Garáž

Zadání

Moorův automat, který bude představovat garáž pro auta. Tento automat bude mít dva stavy, volno, auta mohou přijíždět, nebo stop, kdy už bude parkoviště plné. Na sedmisegmentu bude vysvíceno, kolik aut se právě nachází na parkovišti, jedna led bude pro volno, druhá led bude pro plno.

Teoretický rozbor:

Moorův automat je stavový automat, který má ve stavu určitý výstup, na rozdíl tedy od automatu Mealyho, který posílá výstupní hodnotu při přechodu do jiného stavu, u Moorova automatu je výstup závislý opravdu jen na aktuálním stavu.

Stavy

Vstupní stava- p o

x0	0	0
x1	1	0
x2	0	1
x3	1	1

Výstupní stavy-

	L
y0	0

y1 1

Vnitřní stavy-q2 q1 q0

S0	0	0	0
S1	0	0	1
S2	0	1	0
S3	0	1	1
S4	1	0	0
S5	1	0	1

Vidíme aktivaci sedmisegmentu. Použijeme tlačítka na odjezd a příjezd. Samotný automat je realizovaný pomoci Moorova automatu, z nápovědy si vykopírujeme šalbonu pro moorův automat. Moorův automat obsahuje dva signály: state a next_state, state značí aktuální stav a next_state nadcházející stav, na základě aktuálního stavu state, se podle podmínek rozhodne, jestli state zůstane ve stavu ve kterém je, nebo se naopak přepne do stavu next_state. Podmínky vypadjí takto: Jsem ve stavu s1, auto přijelo. Přesuneme se tedy do stavu 2, auto přijelo. Stav bude tedy 3. Stejně to funguje i pro ostatní stavy. Pokud ale auto odjede, stav se sníži zpátky na 2. Pokud se automat dostane do stavu 5, rozsvíti se ledka která oznamuje že garáž je plná a jíž nemůžou žadné auta přijíždět dokud zase nějaké auto neodjede.

Moorův automat

```
constant S0: STD_LOGIC_VECTOR (2 downto 0) := "000";
```

Příklad konstatovaní stavu, konkretně stavu 0 kdy se v garáží nenachází žadné auto a na sedmisegmentu bude svítít 0 (viz. Aktivace sedmisegmentu).

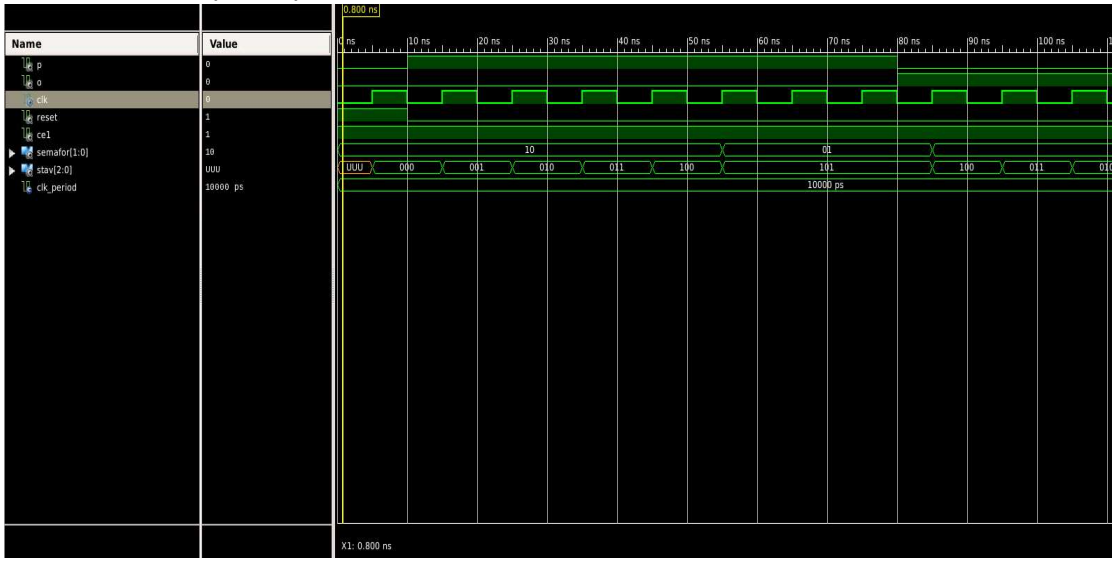
```
OUTPUT_DECODE: process (state)
begin
   if state = s5 then
     semafor <= "01";
```

```
    else
     semafor <= "10";
    end if;
  end process;
```

Proces který nám dáva podmínku že když je stav 5, rozsvíti se ledka stop, jinak svíti ledka volno.

```
when s1 =>
  if (p = '1' and o = '0') then
    next_state <= s2;
    elsif o = '1' and p = '0' then next_state <= s0;
   end if;
```

Úkázka toho že když se automat nachází ve stavu 1 a přijezd je rovné 1 a odjezd 0 tak se automat přepné do dalšího stavu 2, v opačnem případě se přepne do následujícího stavu 0. Signály state a next_state jsou typu std_logic_vektor, jsou 3 bitové, aby se pokryli všechny potřebné stavy „000“ - „101“(0 – 5).



Simulace

Na simulaci můžem vidět začínající stav 0 a postupně přijíždějici auta dokud stav není roven 5, potom se nám změní i stav semaforu který se z „10“(volno) přepne na „01“(plno). A potom naopak kdy začnou auta odjíždět, a hned při přepnuti do stavu 4 se semafor opět přepne do stavu „10“.

Zhodnocení

Garáž fungovala bez chyby v simulaci . Realizace by byla skvěla na parkoviště ke kterému by šly dvě cesty, vjezd a výjezd. Na každé cestě by se nacházelý tlakové desky které by po přejetí autem vyslalý signál, který by měnil ukazatel počtu aut na parkovišti a který by oznamoval zda se zde ještě nějaké auto vleze.