

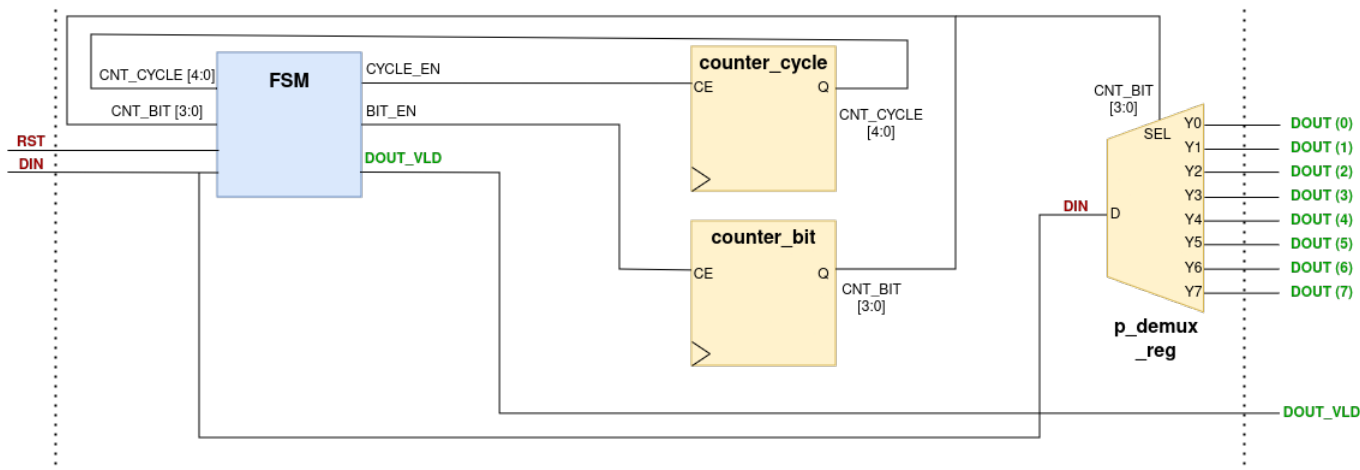
Výstupní zpráva

Jméno: Dmitrii Ivanushkin

Login: xivanu00

Architektura navrženého obvodu (na úrovni RTL)

Schéma obvodu



Popis funkce

Prvky UART obvodu jsou:

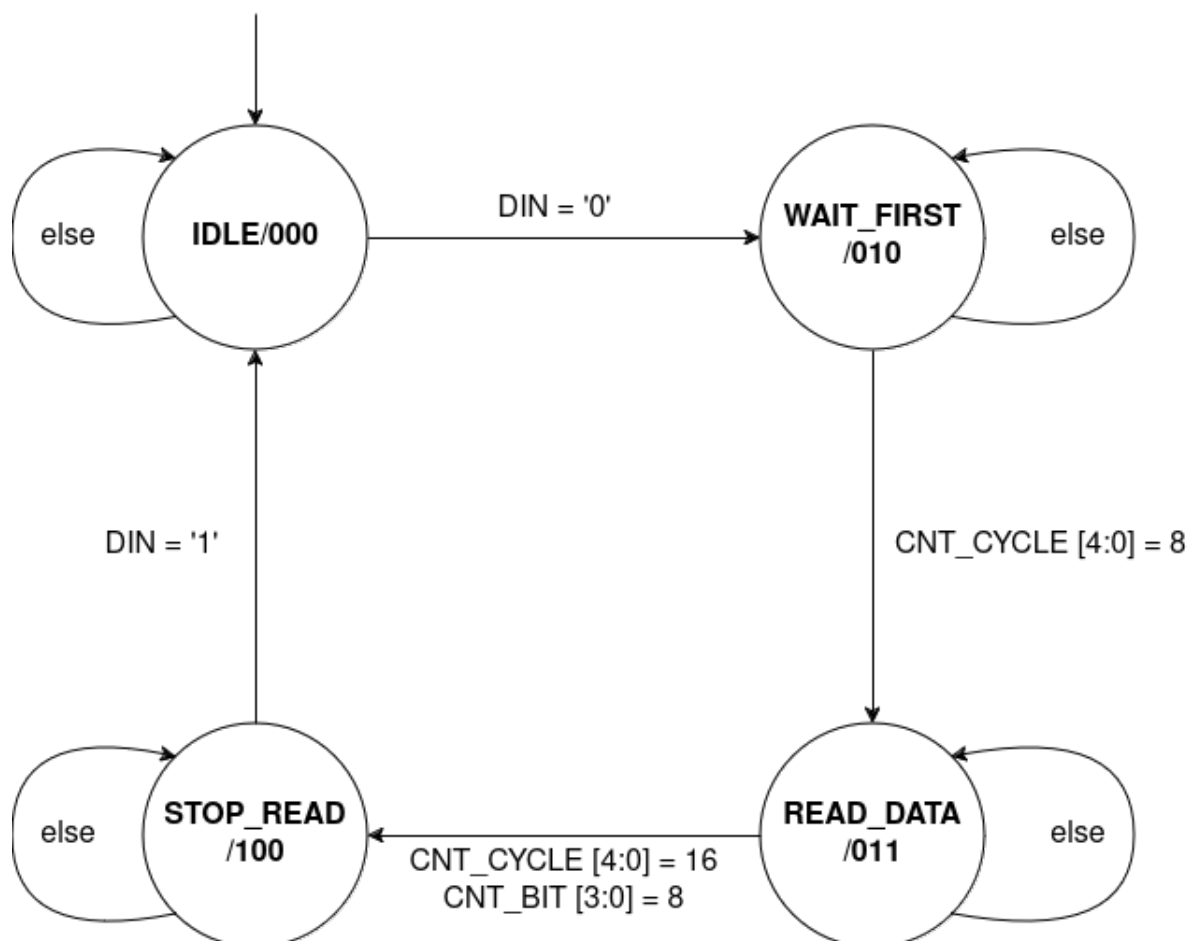
- FSM, který je detailně popsán níže.
- Čítač `counter_cycle` potřebujeme pro správné nalezení MID bitu a posílání hodnoty do FSM. Podmínky resetu jsou popsány níže, navíc resetujeme pokud není `CYCLE_EN`. Inkrementace povolí `CYCLE_EN` pokud:
 1. `CNT_CYCLE != 8` a zároveň **není** nastaven `BIT_EN`. Jinak resetujeme.
 - 1.1. Tato podmínka znamená, že pokud se nacházíme ve stavu `WAIT_FIRST` (poznáme pomocí `BIT_EN`) musíme počkat 8 cyklů, resetovat a můžeme inkrementovat `CNT_CYCLE` znovu.
 2. `CNT_CYCLE != 16` a zároveň **je** nastaven `BIT_EN`. Jinak resetujeme.
 - 2.1. Tato podmínka znamená, že pokud se nacházíme ve stavu `READ_DATA` (poznáme pomocí `BIT_EN`) musíme počkat 16 cyklů, resetovat a můžeme inkrementovat `CNT_CYCLE` znovu.
- Pomocí čítače `counter_bit` předáváme číslo načtených bitů do FSM a v DEMUX (jako selektor). Resetujeme pokud není `BIT_EN`. Inkrementaci povolí `BIT_EN` pokud `CNT_CYCLE == 16`
- Demultiplexor `p_demux_reg`, který na základě čísla bitu (`CNT_BIT`) přiřadí `DIN` do místa ve vektoru `DOUT`. Navíc moment zápisu `DIN` musí být přesně v 16 cyklu (`CNT_CYCLE`)

Návrh automatu (Finite State Machine)

Schéma automatu

Legenda:

- Stavy automatu: IDLE, WAIT_FISRT, READ_DATA, STOP_READ
- Vstupní signály: DIN, CNT_CYCLE [4:0], CNT_BIT [3:0]
- Moorovy výstupy: DOUT_VLD, CYCLE_EN, BIT_EN



Popis funkce

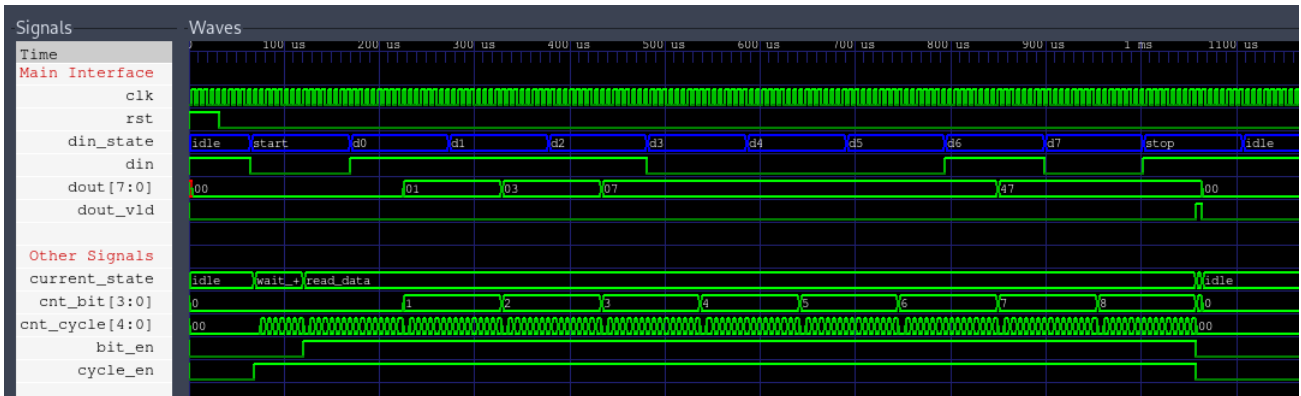
FSM je ve stavu čekání IDLE buď po RST nebo hned po ukončení STOP_READ. FSM čeká pokud DIN nebude 0, což znamená, že je to start a přecházíme do stavu WAIT_FIRST

WAIT_FIRST bude čekat na START bit, který můžeme načíst po 8 cyklů a přejít do stavu READ_DATA. Výstup CYCLE_EN je teď nastaven na 1.

Ve stavu READ_DATA, máme BIT_EN nastavený na 1. Následující skok do stavu STOP_READ nastane když druhý čítač CNT_BIT oznámí, že máme 8 načtených bitů a navíc CNT_CYCLE bude mít hodnotu 16 (potřebujeme pro správné načtení MID STOP bitu)

Ve stavu STOP_READ nastavíme DOUT_VLD na 1 a pokud DIN bude 1, vrátíme se do stavu IDLE

Snímek obrazovky ze simulací



Obrázek 1: Program gtkwave. Soubor wave.ct1 (První slovo)

```
~/D/u/INC - Digital Systems Design  main ± ./uart.sh
##### ANALYSIS #####

##### SYNTHESIS #####

##### SIMULATION #####
Sending data onto DIN with value: 0x47
Output data from DOUT with value: 0x47
Sending data onto DIN with value: 0x55
Output data from DOUT with value: 0x55
Sending data onto DIN with value: 0xAA
Output data from DOUT with value: 0xAA
Sending data onto DIN with value: 0xCA
Output data from DOUT with value: 0xCA
```

Obrázek 2: Simulace (skript uart.sh)