

# Začiatky s VHDL

**Publikované: 23.05.2013, Kategória: Návodý a teória**

**www.svetelektro.com**

Zobrazovanie hodnoty z externého AD prevodníka na 7-segmentovom LCD displeji

Na Žilinskej univerzite sme na inžinierskom stupni absolvovali predmet Návrh zákaznických integrovaných obvodov, kde sme sa zoznámili s FPGA firmy Xilinx, konkrétne sme pracovali s vývojovou doskou Spartan-3A 1800 DSP. Ku FPGA Xilinx je poskytovaná free verzia programu Xilinx ISE, v ktorej sme písali program v jazyku VHDL a simulovali činnosť programu. Mojou úlohou bolo zostrojiť zariadenie schopné čítať hodnotu z paralelného 8-bitového AD prevodníka a zobrazovať ju na 7-segmentovom LCD displeji.

Tento článok má slúžiť hlavne ako názorná ukážka ako prebieha realizácia zapojenia pre FPGA a programovanie v jazyku VHDL.

## 1 Realizácia zapojenia

Pri koncepcii zariadenia som vychádzal zväčša katalógových listov výrobcov a vlastných skúseností.

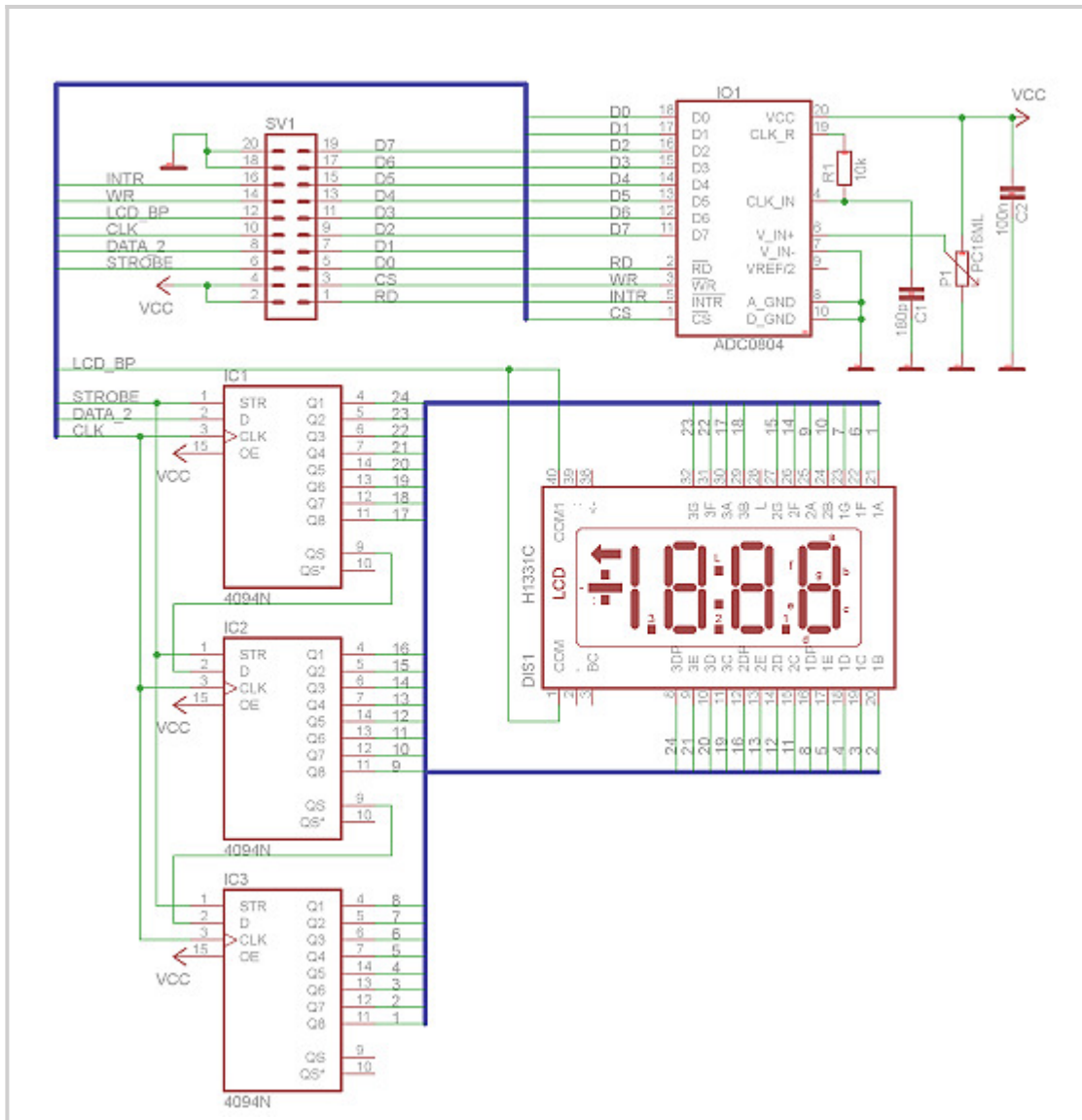
**Celé zariadenie možno rozdeliť na dva hlavné bloky:**

- Blok riadenia a čítania dát z AD prevodníka
- Blok riadenia LCD displeja

Prepojenie s vývojovou doskou Spartan-3A je realizované pomocou 20pin konektora, ktorý slúži zároveň aj na napájanie.

**Schému zapojenia a DPS v programe Eagle 6.2 si môžete stiahnuť: [fpga\\_meter.zip](#)**

**Schéma zapojenia:**



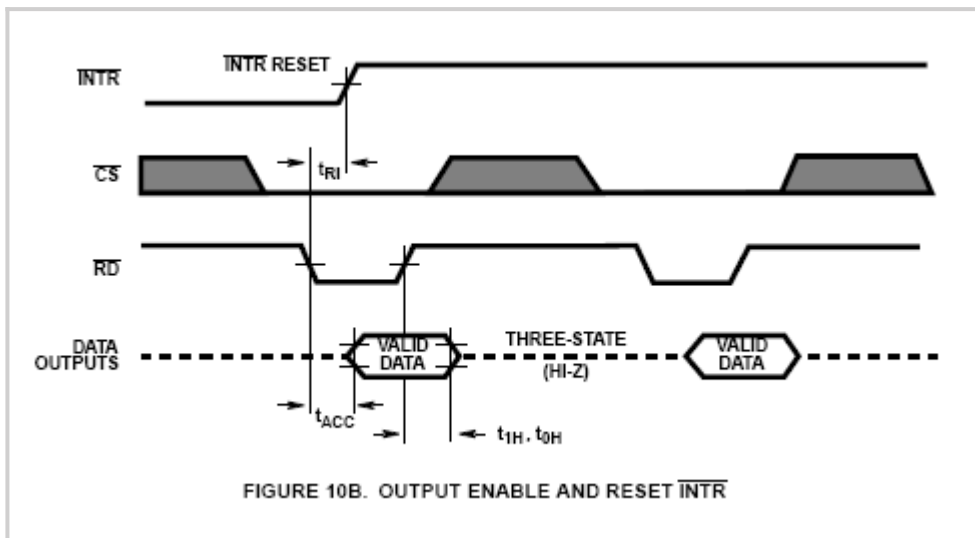
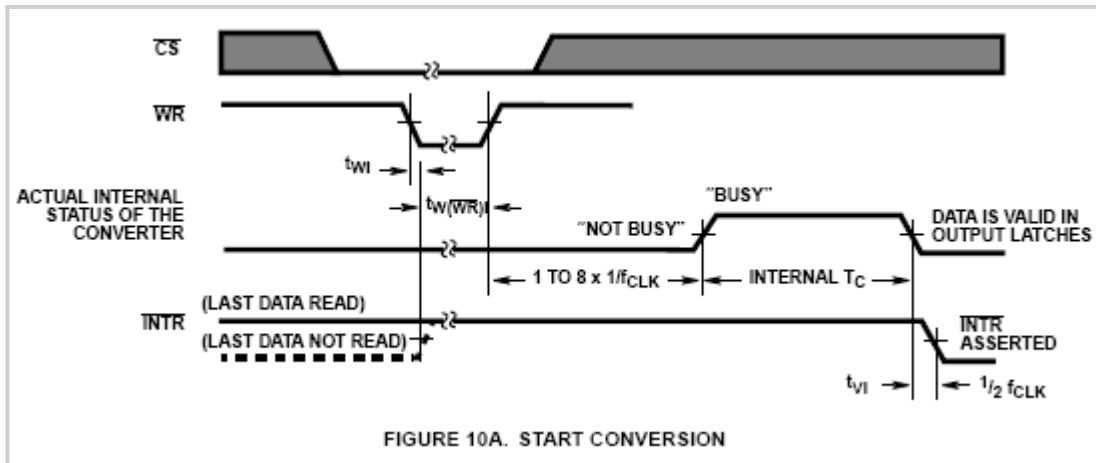
## 1.1 Blok riadenia a čítania dát z AD prevodníka

Tento blok sa stará o riadenie a čítanie dát z AD prevodníka ADC 0804. Na komunikáciu sa využívajú 4 riadiace signály - WR, RD, CS, a INTR a 8 dátových signálov D0-D7. AD prevodník potrebuje pre svoju činnosť minimum externých súčiastok, v minimálnej konfigurácii stačí len RC kombinácia pre generovanie frekvencie vnútorného oscilátora. Štandardné napájanie AD prevodníka je 5V. Pri tomto napájacom napätí sa nastaví pomocou interného deliča referenčné napätie 2,5V. Ja som na dosku doplnil aj odporový trimer pomocou ktorého možno privádzať rôzne hodnoty napätia na vstup AD prevodníka a tým demonštrovať jeho činnosť.

Čítanie dát s AD prevodníka prebieha nasledovne:

- 1) Signály CS a WR sa privedú do log. 0 po dĺžku minimálne 100ns
- 2) Čaká sa na signál INTR z AD prevodníka, ktorý oznamuje riadiacej aplikácii, že prevod je skončený
- 3) Signály CS a RD sa privedú do log. 0 a v tomto stave sa prečítajú dáta z AD prevodníka paralelne pomocou pinov D0-D7.

Časovanie je zrejmé aj s Obr. 1.

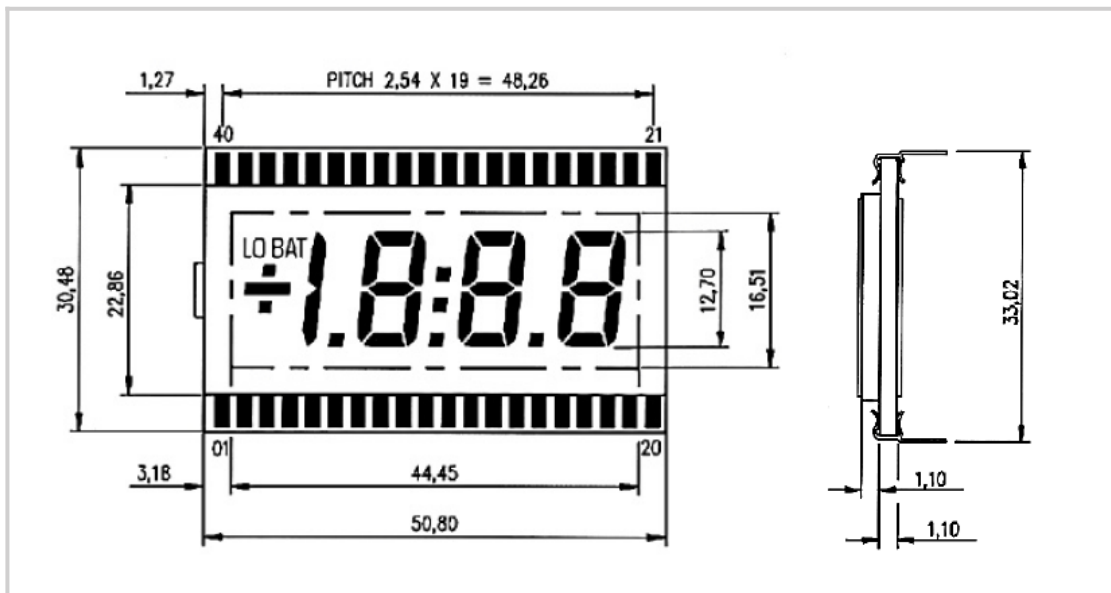


Obr. 1. - Časovanie AD prevodníka

## 1.2. Blok riadenia LCD displeja

Na riadenie 3,5miestneho 7-segmentového LCD displeja som sa rozhodol použiť tri posuvné registre 4094, ktoré dokážu ovládať 3 číslice LCD displeja. Vďaka tomu sú potrebné na riadenie len 4 vodiče. Rozmery a rozmiestnenie pinov displeja je možno vidieť na Obr. 2. Priradenie jednotlivých pinov ku segmentom displeja je zrejmé z Obr. 3. Pin z označením BP je spoločná elektróda všetkých segmentov.

Zobrazenie jednotlivých segmentov sa realizuje striedavým napätím amplitúdy 5Všš a frekvencií 30Hz na daný segment oproti spoločnej elektróde BP.



Obr. 2. – Špecifikácia rozmerov a pinov LCD displeja

SEG.	3A	3F	3G	2B	2A	2F	2G	COL	1B	1A	1F	1G	NC	NC	NC	NC	NC	LOW BAT	:	BP
PIN#	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
SEG.	BP	–	K	NC	NC	NC	NC	DP1	1E	1D	1C	DP2	2E	2D	2C	DP3	3E	3D	3C	3B
PIN#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Obr. 3. – Priradenie segmentov jednotlivým pinom displeja

Na riadenie posuvného registra sú použité signály CLK, DATA a STROBE. Vstupné dáta privádzame na pin DATA, kde sú snímané nábežnou hranou CLK hodinového signálu. Po vyslaní všetkých dát môžeme aktivovať výstupy posuvných registrov privedením signálu STROBE do log. 0.

## 2 POPIS PROGRAMU

Program v jazyku VHDL je písaný systémom stavových automatov.

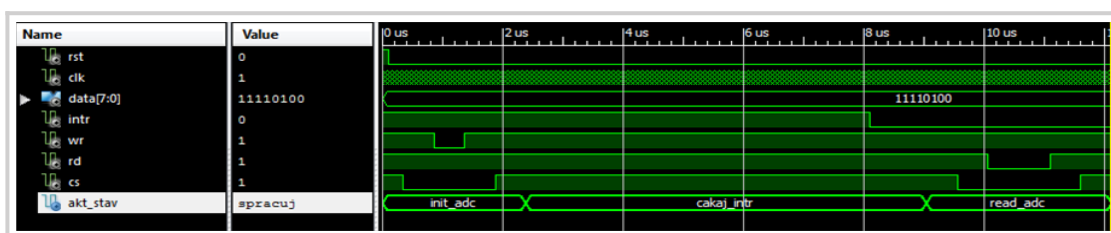
Program obsahuje 6 stavov:

- INIT\_ADC
- CAKAJ\_INTR
- READ\_ADC
- SPRACUJ
- ZOBRAZ
- CAKAJ

Tieto stavy sa prepínajú na základe postupnosti vykonávania daných úloh.

**Kompletný program v jazyku VHDL si môžete prezrieť v nasledovnom odkaze: <http://pastebin.com/FZ6tnkk3>**

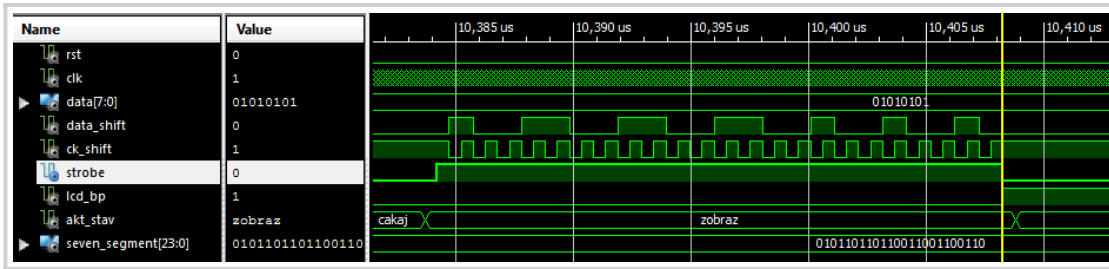
Po signále reset sa nastaví ako východiskový stav INIT\_ADC, ktorý zabezpečí správne vyslanie riadiacich signálov WR a CS pre spracovanie vzorky AD prevodníka. Po vykonaní tohto stavu sa nastaví nasledovný stav na CAKAJ\_INTR, ktorý sa vykonáva dovtedy, pokiaľ sa vstupný signál INTR nenastaví do log. 0, čo nás informuje o skončení prevodu. Po príchode tohto signálu sa ako nasledovný stav nastaví READ\_ADC, ktorý zabezpečí vyslanie riadiacich signálov RD a CS a následne prečítanie hodnoty AD prevodníka pomocou dátových vodičov D0-D7. Vykonávanie tejto sekvencie je zobrazené na Obr. 3.



Obr. 3 – Sekvencia čítania s AD prevodníka

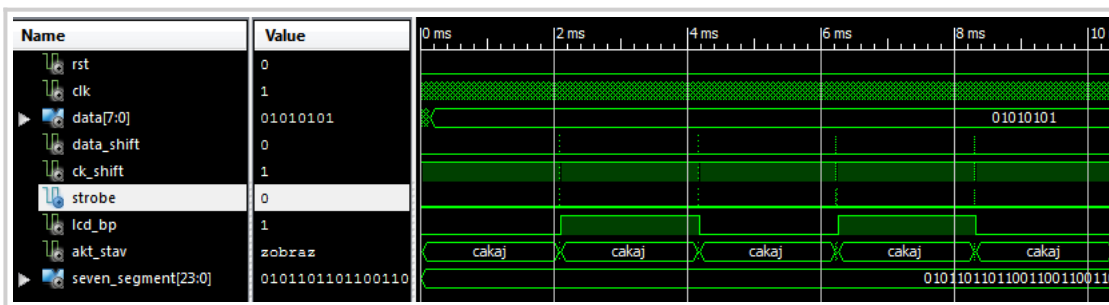
Po skončení tohto stavu nasleduje stav SPRACUJ, ktorý najskôr prevedie 8-bitové binárne číslo do BCD kódu a následne skonvertuje BCD kód do kódu pre riadenie 7-segmentového displeja. Keď sa táto operácia dokončí nastaví sa stav ZOBRAZ, ktorý sa stará o zobrazenie dát na 7-segmentovom displeji.

V stave ZOBRAZ sa najskôr nahrávajú dáta do posuvného registra. Po ich nahraní sa aktivuje výstup posuvných registrov privedením signálu STROBE do log. 0 a signál LCD\_BP sa nastaví tak, aby u segmentov, ktoré sa majú zobraziť bola negovaná logická úroveň. Nasledovný stav sa nastaví na CAKAJ, ktorý zabezpečí čakací čas približne 2ms. Túto sekvenciu možno vidieť na Obr. 4.



Obr. 4 – Sekvencia zápisu do posuvných registrov

Po ich uplynutí sa znova nastaví stav ZOBRAZ, teda znova sa vyšlú dáta do posuvných registrov, tento krát však negované nakoľko musíme zabezpečiť budenie striedavým napätím, teda aj signál LCD\_BP sa zneguje. Tento proces sa opakuje zhruba 200ms. Po tomto čase sa stav nastaví na INIT\_ADC čím sa celý proces znova zopakuje. Na Obr. 5 možno vidieť proces budenia LCD displeja, pomocou signálu LCD\_BP s periódou zhruba 2ms.



Obr. 4 – Budenie LCD displeja

## Fotografia hotového výrobku

