

Hardware Design

Übungsblatt 2

6. Januar 2017

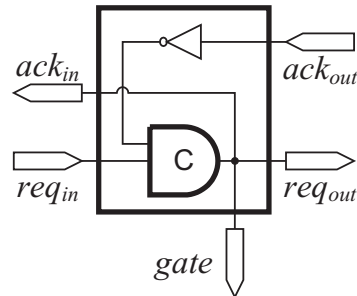
1. Das Muller-C Element ist ein wesentlicher Bestandteil asynchroner Schaltungen. Die Funktion dieser Zellen ist im folgenden Pseudocode dargestellt:

```
proc c_element(a, b : in bit; q : out bit) ≡
  if a = b
    q := a
  fi
end
```

Erstellen Sie eine Entität für das Muller-C Element wie folgt:

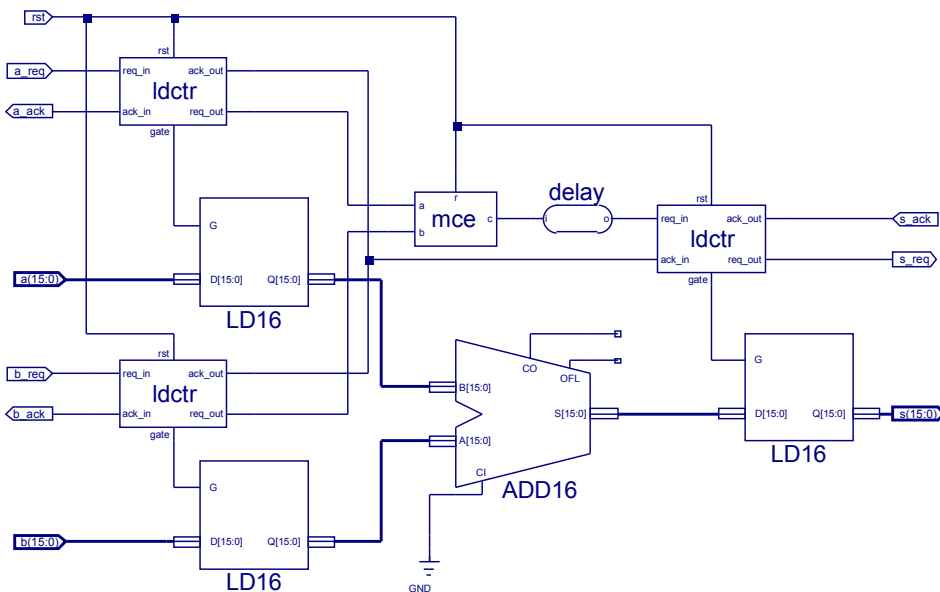
```
1 library IEEE;
  use IEEE.STD_LOGIC_1164.ALL;
  ---- Uncomment the following library declaration for the
  ---- the second implementation of the Muller-C element.
  --library UNISIM;
6 --use UNISIM.VComponents.all;
  entity mce is
    port (
      r, a, b : in std_logic;
      c       : out std_logic);
11 end mce;
  architecture Behavioral of mce is
    -- signal rst, set : std_logic;
  begin
    -----
16    -- Typically this works.
    -----
    c <= '0' after 1 ns when (a or b) = '0' or r = '1'
      else '1' after 1 ns when (a and b) = '1';
    -----
21    -- Use this implementation in case of FPGA design
    -----
    -- rst <= (a nor b) or r after 1 ns;
    -- set <= a and b after 1 ns;
    -- latch : ldcp port map (D => '0', G => '0',
26    --                       CLR => rst, PRE => set, Q => c);
  end Behavioral;
```

- Legen Sie nun eine Entität `ldctr` für einen Latch-Kontroller nach dem nachfolgenden Schema an:



Hinweis: Die Namen der Signale sind an den Port und nicht an deren Richtung angelehnt.

- Erstellen Sie nun ein Symbol für den Latch-Kontroller nach demselben Schema wie in der vorherigen Abbildung gezeigt. Markieren Sie dazu zunächst den Latch-Kontroller im Projektbrowser. Klicken Sie anschließend im Prozessbrowser unter **Design Utilities** auf **Create Schematic Symbol**. Öffnen Sie nun das erstellte Symbol über **File** → **Open...** Selektieren Sie im sich öffnenden Dateibrowser die Datei `ldctr.sym`. Daraufhin können Sie wie in einem Zeichenprogramm, das Symbol für die Entität anpassen.
- Erstellen Sie nun eine neue Entität `delay` mit einem Ein- und einem Ausgang. Bei der Implementierung verzögern Sie das Eingangssignal um 5 ns. (Hinweis: In einer echten Implementierung müssen die Verzögerungen über Puffer- bzw. Inverterketten realisiert werden.)
- Es soll nun ein asynchroner Addierer implementiert werden. Erstellen Sie dazu ein Schematic (**New Source...** → **Schematic**) und legen Sie den Addierer wie in der folgenden Abbildung gezeigt an.



- Simulieren Sie nun ihr Design mit der bereitgestellten Test Bench.