Jan Kaláb, xkalab00

Časový diagram simulace
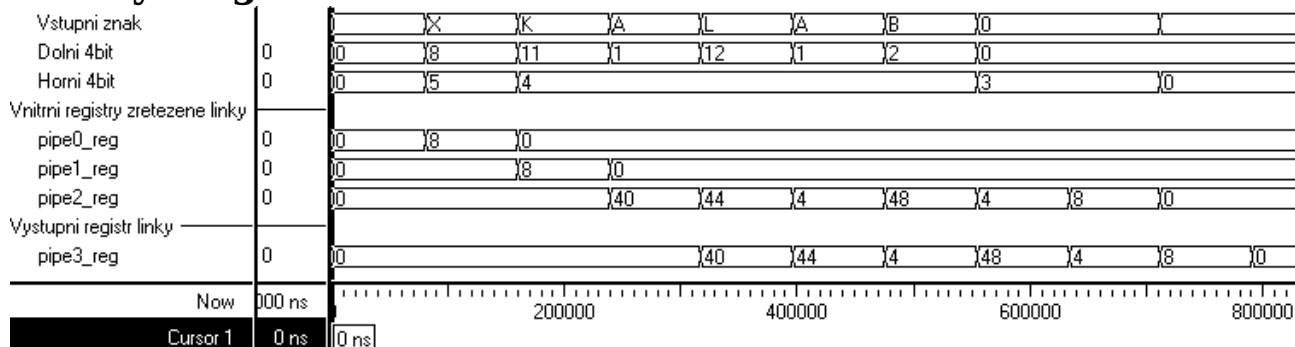
VHDL

```
--========================================================
-- Sem napiste kod, realizujici bloky, ktere jsou ve schematu v
-- zadani podbarveny zelenou barvou.
-- Registry mezi jednotlive faze zretezene linky jsou nadefinovany
-- pipe0_reg, pipe1_reg, pipe2_reg a pipe3_reg.
-- Pro nastaveni techto registru vyuzijte signaly pipe0, pipe1, pipe2 a pipe3.
-- Registry se dvema vstupnimi 4bitovymi operandy jsou ulozeny vzdy pro danou
-- fazi linky v registrech:
--    data0_a, data0_b
--    data1_a, data1_b
--    data2_a, data2_b
--    data3_a, data3_b

-- Prvni stupen
pipe0(0) <= (data0_a(0) and data0_b(0));
pipe0(1) <= (data0_a(1) and data0_b(0));
pipe0(2) <= (data0_a(2) and data0_b(0));
pipe0(3) <= (data0_a(3) and data0_b(0));
pipe0(7 downto 4) <= "0000"; -- doplneni 0

-- Druhy stupen
nula <= "000"; -- pomocny signal
mul1(0) <= (data1_a(0) and data1_b(1));
mul1(1) <= (data1_a(1) and data1_b(1));
mul1(2) <= (data1_a(2) and data1_b(1));
mul1(3) <= (data1_a(3) and data1_b(1));
SumC(mul1(0), pipe0_reg(0), nula(0), pipe1(1), c1(0));
SumC(mul1(1), pipe0_reg(1), c1(0), pipe1(2), c1(1));
SumC(mul1(2), pipe0_reg(2), c1(1), pipe1(3), c1(2));
SumC(mul1(3), pipe0_reg(3), c1(2), pipe1(4), pipe1(5));
pipe1(0) <= pipe0_reg(0); -- doplneni 0
pipe1(7 downto 6) <= pipe0_reg(7 downto 6);

-- Treti stupen
mul2(0) <= (data2_a(0) and data2_b(2));
mul2(1) <= (data2_a(1) and data2_b(2));
mul2(2) <= (data2_a(2) and data2_b(2));
mul2(3) <= (data2_a(3) and data2_b(2));
SumC(mul2(0), pipe1_reg(0), nula(1), pipe2(2), c2(0));
SumC(mul2(1), pipe1_reg(1), c2(0), pipe2(3), c2(1));
SumC(mul2(2), pipe1_reg(2), c2(1), pipe2(4), c2(2));
SumC(mul2(3), pipe1_reg(3), c2(2), pipe2(5), pipe1(6));
pipe2(1 downto 0) <= pipe1_reg(1 downto 0); -- doplneni 0
pipe2(7) <= pipe1_reg(7);
```

```vhdl
-- Ctvrty stupen
mul3(0) <= (data3_a(0) and data3_b(3));
mul3(1) <= (data3_a(1) and data3_b(3));
mul3(2) <= (data3_a(2) and data3_b(3));
mul3(3) <= (data3_a(3) and data3_b(3));
SumC(mul3(0), pipe2_reg(0), nula(2), pipe3(3), c2(0));
SumC(mul3(1), pipe2_reg(1), c3(0), pipe3(4), c2(1));
SumC(mul3(2), pipe2_reg(2), c3(1), pipe3(5), c2(2));
SumC(mul3(3), pipe2_reg(3), c3(2), pipe3(6), pipe1(7));
pipe2(2 downto 0) <= pipe1_reg(2 downto 0); -- doplneni 0

--=====================================================
```