

JasperGold™

High-Level Requirements Based Verification
Leveraging SystemVerilog

A Closer Look...

at the Verification Productivity Problem

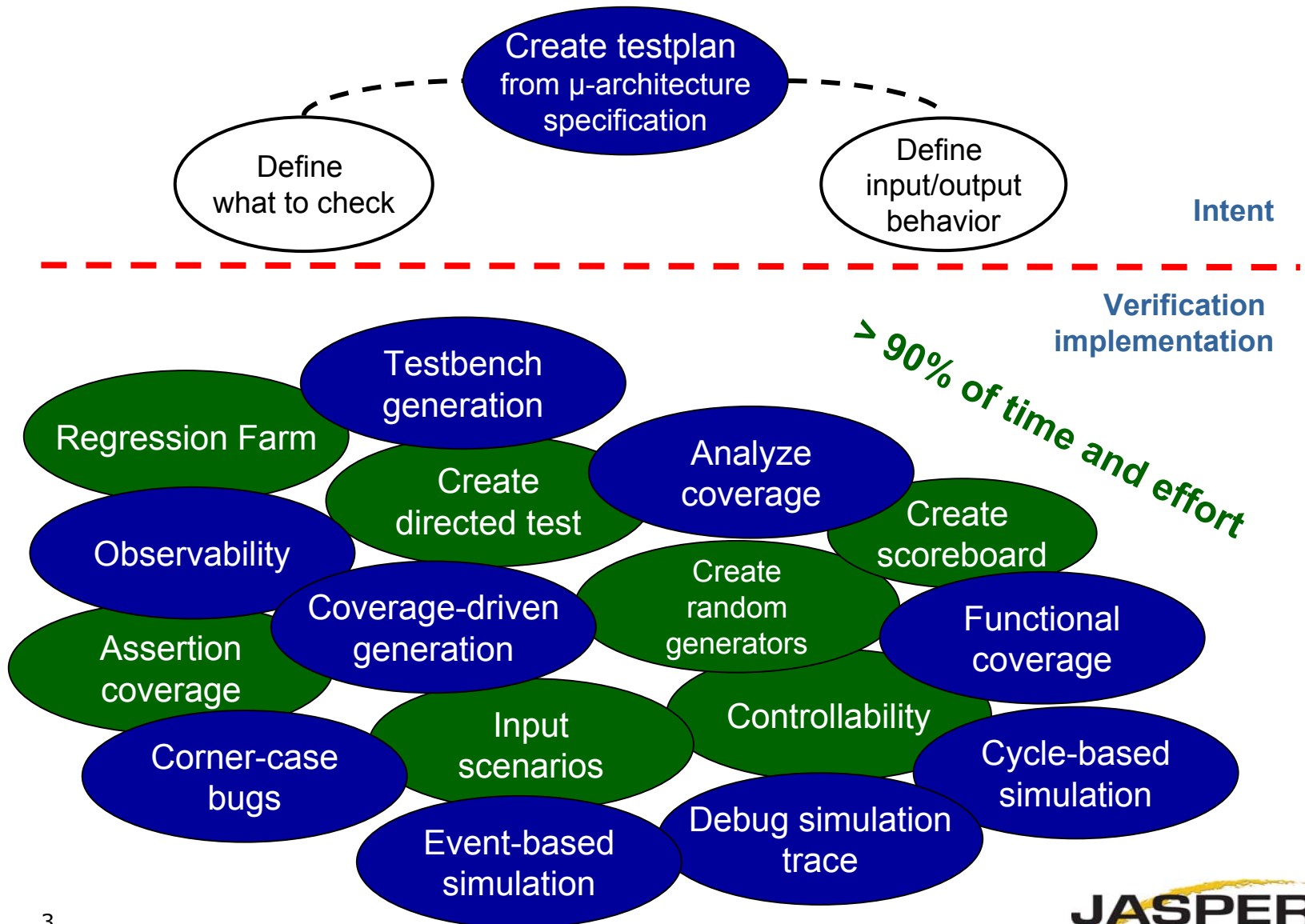
- Verification consumes 30% to 70% of total schedule, depending on design size

-- Study by Gary Smith, Gartner Dataquest

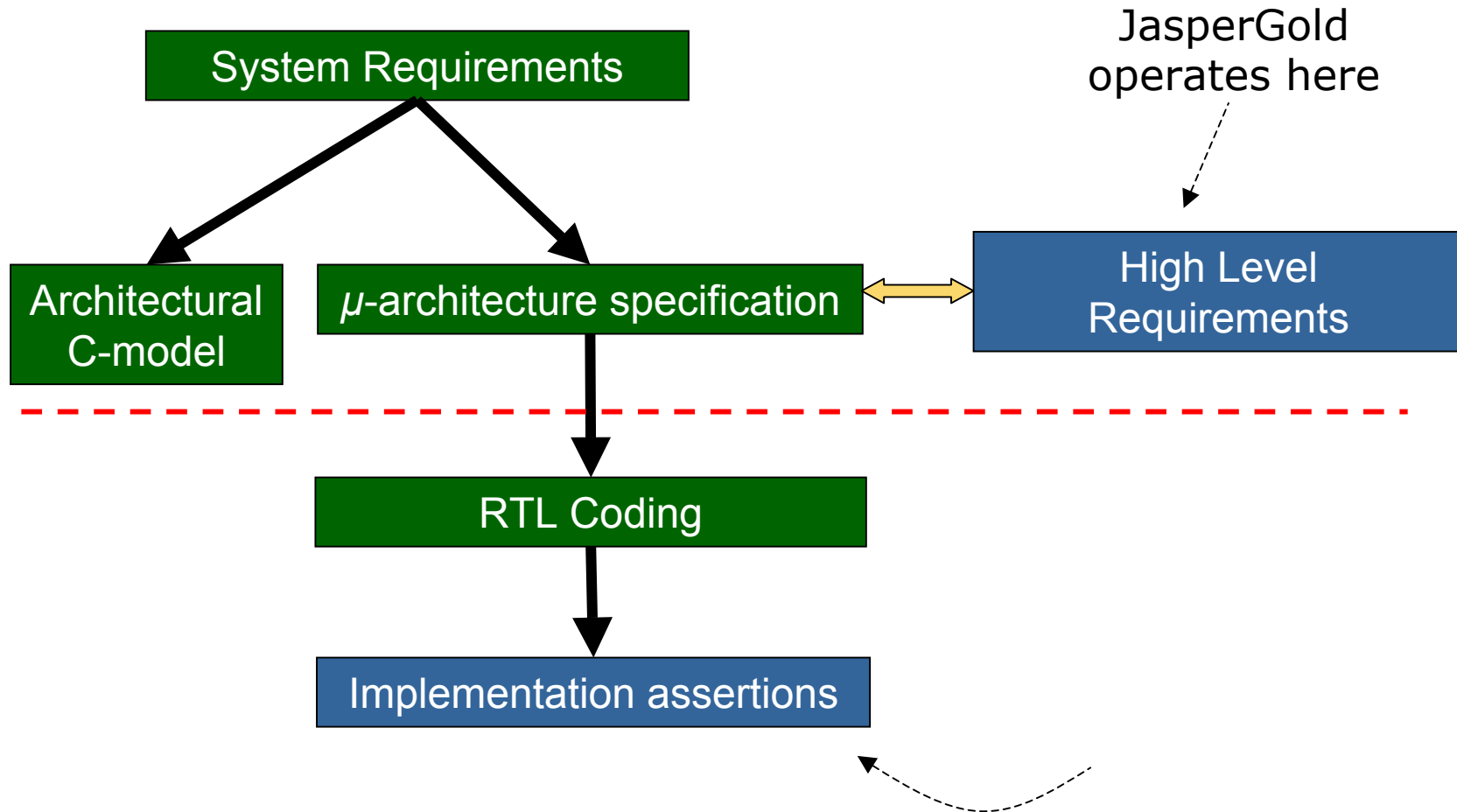
< 10% = Capturing intent

> 90% = Verification implementation

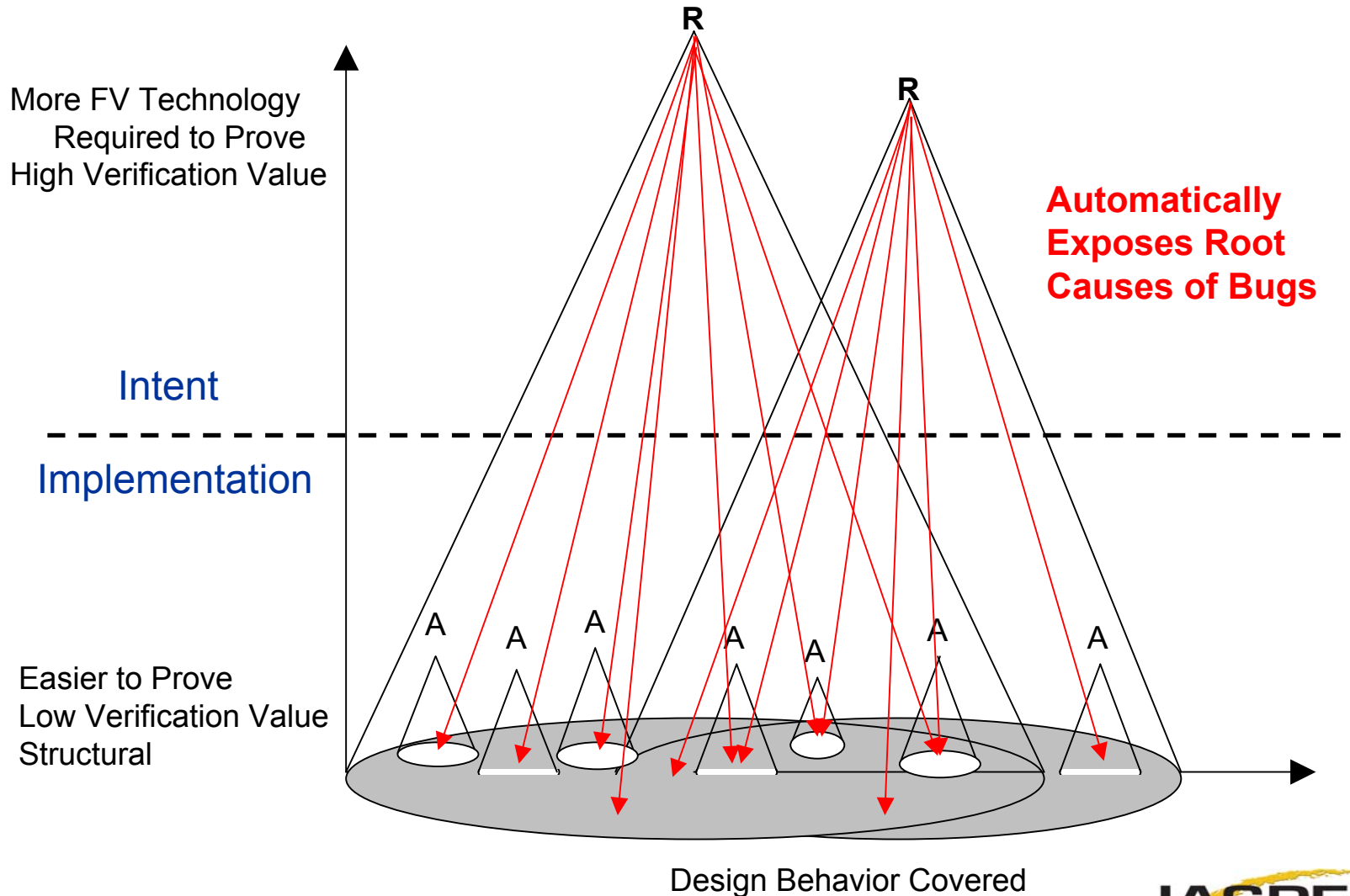
Simulation Effort



Intent Versus Implementation



High-Level Requirements-Based Verification (RBV)



PCI Compliance Checklist, Rev. 2.2

ADDENDUM B: Target Protocol Test Scenarios for Components

2.11. TARGET RECEIVES FAST BACK TO BACK CYCLES

GENERAL FUNCTIONAL DESCRIPTION

Verify the ability of Implementation Under Test (IUT) to successfully complete fast back to back cycles.

CROSS-REFERENCE TO PCI LOCAL BUS SPECIFICATION 2.2

Pages 72 - 74

METHOD OF VERIFICATION

Program the Primary Master to perform the write/write and write/read transfers with the IUT being selected both times and then only on the second transfer.

FLOW

- o Establish known Initialization State

Both transfers to the IUT

- Program the Primary Master to perform a fast back to back transfer with two memory write cycles to the IUT
- Program the Primary Master to perform a memory read cycle from the IUT

→ Verify that the read data is the same as the data written

- Program the Primary Master to perform a fast back to back memory write and read cycle to the IUT

→ Verify that the read data is the same as the data written

Last transfer to the IUT

- Program the Primary Master to perform a fast back to back transfer with two memory write cycles but only the last write selects the IUT
- Program the Primary Master to perform a memory read cycle to the IUT

→ Verify that the read data is the same as the data written

- Program the Primary Master to perform a fast back to back memory write and read cycle but only the read selects the IUT

→ Verify that the read data is the same as the data written

Imaginable Scenarios

Same Property Specified Multiple Times

- 56 pages of Test Plan

- Only 76 functional checks (representing intent)

- Functional checks = high-level requirements

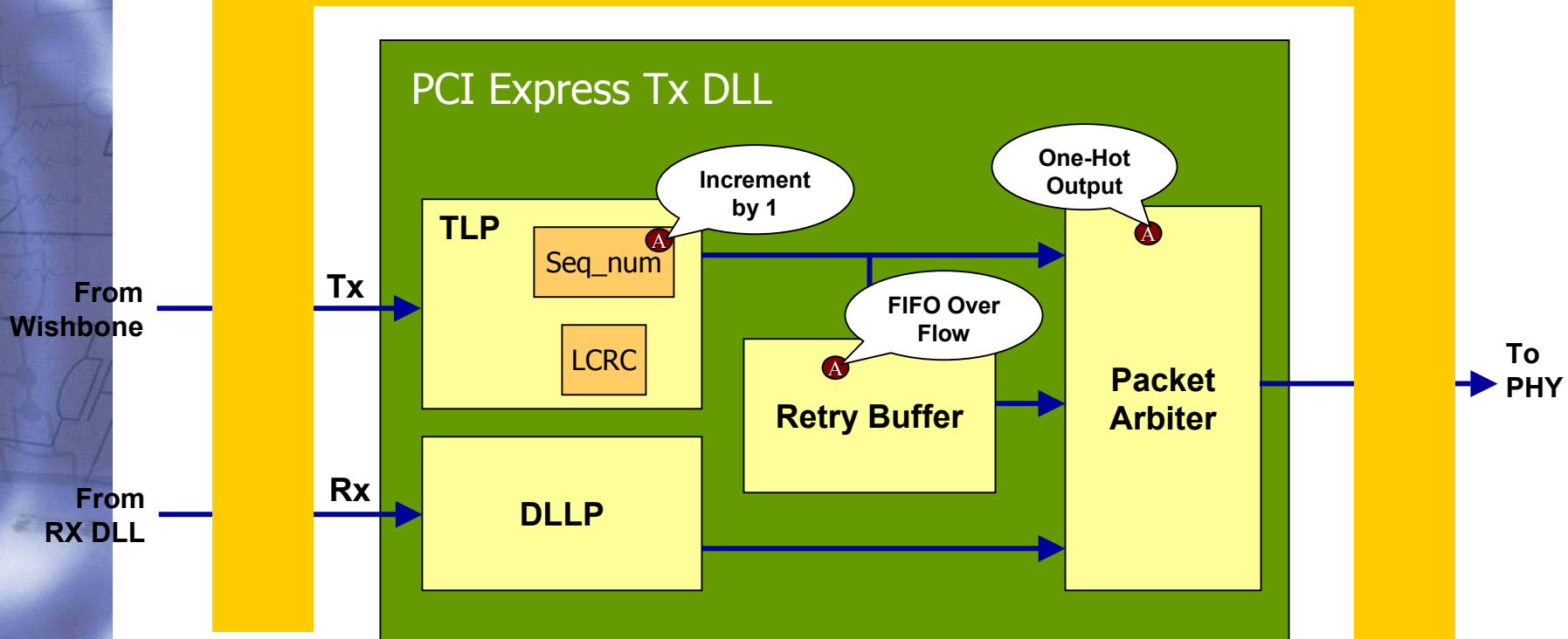
- > 20 to 1 ratio of information related to verification implementation versus intent (checks)

What We've Learned

- There exists a set of high level requirements that completely defines correct behavior of a chip (i.e., μ -Architecture Spec)
- By formally verifying that a design meets its high level requirements, Jasper delivers a quantum leap in verification completeness and productivity
- JasperGold eliminates the need for block-level simulation and dramatically shortens system-level verification

High Level Requirements vs. Assertions

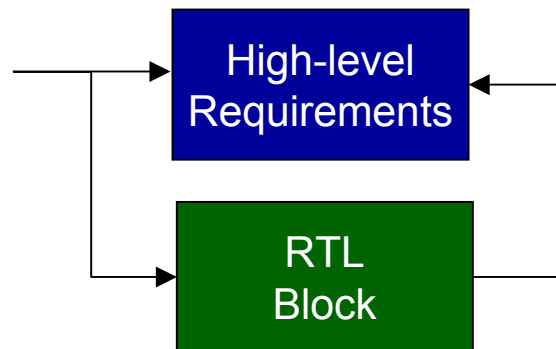
High Level Requirement = no packets are dropped, duplicated, or corrupted



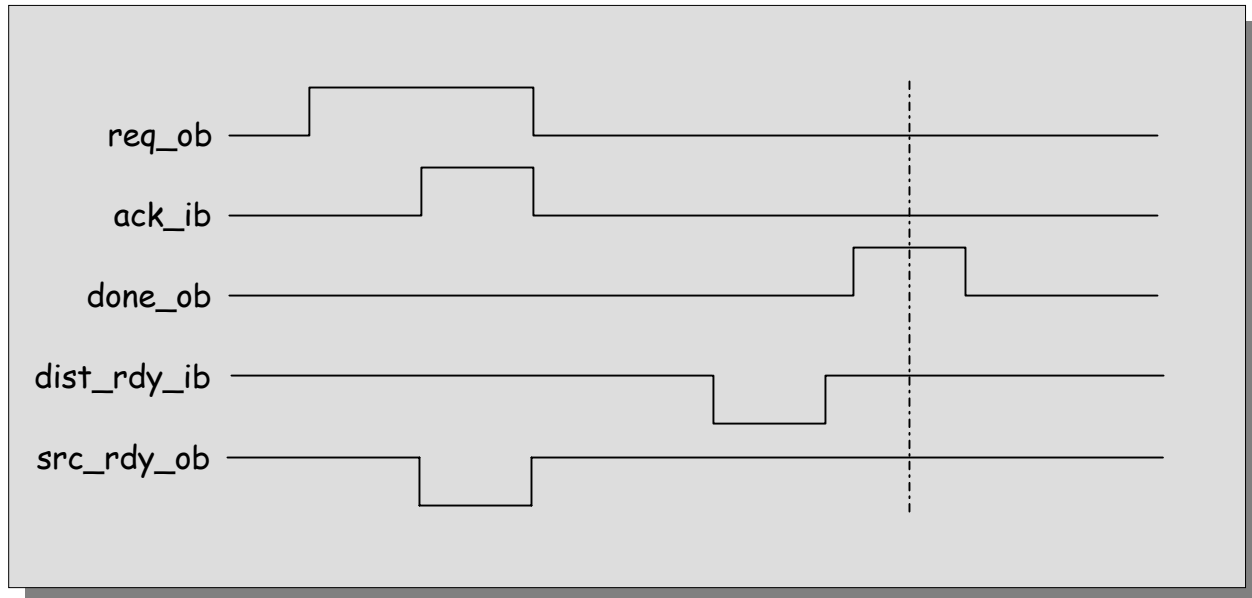
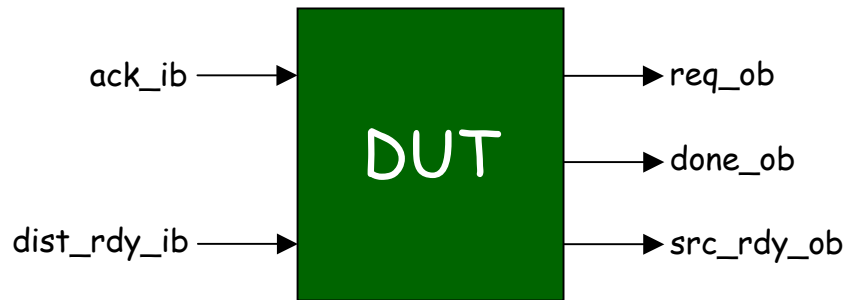
A = Implementation Assertions: low level assertions (similar to writing simulation monitors)

High-level requirements

- *High-level requirements* specify end-to-end micro-architecture behavior
 - Cannot be expressed as simple one-line declarative assertions
- *High-level requirements* consists of three parts:
 - Behavior modeling
 - Input/output relationships/properties
 - “Assertion” to prove



Example of Input/Output Relationship Properties



Modeling Input/Output Relationships

```
// Verilog
// -----
// A Sequence Modeling Conceptual States of the Environment
// -----
always @ (posedge clk or negedge rstN) begin
    if (~rstN)
        state <= `IDLE;
    else begin
        case (state)
            `IDLE:
                if (req_ob) state <= `REQ;
            `REQ:
                if (ack_ib) state <= `DATA;
            `DATA:
                if (done_ob && src_rdy_ob && dest_rdy_ib) state <= `IDLE;
        endcase
    end
end
```

```
// SystemVerilog sequence
((req_ob) |=> (##[0:$] ack_ib
              ##[0:$] done_ob && src_rdy_ob && dest_rdy_ib))
```

SystemVerilog expressiveness

RTL Block-level Design	<i>Verilog</i> High-level Requirements Model	<i>SystemVerilog</i> High-level Requirements Model
5K lines	<500 lines	50-100 lines

Summary

- Objectives of High-Level Requirements-Based Verification:
 - Exhaustively verify most RTL blocks (designer-size chunks) with JasperGold - *eliminating block-level simulation*
 - Dramatically shorten system simulation by integrating “clean” blocks
 - Achieve >30% verification productivity improvement
- SystemVerilog offers 5x or greater efficiency to the modeling of High-Level Requirements