Design of Digital Circuits and Systems, Quiz 2

Memory: ROM, RAM, and Register Files

Solution Outlines

Memory Size

4-bit to 8-bit extender circuit: 32×8

- Circuit has a 4-bit data input and a 1-bit sign (*i.e.*, extension type) input, so $2^5 = 32$ total combinations.
- Output width will match the output data width (extended to 8 bits).

Memory Addressing

Memory capacity of 2 Ki-byte = $2^1 \times 2^{10} \times 2^3 = 2^{14}$ bits.

(A) Each 8×64 RAM chip holds $2^3 \times 2^6 = 2^9$ bits, so we need $2^{14}/2^9 =$ 32 chips.

(B) Since each RAM chip has a depth of 8 words, each one requires $\lceil \log_2 8 \rceil = 3$ address bits. The least significant address bits are connected to the RAM chips because each chip holds neighboring (*i.e.*, consecutive addresses) words, so bits `[2:0]`.

Memory Implementation

(A) We present two possibilities here; others exist. Needed to allow for both ports to write, assuming not a write conflict (*i.e.*, the writing statements are independent, not tied together with an else-if).

a. Add a specific check for a write conflict, and follow that with the normal writing logic code:

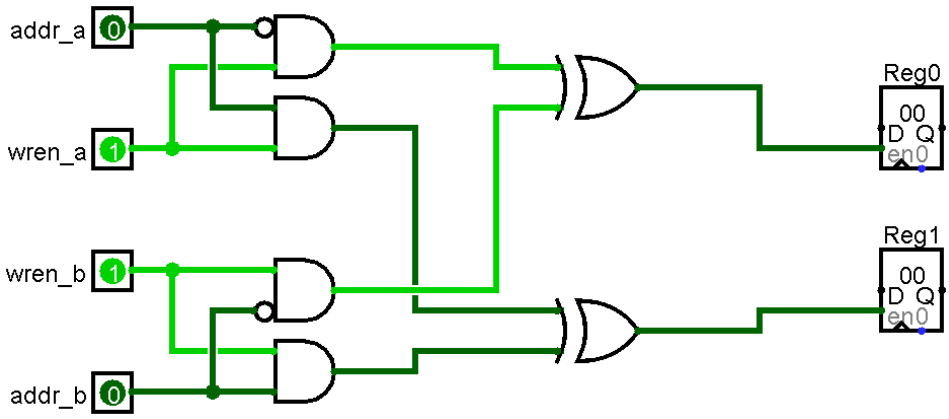
```
always_ff @(posedge clk) begin
  if ( ~(wren_a & wren_b & (addr_a == addr_b)) ) begin
    if (wren_a)
      RAM[addr_a] <= din_a;
    if (wren_b)
      RAM[addr_b] <= din_b;
  end
end
```

b. Use DeMorgan's Law to augment the base conditionals: `~(wren_a & wren_b & (addr_a == addr_b)) = ~wren_a | ~wren_b | (addr_a != addr_b)`

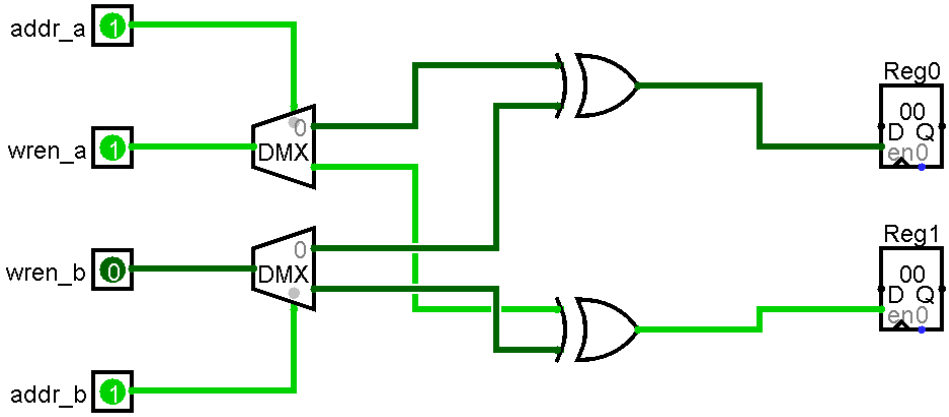
```
always_ff @(posedge clk) begin
  if (wren_a & (~wren_b | (addr_a != addr_b)))
      RAM[addr_a] <= din_a;
  if (wren_b & (~wren_a | (addr_a != addr_b)))
      RAM[addr_b] <= din_b;
end
```

(B) The logic here boils down to: a register should be enabled if one and only one port is writing to it. So for Reg0, this is `(wren_a & ~addr_a) ^ (wren_b & ~addr_b)`. Equivalently, this is `(wren_a & addr_a) ^ (wren_b & addr_b)` for Reg1.

a. Gates drawn out explicitly:



b. Taking advantage of the fact that the four parentheticals are the exact outputs of demultiplexers with the write enable signals as inputs and the address bits as selectors:



c. Could also have built out the write conflict logic to AND out the enable signals (either at the enable inputs or at the demultiplexor inputs):