计算机组成原理 · 实验4报告

计01　容逸朗　2020010869

实验过程

状态机

状态机的设计在实验文档中已经给出，按照思路容易写出下面的代码：

```
// reset and state change
always_ff @(posedge clk_i or posedge rst_i) begin
  if (rst_i) begin
    state <= ST_IDLE;
  end else begin
    state <= state_n;
  end
end

// states
always_comb begin
  state_n = state;
  case (state)
    ST_IDLE: begin
      if (wb_stb_i && wb_cyc_i) begin
        if (wb_we_i) begin
          state_n = ST_WRITE;
        end else begin
          state_n = ST_READ;
        end
      end
    end

    ST_READ: begin
      state_n = ST_READ_2;
    end

    ST_READ_2: begin
      state_n = ST_DONE;
    end

    // 其他省略...

    default: begin
      state_n = ST_IDLE;
    end
```

```
  endcase
end
```

信号处理

由于状态较为简单，因此选用组合逻辑实现：

```
// signals
always_comb begin
  sram_ce_n = (state == ST_IDLE) || (state == ST_DONE);
  sram_oe_n = !((state == ST_READ) || (state == ST_READ_2));
  sram_we_n = !(state == ST_WRITE_2);
  wb_ack_o = (state == ST_DONE);
  sram_addr = wb_adr_i[21: 2];
  sram_be_n = (state == ST_WRITE || state == ST_WRITE_2 || state == ST_WRITE_3) ? ~wb_sel_i : '0;
end

// data processing
assign sram_data = sram_we_n ? 32'bz : wb_dat_i;
assign wb_dat_o = sram_data;
```

唯一需要注意的是 SRAM 的单位是 4 字节而非总线的 1 字节。

思考题

> 1. 静态存储器的读和写各有什么特点？

- SRAM 的读和写过程都是通过读写信号控制，且读写共用一条数据线。
- SRAM 的数据是以行为单位存储的，每一行有 32 列（视位数而定，此处为 32 位），因此需要额外的时间来找到地址对应的行才能进行下一步的读 / 写操作。

> 2. 什么是 RAM 芯片输出的高阻态？它的作用是什么？

- 高阻态是电路的一种输出状态，既不是高电平也不是低电平，类似于引脚悬空的情况。
- RAM 芯片输出高阻态是因为 RAM 芯片为了节省引脚数量，导致了同一个信号在不同的时间传输不同方向的数据的现象。此时为了防止两端设备同时输出造成数据不稳定的情况，因此设备在不输出信号时需要设置高阻态。

> 3. 本实验完成的是将 BaseRAM 和 ExtRAM 作为独立的存储器单独进行访问的功能。如果希望将 Base_RAM 和 Ext_RAM 作为一个统一的 64 位数据的存储器进行访问，该如何进行？

- 由于 Base_RAM 和 Ext_RAM 都是 32 位的，如果要作为一个统一的 64 位数据的存储器访问，可以进行如下操作：
  - 地址缩小为 0x0 - 0x3FFFFF，且每个地址都由 Base_RAM 和 Ext_RAM 共用；
  - 数据的高 32 位存入 Base_RAM 对应地址行，低 32 位存入 Ext_RAM 的同一行；
  - 更改 1M2S 复用器接口，在此处对数据进行分割 / 合并操作。每次操作时，除了数据线以外需要给 Base_RAM 和 Ext_RAM 相同的信号，这样才能正确进行读 / 写操作，当两者的操作均完成后才返回 `ack_o = 1`。

实验总结

本次实验中，我只提交了一次评测就通过本次实验了:-)