A Simple Network Switch (In Verilog)

EE685, Fall 2021

Hank Dietz

http://aggregate.org/hankd/

University of Kentucky

Clocking Over A Network

- Lots of wires increases cost
  - Data paths narrower than transfer unit
  - Bidirectional links reduce wire count, but require a protocol to avoid conflicts
- Often not practical to have a global clock
  - Drive problems with high fanout
  - Long wires cause clock skew
- Clock is generally sent with/determined by data

Basic Serial Transmission

- Serial Peripheral Interface (SPI)
  - 4-wire bidirectional bus with clock
  - https://www.circuitbasics.com/basics-of-the-spi-communication-protocol
- Universal Asynchronous Reciever/Transmitter (UART)
  - 2-wire w/o clock; start bit & baud rate
  - https://www.circuitbasics.com/basics-uart-communication/
- Inter-Integrated Circuit (I2C)
  - 2-wire with clock from current master
  - https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/

Verilog Bidirectional Links

- Explicitly use assignment to `z` to disconnect:

```
module tribuf(o, i, en);
parameter BITS=1;
output [BITS-1:0] o;
input [BITS-1:0] i;
input en;

assign o = (en ? i : {BITS{1'bz}});
endmodule
```

- External assignments will alter `o`'s value here

Routing

- Direct connection it's just direction
- 2-way-switch networks can use test & shift
  - Use (source ^ destination) & bitmask
  - Shift routing tag one position at each level
- Internet, Ethernet, etc. use table lookup
  - Domain Name System (DNS) → IP address
  - Route[destination] gives output port
  - Not in table? Send it everywhere but port request came in on… learn from response

Routing: Are We There Yet?

- Many schemes send requests to devices they are not trying to talk to (e.g., I2C, Ethernet)
- All devices receive address
- Is it mine?
  Yes: Ack, process request, respond
  No: keep my outputs disabled (e.g., z state)

A Simple (Blocking) Switch

- 2 top inputs, 2 bottom outputs… all bidirectional
- Wires for 1 connection contain:
  - Clock (from above; a single clock is allowed)
  - 8-bit address (always from above)
  - 32-bit data (to be written / as read)
  - Read (always from above)
  - Write (always from above)
  - Ready (always from below)
- Everything happens on `posedge` of clock

A Simple (Blocking) Switch

- Protocol for Write (from above):
  - When Ready: Address=?, Data=?, Write=1
  - Address=z, Data=z, Write = 0
- Protocol for Read (from above):
  - When Ready: Address=?, Read=1
  - Address=z, Read=0
  - When Ready: Data holds value from memory

To Test It

- Make two instances of processor from: http://aggregate.org/EE380/onebeq.html
  - Have a word cache each for instruction, data; nothing happens until all is ready, so best case is really several clocks per instruction
  - Fetch instruction, then do data access… both only as needed
  - Make $1 be processor number, 0 or 1
- Hook each to a top input of one switch

Memory Module(s)?

- Have one 256x32 memory module with 2 port connections acting on disjoint halves of memory
- One memory module can easily be initialized using output from AIK
  - Have code start at memory location 0
  - Have data start at memory location 128
- You'll be testing two different memory mappings
  - Port *p* is 8'bxxxxxxx*p*
  - Port *p* is 8'b*p*xxxxxxx

Your Project, Due Dec. 6

- You are all one team
  - All will not work on every part, but each must be "aware" of all portions of the project
  - Only one submission…
- What you will submit: a tar containing
  - Implementor's notes
  - Verilog implementation (*Hint: memory module is a simplified switch*)
  - AIK code for your test program(s)

Your Test Program(s)

- Open choice for you… except:
  - Both PE0 and PE1 start with PC=0, but at some point they take different paths based on contents of register $1
  - The code in PE0 and PE1 should access at least a few shared variables and some that are not shared for both loads and stores
- Implementor's notes should say a bit about which of the two address mappings worked best

Implementor's Notes Notes

- Say a bit about what needed to change in module processor
- Which of the address mappings worked best?
- How much did the processors slow down? Was it more or less than you expected?