# Programozható LED-fűzéren alapuló reklámpanel - LED fűzér vezérlése, adatok kiírása

Patka Zsolt-András | Számítástechnika BSc

2019.11.11

## 1. Beveztő

A projekt célja egy LED-fűzér vezérlése és ennek segítségével egy reklámszöveg megjelenítése. Ehhez egy FPGA lap és egy Worldsemi WS2813 ledfűzér lesz felhasználva.

## 2. Követelmények

### 2.1. Funkcionális követelmények

- Lehetséges legyen egy reklámszöveget kiírni a ledfűzérek által létrehozott mátrix-ra.
- Egy sorban minimum 100 LED található
- Az egyes ledfűzérek szinkronba működjenek
- A kiírás párhuzamosan történik az adott ledfűzéreken
- Másodpercenként 60 frissítés (60 Hz)
- Mozgó szöveg

### 2.2. Nem funkcionális követelmények

- Benti használatra van tervezve
- Ha az FPGA-lapnak lesz készítve külön tokozat, akkor IP31-es standardnak kell megfeleljen
- Ha az FPGA-lapnak nem lesz készítve külön tokozat, akkor nem felel meg IP standardnak (IP00)

### 2.3. Fejlesztési követelmények

- Implementáció VHDL nyelvben
- Szimulációs állomány a rendszer tesztelésére
- Adatok kiírása lehetséges a LED fűzérre
- Modularitás
  - Külön modul egy 24 bit-es blokk küldésére
  - Külön modul a 24 bit-es blokkokat küldő modulok használatára (paraméterezhető, bővíthető)
- Opcionális:
  - Pár betű kódolása (3-4)
  - Betük tárolása BRAM memóriában

## 3. Rendszer-specifikáció

- WS2813 egyszálú adatátvitel protokolljának helyes használata
- Worldsemi WS2813 100 LED-es ledfűzér van felhasználva
- Digilent Basys3 FPGA vezérli a ledfűzért

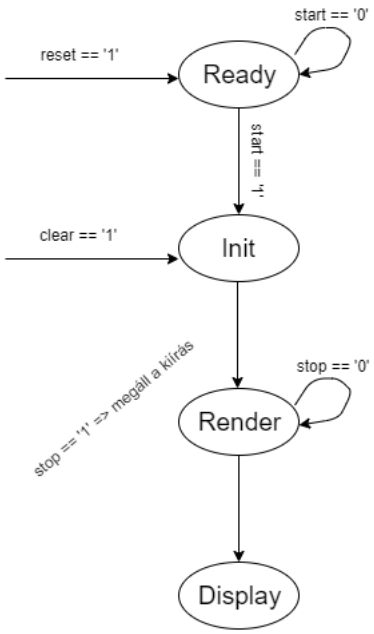
# 4. Tervezés

## 4.1. Állapotok

**2019.10.14**

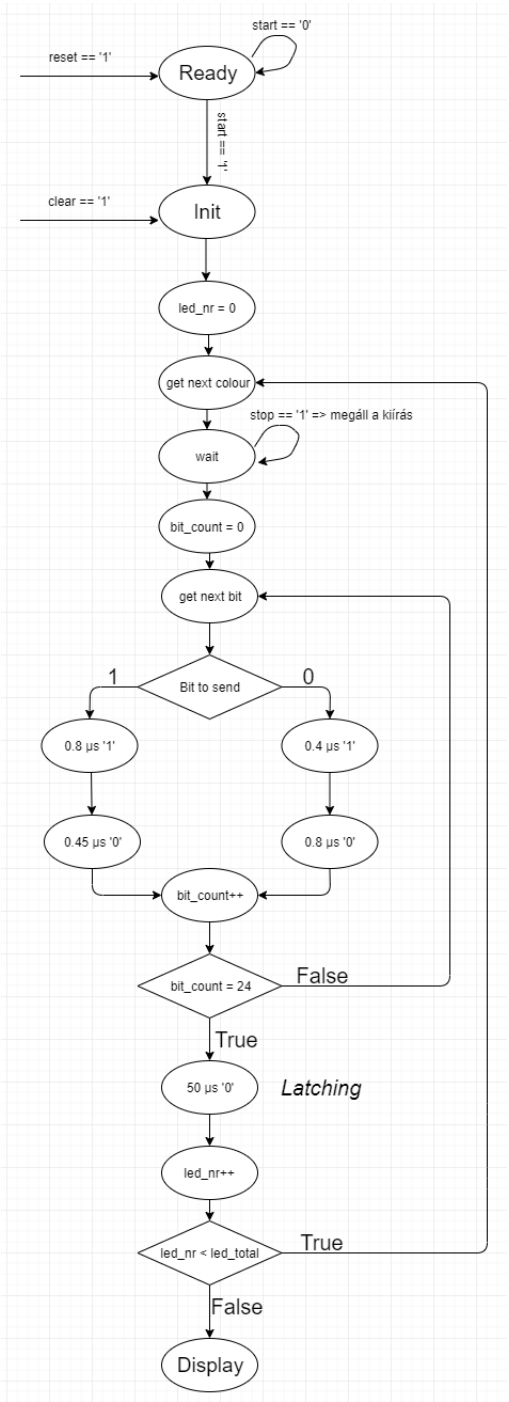
Állapotok:

- READY
  - Alap állapot
  - "reset" jel esetén ide kerül vissza az automata
- INIT
  - minden LED-et kikapcsol (0x000000-t ír)
  - "clear" jel esetén ide kerül az automata
- RENDER
  - egyenként küldi a szín információt a LED-ekre
  - annyiszor végződik el itt a művelet, ahány LED-ünk van
  - "stop" jel esetén megáll a kiírás
- DISPLAY
- megtörtént a kiírás

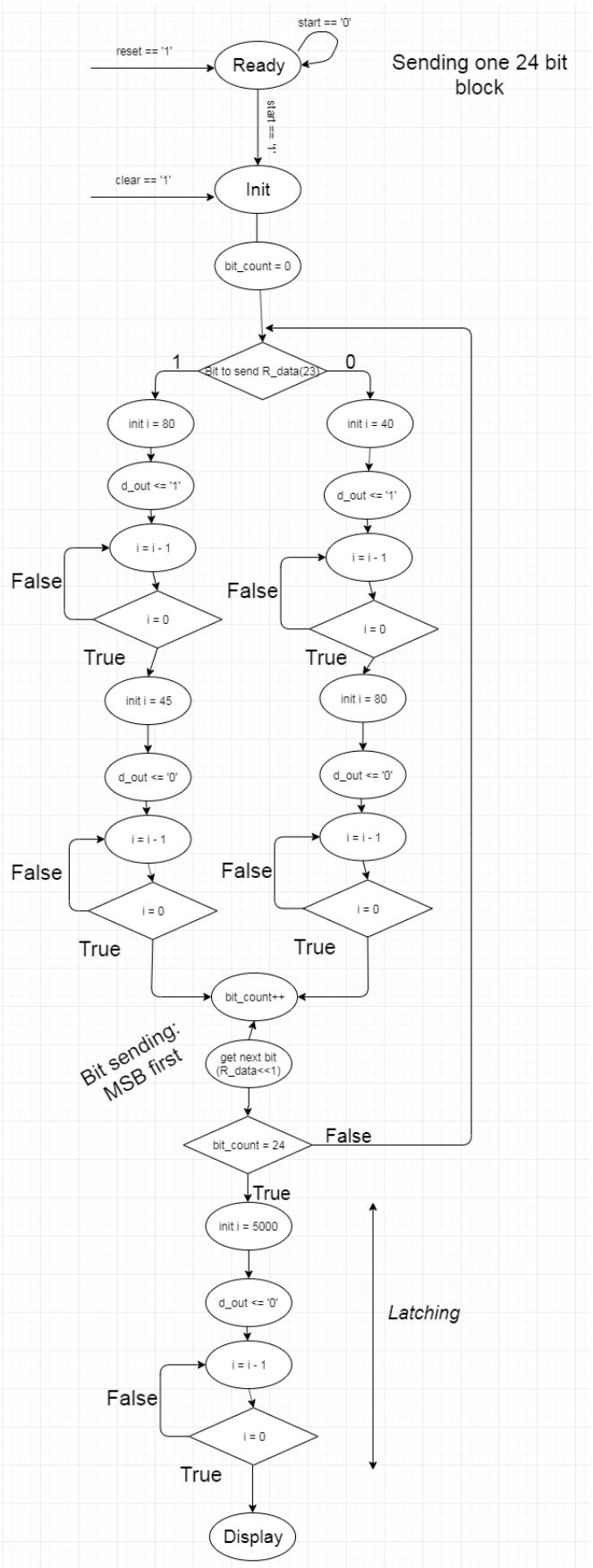



**2019.10.28**

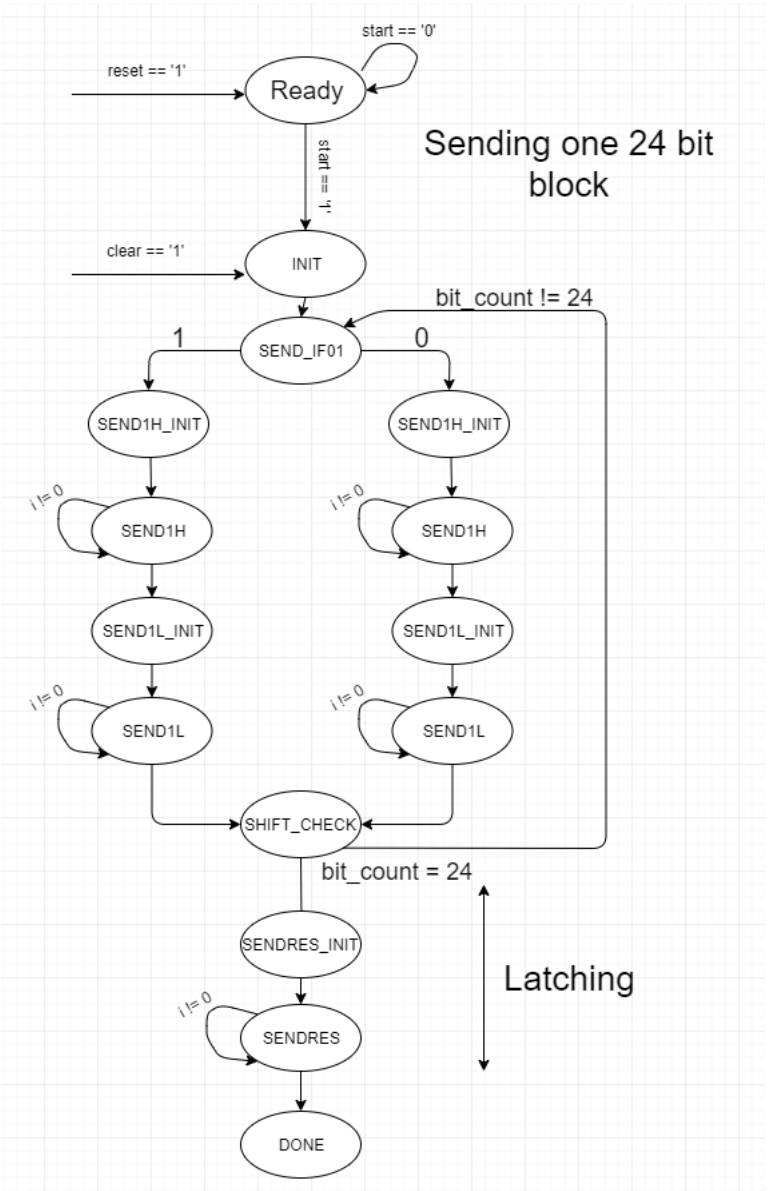
Állapotdiagram átírva úgy, hogy a küldési logikát is tartalmazza:




Állapotdiagram egy 100 LED-et vezérlő modulra




**2019.11.11**
Végleges állapotdiagram, az implementációban is használt.



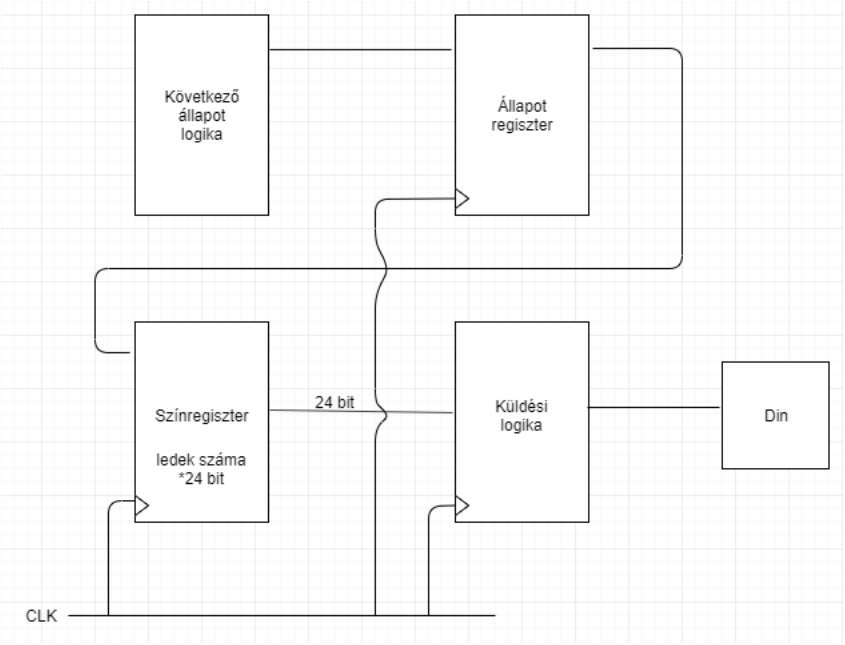

- READY
  - Alap állapot
  - "reset" jel esetén ide kerül vissza az automata
- INIT
  - inicializálja a bit-count-ot nullára
- SENDIF_01
  - Megvizsgálja data[23]-as bit-et. Ha 1, akkor a következő állapot a **SEND1H_INIT**, ha 0 akkor SEND0H_INIT
- SEND1H_INIT
  - Inicializálja az **i** változót a T1H értékre
  - A d_out output-ot 1-re állítja
- SEND1H
  - Dekrementálja az **i** változót
  - Amikor az **i** változó nulla lesz, akkor tovább megy a **SEND1L_INIT** állapotra
- SEND1L_INIT

  - Inicializálja az **i** változót a T1L értékre
  - A d_out output-ot 0-ra állítja
- SEND1L
  - Dekrementálja az **i** változót
  - Amikor az **i** változó nulla lesz, akkor tovább megy a **SHIFT_CHECK** állapotra
- SEND0H_INIT
  - Inicializálja az **i** változót a T0H értékre
  - A d_out output-ot 1-re állítja
- SEND0H
  - Dekrementálja az **i** változót
  - Amikor az **i** változó nulla lesz, akkor tovább megy a **SEND0L_INIT** állapotra
- SEND0L_INIT
  - Inicializálja az **i** változót a T0L értékre
  - A d_out output-ot 0-ra állítja
- SEND0L
  - Dekrementálja az **i** változót
  - Amikor az **i** változó nulla lesz, akkor tovább megy a **SHIFT_CHECK** állapotra
- SHIFT_CHECK
  - Megnézi, hogy a **bit_count** változó 24-e, ha igen, akkor tovább megy a **SENDRES_INIT** állapotra
  - Ha a **bit_count** változó nem egyenlő 24-el, akkor shift-eli a **data** std logic vectort balra eggyel; vissza megy a **SEND_IF01** állapotra
- SENDRES_INIT
  - Inicializálja az **i** változót a TRES értékre
  - A d_out output-ot 0-ra állítja
- SENDRES
  - Dekrementálja az **i** változót
  - Amikor az **i** változó nulla lesz, akkor tovább megy a **SEND_DONE** állapotra
- SEND_DONE
  - Befejeződött a 24 bit-es blokk kiírása
  - Beállítja a **done** kimenetet 1-esre

## 4.2. Alegységek

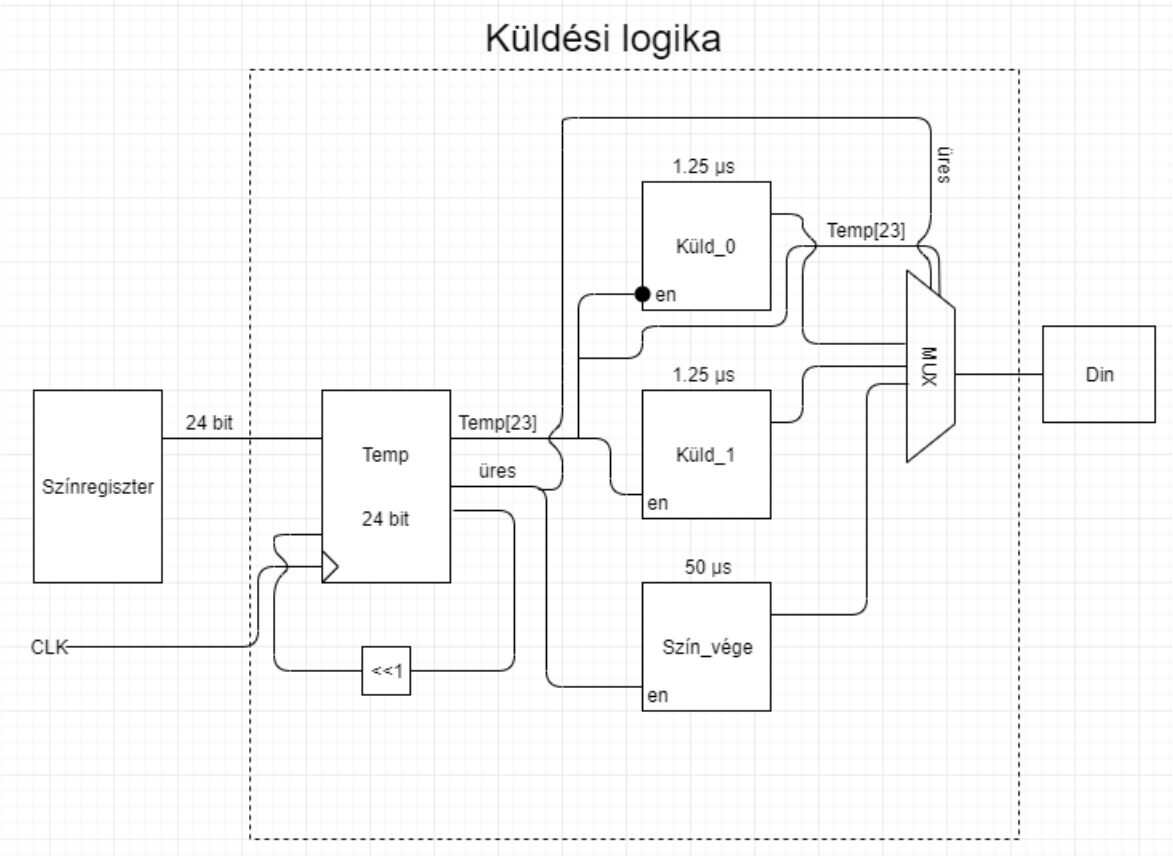
**2019.10.14**

- Következő állapot regiszter ***Next State Register***
- Állapot regiszter ***State Register***
- Szín regiszter ***Colour Register***
- Küldési logika regiszter ***Transmission Logic Register***



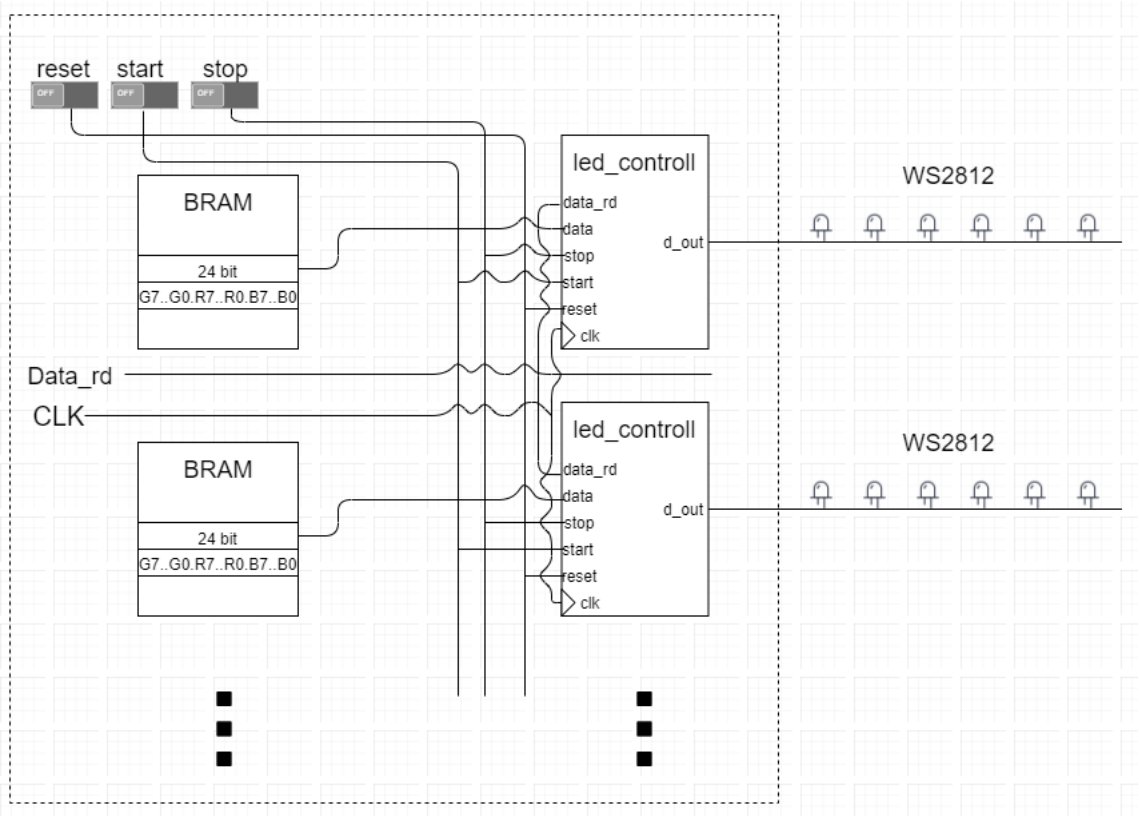

#### 4.2.1. Küldési logika regiszter

A küldési logika modul részletesebb lebontása:




**2019.10.28**

Minden **led_controll** modulhoz tartozik egy BRAM blokk és minden ilyen modul egy ledfűzért vezérel meg. Öt ilyen blokk megvezérel öt ledfűzért, ezáltal létrehozva a ledmátrixot. Az órajel, start, reset, stop és data-rd jelek közösek minden modulnak.




## 4.3. WS2813 egyszálú adatátvitel protokoll leírása

A LED-eket vezérlő áramkörök egymás után vannak bekötve úgy, hogy az egyik áramkörnek az adatkimenete a következő áramkörnek az adatbemenetét képzi. Egyszálú az adatátvitel, fontos a protokoll betartása, ahhoz, hogy adatokat tudjunk megjeleníteni a LED-fűzéren.

Amikor egy áramkör megkap egy 24 bit-es kódot, akkor ezt addig tárolja amíg más kódot nem kap, vagy a tápforrást el nem veszti.

### 4.3.1. A 24 bit-es kód

A 24 bit-es kód a következőképpen kell kinézzen:
8 bit GREEN | 8 bit RED | 8 bit BLUE
Az adatátvitel a következő sorrendben kell történjen:

1. GREEN
2. RED
3. BLUE

### 4.3.2. Bit-ek küldési sorrendje

**Az egyes byte-ok küldését úgy kell elvégezni, hogy az MSB-vel kell kezdeni és haladni az LSB fele.**
24 bit-es kód részletesebb felbontása:

- *G7 G6 G5 G4 G3 G2 G1 G0 | R7 R6 R5 R4 R3 R2 R1 R0 | B7 B6 B5 B4 B3 B2 B1 B0*

A küldés a következő sorrendben kell elvégződjön:

- **G7 G6 G5 G4 G3 G2 G1 G0 | R7 R6 R5 R4 R3 R2 R1 R0 | B7 B6 B5 B4 B3 B2 B1 B0**

### 4.3.3. Időzítések

Minden 24 bit-es adatátvitel után kell legalább 50 µs-ot várakozni, alacsony feszűltségen. Ez jelzi azt, hogy egy 24 bit-es blokk továbbítása megtörtént.
Az egyes bit-ek átvitele a következőképp történik:

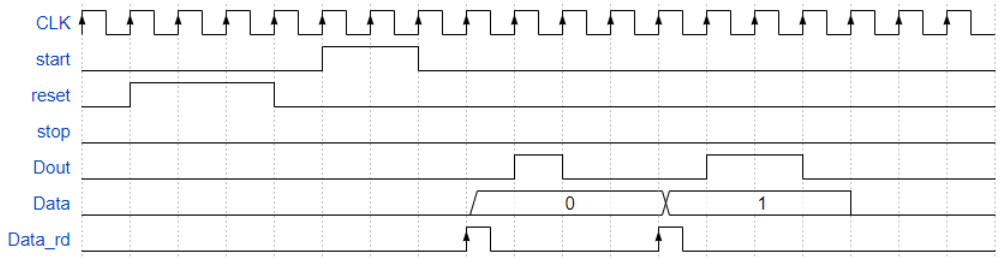
- Logikai 1-es
  - 0.8 µs-ot magas feszültségen
  - 0.45 µs-ot alacson feszültségen
- Logikai 0-ás
  - 0.4 µs-ot magas feszültségen
  - 0.85 µs-ot alacson feszültségen
- 24 bit-es adatblokk küldése után:
  - $>$ 50 µs-ot alacsony feszültségen

A bit-ek továbbításánál egy +/- 150 ns-os eltérés megengedett.

A várakozási értékeket nem az adatlapból, hanem az alábbi útmutatóból vettem. Az útmutató szerint az adatlapban levő értékek rosszul vannak kiszámolva.

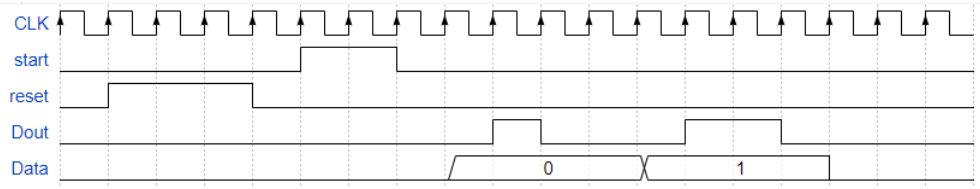
Egyelőre megpróbálok az útmutatóban megadott értékekkel dolgozni. Ha ez nem megfelelő működéshez vezet, akkor veszem az adatlapban levő értékeket.

## 4.4. Idődiagram




- CLK: 100 MHz-es órajel
- start: jel a folyamat elindításához
- reset: jel a folyamat resetálásához
- stop: jel a kiírás megállításához. Csak két 24 bit-es blokk kiírása közben tudja megállítani a kiírást
- Dout: Egyszálú adatsín a LED-ekre.
- Data: Kiírandó adat, Data_rd felmenő órajelére olvassa be az adatot.
- Data_rd: Aktiváló bit az adat beolvasására

**2019.11.11**




- CLK: 100 MHz-es órajel
- start: jel a folyamat elindításához
- reset: jel a folyamat resetálásához
- Dout: Egyszálú adatsín a LED-ekre.
- Data: Kiírandó adat