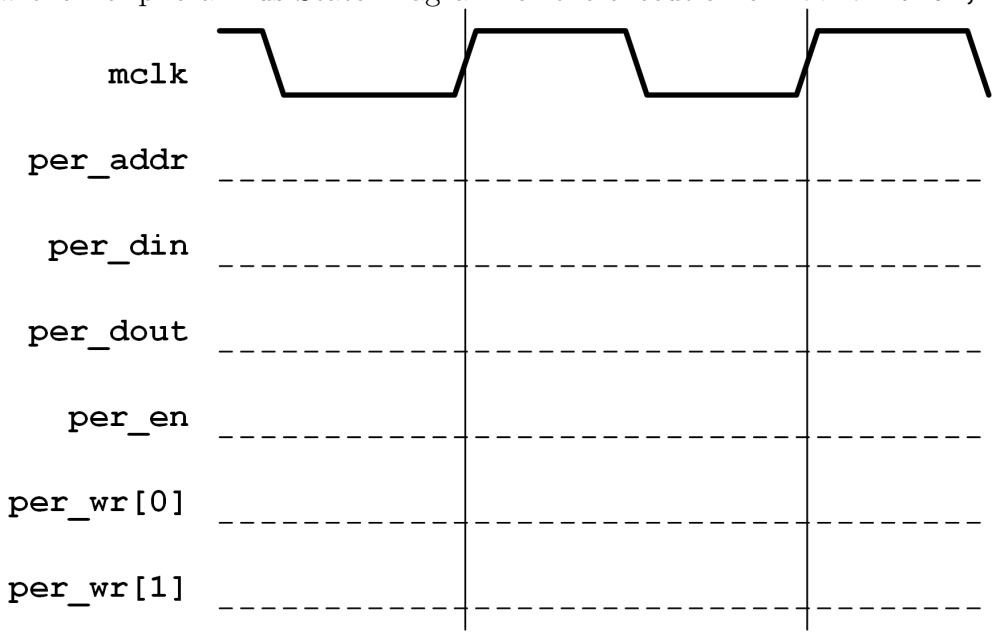
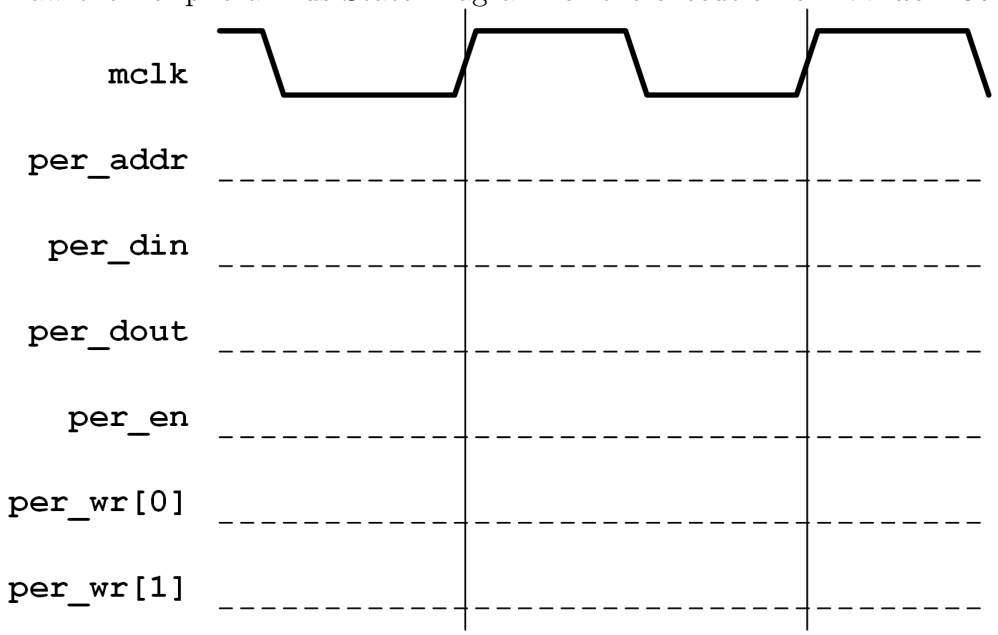
ECE4530 Fall 2019: Handout Lecture 6

PER Bus Timing

Draw the Peripheral Bus State Diagram for the execution of `mov.b #0xCD, &0xA1`



Draw the Peripheral Bus State Diagram for the execution of `mov &0x130, R5`



8-bit Memory-mapped Write-only Register at 0x80

```
module  my_hardware_register8 (output      [15:0] per_dout,
                               input              mclk,
                               input       [13:0] per_addr,
                               input              per_din,
                               input              per_en,
                               input        [1:0] per_we,
                               input              puc_rst  );
  reg         [7:0]  my_reg;
  wire        [7:0]  my_regnext;
  wire               valid_write;
  always @(posedge mclk or posedge puc_rst)
    my_reg <= (puc_rst) ? 8'b0 : my_regnext;

  my_regnext = valid_write ? per_din[__:__] : myreg;

  assign valid_write = ____________________________________________________;

  assign per_dout    = 8'h0;
endmodule
```

16-bit Memory-mapped R/W Register at 0x140

```
module  my_hardware_register16 (output      [15:0] per_dout,
                                input              mclk,
                                input       [13:0] per_addr,
                                input              per_din,
                                input              per_en,
                                input        [1:0] per_we,
                                input              puc_rst  );
  reg         [15:0]  my_reg;
  wire        [15:0]  my_regnext;
  wire                valid_write;
  always @(posedge mclk or posedge puc_rst)
     my_reg <= (puc_rst) ? 8'b0 : my_regnext;

  my_regnext = valid_write ? per_din[__:__] : myreg;

  assign valid_write = ____________________________________________________;

  assign per_dout    = ____________________________________________________;
endmodule
```