



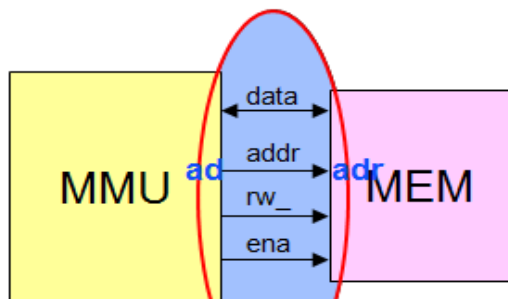
Centurion
UNIVERSITY

Shaping Lives

Interfaces

Great coding efficiency can be achieved by modeling the blocks of a system at different levels of abstraction, behavioral, rtl, gate, etc.

In Verilog, the I/O between these blocks has always remained at the lowest “wire” level. High-performance system-level simulation requires the abstraction of inter-block communication.



interface

At it's simplest
an interface is
like a module
for ports/wires

```
module mmu(d, a, rw_, en);
    output [15:0] a;
    output rw_, en;
    inout [7:0] d;
    ...
endmodule

module mem(d, a, rw_, en);
    input [15:0] a;
    input rw_, en;
    inout [7:0] d;
    ...
endmodule

module system;
    wire [7:0] data;
    wire [15:0] addr;
    wire ena, rw_;

    mmu U1 (data, addr, rw_, ena);
    mem U2 (data, addr, rw_, ena);
endmodule
```

Traditional
Verilog

```
interface interf;
    logic [7:0] data;
    logic [15:0] addr;
    logic ena, rw_;
endinterface

module mmu(interf io);
    io.addr <= ad;
    ...
endmodule

module mem(interf io);
    adr = io.addr;
    ...
endmodule

module system;
    interf i1;
    mmu U1 (i1);
    mem U2 (i1);
endmodule
```

SystemVerilog



Centurion
UNIVERSITY

Shaping Lives...

Improving Communities...

Interfaces – Characteristics

Brings abstraction-level enhancements to ports and internals

Interface may contain any legal System Verilog code except module definitions and/or instances.

tasks, functions, initial/always blocks, parameters, etc.

Bus timing, pipelining, etc. may be captured in an interface rather than the connecting modules.

Interfaces are defined once and used widely, so it simplifies design.

Interfaces are synthesizable.



Centurion
UNIVERSITY

*Shaping Lives...
Empowering Communities...*

ModPorts

Different users of interface need different views

Master/Slave

Monitor/driver

Modport construct allows you to group signals and specify directions



Centurion
UNIVERSITY
Shaping Lives

Mod port example

Example 5-9 Interface with modports

```
interface arb_if(input bit clk);
    logic [1:0] grant, request;
    logic reset;

    modport TEST (output request, reset,
                 input grant, clk);

    modport DUT (input request, reset, clk,
                output grant);

    modport MONITOR (input request, grant, reset, clk);
endinterface
```



Centurion
UNIVERSITY

Shaping Lives
Empowering the Future

Mod port example - usage

Example 5-12 Arbiter model with interface using modports

```
module monitor (arb_if.MONITOR arbif);

    always @(posedge arbif.request[0]) begin
        $display("@%0d: request[0] asserted", $time);
        @(posedge arbif.grant[0]);
        $display("@%0d: grant[0] asserted", $time);
    end

    always @(posedge arbif.request[1]) begin
        $display("@%0d: request[1] asserted", $time);
        @(posedge arbif.grant[1]);
        $display("@%0d: grant[1] asserted", $time);
    end
endmodule
```