



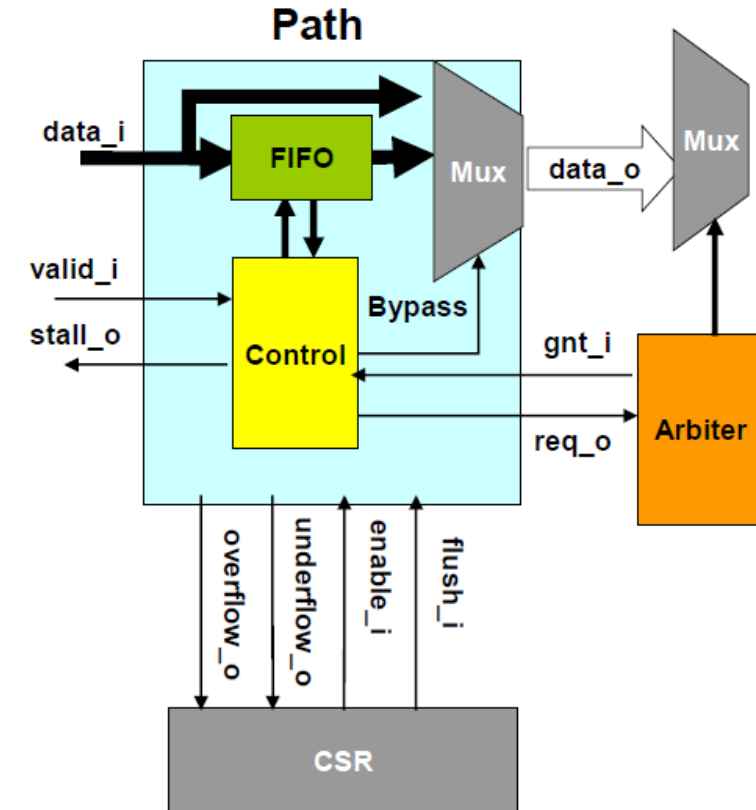
Computer-Aided VLSI System Design - **HW5**

SVA Properties for Formal Testbench



Spec – Path Block Description

- ❖ The path block is a data buffer connecting two blocks:
 - ❖ Data enters the path when *valid_i* is asserted and *stall_o* is not asserted.
 - ❖ The data is stored in a FIFO if it is not immediately transmitted (bypass).
 - ❖ Data leaves the path by way of an arbitration (req/gnt) mechanism.
 - ❖ We only have the req/gnt arbitration if the path has been supplied with valid data.
 - ❖ Several control and status signals are also associated with the path.





Spec – Path Block Components

❖ FIFO Control block that deals with

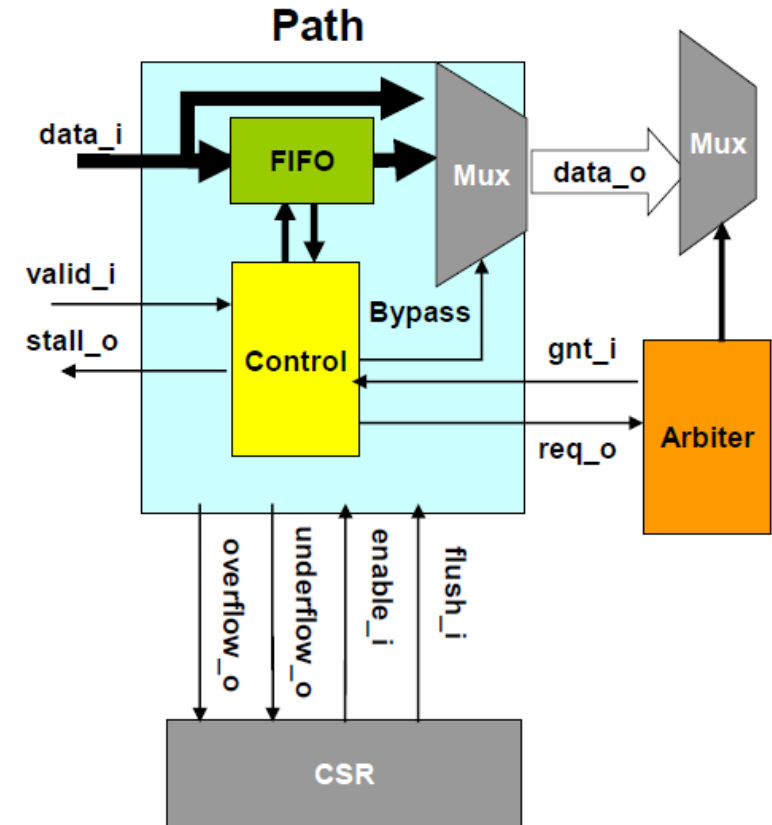
- ❖ Arbitration
- ❖ Bypass
- ❖ I/F to CSR
- ❖ `stall_o = (full && !gnt_i) || !enable;`

❖ Mux for bypass

- ❖ `bypass = empty && gnt_i && valid_i;`

❖ Parameters

- ❖ FDEPTH: FIFO depth
 - Default: 5
- ❖ DWIDTH: Data width
 - Default: 8





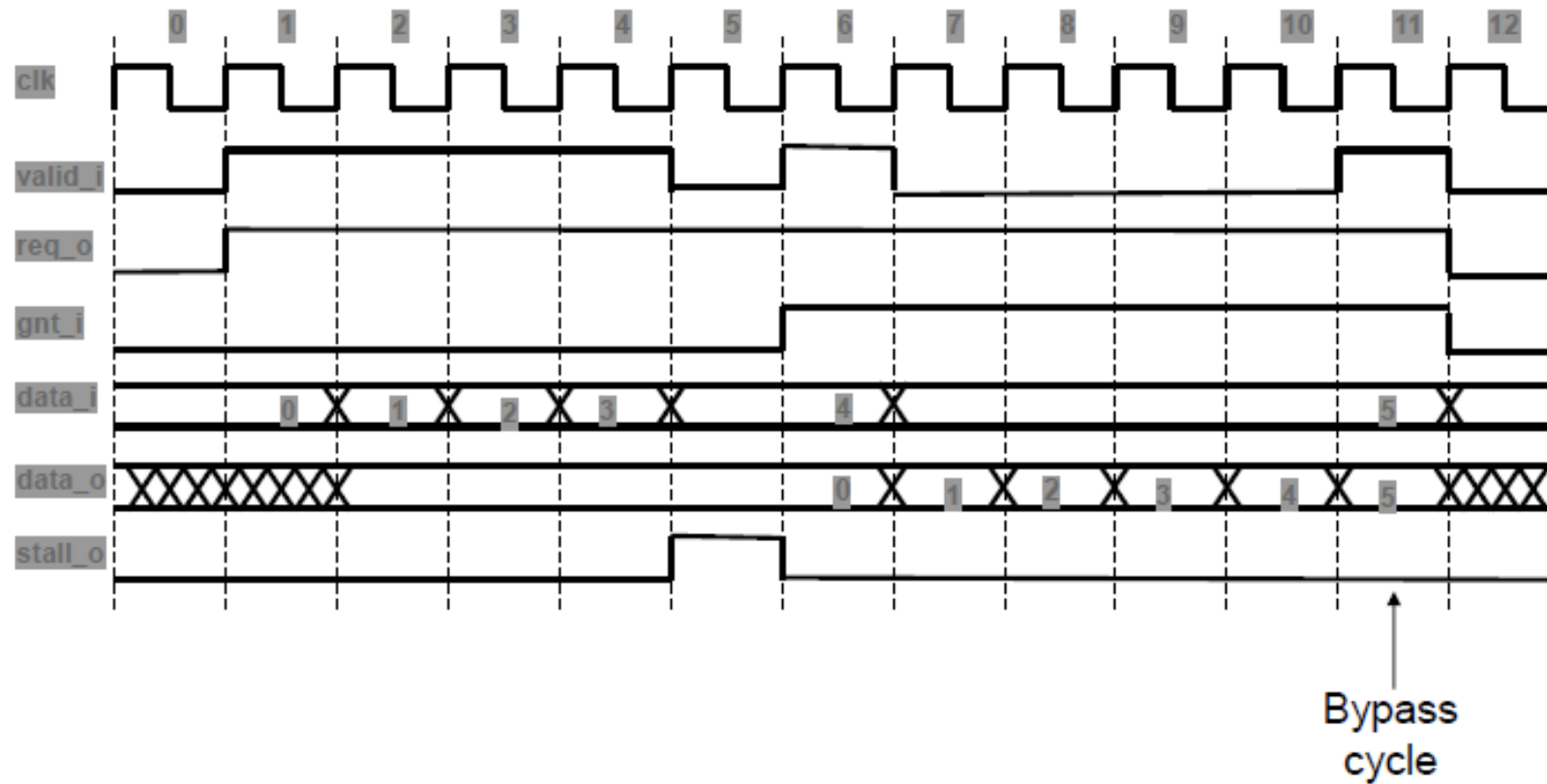
Spec – Path Block I/O

Name	I/O	Width	Description
Global Signals			
clk	I	1	System clock
rst_n	I	1	System reset (active low)
CSR I/F			
enable_i	I	1	Allow path to accept data
flush_i	I	1	Signal to flush FIFO by resetting read/write pointers
overflow_o	O	1	Status bit indicating overflow has occurred
underflow_o	O	1	Status bit indicating underflow had occurred
Arbiter I/F			
req_o	O	1	Request from path to send data out. Must occur on same cycle as data is accepted. Will not be asserted if Path has not accepted data. Will stay high as long as the path has data to transmit. If we observe $\neg \text{rose}(\text{req})$ then implicitly the FIFO will be empty
gnt_i	I	1	Grant from arbiter accepting data. Grant only asserted in response to a request from the path to transmit data
Data I/F			
data_i	I	DWIDTH	Data coming into path
valid_i	I	1	Indicates when data coming into path is valid
data_o	O	DWIDTH	Data coming out of path
stall_o	O	1	Feedback to block driving path to stop sending valid data



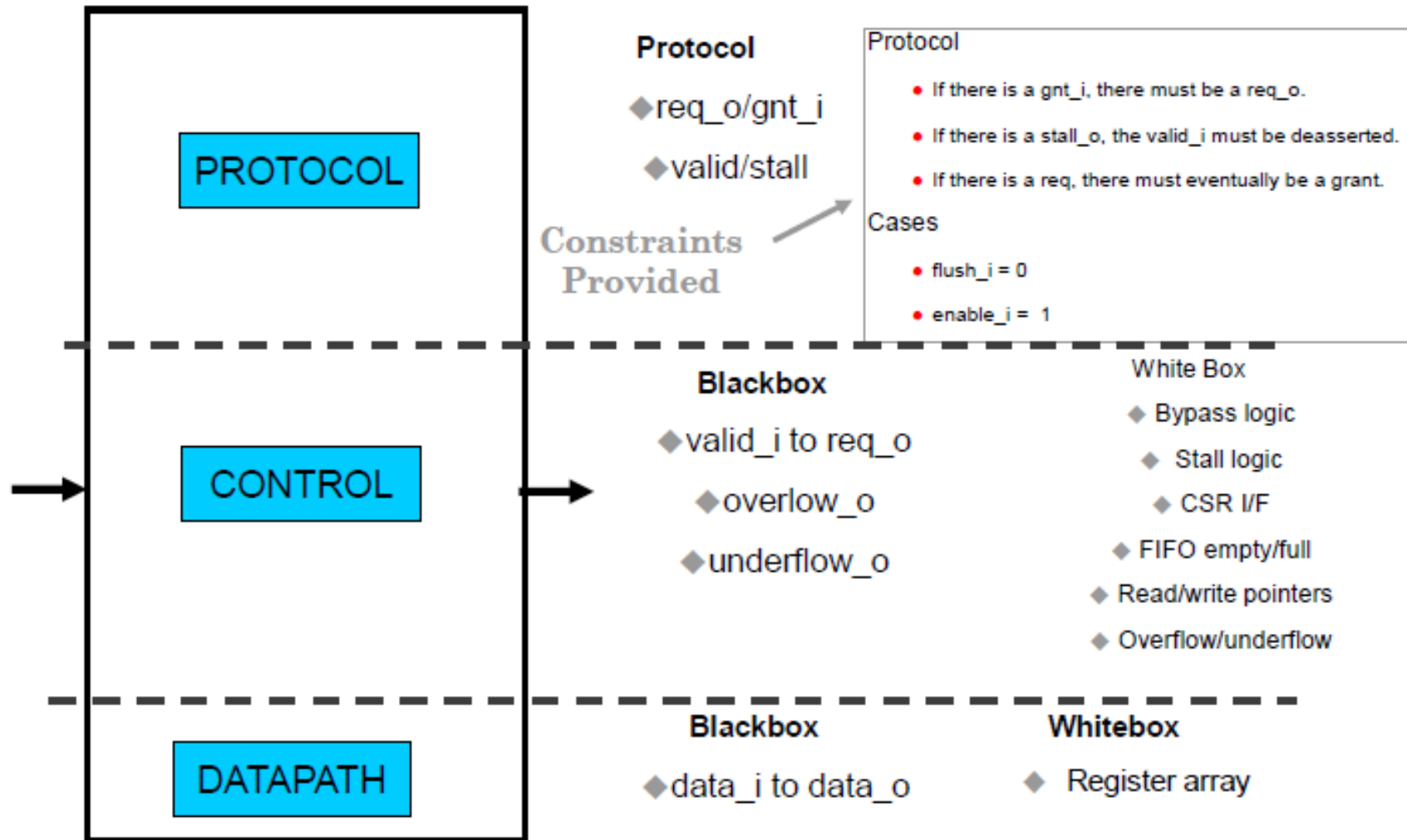
Spec – Path Block

FIFO_DEPTH = 4





Spec – Breaking Down the PATH Design





Spec – FIFO

- ❖ This is non exhaustive list of FIFO signals
- ❖ This list is useful when you define your whitebox properties
- ❖ For more detail, see the RTL code in the lab file indicated in the lab book

Name	I/O	Width	Description
clk	I	1	System clock
rst_n	I	1	System reset (active low)
read_i	I	1	When asserted indicates a read request
write_i	I	1	When asserted indicates a write request
full_o	O	1	Asserted when FIFO is FULL
empty_o	O	1	Asserted when FIFO is EMPTY
rd_ptr	O	ADDR_WIDTH	Internal write pointer in the FIFO. Points to next location to be written to in the FIFO. Counts from 0 to ADDR_WIDTH-1
wr_ptr	O	ADDR_WIDTH	Internal read pointer in the FIFO. Points to next location to be read from the FIFO. Counts from 0 to ADDR_WIDTH-1



Files

- ❖ define/cfunctions.h: define functions of system Verilog
- ❖ Source.txt: Get access to JasperGold
- ❖ sva/rtl
 - ❖ fifo.v: RTL code of FIFO module
 - ❖ Path.v: RTL code of PATH module
- ❖ sva/jaspergold
 - ❖ Vcomp_path.v: Formal module to add properties
 - ❖ jg.tcl: Tcl file for running JasperGold
 - ❖ RUNME.sh: `bash RUNME.sh` to run jg.tcl
 - ❖ CLEAN: `bash CLEAN` to delete JasperGold directory



Objective

- ❖ 1. Finish vcomp_path.v
 - ❖ Add formal properties based on hints in vcomp_path.v
 - ❖ Run JasperGold to verify design

- ❖ 2. Correct fifo.v and path.v
 - ❖ Find bugs in fifo.v and path.v by properties you added in JasperGold
 - ❖ Correct bugs in fifo.v and path.v
 - ❖ Verify again until all properties are proved



Grading Policy

- ❖ (50%) For completeness of properties in vcom_path.v
- ❖ (20%) For correctness of fifo.v
- ❖ (20%) For correctness of path.v
- ❖ (10%) Report



Submission

❖ Files

- ❖ vcomp_path.v
- ❖ fifo.v
- ❖ path.v
- ❖ Report_StudentID.pdf

❖ Naming convention:

- ❖ StudentID_HW5_vk.zip
(k is number of version, $k = 1, 2, \dots$)

Deadline	2018/5/29 12:00 p.m.
IP	140.112.18.84
Port	1232
Account	CVSD_STUDENT
Password	cvsd2018



Report

- ❖ 1. 請指出path.v裡有bug的地方，該如何修正，解釋原因，以及會使那些assertion有counter example。
- ❖ 2. 請指出fifo.v裡有bug的地方，該如何修正，解釋原因，以及會使那些assertion有counter example。
- ❖ 3. 請附上用13條assertion的verify (1) 初始有bug的RTL檔案 以及 (2)修掉bug的RTL檔案 在JasperGold上prove的結果 (2張截圖)。
- ❖ 4. 請完整地列出(包含code)哪幾條assertion，即使是有bug的RTL code，也會被proved。