



APRIL 16, 2023

# UART RECEIVER

INC PROJECT

KHODAREVSKYI DMYTRO, XKHODA01

VUT FIT



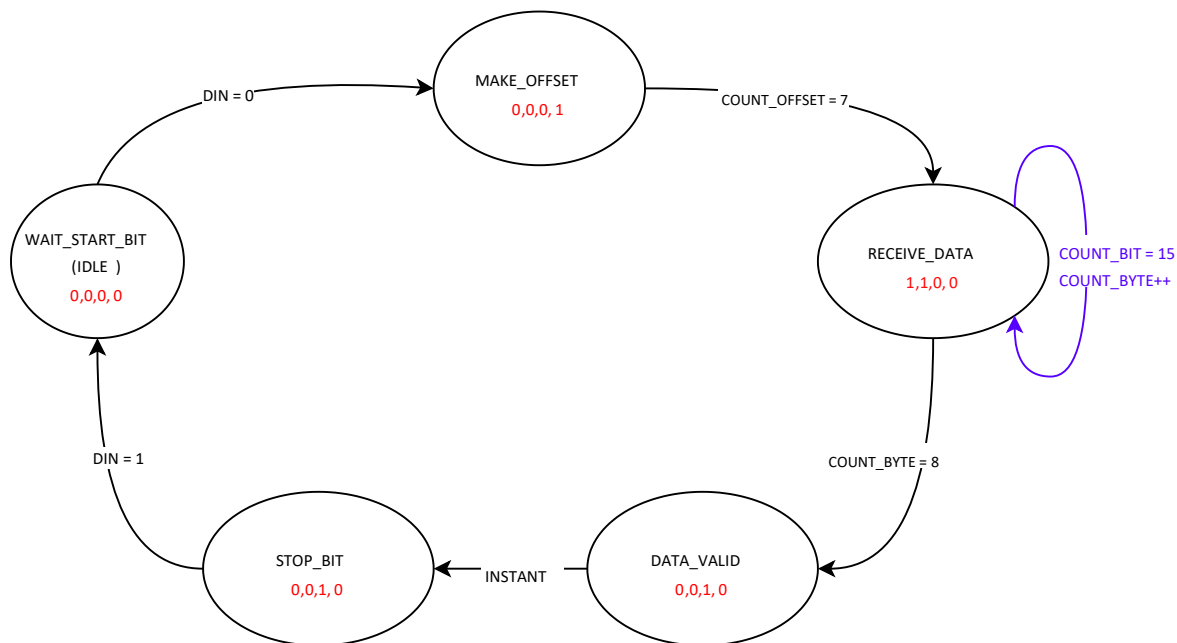
# Obsah

<b>Návrh .....</b>	<b>2</b>
<b>Návrh FSM .....</b>	<b>2</b>
<b>Návrh RTL.....</b>	<b>3</b>
<b>Výstupy ve GTKWave .....</b>	<b>6</b>

# Návrh

## Návrh FSM

BITY STAVU: (ENABLE\_REGISTER, COUNT\_ENABLE, DOUT\_VALID, OFFSET\_BIT)



Přijímač se spustí ve stavu WAIT\_START\_BIT, což je v podstatě stav IDLE, a čeká na reset (0) v bitu DIN, což je podmínka pro vstup do stavu MAKE\_OFFSET.

Stav MAKE\_OFFSET je určen pro počítání 7 tiků offsetu CLK, pro následné čtení přesně uprostřed, aby se předešlo chybám v případě posunu informace. Jakmile dorazí 7. tik, automat přejde do stavu RECEIVE\_DATA a spustí se přesně uprostřed prvního bitu informace.

Ve stavu RECEIVE\_DATA bude počítat každých 15 tiků CLK, aby skočil na další bit a podle toho navýšil COUNT\_BYTE o 1 (na diagramu jsem to ukázal modře, protože to není přímou součástí stroje, je to potřeba pouze pro lepší pochopení toho, co se děje).

Když se COUNT\_BYTE rovná 8, je to signál, že uplynul poslední bit a můžeme potvrdit svá data bitem DATA\_VLD, respektive přechodem do stavu DATA\_VALID.

Ze stavu DATA\_VALID jdeme momentálně na STOP\_BIT, to se děje výhradně v informaci, že naše informace jsou hotové a jdeme končit, takže jsem tomu stavu udělal takový dvojitý název. Poté aplikujeme na DIN 1, čímž se vrátíme do předchozího stavu WAIT\_START\_BIT (IDLE)

## Návrh RTL

Níže uvedené schéma je udělaná v dost abstraktní formě (nepsal jsem podrobně všechny počítadla, protože neměl jsem dostatek místa a mohl jsem udělat ještě více chyb) a bohužel se mi to nevešlo na stránku tak jsem to musel otočit, doufám, že to není moc velký problém. Některá jména na stroji jsem také zkrátil, protože jsem trochu nekalkuloval s jeho velikostí, ale doufám, že všechny názvy budou dostatečně jasné. Tlustými šipkami jsou označené vektory bitů.

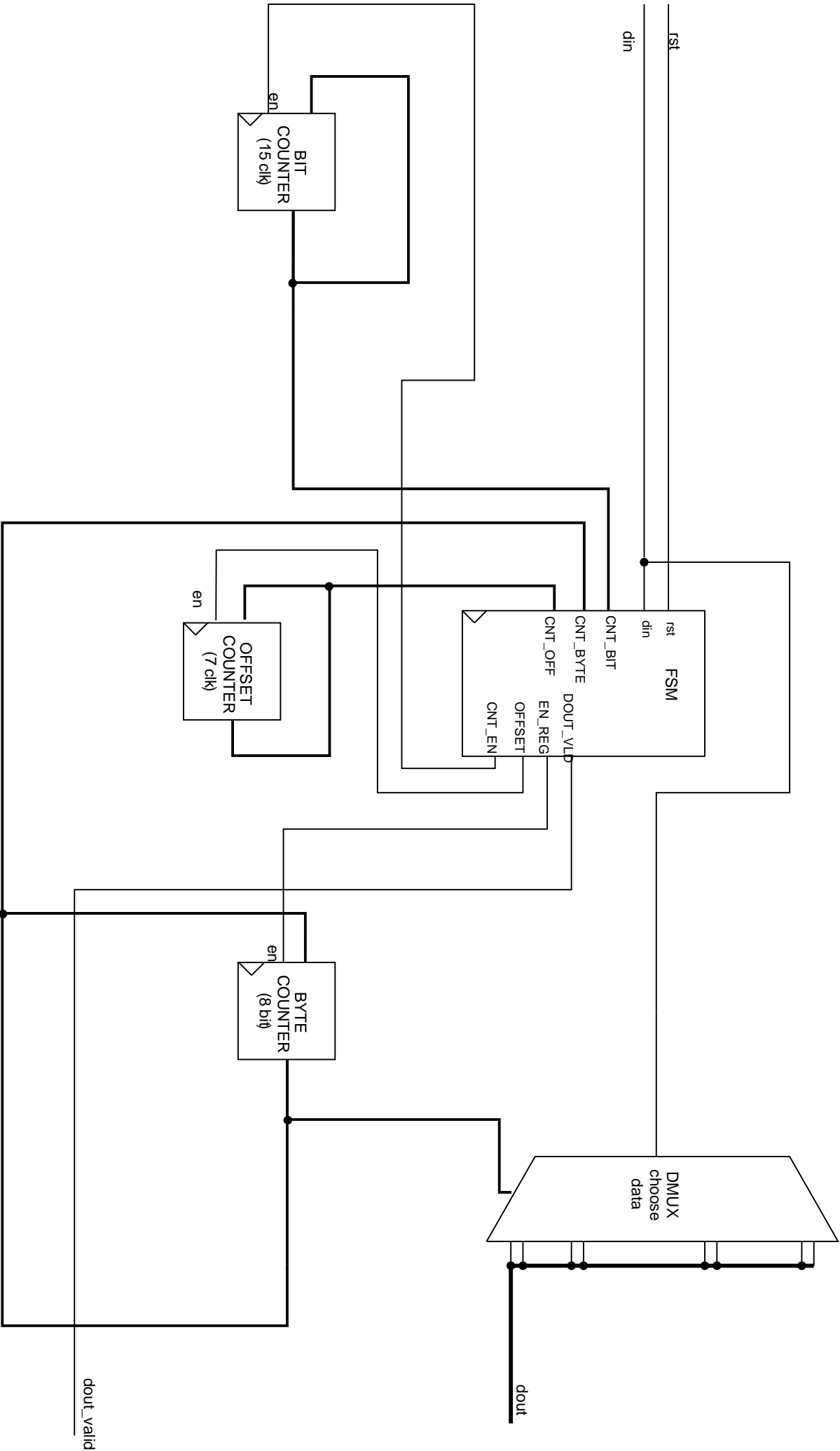
Na vstupu máme DIN s našimi daty, jelikož čtení naše data začínají od bitu 0, který převádí automat do dalšího stavu, připojil jsem din přímo na vstup automatu, také jsem ho oddělil a poslal na vstup demultiplexoru, protože až tam budou data, budeme je muset přečíst.

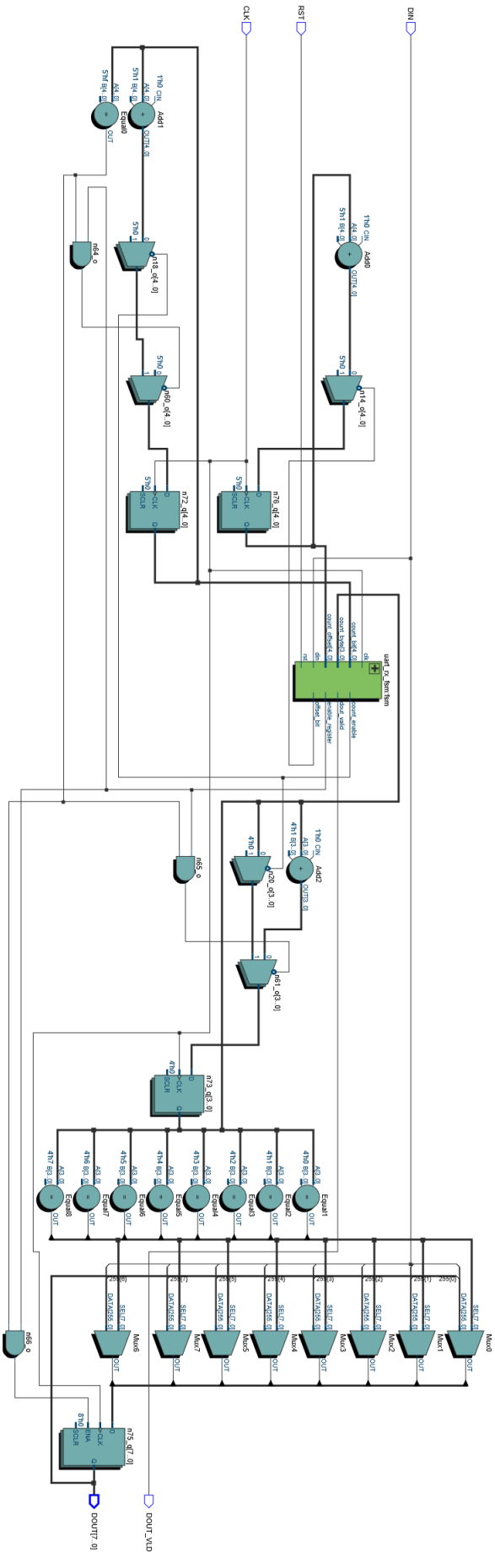
Pak jsme ve druhém stavu OFFSET a náš automat zapne OFFSET\_COUNTER a začne počítat 7 clk (je synchronní, což znamená jeho trojúhelník dole). U všech čítačů jsem připojil zpětnou vazbu z výstupu, aby si nějak pamatovaly svou poslední hodnotu, ale tady jsem se moc do podrobností včetně nulování hodnoty nezabýval, v tomto případě si představte, že k resetu dojde při ENABLE=0.

Když COUNT\_OFFSET je 7, fsm to uvidí na vstupu a přejde do dalšího stavu RECEIVE\_DATA, zapne 2 (BIT a BYTE) čítače, které začnou počítat clk pro příjem dat, každý bit trvá 15 clk, BYTE\_COUNTER odešle výstup do demultiplexoru, který vybere, který z výstupů bude aktuálně posílat bit. Když je 8 bitů, naše čtení skončí, čehož si náš stroj všimne prostřednictvím vstupu COUNT\_BYTE. Tyto čítače mají stejné schéma jako předchozí.

Když čtení se skončí a automat se přesune do dalšího stavu, vyšle signál DOUT\_VALID, který indikuje, že data jsou připravena a din se vrátí na 1, čímž se náš automat vrátí do stavu WAIT\_START\_BIT (IDLE).

Také o stránku později je z kódu vygenerován diagram, který může opravit mé chyby a objasnit několik bodů.





## Výstupy ve GTKWave

