AXI Protocol Project

About

This project is on AXI3(Advances eXtensible Interface) protocol using System Verilog language simulated on Xilinx Vivado.

AMBA AXI3 Protocol

Overview

The AMBA AXI3 (Advanced eXtensible Interface 3) protocol is a widely adopted on-chip interconnect standard developed by ARM. It forms part of the AMBA 3 specification and is designed to provide a high-performance, flexible, and efficient communication protocol for system-on-chip (SoC) designs.

Architecture

AXI3 employs a five-channel architecture, separating the communication paths for address and data to maximize throughput and reduce latency. These channels include:

- Write Address Channel (AW): Transfers write addresses and control signals from master to slave.
- Write Data Channel (W): Carries write data and write strobes to the slave.
- Write Response Channel (B): Returns acknowledgments and status responses for write operations.
- Read Address Channel (AR): Transfers read addresses and control signals from master to slave.
- Read Data Channel (R): Returns read data and response signals from slave to master.

This separation of addresses and data phases enables pipelined and overlapped transactions, facilitating higher bus utilization and efficiency.

Burst Transfers and Transaction Handling

AXI3 supports burst transfers, where the master sends a single start address accompanied by the burst length and type. The slave then completes the requested sequence of data beats accordingly, without requiring master retransmission of each address.

The protocol supports all commonly required burst types:

- FIXED: Address remains constant throughout the burst.
- INCR (Incrementing): Address increments with each beat.
- WRAP: Address narrows and wraps around at the burst boundary.

Furthermore, AXI3 allows multiple outstanding and out-of-order transactions, identified and tracked using ID tags (AWID, ARID, WID, RID, BID). This facilitates concurrent processing of multiple reads and writes, improving throughput and reducing stalls.

Response and Error Handling

Each read and write operation concludes with response signals indicating the success or failure status:

- BRESP (Write Response): Indicates write transaction status for the entire burst.
- RRESP (Read Response): Indicates read data validity and error information for each data beat.

The protocol defines four standard response types:

- OKAY: Transaction completed successfully.
- EXOKAY: Exclusive access succeeded (used in atomic operations).
- SLVERR: Slave reported an error during transaction.
- DECERR: Decode error; no slave responded to the address.

These detailed responses provide comprehensive status reporting, aiding robust transaction handling and error recovery.

Other Important Features

- Support for exclusive accesses and atomic operations enables safe resource sharing and synchronization in multi-master systems.
- Unaligned data accesses are supported through byte strobes, enhancing flexibility for various data sizes and memory architectures.
- Power efficiency features include optional protocol support for low-power states and clock management.
- Handshake mechanism**:** All channels use VALID and READY signals to manage data flow reliably, allowing dynamic throttling and back-pressure handling.

Applications

The AXI3 protocol serves as a foundational interconnect in modern SoCs, facilitating communication among CPUs, memory controllers, peripherals, and DMA engines. Its high efficiency and flexibility make it ideal for complex, high-bandwidth designs that demand low latency and high throughput.

- High-performance on-chip bus communication.
- Connecting processors, memory controllers, peripherals in SoCs.
- DMA engines, cache coherent interfaces, high-bandwidth data streaming.

Implementation

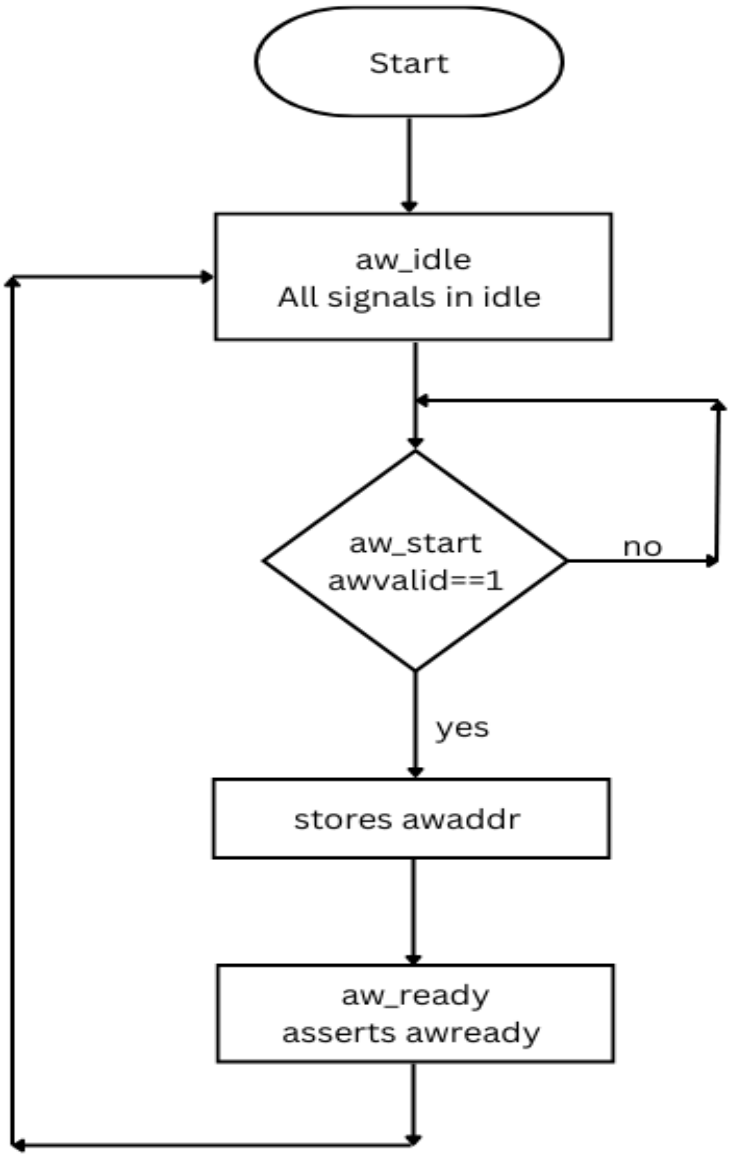
Slave is memory of 128 locations with width 8 bits. The burst lengths are kept as 2, 4, 8 or 16.

The address widths and data widths are 32 bits.

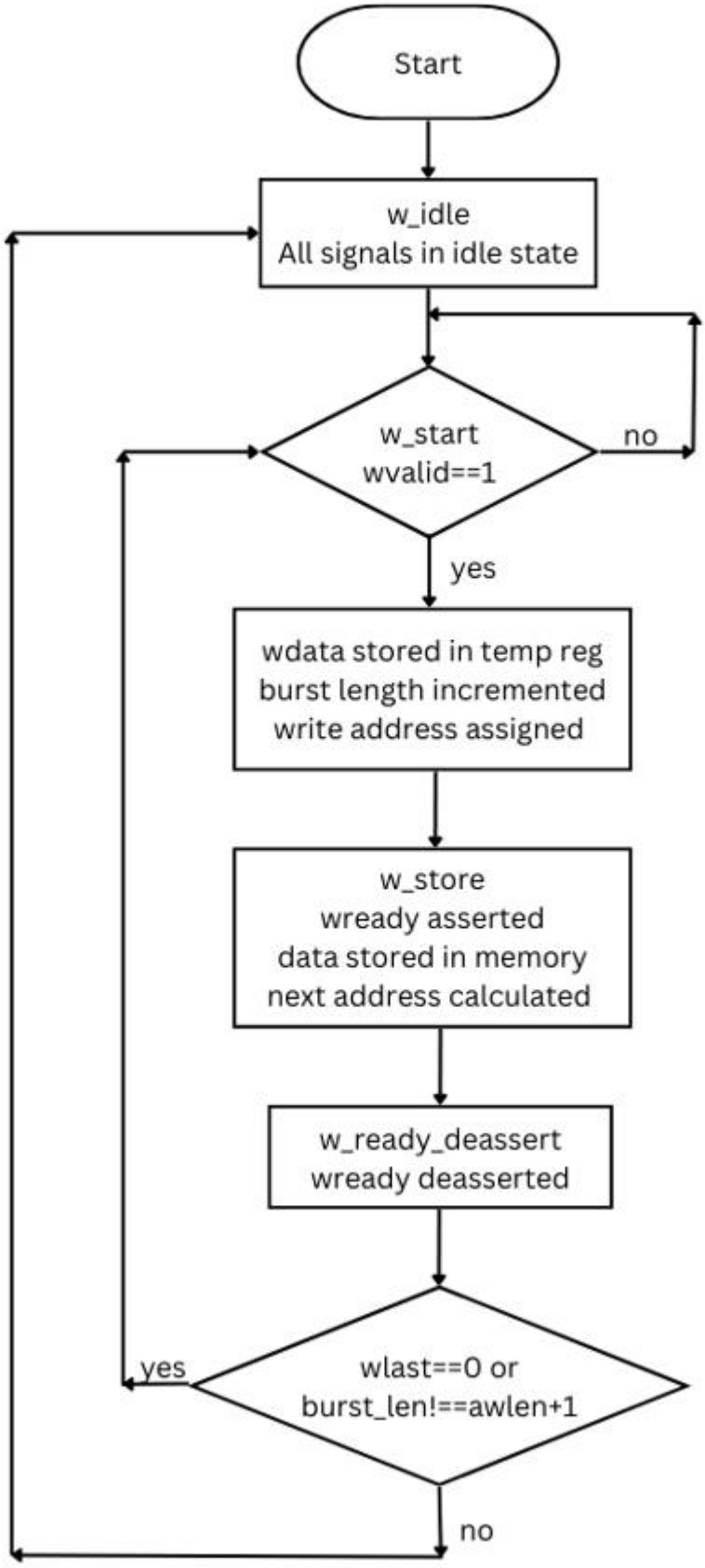
Signals for cache, protection and Lock are ignored.

For each channel separate finite state machines are developed.

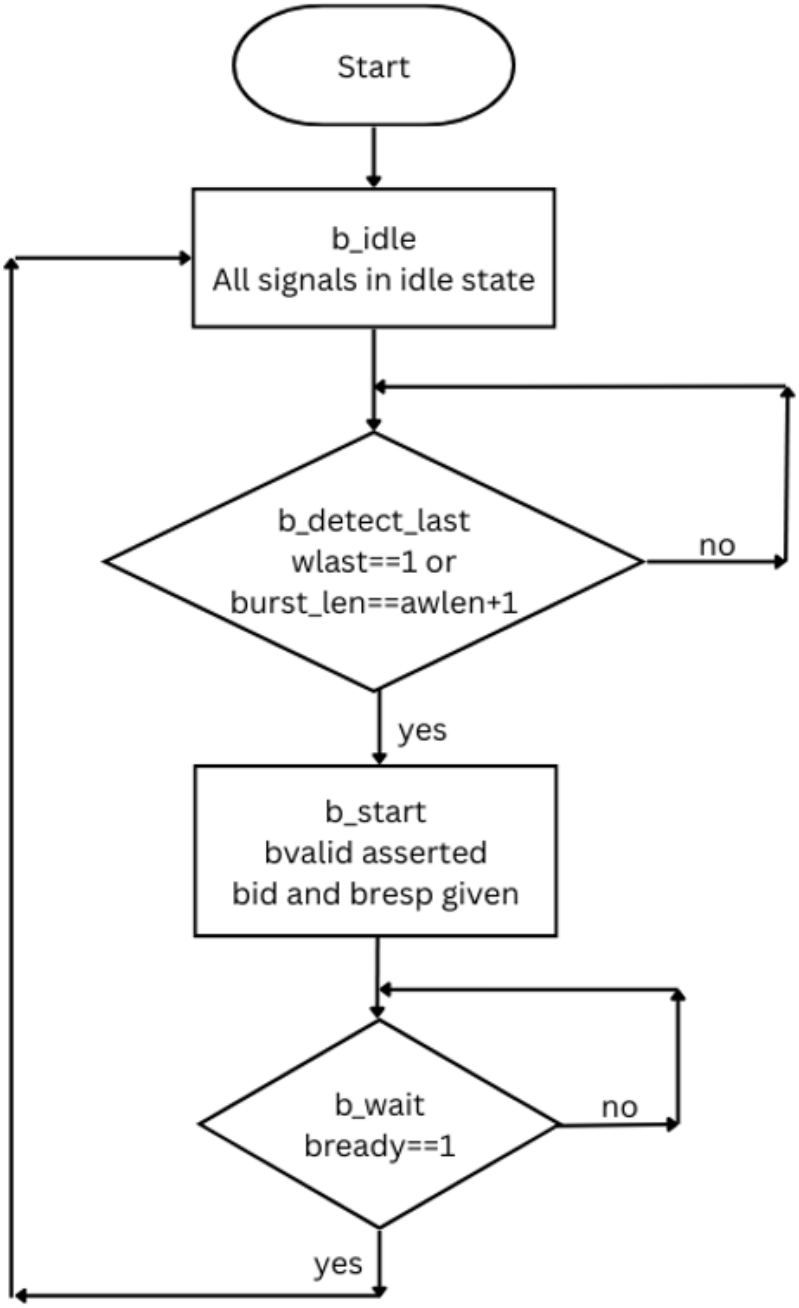
Write address channel flowchart:-



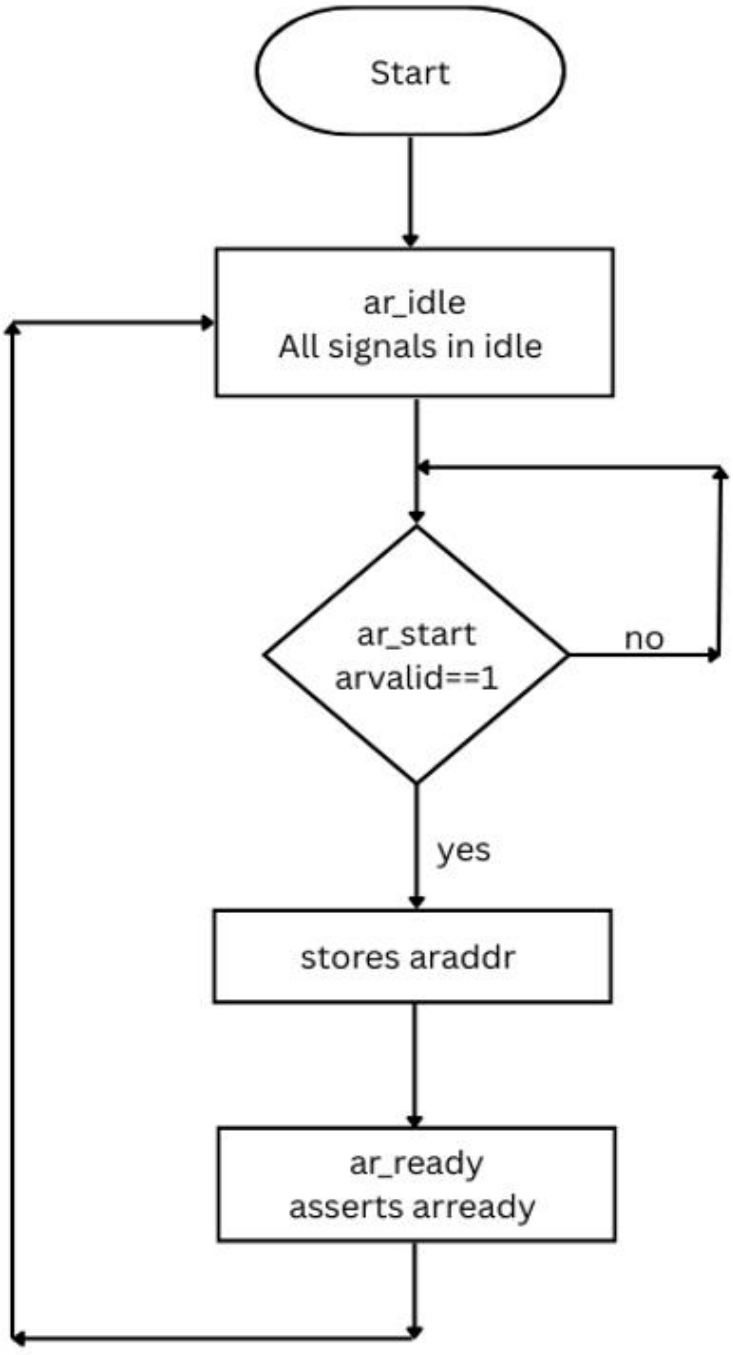
Write Data channel flowchart:-



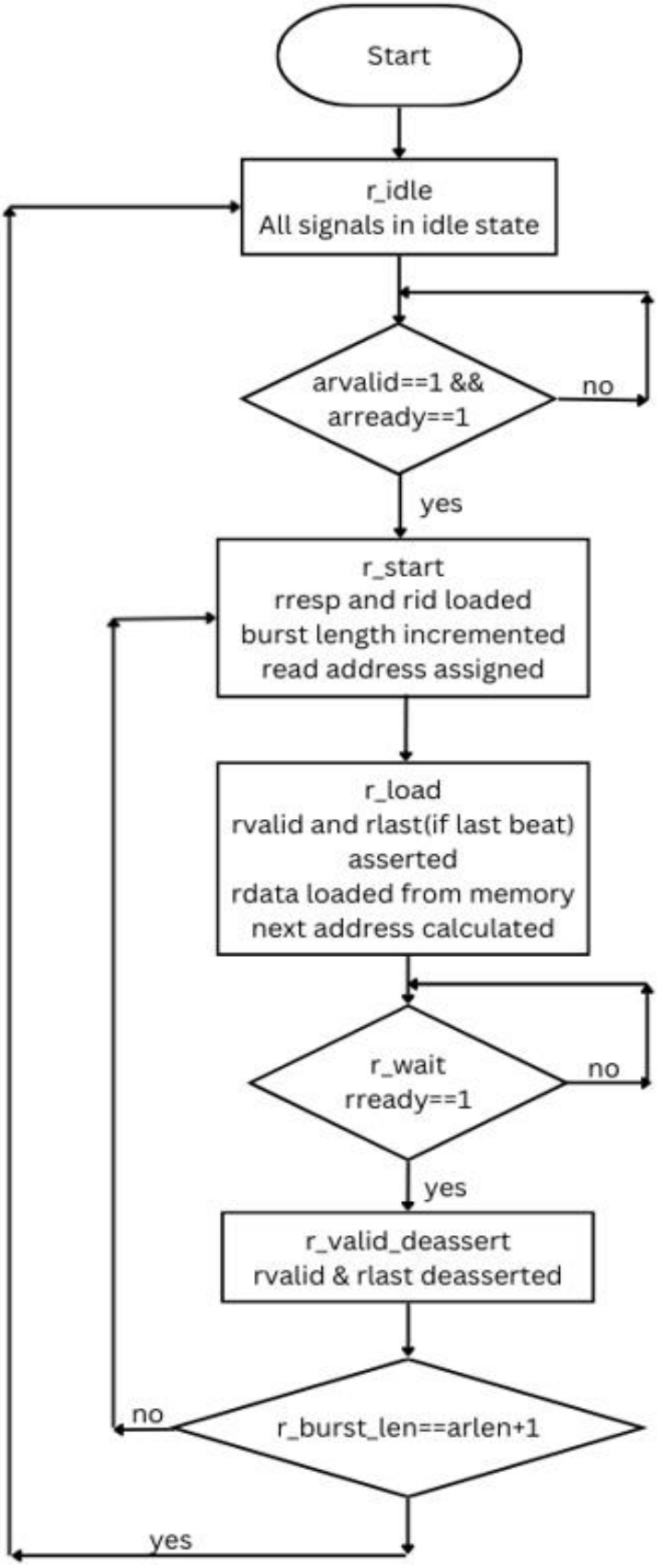
Write Response channel flowchart:-



Read address channel flowchart:-



Read data channel flowchart:-



Bursts functions are created each for read and write. These return the next address after write/read.

Testbench is created using UVM structure:

- Generator generates random stimuli for given number of iterations and sent to driver.
- Driver drives the signals of DUT(design under test) via interface.
- Monitor collects the output of DUT at proper handshake signals timing and sent to scoreboard.
- Scoreboard checks the output with reference model and displays outcome.
- Scoreboard and Agent (includes generator/sequencer, driver, monitor) are under Environment which connects all.
- Testbench (here, top module and test module are same) contains design under test (DUT) and Environment.

Simulation Results

Below is example of simulation waveform and TCL console output when number of iterations given are 5.

```
[DRV] : Reset done
------------------------------------------------------------
1/5
[GEN] : arvalid = 1, awvalid = 0, araddr = 3e, arburst = 1, arsize = 0, arlen = 15
[DRV] : Read Incr Burst
[DRV] : arvalid=1, araddr=62, arlen=15, arsize=0, arburst=1
[MON] : arvalid=1, arburst=1, arlen=15, arsize=0
[MON] : i=1, rdata=3e, read_addr=62, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[62]=3e, mem[63]=3f, mem[64]=40, mem[65]=41
[SCO] : read_addr=62, temp=3e, rdata=3e, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=1, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=2, rdata=3f, read_addr=63, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[63]=3f, mem[64]=40, mem[65]=41, mem[66]=42
[SCO] : read_addr=63, temp=3f, rdata=3f, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=2, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=3, rdata=40, read_addr=64, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
```

```
[SCO] : mem[64]=40, mem[65]=41, mem[66]=42, mem[67]=43
[SCO] : read_addr=64, temp=40, rdata=40, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=3, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=4, rdata=41, read_addr=65, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[65]=41, mem[66]=42, mem[67]=43, mem[68]=44
[SCO] : read_addr=65, temp=41, rdata=41, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=4, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=5, rdata=42, read_addr=66, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[66]=42, mem[67]=43, mem[68]=44, mem[69]=45
[SCO] : read_addr=66, temp=42, rdata=42, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=5, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=6, rdata=43, read_addr=67, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[67]=43, mem[68]=44, mem[69]=45, mem[70]=46
[SCO] : read_addr=67, temp=43, rdata=43, rlast=0
```

```
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=6, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=7, rdata=44, read_addr=68, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[68]=44, mem[69]=45, mem[70]=46, mem[71]=47
[SCO] : read_addr=68, temp=44, rdata=44, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=7, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=8, rdata=45, read_addr=69, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[69]=45, mem[70]=46, mem[71]=47, mem[72]=48
[SCO] : read_addr=69, temp=45, rdata=45, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=8, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=9, rdata=46, read_addr=70, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[70]=46, mem[71]=47, mem[72]=48, mem[73]=49
[SCO] : read_addr=70, temp=46, rdata=46, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
```

```
[SCO] : len=9, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=10, rdata=47, read_addr=71, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[71]=47, mem[72]=48, mem[73]=49, mem[74]=4a
[SCO] : read_addr=71, temp=47, rdata=47, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=10, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=11, rdata=48, read_addr=72, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[72]=48, mem[73]=49, mem[74]=4a, mem[75]=4b
[SCO] : read_addr=72, temp=48, rdata=48, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=11, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=12, rdata=49, read_addr=73, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[73]=49, mem[74]=4a, mem[75]=4b, mem[76]=4c
[SCO] : read_addr=73, temp=49, rdata=49, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=12, arlen=15, awlen=0
------------------------------------------------------------
```

```
[MON] : i=13, rdata=4a, read_addr=74, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[74]=4a, mem[75]=4b, mem[76]=4c, mem[77]=4d
[SCO] : read_addr=74, temp=4a, rdata=4a, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=13, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=14, rdata=4b, read_addr=75, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[75]=4b, mem[76]=4c, mem[77]=4d, mem[78]=4e
[SCO] : read_addr=75, temp=4b, rdata=4b, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=14, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=15, rdata=4c, read_addr=76, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
[SCO] : mem[76]=4c, mem[77]=4d, mem[78]=4e, mem[79]=4f
[SCO] : read_addr=76, temp=4c, rdata=4c, rlast=0
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : len=15, arlen=15, awlen=0
------------------------------------------------------------
[MON] : i=16, rdata=4d, read_addr=77, rlast=1
[SCO] : arvalid=1, araddr=0, arburst=1, arlen=15, arsize=0
```

```
[SCO] : mem[77]=4d, mem[78]=4e, mem[79]=4f, mem[80]=50
[SCO] : read_addr=77, temp=4d, rdata=4d, rlast=1
[SCO] : Match
[SCO] : rid=1, rresp=0
[SCO] : ->sconext
------------------------------------------------------------
[DRV] : Read Incr Burst DONE
[DRV] : ->drvnext
------------------------------------------------------------
2/5
[GEN] : arvalid = 1, awvalid = 0, araddr = 47, arburst = 0, arsize = 0, arlen = 3
[DRV] : Read Fixed Burst
[DRV] : arvalid=1, araddr=71, arlen=3, arsize=0, arburst=0
[MON] : arvalid=1, arburst=0, arlen=3, arsize=0
[MON] : i=1, rdata=47, read_addr=71, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=0
[SCO] : mem[71]=47, mem[72]=48, mem[73]=49, mem[74]=4a
[SCO] : read_addr=71, temp=47, rdata=47, rlast=0
[SCO] : Match
[SCO] : rid=0, rresp=0
[SCO] : len=1, arlen=3, awlen=0
------------------------------------------------------------
[MON] : i=2, rdata=47, read_addr=71, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=0
[SCO] : mem[71]=47, mem[72]=48, mem[73]=49, mem[74]=4a
[SCO] : read_addr=71, temp=47, rdata=47, rlast=0
```

```
[SCO] : Match
[SCO] : rid=0, rresp=0
[SCO] : len=2, arlen=3, awlen=0
------------------------------------------------------------
[MON] : i=3, rdata=47, read_addr=71, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=0
[SCO] : mem[71]=47, mem[72]=48, mem[73]=49, mem[74]=4a
[SCO] : read_addr=71, temp=47, rdata=47, rlast=0
[SCO] : Match
[SCO] : rid=0, rresp=0
[SCO] : len=3, arlen=3, awlen=0
------------------------------------------------------------
[MON] : i=4, rdata=47, read_addr=71, rlast=1
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=0
[SCO] : mem[71]=47, mem[72]=48, mem[73]=49, mem[74]=4a
[SCO] : read_addr=71, temp=47, rdata=47, rlast=1
[SCO] : Match
[SCO] : rid=0, rresp=0
[SCO] : ->sconext
------------------------------------------------------------
[DRV] : Read Fixed Burst DONE
[DRV] : ->drvnext
------------------------------------------------------------
3/5
[GEN] : awvalid=1, arvalid=0, awaddr=4a, awburst=0, awsize=1, awlen=7, wdata=340a2ffc
[DRV] : Write Fixed Burst
```

```
[MON] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[DRV] : i=1, wvalid=1, wdata=7ab3dfb, wstrb=1010
[MON] : i=1, wdata=7ab3dfb, write_addr=74, wlast=0
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[SCO] : mem[74]=3d, mem[75]=7, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=7ab3dfb, write_addr=74, wstrb=1010, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=1, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=2, wvalid=1, wdata=88e7ea2b, wstrb=1010
[MON] : i=2, wdata=88e7ea2b, write_addr=74, wlast=0
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[SCO] : mem[74]=ea, mem[75]=88, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=88e7ea2b, write_addr=74, wstrb=1010, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=2, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=3, wvalid=1, wdata=8073062b, wstrb=1010
[MON] : i=3, wdata=8073062b, write_addr=74, wlast=0
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[SCO] : mem[74]=6, mem[75]=80, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=8073062b, write_addr=74, wstrb=1010, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=3, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=4, wvalid=1, wdata=17a64d61, wstrb=1010
```

```
[MON] : i=4, wdata=17a64d61, write_addr=74, wlast=0
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[SCO] : mem[74]=4d, mem[75]=17, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=17a64d61, write_addr=74, wstrb=1010, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=4, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=5, wvalid=1, wdata=5ea25d31, wstrb=1010
[MON] : i=5, wdata=5ea25d31, write_addr=74, wlast=0
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[SCO] : mem[74]=5d, mem[75]=5e, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=5ea25d31, write_addr=74, wstrb=1010, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=5, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=6, wvalid=1, wdata=5b0c358f, wstrb=1010
[MON] : i=6, wdata=5b0c358f, write_addr=74, wlast=0
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[SCO] : mem[74]=35, mem[75]=5b, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=5b0c358f, write_addr=74, wstrb=1010, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=6, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=7, wvalid=1, wdata=492890cc, wstrb=1010
[MON] : i=7, wdata=492890cc, write_addr=74, wlast=0
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
```

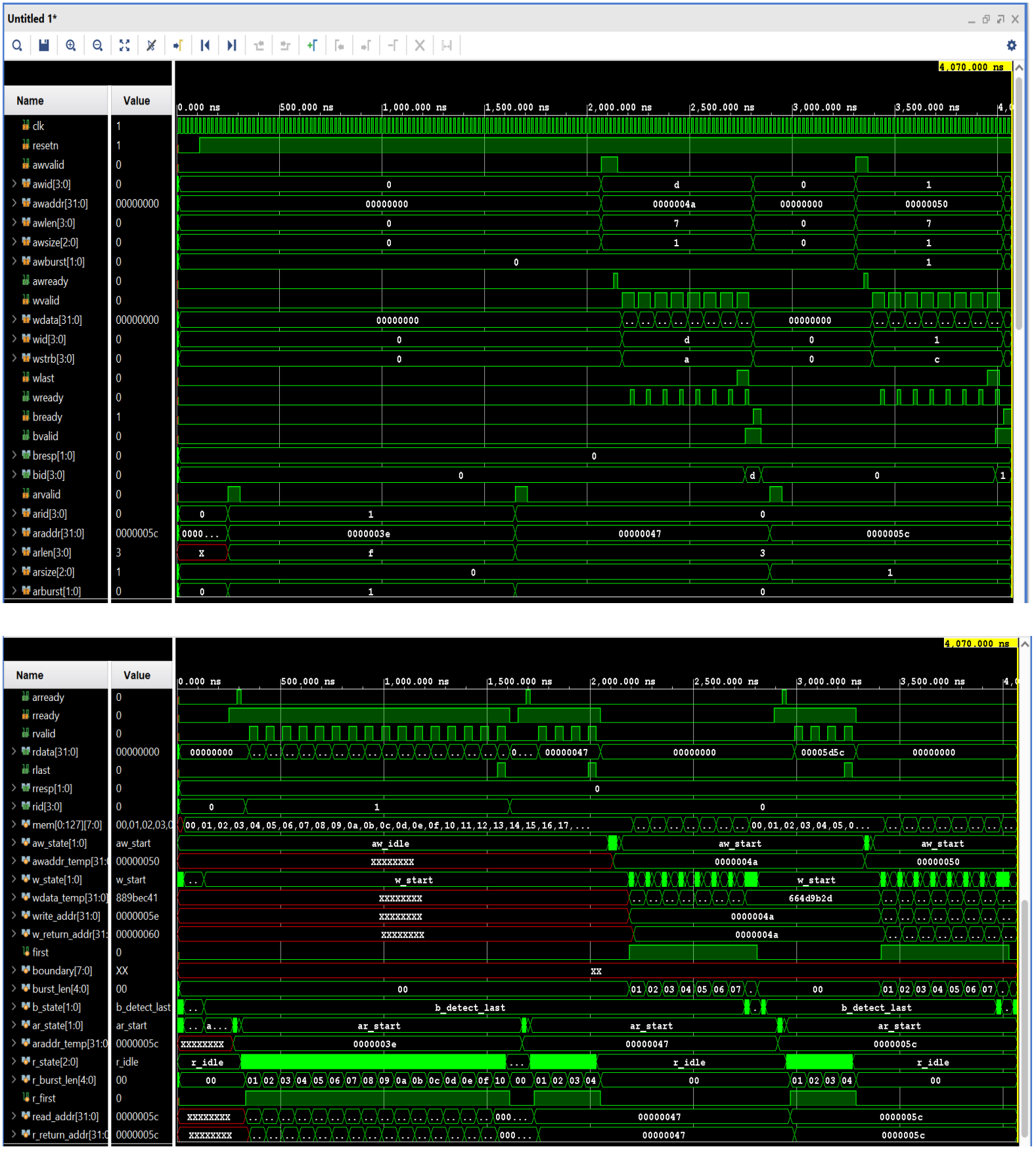
```
[SCO] : mem[74]=90, mem[75]=49, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=492890cc, write_addr=74, wstrb=1010, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=7, awlen=7, arlen=3
----------------------------------------------------------------
[DRV] : i=8, wvalid=1, wdata=664d9b2d, wstrb=1010
[MON] : i=8, wdata=664d9b2d, write_addr=74, wlast=1
[SCO] : awvalid=1, awaddr=74, awburst=0, awlen=7, awsize=1
[SCO] : mem[74]=9b, mem[75]=66, mem[76]=4c, mem[77]=4d
[SCO] : DATA WRITTEN wdata=664d9b2d, write_addr=74, wstrb=1010, wlast=1
[SCO] : bid=13, bresp=0
[SCO] : ->sconext
----------------------------------------------------------------
[DRV] : Write Fixed Burst DONE
[DRV] : ->drvnext
----------------------------------------------------------------
4/5
[GEN] : arvalid = 1, awvalid = 0, araddr = 5c, arburst = 0, arsize = 1, arlen = 3
[DRV] : Read Fixed Burst
[DRV] : arvalid=1, araddr=92, arlen=3, arsize=1, arburst=0
[MON] : arvalid=1, arburst=0, arlen=3, arsize=1
[MON] : i=1, rdata=5d5c, read_addr=92, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=1
[SCO] : mem[92]=5c, mem[93]=5d, mem[94]=5e, mem[95]=5f
[SCO] : read_addr=92, temp=5d5c, rdata=5d5c, rlast=0
[SCO] : Match
```

```
[SCO] : rid=0, rresp=0
[SCO] : len=1, arlen=3, awlen=7
----------------------------------------------------------------
[MON] : i=2, rdata=5d5c, read_addr=92, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=1
[SCO] : mem[92]=5c, mem[93]=5d, mem[94]=5e, mem[95]=5f
[SCO] : read_addr=92, temp=5d5c, rdata=5d5c, rlast=0
[SCO] : Match
[SCO] : rid=0, rresp=0
[SCO] : len=2, arlen=3, awlen=7
----------------------------------------------------------------
[MON] : i=3, rdata=5d5c, read_addr=92, rlast=0
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=1
[SCO] : mem[92]=5c, mem[93]=5d, mem[94]=5e, mem[95]=5f
[SCO] : read_addr=92, temp=5d5c, rdata=5d5c, rlast=0
[SCO] : Match
[SCO] : rid=0, rresp=0
[SCO] : len=3, arlen=3, awlen=7
----------------------------------------------------------------
[MON] : i=4, rdata=5d5c, read_addr=92, rlast=1
[SCO] : arvalid=1, araddr=0, arburst=0, arlen=3, arsize=1
[SCO] : mem[92]=5c, mem[93]=5d, mem[94]=5e, mem[95]=5f
[SCO] : read_addr=92, temp=5d5c, rdata=5d5c, rlast=1
[SCO] : Match
[SCO] : rid=0, rresp=0
[SCO] : ->sconext
```

```
----------------------------------------------------------------
[DRV] : Read Fixed Burst DONE
[DRV] : ->drvnext
----------------------------------------------------------------
5/5
[GEN] : awvalid=1, arvalid=0, awaddr=50, awburst=1, awsize=1, awlen=7, wdata=35317c67
[DRV] : Write Incr Burst
[DRV] : awvalid=1, awaddr=80, awlen=7, awsize=1, awburst=1, bready==0
[MON] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[DRV] : i=1, wvalid=1, wdata=63f41fba, wstrb=1100
[MON] : i=1, wdata=63f41fba, write_addr=80, wlast=0
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[SCO] : mem[80]=f4, mem[81]=63, mem[82]=52, mem[83]=53
[SCO] : DATA WRITTEN wdata=63f41fba, write_addr=80, wstrb=1100, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=1, awlen=7, arlen=3
----------------------------------------------------------------
[DRV] : i=2, wvalid=1, wdata=499b96f7, wstrb=1100
[MON] : i=2, wdata=499b96f7, write_addr=82, wlast=0
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[SCO] : mem[82]=9b, mem[83]=49, mem[84]=54, mem[85]=55
[SCO] : DATA WRITTEN wdata=499b96f7, write_addr=82, wstrb=1100, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=2, awlen=7, arlen=3
----------------------------------------------------------------
[DRV] : i=3, wvalid=1, wdata=7df3e88e, wstrb=1100
```

```
[MON] : i=3, wdata=7df3e88e, write_addr=84, wlast=0
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[SCO] : mem[84]=f3, mem[85]=7d, mem[86]=56, mem[87]=57
[SCO] : DATA WRITTEN wdata=7df3e88e, write_addr=84, wstrb=1100, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=3, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=4, wvalid=1, wdata=42967cec, wstrb=1100
[MON] : i=4, wdata=42967cec, write_addr=86, wlast=0
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[SCO] : mem[86]=96, mem[87]=42, mem[88]=58, mem[89]=59
[SCO] : DATA WRITTEN wdata=42967cec, write_addr=86, wstrb=1100, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=4, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=5, wvalid=1, wdata=24a7cc86, wstrb=1100
[MON] : i=5, wdata=24a7cc86, write_addr=88, wlast=0
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[SCO] : mem[88]=a7, mem[89]=24, mem[90]=5a, mem[91]=5b
[SCO] : DATA WRITTEN wdata=24a7cc86, write_addr=88, wstrb=1100, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=5, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=6, wvalid=1, wdata=471c795a, wstrb=1100
[MON] : i=6, wdata=471c795a, write_addr=90, wlast=0
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1


[SCO] : mem[90]=1c, mem[91]=47, mem[92]=5c, mem[93]=5d
[SCO] : DATA WRITTEN wdata=471c795a, write_addr=90, wstrb=1100, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=6, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=7, wvalid=1, wdata=a8e41819, wstrb=1100
[MON] : i=7, wdata=a8e41819, write_addr=92, wlast=0
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[SCO] : mem[92]=e4, mem[93]=a8, mem[94]=5e, mem[95]=5f
[SCO] : DATA WRITTEN wdata=a8e41819, write_addr=92, wstrb=1100, wlast=0
[SCO] : bid=0, bresp=0
[SCO] : len=7, awlen=7, arlen=3
------------------------------------------------------------
[DRV] : i=8, wvalid=1, wdata=889bec41, wstrb=1100
[MON] : i=8, wdata=889bec41, write_addr=94, wlast=1
[SCO] : awvalid=1, awaddr=80, awburst=1, awlen=7, awsize=1
[SCO] : mem[94]=9b, mem[95]=88, mem[96]=60, mem[97]=61
[SCO] : DATA WRITTEN wdata=889bec41, write_addr=94, wstrb=1100, wlast=1
[SCO] : bid=1, bresp=0
[SCO] : ->sconext
------------------------------------------------------------
[DRV] : Write Incr Burst DONE
[DRV] : ->drvnext
$finish called at time : 4070 ns : File "D:/Xilinx Projects/axi/axi.srcs/sim_1/new/tb.sv" Line 930
```


Untitled 1*
4,070.000 ns
Name
Value
0.000 ns
500.000 ns
1,000.000 ns
1,500.000 ns
2,000.000 ns
2,500.000 ns
3,000.000 ns
3,500.000 ns
clk 1
resetn 1
awvalid 0
awid[3:0] 0
awaddr[31:0] 00000000
awlen[3:0] 0
awsize[2:0] 0
awburst[1:0] 0
awready 0
wvalid 0
wdata[31:0] 00000000
wid[3:0] 0
wstrb[3:0] 0
wlast 0
wready 0
bready 1
bvalid 0
bresp[1:0] 0
bid[3:0] 0
arvalid 0
arid[3:0] 0
araddr[31:0] 0000005c
arlen[3:0] 3
arsize[2:0] 1
arburst[1:0] 0
4,070.000 ns
Name
Value
0.000 ns
500.000 ns
1,000.000 ns
1,500.000 ns
2,000.000 ns
2,500.000 ns
3,000.000 ns
3,500.000 ns
aready 0
rready 0
rvalid 0
rdata[31:0] 00000000
rlast 0
rresp[1:0] 0
rid[3:0] 0
mem[0:127][7:0] 00,01,02,03,0
aw_state[1:0] aw_start
awaddr_temp[31:0] 00000050
w_state[1:0] w_start
wdata_temp[31:0] 889bec41
write_addr[31:0] 0000005e
w_return_addr[31:0] 00000060
first 0
boundary[7:0] XX
burst_len[4:0] 00
b_state[1:0] b_detect_last
ar_state[1:0] ar_start
araddr_temp[31:0] 0000005c
r_state[2:0] r_idle
r_burst_len[4:0] 00
r_first 0
read_addr[31:0] 0000005c
r_return_addr[31:0] 0000005c

Differences between AXI3 and AXI4 Full Signals

Aspect	**AXI3 Protocol**	**AXI4 Protocol**
AWLEN	Narrower signal width	Wider AWLEN signal allowing longer bursts
AWLOCK	Multi-bit signal supporting locked transfers	Reduced to a single bit supporting exclusive transfers only (locked transfers not supported)
AWQOS	Not present	Added signal supporting Quality of Service (QoS)
AWREGION	Not present	Added signal supporting subordinate regions allowing multiple logical interfaces on one physical subordinate
WID	Present, supports write data reordering	Removed because write data reordering is no longer allowed
User-defined signals	Not supported	Added to each channel
ARLEN	Narrower signal width	Wider ARLEN allowing longer read bursts
ARLOCK	Multi-bit signal supporting locked transfers	Reduced to a single bit supporting exclusive transfers only (locked transfers not supported)
ARQOS	Not present	Added signal supporting Quality of Service (QoS)
ARREGION	Not present	Added signal supporting subordinate regions same as AWREGION
User-defined signals	Not supported	Added to both read address and read data channels

Thank you

(by Vanashree Parate)