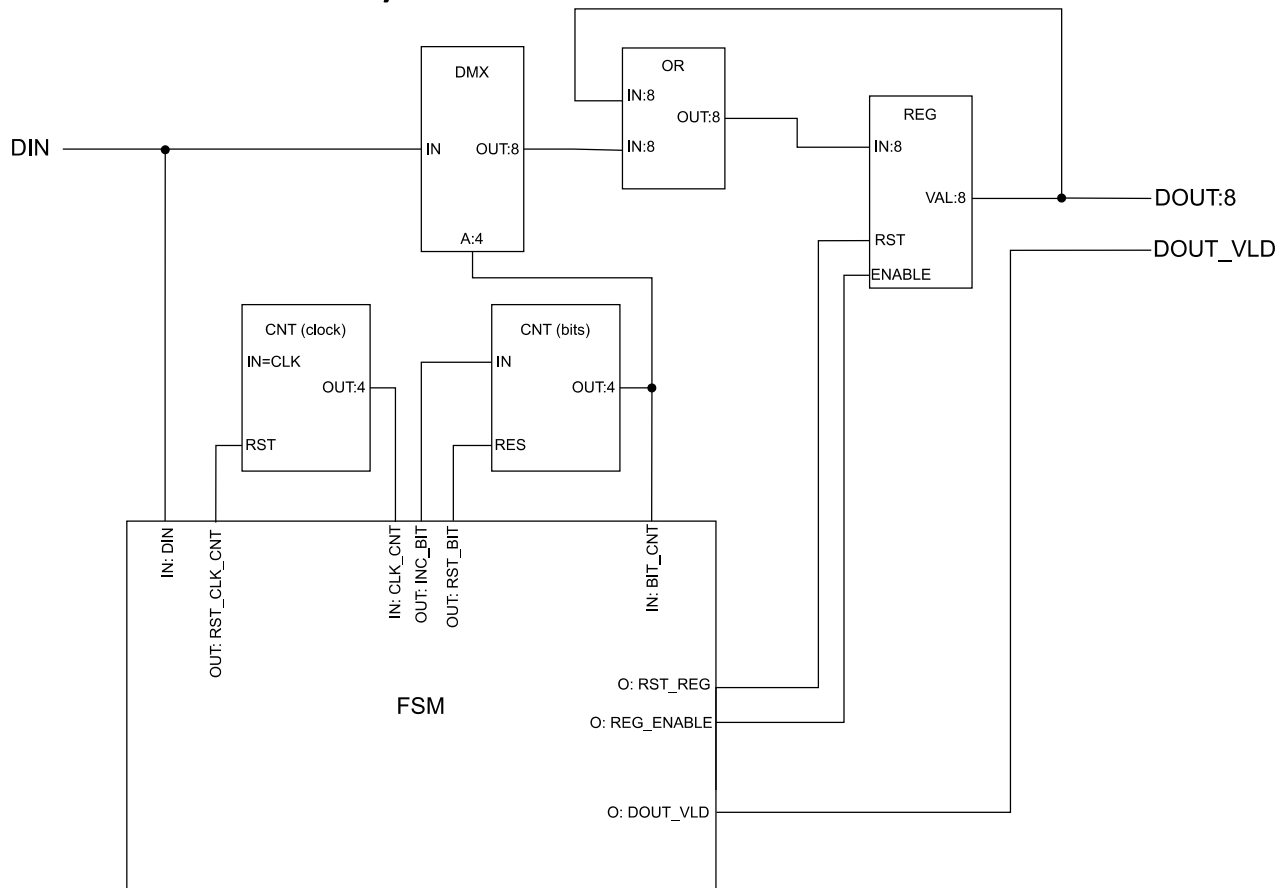


INC – Návrh číslicových systémů Implementace projektu

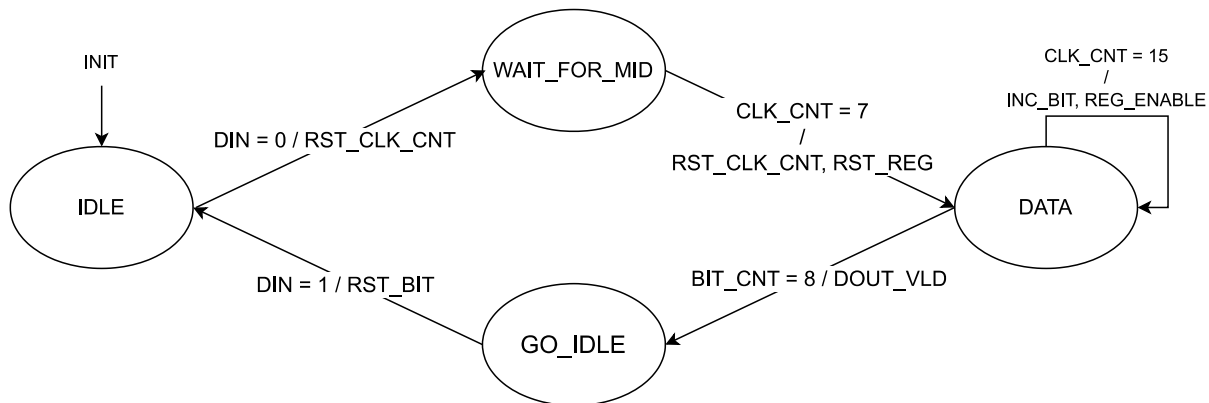
Návrh architektury



Poznámky:

- Návrh se oproti původnímu liší dost, vynechal jsem veškerou kombinační logiku, o kterou jsem se snažil v původním návrhu
 - např. Výpočet binárních vstupů do automatu na základě složitých podmínek realizovaných různými porovnávacími prvky – vizte původní návrh
 - moje motivace k přepracování byla taková, že původní návrh by byl jen velice obtížně realizovatelný ve VHDL
- koncepce činnosti přijímače je však pro obě verze návrhu totožná

Návrh automatu



Poznámky:

- oproti původnímu návrhu jsem zaměnil názvy a poupravil vstupy a výstupy, aby návrh co nejvěrněji odpovídal skutečné implementaci
- nijak se však nezměnila podstata činnosti

Implementace

Návrh jsem zdárně realizoval tak, aby fungoval v poskytnutém prostředí. Obvod jsem vyvíjel a ladil s poskytnutou kostrou, test-benchem a překladovým skriptem na OS **Windows 11** uvnitř subsystému **WSL**, kde mám nainstalovaný systém **Debian 11**, s použitím standardních nástrojů **ghdl** a **gtkwave**.

Zde budiž verze software, které používám:

WSL version: 1.2.0.0

kernel-release: 5.15.90.1-microsoft-standard-WSL2

ghdl: Compiled with GNAT Version: 10.2.1 20210110

gtkwave: v3.3.104

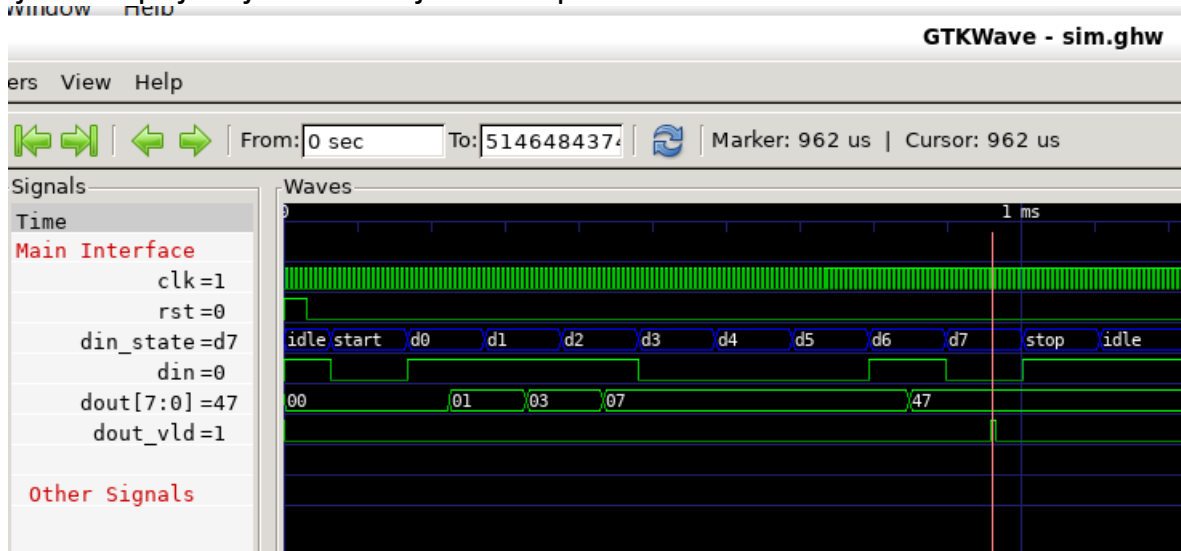
Implementaci jsem vyzkoušel i na serveru **fitkit-build-fit.vutbr.cz**, odkud se mi sice nepodařilo vzdáleně spustit GUI **gtkwave**, ale jinak výstup vypadal stejně.

Během implementace se skutečně nekonalo mnoho invence. Realizoval jsem navržený obvod. Potýkal jsem se však s problémem, že byl-li čítač navržen jako asynchronní, automatu nikdy nepřešel z jednoho stavu do druhého a nevěděl jsem proč. Po přepracování čítače na jeho synchronní variantu byl problém vyřešen.

Jedna implementační zajímavost také je, že registr se resetuje až před samotným začátkem příjmu dalšího bajtu. To jsem původně nezamýšlel, ale když už jsem měl

implementaci hotovou a koukal na časové diagramy výstupů obvodu, řekl jsem si, že pro přijímací stranu by mohlo být dobré, kdyby data na výstupu byla platná déle, než je garantováno výstupem **DOUT_VLD**. Jeho trvání jsem ponechal 1 takt hned po dokončení příjmu bajtu.

Zde je vidět příjem jednoho bajtu **0x47** po lince:



Zde výstup poskytnutého testbenche:

- výstup byl zdárný lokálně u mě i na serveru fitkit-build

```
Sending data onto DIN with value: 0x47
Output data from DOUT with value: 0x47
Sending data onto DIN with value: 0x55
Output data from DOUT with value: 0x55
Sending data onto DIN with value: 0xAA
Output data from DOUT with value: 0xAA
Sending data onto DIN with value: 0xCA
Output data from DOUT with value: 0xCA
```

Závěr:

Podařila se mi implementace přijímače asynchronní sériové linky zjednodušené formy protokolu UART. Seznámení s jazykem VHDL bylo postupné a v jednom momentě jsem zjistil, že budu muset přepracovat návrh. Všechno dobře dopadlo a implementace funguje zamýšleným způsobem.