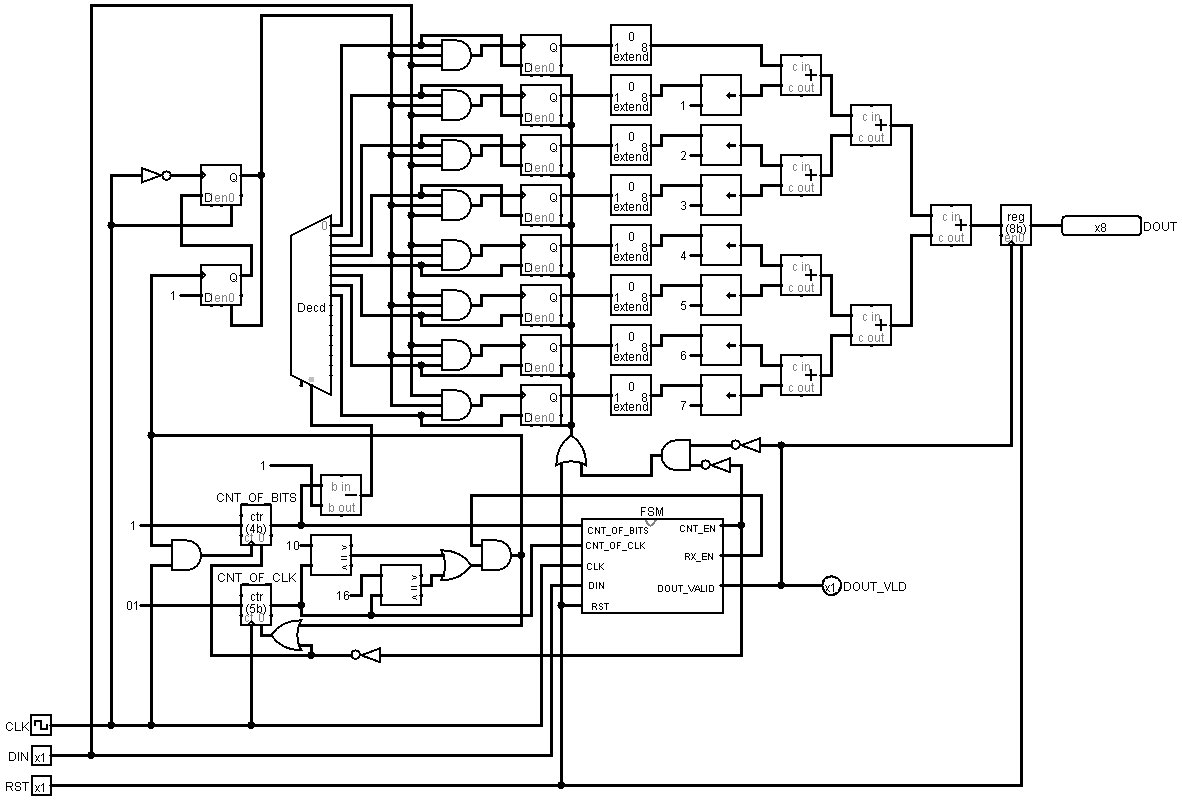
# Příloha: Výstupní zpráva

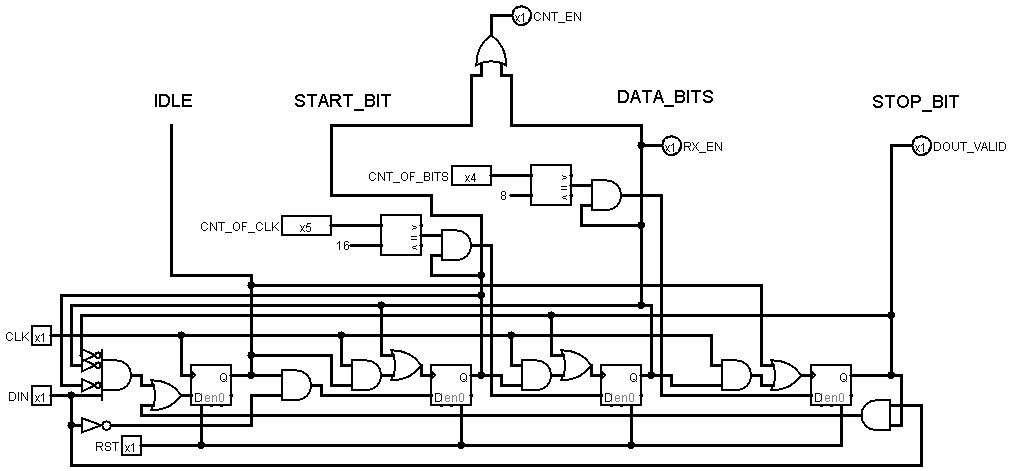
**Jméno:** Michal Blažek

**Login:** xblaze38

## Architektura navrženého obvodu (na úrovni RTL)

### Schéma obvodu

**Main:**

**FSM:**

### Popis funkce

FSM automat pouze přepíná mezi stavy IDLE, START\_BIT, DATA\_BITS a STOP\_BIT, přičemž nastavuje výstupy RX\_EN, CNT\_EN a DOUT\_VALID podle daného stavu. Pro přechod mezi stavy využívá různé podmínky (viz Schéma automatu) na základě jejich vyhodnocení se přepíná.

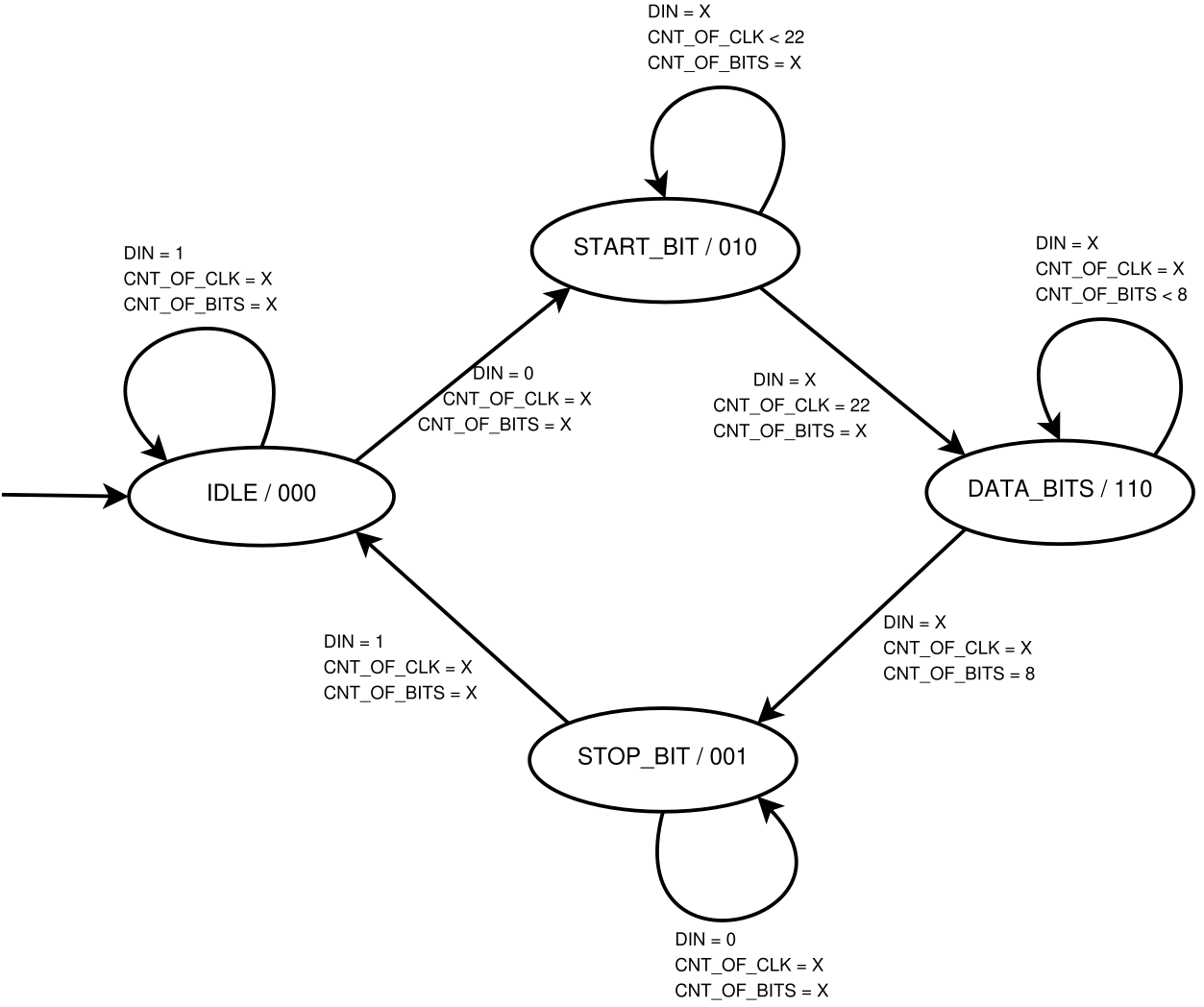
Hlavní obvod využívá FSM, dvou čítačů CNT\_OF\_CLK a CNT\_OF\_BITS a dekodéru, který vybírá určitý výstup. Podle výstupních bitů z FSM je obvod schopen rozpoznat stav FSM a určit který čítač bude aktivní. Čítač CNT\_OF\_CLK se pokaždé, když dosáhne požadované hodnoty, vynuluje a při stavu DATA\_BITS inkrementuje čítač CNT\_OF\_BITS, který se také při požadované hodnotě nuluje. Jakmile dojde ke změně v čítači CNT\_OF\_BITS během čtení dat, odešle čítač svoji informaci do dekodéru a z něj se na příslušný bit zapíše hodnota vstupu DIN. Tyto hodnoty se postupně ukládají do registru, který ve stavu STOP\_BIT, propustí svoje data, pokud je aktivní DOUT\_VLD, na výstup DOUT.

## Návrh automatu (Finite State Machine)

### Schéma automatu

Legenda:

* Stavy automatu: IDLE, START\_BIT, DATA\_BITS, STOP\_BIT
* Vstupní signály: DIN, CNT\_OF\_CLK, CNT\_OF\_BITS
* Moorovy výstupy: RX\_EN, CNT\_EN, DOUT\_VALID



### Popis funkce

Nejprve je automat ve stavu IDLE, kde pouze čeká, dokud se na vstupu DIN neobjeví 0. Díky tomuto je automat schopný poznat, že přichází informace, a přepne se do stavu START\_BIT. Jelikož máme přenosovou rychlost 9600 baudů a používáme technologii midbit, hodinový signál máme pro vzorkování 16x rychlejší. To znamená, že ve stavu START\_BIT se pomocí hodinového signálu odpočítá 22 period, kde 8 period je abychom se dostali přibližně doprostřed bitu a měli tak záruku kvalitního přečtení signálu a 16 period pro přesun na první datový bit. Dohromady to dává 24 period, jelikož při testování bylo zjištěno, že data v posledních 2 bitech lehce utíkají, zvolil jsem pro tuto implementaci pouze 22 period. Nyní se přejde do stavu DATA\_BITS, kde se postupně po 16 periodách čte každý bit, dokud jich není všech 8. Poté automat přechází do stavu STOP\_BIT a nastaví signál DOUT\_VALID pro určení platnosti dat. Nakonec automat, za pomoci signálu DIN nastaveného na 1, přechází zpět do stavu IDLE a proces se může opakovat.

## Snímky obrazovky ze simulací

