

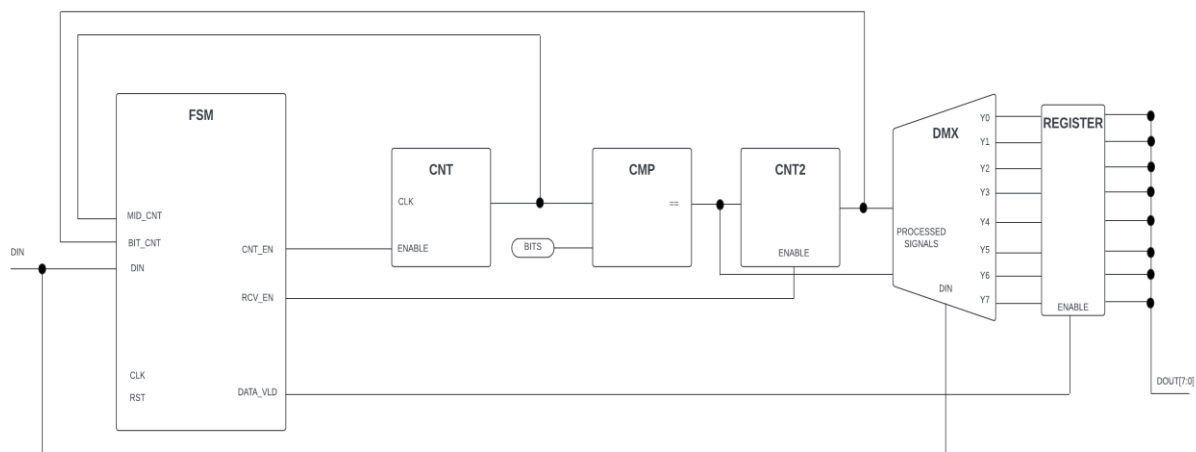
# Výstupní zpráva

Jméno: Tadeáš Novotný

Login: xnovott00

## Architektura navrženého obvodu (na úrovni RTL)

### Schéma obvodu



### Popis funkce.

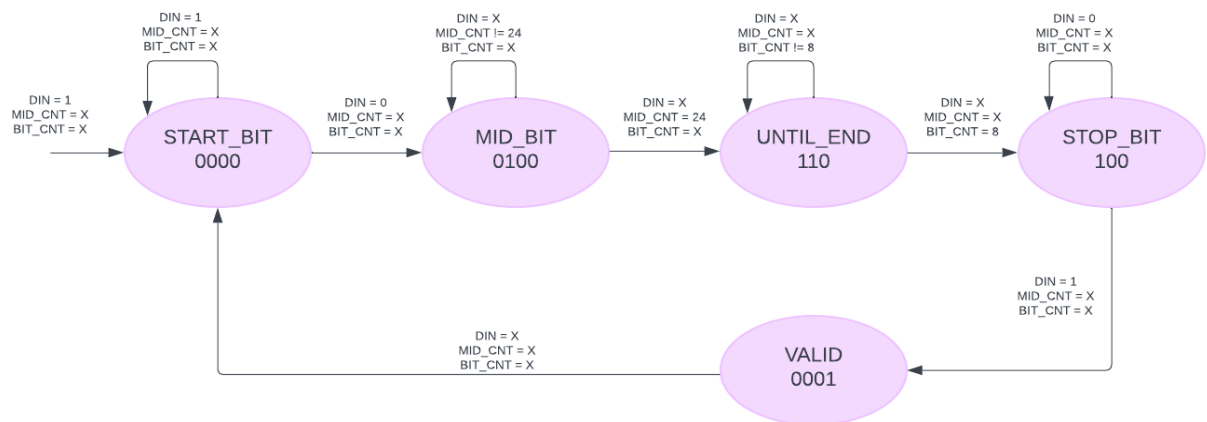
Moje RTL schéma je na bázi prvku FSM, dvou sčítačů CNT, CNT2, DMX (demultiplexor), REGISTER. Obvod by se dalo říci, že začíná, kdy FSM přijímá vstup DIN a díky němu odevzdá signal a reguluje sčítače a výstup. CNT náhle když **přijme** od FSM signál ENABLE – CNT\_EN. Výstupem je 5-bitová hodnota MID\_CNT, ta se porovná s 24, když je rovna, posílá signal, zda se nachází v midbitu. Signál s ENABLE z FSM je přesunut do CNT2 mého druhého sčítače, **kde** výstupem je 4 bitová hodnota, která určuje kolik se načetlo bitů. Dále DMX sesbírá spracované signály a **podle** signálu D\_OUT bity projdou registrem a přejd

# Návrh automatu (Finite State Machine)

## Schéma automatu

Legenda:

- – Stavy automatu: START\_BIT, MID\_BIT, END, STOP\_BIT, VALID
- – Vstupní signály: DIN, MID\_CNT, BIT\_CNT
- – Moorovy výstupy: MID\_CNT\_EN, DATA\_RCV\_EN, DATA\_VLD



## Popis funkce

Automat FSM probíhá v cyklu o 5ti stádiích. Prvním stavem funkce je **START\_BIT**, pokud tato funkce dostane na vstupu logickou 0 (DIN), tak přejde do stavu **MID\_BIT** a v něm bude 23 hodinových cyklů. Poté, když  $MID\_CNT = 24$ , tak se přejde do stavu **UNTIL\_END**. Ve stavu **UNTIL\_END** zůstává dokud počítadlo **BIT\_COUNT** nenapočítá 8. v pořadí a tím se přesouvá na konečný bit. **STOP\_BIT** stav **zkontroluje, zda** se na vstupu objevil signal DIN 1, poté se přejde do stavu **VALID**, kde se následně **potvrdí** validita přijatého signálu. Signál je přijat a vypsán, cyklus se může zopakovat.

## Snímek obrazovky ze simulací

