

# پروژه state machine درس طراحی سیستم های دیجیتال

ارایه دهندگان:

محمدحسین حسینی

علی سلطانی

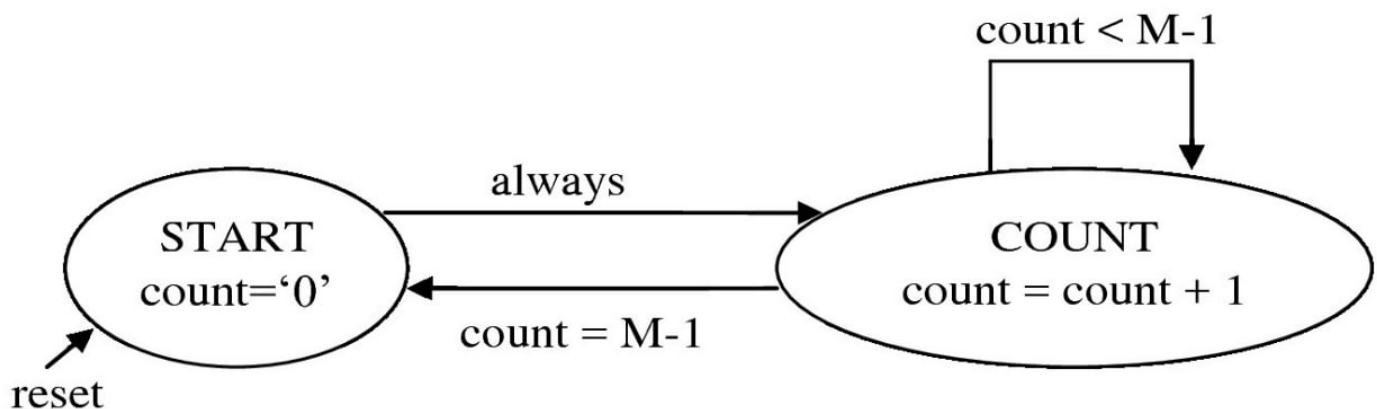
عارف علیجان نژاد

مهدی حسینی زاده

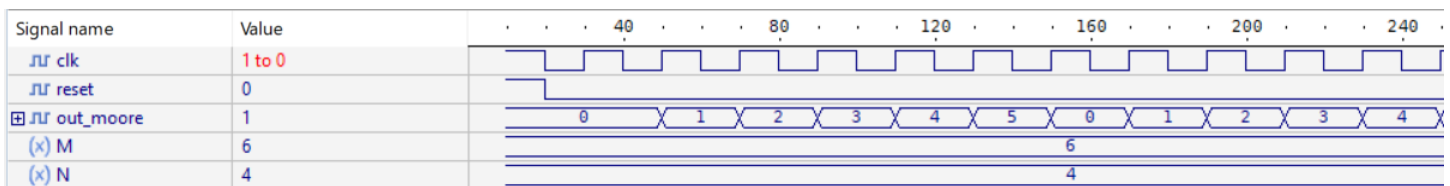
موضوع پروژه:

## Recursive Machine : Mod-m counter

عکس state machine:



عکس Waveform:



```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity counterEx is
  generic (
    M : natural := 6;
    N : natural := 4 -- N bits are required for M
  );
  port (
    clk, reset : in std_logic;
    out_moore : out std_logic_vector(N-1 downto 0)
  );
end entity;

architecture arch of counterEx is
  type stateType_moore is (start_moore, count_moore); -- Define an enumeration type for Moore state
  signal state_moore_reg, state_moore_next : stateType_moore; -- Registers to hold current and next Moore state
  signal count_moore_reg, count_moore_next : unsigned(N-1 downto 0); -- Registers to hold current and next count value
begin
  process (clk, reset)
  begin
    if reset = '1' then -- Asynchronous reset condition
      state_moore_reg <= start_moore; -- Reset the state to start_moore
      count_moore_reg <= (others => '0'); -- Reset the count to all zeros
    elsif rising_edge(clk) then -- Synchronous behavior on rising edge of the clock
      state_moore_reg <= state_moore_next; -- Update the current state with the next state
      count_moore_reg <= count_moore_next; -- Update the current count with the next count
    end if;
  end process;

  process (count_moore_reg, state_moore_reg)
  begin
    case state_moore_reg is
      when start_moore =>
        count_moore_next <= (others => '0'); -- Reset the count to all zeros
        state_moore_next <= count_moore; -- Transition to count_moore state
      when count_moore =>
        count_moore_next <= count_moore_reg + 1; -- Increment the count by 1
        if (count_moore_reg + 1) = M - 1 then -- Check if the count reached (M - 1)
          state_moore_next <= start_moore; -- Transition back to start_moore state
        else
          state_moore_next <= count_moore; -- Stay in the count_moore state
        end if;
      end case;
    end process;
  end architecture arch;

```

```

        end if;
    end case;
end process;

out_moore <= std_logic_vector(count_moore_reg); -- Convert the count to
std_logic_vector and assign it to the output
end arch;

```

این کد، یک ماشین شمارنده Recursive با نام "Mod-m counter" را پیاده‌سازی می‌کند. این شمارنده، به طور پیوسته از صفر شروع می‌کند و به تعدادی (M) دفعات به بالا شمارش می‌کند، سپس به صفر باز می‌گردد و این فرایند را تکرار می‌کند. کتابخانه‌های `ieee.std_logic_1164` و `ieee.numeric_std` در این کد استفاده شده‌اند. اولی برای استفاده از نوع داده `std_logic` و دومی برای استفاده از نوع داده `unsigned` و توابع مرتبط با آن.

پارامترهای `generic` که در تعریف `entity` وجود دارند، شامل M و N هستند. M بیانگر تعداد دفعاتی است که شمارنده به بالا می‌شمارد تا به مقدار (M - 1) برسد، و N تعداد بیت‌های مورد نیاز برای نمایش این مقدار است.

در بخش `architecture`، یک `enumeration type` به نام `stateType_moore` تعریف شده است که دو حالت "start\_moore" و "count\_moore" را دارا می‌باشد. سیگنال‌های `state_moore_reg` و `state_moore_next` برای نگهداری حالت فعلی و حالت بعدی ماشین شمارنده استفاده می‌شوند. همچنین، سیگنال‌های `count_moore_reg` و `count_moore_next` برای نگهداری مقدار فعلی و مقدار بعدی شمارنده به کار می‌روند.

در دو فرآیند (`process`) اصلی که در کد وجود دارند، رفتار سیگنال‌ها به صورت همزمان و ناهمزمان تعیین می‌شود. در فرآیند اول، سیگنال `reset` و `clk` بررسی می‌شوند. اگر سیگنال `reset` برابر با '۱' باشد، یعنی شرط ریست ناهمزمان برقرار است و در این صورت حالت شمارنده به "start\_moore" تنظیم می‌شود و شمارنده به صفر باز می‌گردد. اگر سیگنال `reset` برابر با '۰' باشد و در لبه بالارونده (`rising edge`) سیگنال `clk` باشیم، حالت و مقدار فعلی شمارنده با حالت و مقدار بعدی جایگزین می‌شوند. در فرآیند دوم، بر اساس حالت فعلی ماشین شمارنده، عملکرد مشخصی انجام می‌شود. در حالت "start\_moore"، مقدار شمارنده به صفر تنظیم می‌شود و حالت بعدی به "count\_moore" تغییر می‌کند. در حالت "count\_moore"، مقدار شمارنده یک واحد افزایش می‌یابد و سپس بررسی می‌شود که آیا مقدار شمارنده برابر با (M - 1) است یا خیر. اگر برابر باشد، حالت بعدی به "start\_moore" تغییر می‌کند و در غیر این صورت، در حالت "count\_moore" باقی می‌ماند.

در نهایت، مقدار فعلی شمارنده در قالب `std_logic_vector` تبدیل شده و به سیگنال `out_moore` اختصاص داده می‌شود. این سیگنال مقدار شمارنده را به صورت بیت‌ها نمایش می‌دهد.