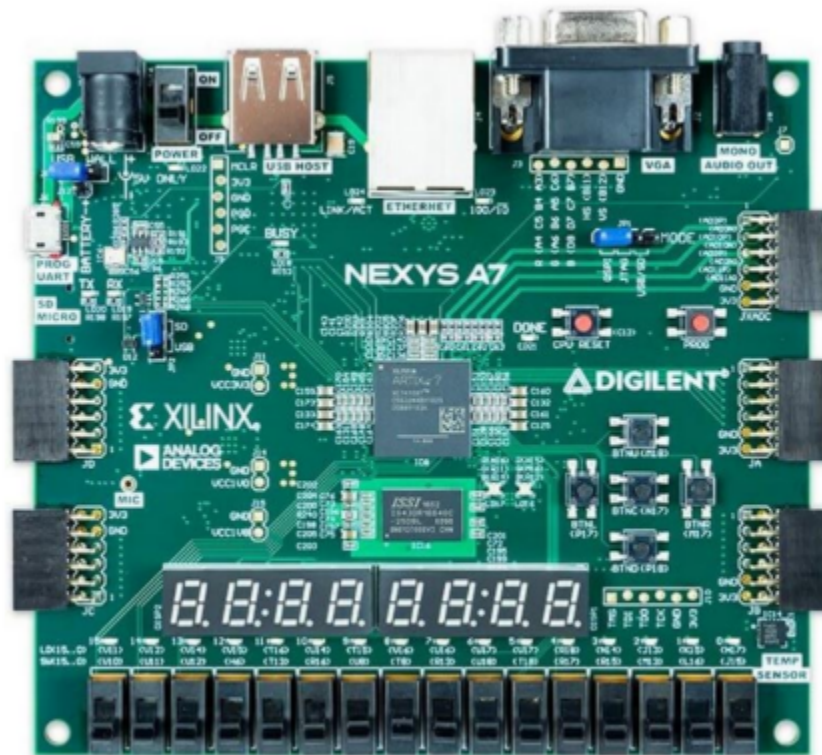


Projekt dokumentáció

Újrakonfigurálható digitális áramkörök

ADCS7476 nyomásmérő szenzor 16 bites A/D konverter

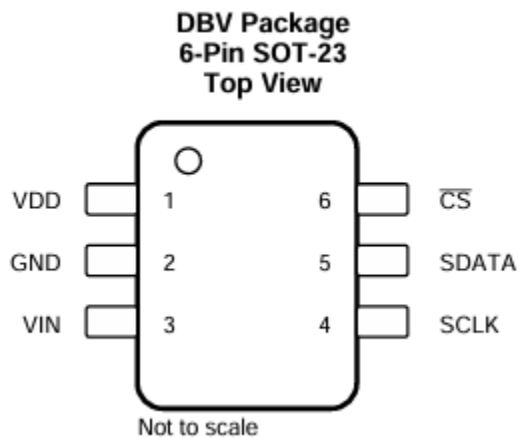
A projekt során egy FPGA újrakonfigurálható digitális áramkörre terveztünk. Az FPGA napjainkban egy fontos eszköz az automatizálás területén. Egy igen jó alternatívát nyújthatnak a mikrovezérlős implementációk helyett. Számos területen használják, például: mesterséges intelligencia és gépi tanulás terén, AI modellek implementálása FPGA-n, autóiparban, egészségügyben vagy űrkutatás és katonai alkalmazásokban. Nagy előnye a gyorsaság, például a mi fejlesztőlapunk képes 100MHz gyorsasággal dolgozni. Egyén előnyök még például az újra programozhatóság, nagy flexibilitás vagy az olcsó kihozatali költség.



1.ábra FPGA fejlesztőlap

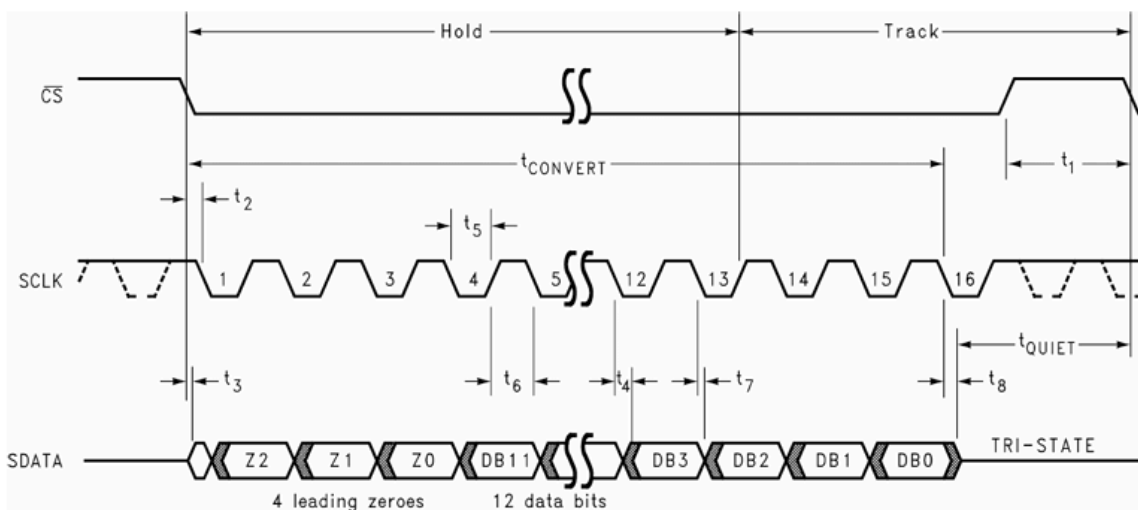
A projekt feladatomban volt egy ADCS7476 nyomásmérő szenzorral dolgozni. Az áramkör 12 biten alakítja át az analóg jelet digitális jelre. Alkalmazási területei: autóipar, ipari automatizálás, hordozható rendszerek, orvosi eszközök, műszer- és vezérlőrendszerek. Főleg olyan területekre tervezték ahol tehát olyan területre tervezték a szenzort, ahol kritikusan kevés

a hely a szenzornak és a hőmérséklet -40 és 125 C között van. Egy alacsony tápellátású analog digitál konverter, lehetőség van kikapcsolási módba állítani, amikor minimális a fogyasztása. Az analóg-digitál átalakítási sebesség függ *seria clock* bemenetén érkező órajeltől, az eszköz képes 20 MHz sebességre.



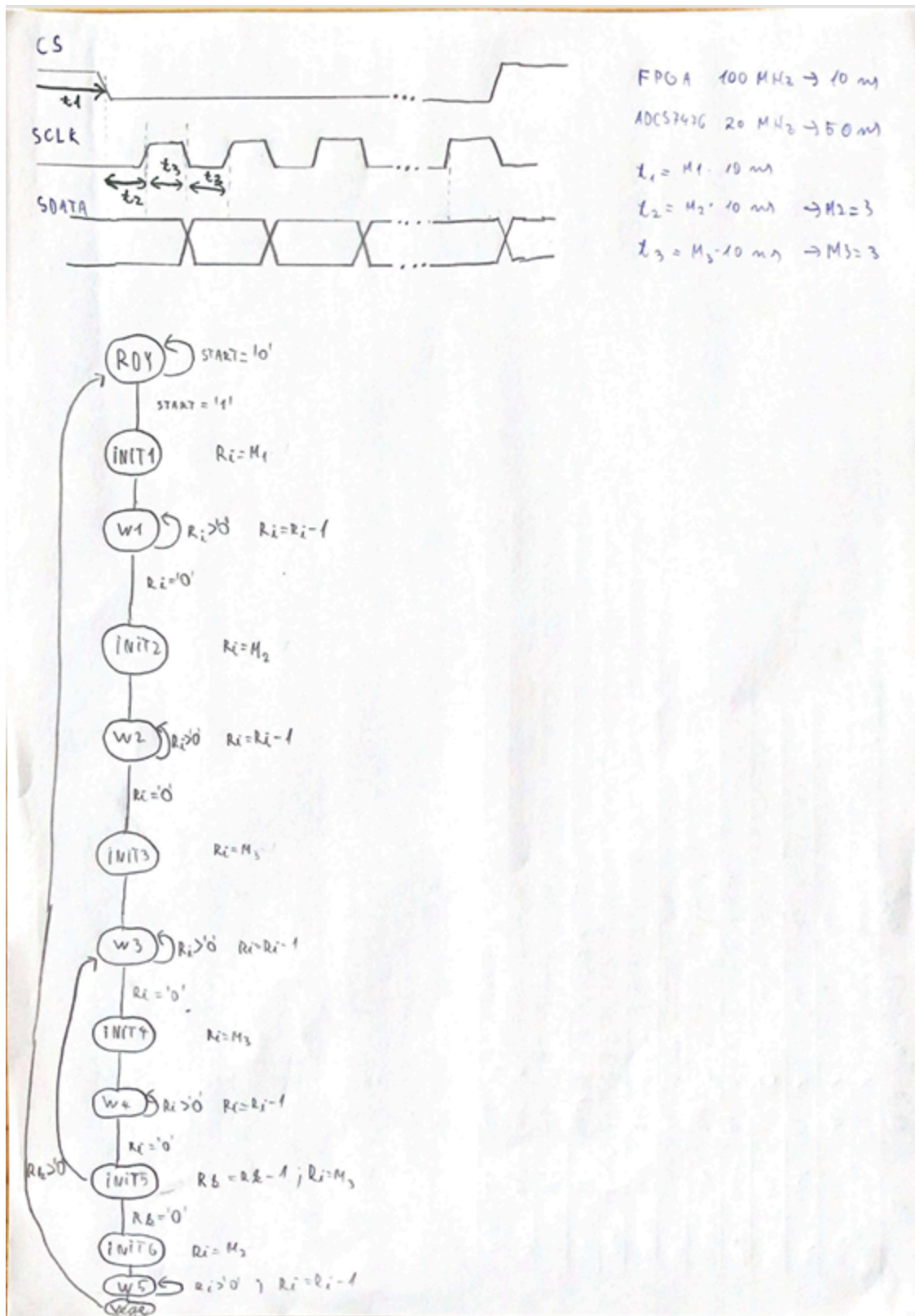
2.ábra lábkiosztás

A soros interfész kompatibilis az SPI kommunikációs protokollal így azt használtam a projekt során.



3.ábra soros interfész idődiagramja

A soros interfész idődiagramján látható, hogy a kommunikáció kezdetét veszi, ha a CS bemenet magas feszültségi állapotból alacsonyba lépik. Az adat 16 biten érkezik, az első 4 biten zérós értékek, majd az ezt követő 12 bit, MSB -vel kezdődően, hordozza az adatot (a levegőnyomás értékét)

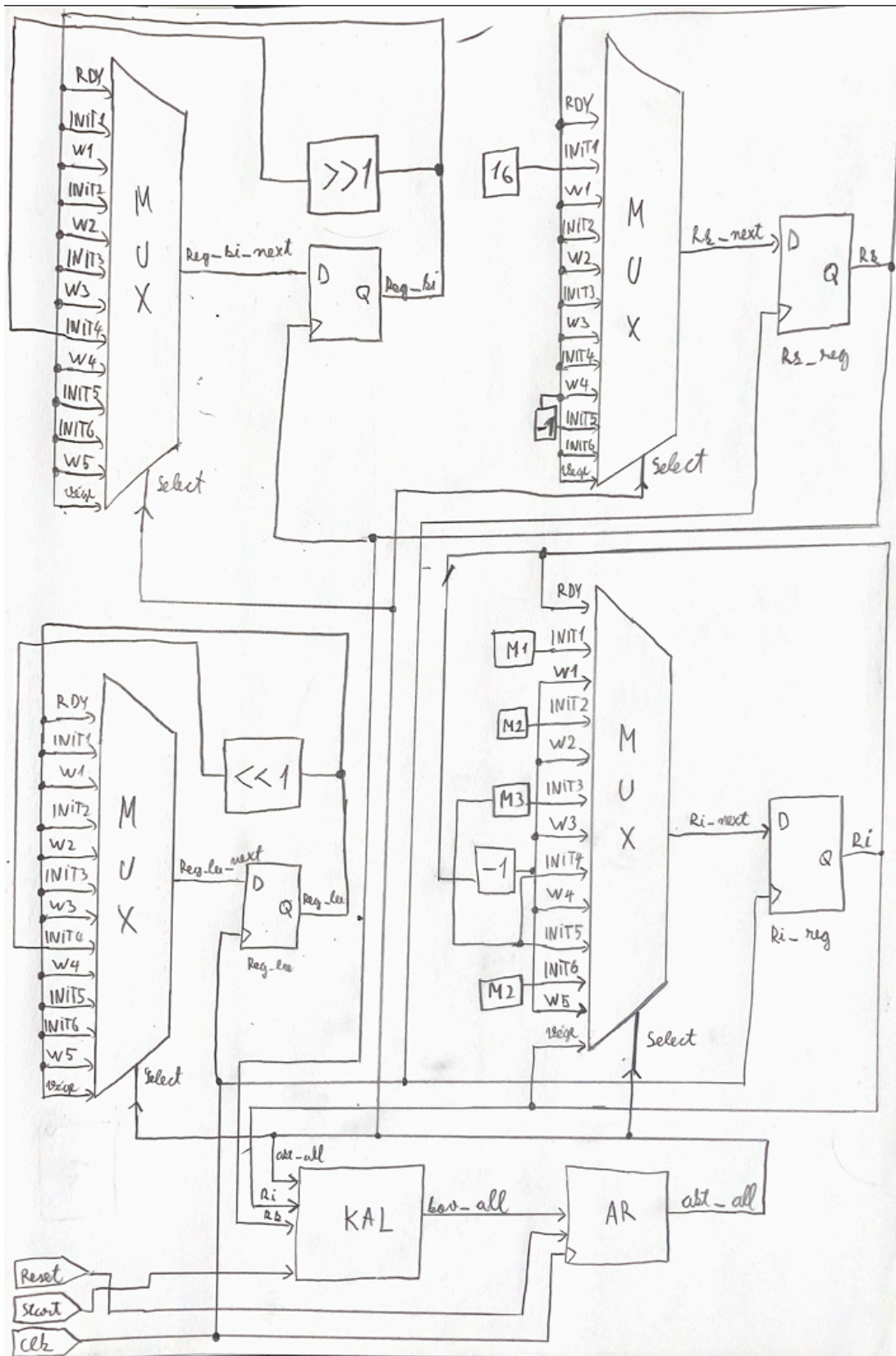


4. ábra idődiagram és állapotdiagram

A megvalósításhoz használtam FSMD komplex állapotgépet adatúttal. Az ábrán látható az állapotdiagram, 13 állapoton megy végig, a "start" aktiválása után lépik ki a "RDY" állapotból és végigmegy egy "INIT" "WAIT" sorozaton. Az Ri, Rk változók segítségével történik az időzítés, az SCLK meghatározása. Az Rk egy külső ciklus, azt vizsgálja, hogy végigmentünk-e a 16 biten, az Ri pedig előállítja a magas és alacsony állapotokat, késlelteti az FPGA 10 ns- os jelét.

	Ri	Rk	RCS	RSCLK	Reg-hi	Reg-lu	Rstatus
RDY	Ri	Rk	1	0	valid-hi	Reg-lu	0
INIT1	M1	16	1	0	Reg-hi	Reg-lu	0
W1	Ri-1	Rk	1	0	Reg-hi	Reg-lu	0
INIT2	M2	Rk	0	0	Reg-hi	Reg-lu	0
W2	Ri-1	Rk	0	0	Reg-hi	Reg-lu	0
INIT3	M3	Rk	0	0	Reg-hi	Reg-lu	0
W3	Ri-1	Rk	0	1	Reg-hi	Reg-lu	0
INIT4	M3	Rk	0	1	Reg-hi	Reg-lu < 1	0
W4	Ri-1	Rk	0	0	Reg-hi	Reg-lu	0
INIT5	M3	Rk-1	0	0	Reg-hi > 1	Reg-lu	0
INIT6	M2	Rk	0	1	Reg-hi	Reg-lu	0
W5	Ri-1	Rk	0	1	Reg-hi	Reg-lu	0
Vége	Ri	Rk	1	0	Reg-hi	Reg-lu	1

5.ábra állapot táblázat



6. ábra blokk diagram

A következő kódrészlet mutatja az Ri, Rk változók meghatározását minden állapotban:

```
with akt_all select
    Rk_next<=Rk when RDY,
        "10000" when INIT1,
    Rk when W1,
    Rk when INIT2,
    Rk when W2,
    Rk when INIT3,
    Rk when W3,
    Rk when INIT4,
    Rk when W4,
    Rk - 1 when INIT5,
    Rk when W5,
    Rk when INIT6,
    Rk when Vege;
```

7.ábra Rk regiszter minden állapotban

```
with akt_all select
    Ri_next<=Ri when RDY,
        M1 when INIT1,
    Ri-1 when W1,
    M2 when INIT2,
    Ri-1 when W2,
    M3 when INIT3,
    Ri-1 when W3,
    M3 when INIT4,
    Ri-1 when W4,
    M3 when INIT5,
    Ri-1 when W5,
    M2 when INIT6,
    Ri-1 when Vege;
```

8.ábra Ri regiszter minden állapotban

Források

- „Brassai Tihamer_UKDA_REAL”
- http://real.mtak.hu/122602/1/Brassai%20Tihamer_UKDA_REAL.pdf
- <https://nandland.com/lesson-1-what-is-an-fpga/>