

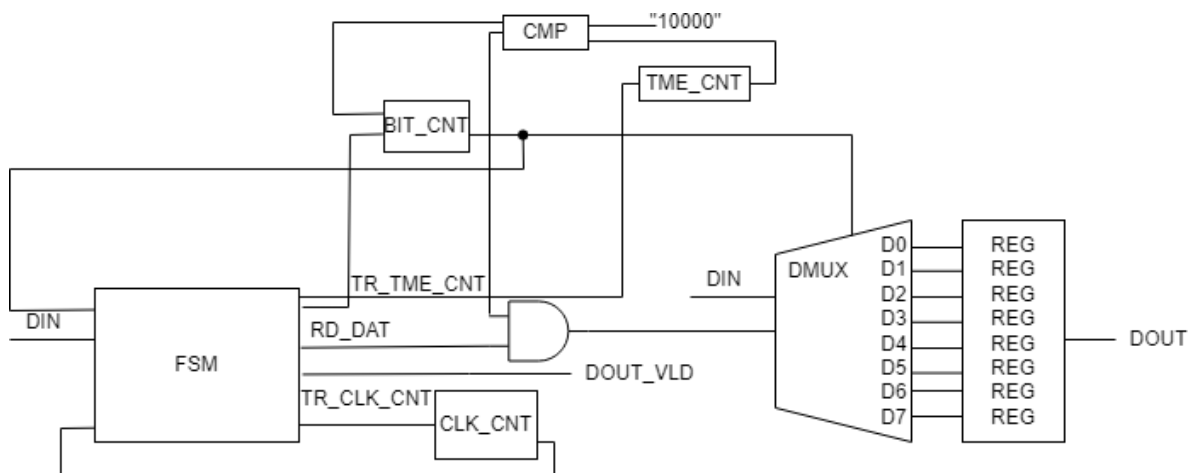
# Výstupní zpráva

Jméno: Samuel Lencsés

Login: xlencs00

## Architektura navrženého obvodu na úrovni RTL

### Schéma obvodu



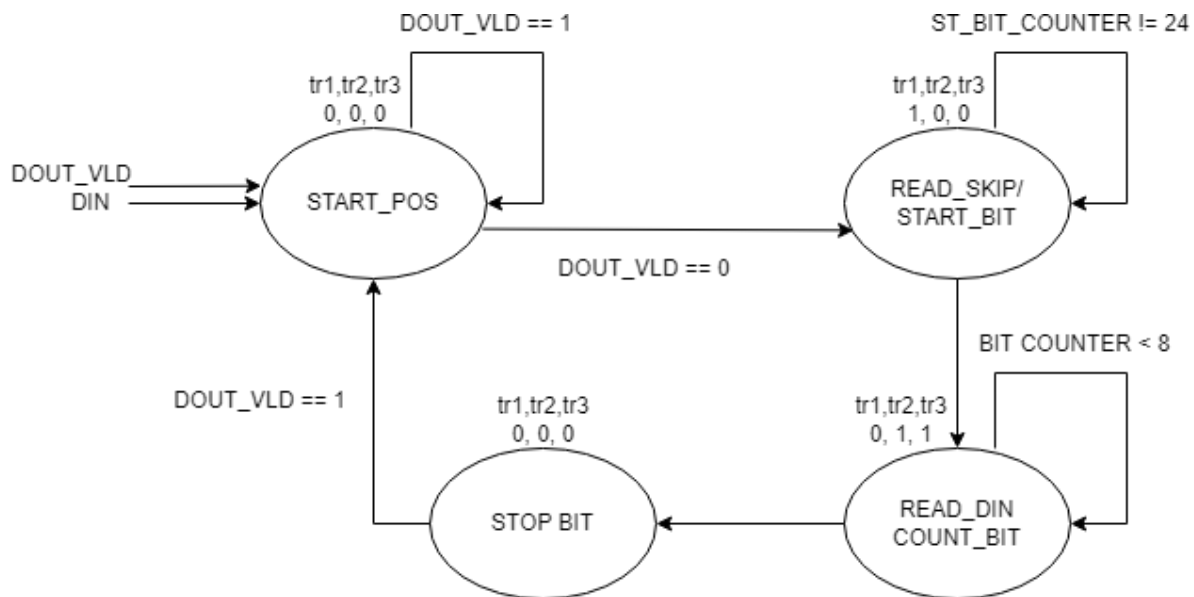
### Stručný popis

Daný obvod načítava zo vstupu **DIN** a ukladá do **DOUT** pomocou demultiplexoru **DMUX** a registrov. V prvej fáze sa spustí **CLK\_CNT** pomocou **TR\_CLK\_CNT** (trigger), ktorý napočíta 24 hodinových cyklov a tým preskočí start bit a dostane sa stredu prvého bitu slova, ktoré chceme uložiť. V druhej fáze sa spustí **BIT\_CNT**, ktorý indexuje a pamätá si pozíciu bitu slova, ktoré uložíme potom do registrov poporadí.

Ďalší bit získame pomocou **TME\_CNT**, ktorý napočíta do 16 (tým sa dostaneme do stredu nasledujúceho bitu) a proces sa zopakuje. Do registru zapisujeme iba vtedy, ak je **RD\_DAT** (povolenie na zápis) a **CMP** nastavené na 1 (AND hradlo). Ak sme dokončili proces, nastavíme hodnotu **DOUT\_VLD** na 1 a tým ukončíme dané slovo.

## Návrh automatu (FSM)

### Schéma automatu



**Legenda:**

- tr1 = tr\_clk\_cnt (počiatočné počítadlo hodinových cyklov)
- tr2 = bit\_cnt (počíta počet bitov v slove[index])
- tr3 = tme\_cnt (počítadlo času [16 hodinových cyklov])

### Implementácia:

**START\_POS = START POS**

**READ\_SKIP = READ\_DIN**

**STOP\_BIT = STOP\_BIT**

### Stručný popis

Na začiatočnú pozíciu **START\_POS** prídu údaje a čaká sa, kým hodnota **DOUT\_VLD** nebude 0 (indikuje začiatok čítania slova). Ak sa splní daná podmienka, prejde do pozície 2 v ktorej je počítadlo **ST\_CLK\_CNT** ktoré preskočí prvých 16 hodinových cyklov (**START BIT**) a nastaví ho do stredu prvého bitu ktorý čítame, čiže 8. hodinový cyklus bitu ktorý čítame (**MID BIT**)[16+8=24]. V ďalšej pozícii čítame dané slovo, ktoré má 8 bitov(prechádzame ho pomocou cyklu) a uložíme ho. V poslednej fáze vyresetujeme **DOUT\_VLD** na 1 čím indikujeme **STOP BIT** a ukončenie daného slova.

## Snímek obrazovky ze simulací

