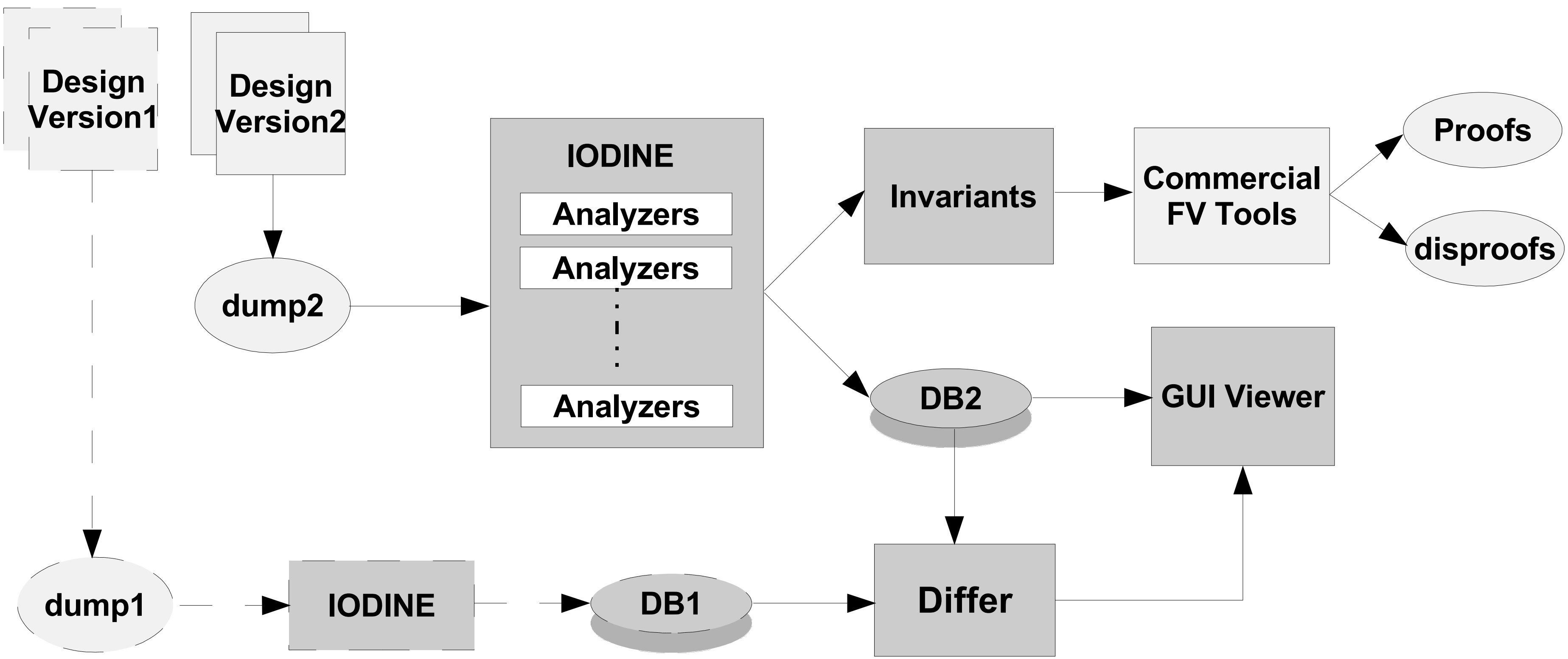
Dynamic Invariants

- Proposed by Michael Ernst for software programs
  - Worked well for small programs
- Basic Idea:
  - Hypothesize a space of invariants
  - Analyze invariants on test inputs (assumed to pass)
  - Rule out invariants which do not hold
- Invariants detected may be unsound!
  - That's ok – incorrect invariants indicate a coverage hole
  - For hardware designs, testsuites are expected to be nearly complete; exposing missing coverage is also valuable

A Hardware Invariant Detector

- IODINE: dynamic analysis for common hardware invariants
  - trade-off: complexity & cost v/s exhaustiveness & noise
- Invariants are *pervasive* and *complete*
  - reduced search depth (useful for bounded model checking)
- Easier for designers to certify rather than write invariants
- Invariants on block's inputs can become constraints

Framework



- Analyzers extract different kinds of information
- Analyzers can query other analyzers

IODINE Features

- Invariants can be ranked on the basis of confidence
- Extensible set of analyzers
- Invariants involving up to 4 variables considered
  - User can specify own expressions of interest
- Smart multi-pass analysis algorithms to speed analysis
  - Analysis time is O(few hours), i.e. O(simulation time)

IODINE Analyzer Library

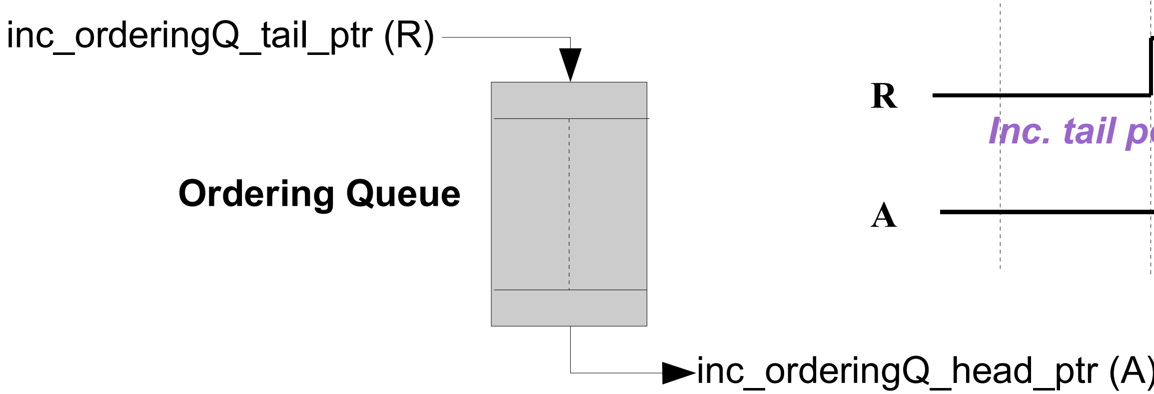
Analyzers	Invariants
Onehot	Onehots, Onecolds, # Bits on/off
ReqAck	Req-Ack pairs or triples,Multi-reqs/Multi-acks, Event Protocols, Fixed Delay pairs
Fifo	Fifo's, Out of order scoreboards, Pipeline data flow
State	FSM transitions, Up/Down Counters
CrossProduct	Mutual exclusion conditions
Equals	Dynamic Aliases
Constants	Constant Signals
Freq	Support Analyzer
Vector	Support Analyzer

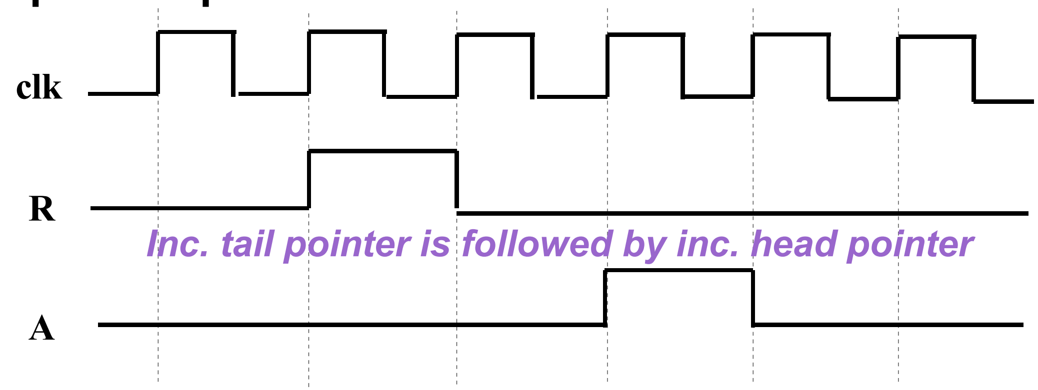
Examples from Dual-core UltraSPARC™ microprocessor follow

Req-Ack Analyzer

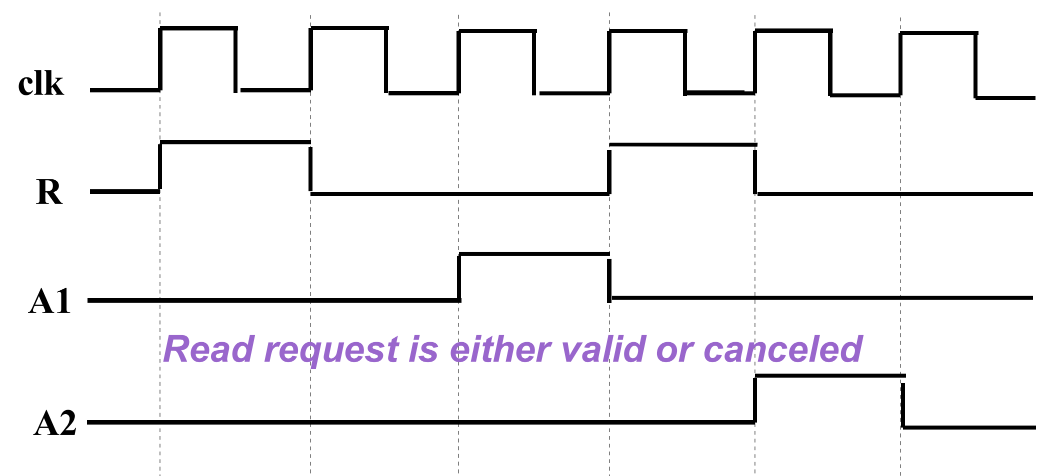
- Exhaustively extracts Req-Ack pairs

$- R \rightarrow A$



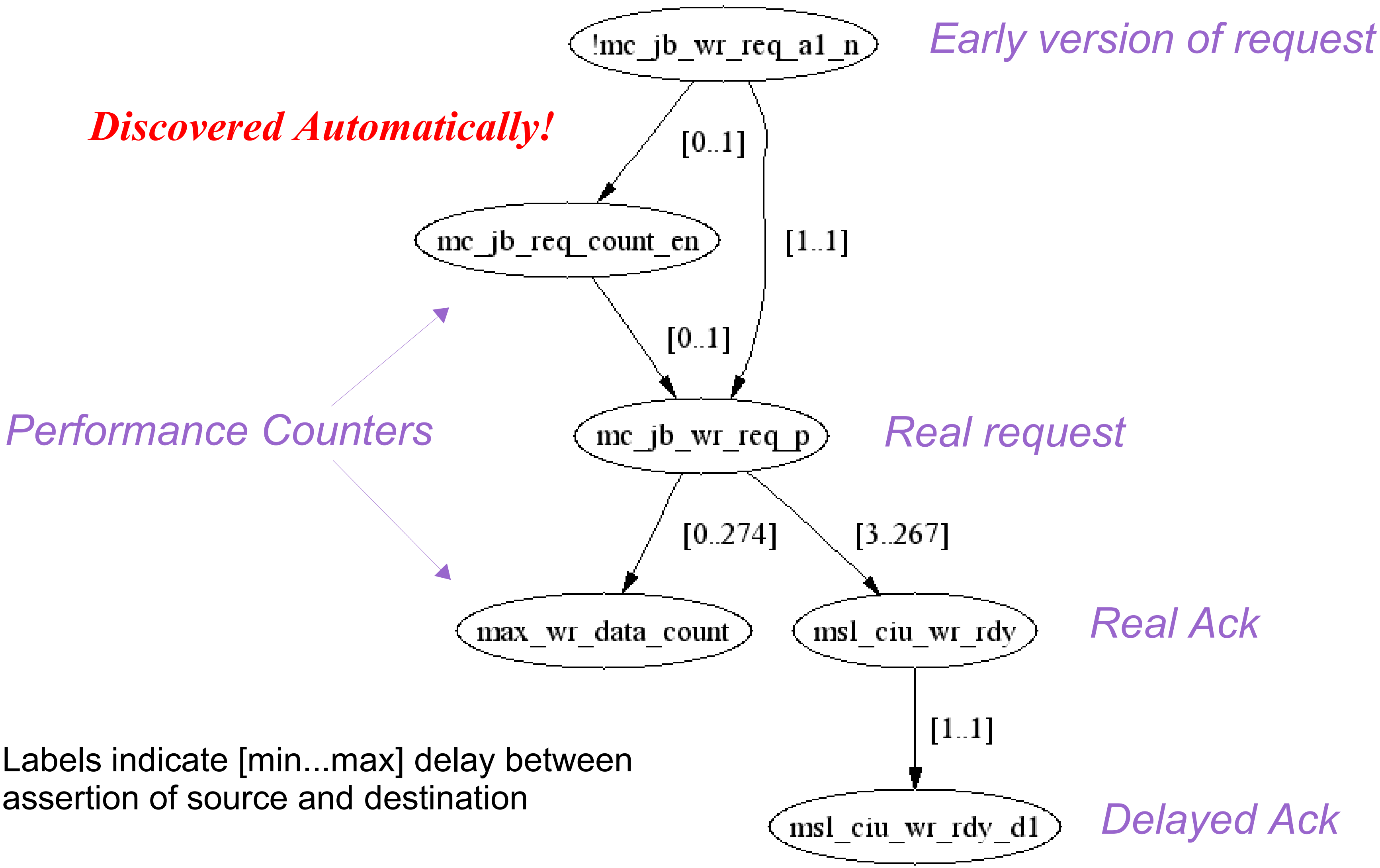


$- R \rightarrow (A1 \mid A2)$



Examples from Dual-core UltraSPARC™ microprocessor

Memory Controller Write Protocol



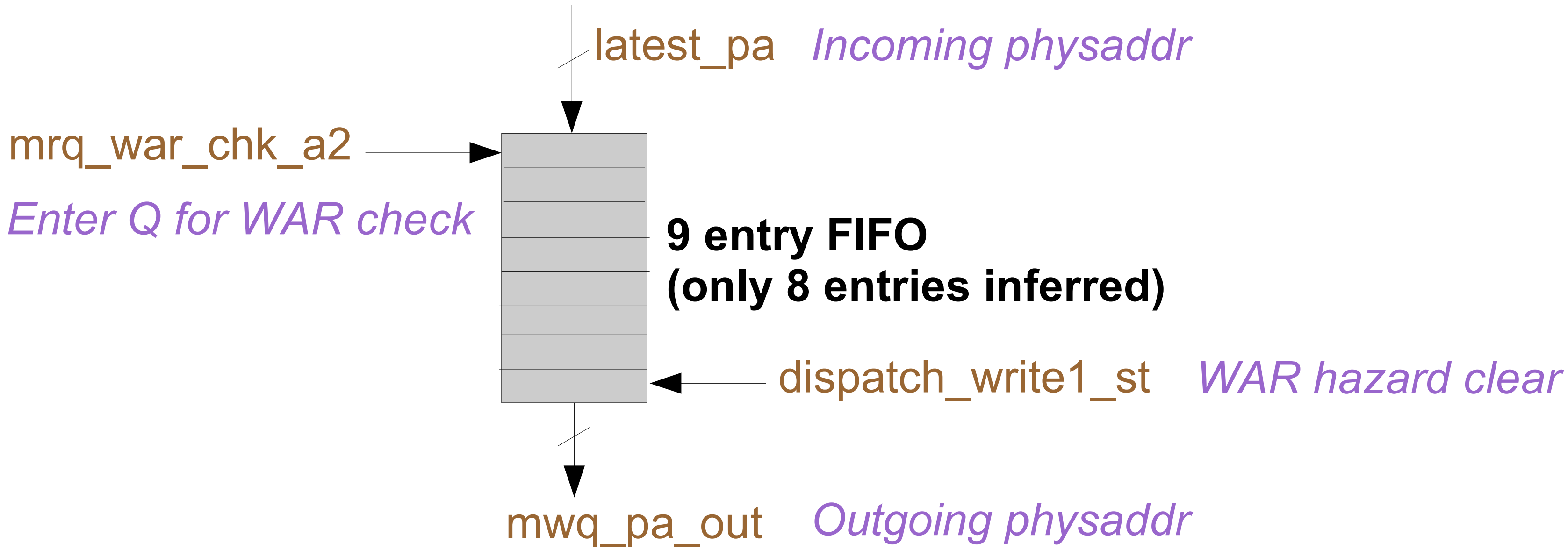
Labels indicate [min...max] delay between assertion of source and destination

Example from Dual-core UltraSPARC™ microprocessor

FIFO Analyzer: Example

Following invariant exposes functional coverage hole:

fifo -depth ***8*** -req mrq_war_chk_a2 -deq dispatch_write1_st
-enq_data latest_pa -deq_data mwq_pa_out



Example from Dual-core UltraSPARC™ microprocessor

Summary

- IODINE extracts dynamic invariants automatically
  - Useful for formal property checking
  - Automatic functional coverage information
  - Aid for design understanding, evolution
- Invariants are pervasive and complete
  - Reduces search depth for FV tools
- See paper for details on analyzers and invariants

Related Work & Thanks

- [Nimmer,Ernst] Software work for Daikon and ESC-Java
- [Yang,Evans] Temporal properties (2 vars) using QREs

Thanks to:

Monica Lam, Stanford University

Michael Ernst, MIT

Ajit Pasi, IIT Delhi

For more information:

http://xenon.stanford.edu/~hangal/iodine.html

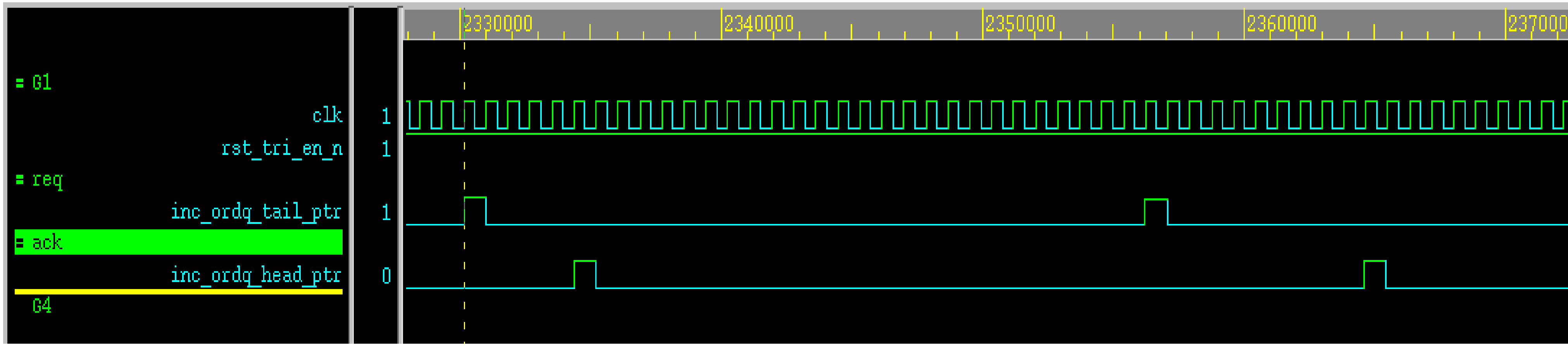
iodine@sun.com

Backup Slides

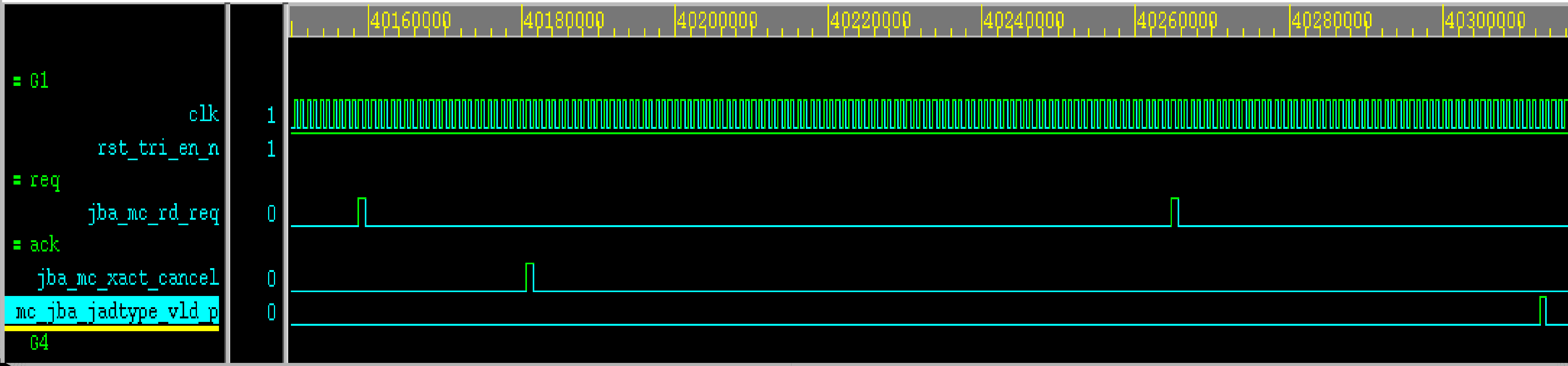
Req-Ack Analyzer

- Exhaustively extracts Req-Ack pairs

  - R → A



  - R → (A1 | A2)



Fifo Analyzer

Extracts Fifo's and Outstanding-id's

0in fifo -depth ***8*** -req ***mrq_war_chk_a2***
-deq ***dispatch_write1_st***
-enq_data ***latest_pa***
-deq_data ***mwq_pa_out***