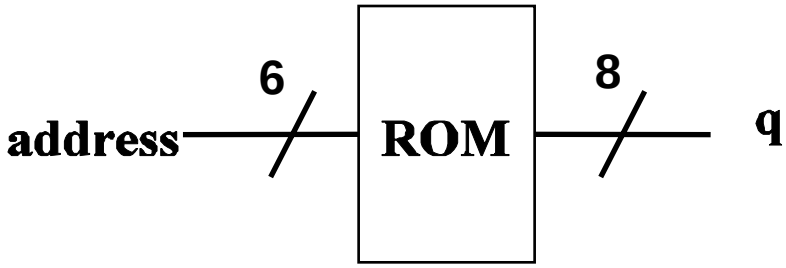
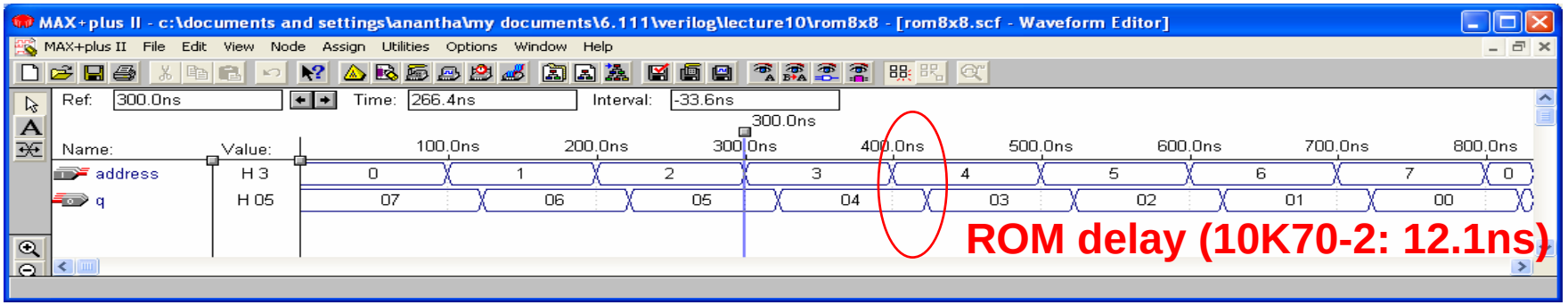
rom64x8.v (generated automatically)

```
//=============================================
// File Name: rom64x8.v
// Megafunction Name(s):
//                                    lpm_rom
//=============================================
module rom8x8 (
        address,
        q);
        input       [5:0]  address;
        output      [7:0]  q;

        wire [7:0] sub_wire0;
        wire [7:0] q = sub_wire0[7:0];

        lpm_rom     lpm_rom_component (
                                    .address (address),
                                    .q (sub_wire0));
        defparam
                    lpm_rom_component.lpm_width = 8,
                    lpm_rom_component.lpm_widthad = 6,
                    lpm_rom_component.lpm_address_control = "UNREGISTERED",
                    lpm_rom_component.lpm_outdata = "UNREGISTERED",
                    lpm_rom_component.lpm_file = “impluses.mif";
endmodule
```

Example: 64 deep by 8 bits wide

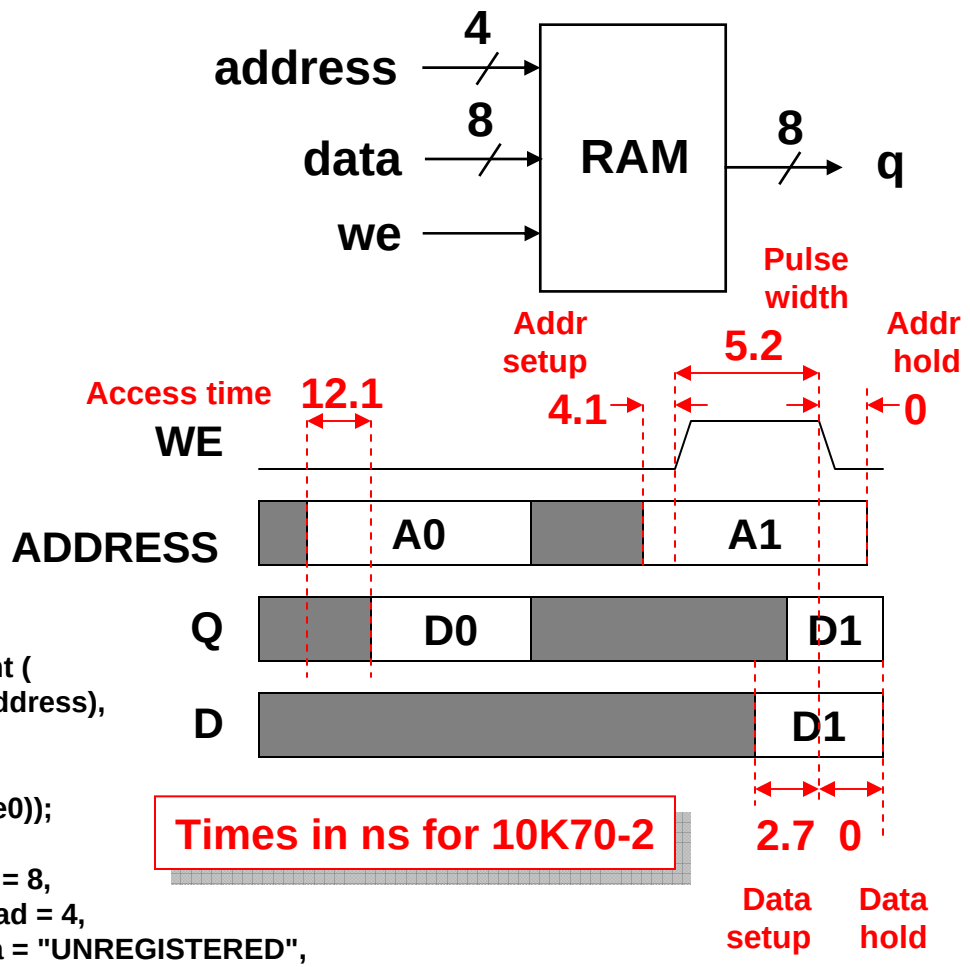


Path to location of Rom data



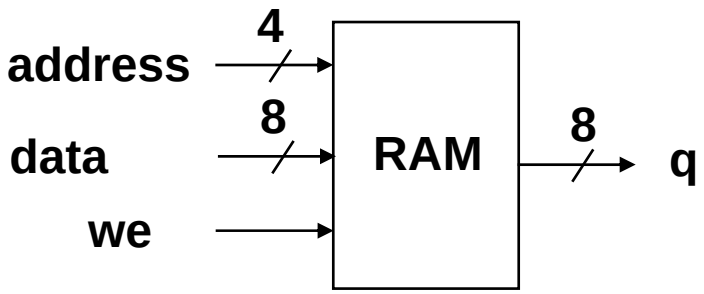
ram16x8.v

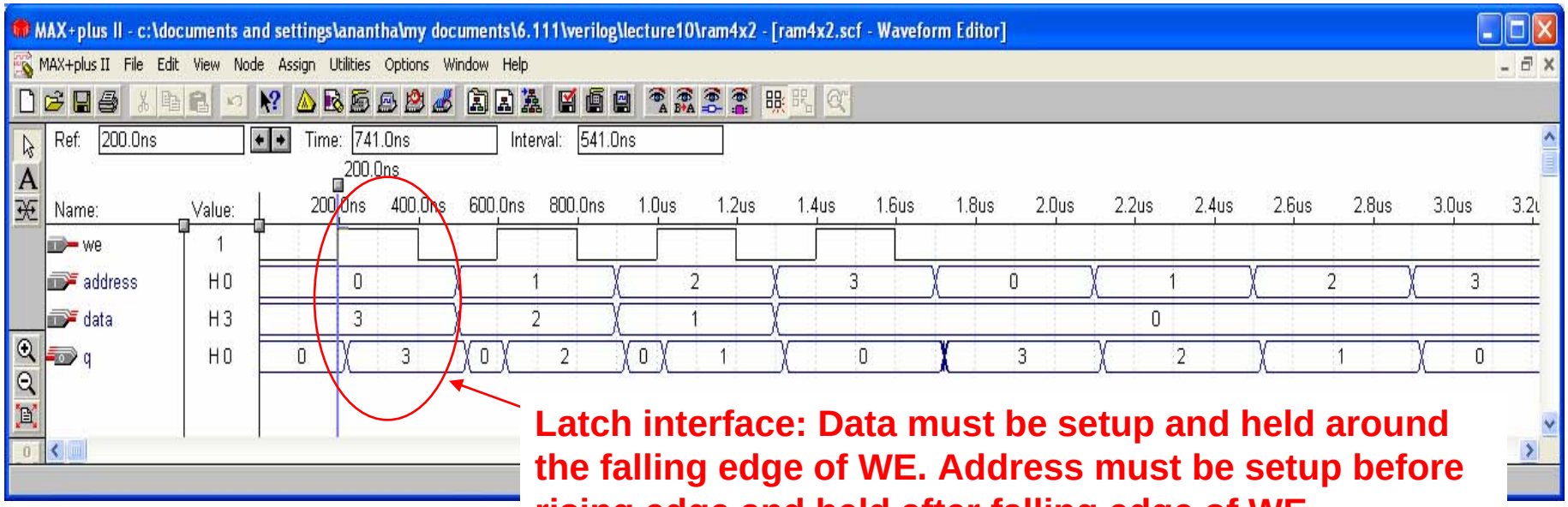
```
// megafunction wizard: %LPM_RAM_DQ%
module ram4x2 (
        address,
        we,
        data,
        q);

        input   [3:0]  address;
        input     we;
        input   [7:0]  data;
        output  [7:0]  q;

        wire [7:0] sub_wire0;
        wire [7:0] q = sub_wire0[7:0];

        lpm_ram_dq      lpm_ram_dq_component (
                                .address (address),
                                .data (data),
                                .we (we),
                                .q (sub_wire0));

        defparam
                lpm_ram_dq_component.lpm_width = 8,
                lpm_ram_dq_component.lpm_widthad = 4,
                lpm_ram_dq_component.lpm_indata = "UNREGISTERED",
                lpm_ram_dq_component.lpm_address_control = "UNREGISTERED",
                lpm_ram_dq_component.lpm_outdata = "UNREGISTERED",
                lpm_ram_dq_component.lpm_hint = "USE_EAB=ON";

endmodule
```



Asynchronous RAM Simulation

```
module ram16x8 (
            address,
            we,
            data,
            q);

            input       [3:0]  address;
            input        we;
            input       [7:0]  data;
            output      [7:0]  q;

endmodule
```

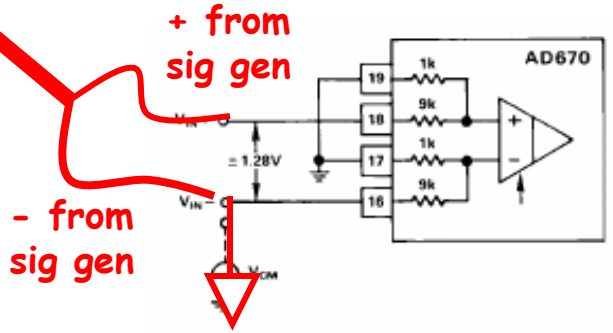




Latch interface: Data must be setup and held around the falling edge of WE. Address must be setup before rising edge and held after falling edge of WE.

Expected results

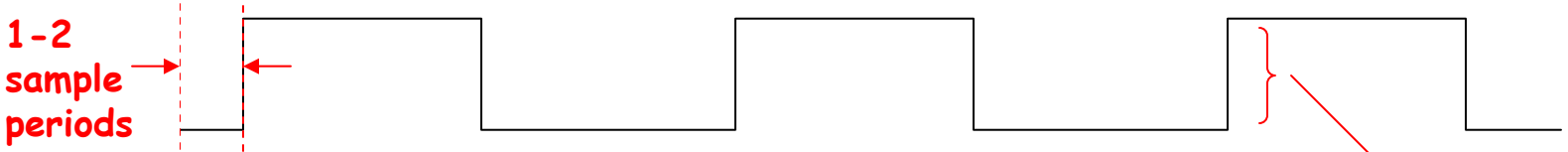
Sig. gen. output adjusted for 2.5V peak-to-peak (try -20db switch!)



3a. ±1.28 V Range

Test input: ~600 Hz square wave (20Khz sample rate)

All-pass (filter #0): coeffs = {0x7F,0,0,0,0,…,0}



Box car (filter #2): coeffs = {8,8,8,8,...,8}