

ابتدا یک کامپیوتر تعریف می کنیم و در آن یک حافظه و یک پردازنده قرار می دهیم.

```
begin
memory: ram port map(input_data => bdata, address => badress, read => boe,
                      write => bwe, reset => rrst, output_data => bdata); -- initialize memory

cpu: CPU8BIT2 port map(data => bdata, adress => badress, oe => boe, we => bwe, rst => rst, clk => clk);
```

```
32      elsif rising_edge(clk) then
33
34          -- PC / Address path
35          If (states = "000") then
36              pc <= adreg + 1;
37              adreg <= data(4 downto 0);
38          elsif (states = "100") then
39              adreg <= pc - 1;
40          else
41              adreg <= pc;
42          end If;
43
```

در خط ۳۸ یک elsif اضافه میکنیم تا هنگامی که دستور حالت داریم جولو تر از جایی که هستیم نرویم.

```
-- State machine
If (states /= "000") then states <= "000"; -- fetch next opcode
elsif (data(7 downto 5) = "111" and akku(8)='1') then states <= "101"; -- branch not taken
elsif (data(7 downto 5) = "010") then states <= "010"; akku <= "000000000"; -- load or add data with zero
elsif (data(7 downto 5) = "011") then states <= "100"; -- halt
else states <= not data(7 downto 5); -- execute instruction i have set opcode in a way that negation of them correspond to the proper state
end If;
end If;
end Process;
```

برای load و halt در استیت ماشین حالاتی را تولید میکنیم. برای Load اکومولاتور را خالی میکنیم تا دیتای جدیدی که می آوریم با صفر جمع شود و در واقع خودش را در اکومولاتور قرار میدهیم. برای حالت هم فقط کافی است که به استیت ۱۰۰ برویم تا همانطور که در عکس قبل دیدید جولو تر از جایی که هستیم نرویم.

```

Case states is
  when "010" => akku <= ("0" & akku(7 downto 0)) + ("0" & data); -- add | load
  when "011" => akku(7 downto 0) <= akku(7 downto 0) nand data; -- nand
  when "101" => akku(8) <= '0'; -- branch not taken, clear carry
  when "111" => akku <= std_logic_vector(shift_left(unsigned(akku), to_integer(unsigned(adreg(4 downto 0))))); -- shift left
  when "110" => akku <= std_logic_vector(shift_right(unsigned(akku), to_integer(unsigned(adreg(4 downto 0))))); -- shift right
  when others => null; -- instr. fetch, jcc taken (000), sta (001)
end Case;

```

۱۱۰ و ۱۱۱ را به ترتیب برای شیفت چپ و شیفت راست استفاده کرده ایم و در دستور nor با nand جایگزین کرده.