

Světelné noviny na FPGA a FITkitu

projekt do předmětu IVH 2020 / 2021

Změny:

12. 4. úprava třetího projektu

14. 5. odstranění zmínek o klávesnici

Zadání

Implementujte světelné noviny na maticovém displeji pouze s využitím FPGA. Aplikace se bude ovládat pomocí vestavěné vnitřního stavového automatu. Zobrazování bude realizováno pomocí maticového displeje 16x8 bodů, který bude možné zapůjčit. Na displeji se zobrazí obrázek, který se bude posouvat (rotovat) doleva či doprava. Směr se bude po třech iteracích měnit a na závěr proběhne jednoduchá animace

Odevzdávání

Celý projekt je nutné odevzdat do **19. května 2021**; jednotlivé podúkoly do termínů uvedených níže.

Ovládání

bez ovládání uživatelem, dokola se bude pouštět tato smyčka

- 3x rotace vašeho obrázku doprava (tj. 3x16 kroků)
- 3x rotace vašeho obrázku doleva (tj. 3x16 kroků)
- spuštění vlastní animace

smyčka by měla proběhnout **do jedné minuty**.

Specifikace

Při implementaci je možné použít nebo se inspirovat libovolným kódem z repozitáře projektu FITkit. Při přednáškách předmětu IVH budou všechny nutné prerekvizity vysvětleny.

Aplikace nebude mít žádný centrální řídicí prvek, ale bude obsahovat soustavu 128 buněk, které budou navzájem propojeny a které spolu budou lokálně komunikovat. Každá buňka bude mít na starosti řízení svitu jedné LED diody maticového displeje. Chování jednotlivých součástí je popsáno následovně (jedná se o jednu z možných variant).

Buňka

- synchronní prvek
- stav buňky bude vyjádřen pomocí jednoho bitu (svítí / nesvítí)
- signály entity (+ případně další dle potřeby)
 - CLK - hodinový signál

- RESET - resetovací signál
- STATE - výstupní stav
- INIT_STATE - vstupní stav
- NEIGH_LEFT - stav souseda nalevo
- NEIGH_RIGHT - stav souseda napravo
- DIRECTION - posun vlevo nebo vpravo - typu DIRECTION_T (viz úkol 1)
- EN - povolení provedení akce posunu

Ovladač chování

Navrhněte patřičný stavový automat.

Časovač

Pro generování povolovacích signálů, použijte generický zápis a správně vypočtěte, jakou hodnotu pro jakou frekvenci změn potřebujeme. Detaily viz úkol 2.

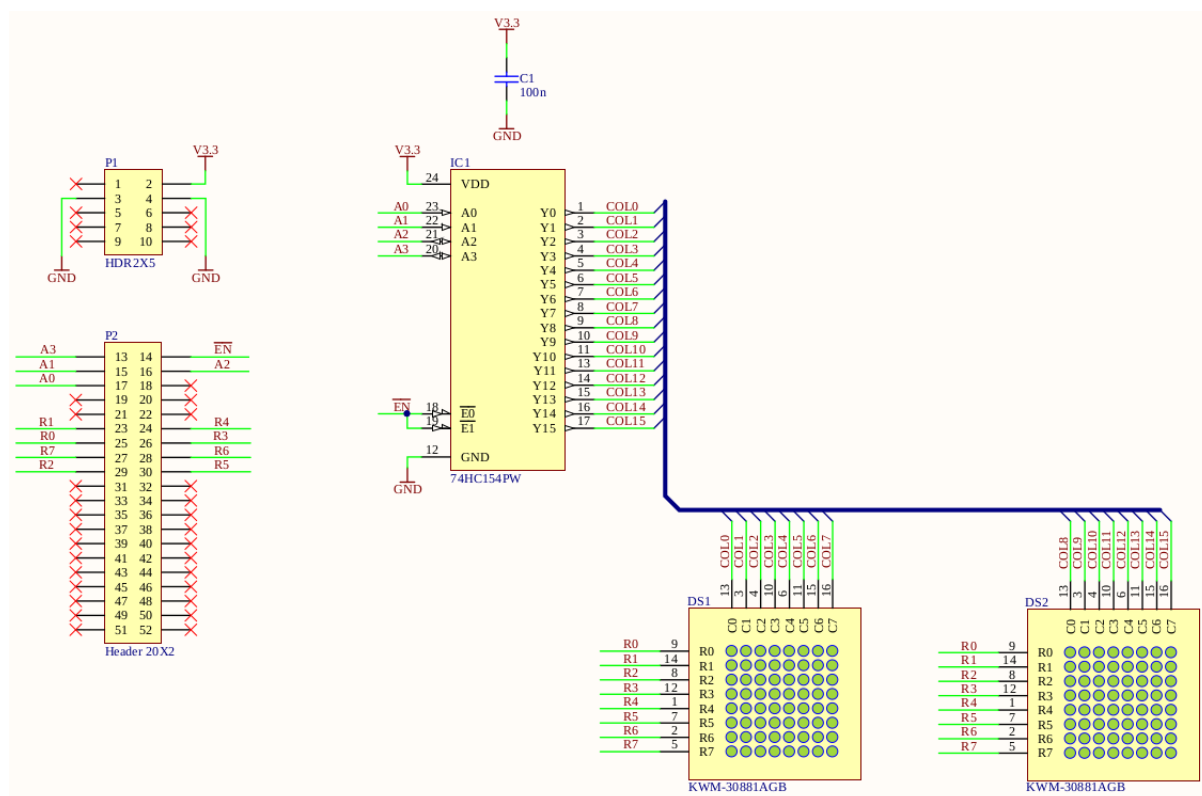
Výstupní řadič

Je použit pro řízení maticového displeje.

Maticový displej má vstupy:

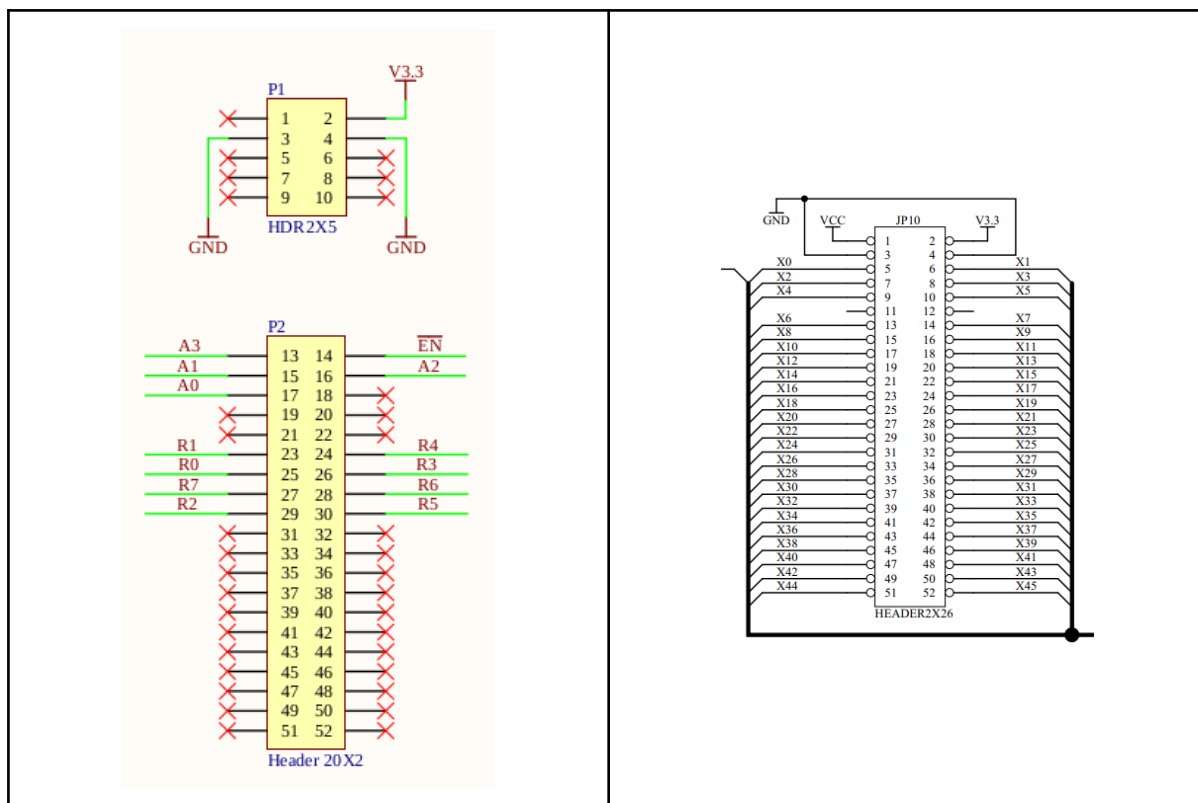
- A0-A3: 4bitová identifikace sloupce, se kterým se pracuje - řádky jsou propojeny přes hardwarový dekodér 74HC154W (4-to-8 decoder) - plníte čísla 000 až 111
- R0-R7: 8bitová hodnota udávající, který řádek (ve vybraném sloupci) má svítit

Zobrazování využívá tzv. multiplexingu, neboli zobrazování jednotlivých sloupců za sebou. V jednom taktu nastavíte A3-A0 na 000 a R7-R0 na požadovanou konfiguraci a určitou dobu počkáte (zvolte vhodnou obnovovací frekvenci, aby displej neblíkal, ale byl čas pro rozsvícení LED). Pak nastavíte A3-A0 na 001 a R7-R0 na (jinou) hodnotu. A tak dále. Periodicky tento děj opakuje.



Pro propojení na fitkitu využijte mapování vektorů pro adresu sloupce a řádku na výstupní port X. Vycházejte ze schématu zapojení FITkitu.

| | |
|-----------|------------|
| Přípravek | FitKit 2.0 |
|-----------|------------|



Animace

Tato funkce může být implementovaná pomocí speciálního signálu do buňky (povolení animace) a např. generické proměnné, která říká, jestli buňka bude při animaci svítit, nebo bude zhaslá. Příkladem animace je to, že se např na 1 sekundu zobrazí “šachovnice” nebo “sudé řádky budou svítit”.

Dokumentace

Součástí řešení je stručná dokumentace, obsahující zejména odůvodnění použití různých konstant (zejména pro časovače). Dále by měla obsahovat výstup (screenshot) simulace, kde budou zvýrazněny určité stavy a v textu zdůvodněno, jestli se komponenta chová správně. Součástí dokumentace bude odkaz na vámi natočené video funkční implementace s komentářem o délce max 2 minuty. Pro uložení můžete použít např. <http://nextcloud.fit.vutbr.cz/>, youtube.com, drive.google.com, atd.

Hodnocení

Projekt bude hodnocen následovně:

- | | |
|------------------------------|------------|
| • První podúkol | až 10 bodů |
| • Druhý podúkol | až 10 bodů |
| • Třetí podúkol | až 25 bodů |
| • Funkčnost posuvů | až 35 bodů |
| • Funkčnost výstupní animace | až 10 bodů |
| • Další rozšíření projektu | až 10 bodů |

Hodnocení bude zohledňovat funkčnost, efektivitu, přehlednost, dodržení zvyklostí jazyka VHDL a případná implementovaná rozšíření nad rámec zadání. V případě nejasností může být vyžadována osobní prezentace řešení.

Podúkoly

1. Podpúrné funkce

Navrhněte a odevzdejte do WISu implementaci balíku `matrix_pack` v souboru `matrix_pack.vhd`, který bude obsahovat

- funkci GETID, která bere na vstupu čtyři čísla typu integer: X, Y, ROWS, COLS s tím, že bude provádět výpočet adresy použité pro indexaci vektoru o délce ROWS * COLS (neboli mapování souřadnic 2D pole do souřadnic 1D pole). Vrací tedy čísla 0 až ROWS * COLS - 1. Pozor! Pokud bude X mimo limit (<0 nebo >= COLS), funkce vrací adresu prvku z druhého konce (t.j. pro X = -1 se má vrátit hodnota COLS - 1). Pro vstupy Y se očekává analogické chování. Očekávejte, že hodnoty ROWS a COLS budou nenulové, ne nutně však stejně velké.

Ukázka mapování pro COLS=3 a ROWS=2. Pozor, kde je X a kde Y!

| | | |
|---|---|---|
| 0 | 2 | 4 |
| 1 | 3 | 5 |

- Výčtový typ DIRECTION_T, který bude obsahovat směr doleva (DIR_LEFT) a doprava (DIR_RIGHT)

Dále odevzdejte soubor `matrix_pack_tb.vhd`, který bude pomocí konstrukce ASSERT testovat správnou funkci funkce GETID.

Termín: 15. 3. 2021

2. Čítač / časovač

Navrhněte a odevzdejte do WISu implementaci časovače (`counter.vhd`) a jeho testbench (`counter_tb.vhd`), který bude testovat jeho funkčnost. Kostru entit stáhněte z dokumentového skladu WISu.

Čítač bude mít 2 generické parametry: frekvenci hodinového signálu (CLK_FREQ) a výstupní frekvenci (OUT_FREQ) (obě dvě zadané v Hz). Čítač s frekvencí odpovídající OUT_FREQ (t.j., např 2x za sekundu) aktivuje na jeden hodinový cyklus signál EN.

Napište testbench, který otestuje, že čítač opravdu za zvolený časový interval aktivuje signál EN. Testbench musí obsahovat funkci ASSERT pro testování. Počítejte s tím, že při zkoušení poběží testbench 100 ms (tj výstupní frekvence musí být větší než 10 Hz).

Termín: 5. 4. 2021

3. Maticový displej

Vytvořte samostatný projekt (spustitelný přes QDevKit), který bude na zapůjčeném přípravku s dvěma maticovými displeji zobrazovat vaše iniciály (bez diakritiky). Kostru projektu stáhněte v dokumentovém skladu, neměňte entitu display (maximálně můžete přidat pro účely testování generické parametry, které budou mít definované výchozí hodnoty).

Iniciály si definujte sami ve 128-bitovém vektoru DATA. Napíšete “Výstupní řadič” popsany výše. Je nutné použít tzv. multiplexing - budete postupně vybírat sloupec, se kterým pracujete a budete nastavovat 8-bitový registr řádků.

Pro řízení časového děje byste měli využít čítač, který jste navrhli ve druhém úkolu. Pokud nebyl správně, je nutné jej opravit. Odevzdávejte soubor [display.zip](#), který bude obsahovat

- složku proj3 obsahující projekt pro QDevKit - **neodevzdávejte složku build!, je však nutné vložit soubor project.xml.**
- soubor display_tb.vhd, který bude testovat váš display (není nutné, aby tam byly asserty)
- soubor **doc.pdf**, který bude obsahovat
 - screenshot/fotka funkčního řešení (fyzického)
 - screenshot simulace - komentovaný - s využitím display_tb.vhd
 - zdůvodnění výpočtu časových konstant (každá hodnota čítače musí být odvozená)

Zobrazovaný vzor definujte pomocí signálu o 128 bitech, přičemž jednotlivé bity budou odpovídat mapování popsáném v případě funkce GETID.

Termín: 27. 4. 2021

Konzultace

Všechny potřebné informace k řešení projektu se dozvíte na přednáškách - některé přednášky budou organizované jako demonstrační cvičení, kde si na příkladu detailně ukážeme, jak při řešení projektu postupovat.

Další dotazy je možné řešit po přednášce nebo online po domluvě předem s Vojtěchem Mrázkem (mrazek@fit.vutbr.cz)