

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií



IVH
Dokumentace k projektu

Hodiny/ Stopky

Obsah

| | |
|--------------------------------------|---|
| 1. Popis zvoleného problému | 3 |
| 2. Popis navrženého řešení | 3 |
| 1. Specifikace časové jednotky | 3 |
| 2. Realizace času | 3 |
| 3. Rom paměť | 4 |
| 4. Řadič VGA | 5 |
| 5. Řízení z klávesnice | 5 |
| 6. Ovládání/Souhrn | 5 |
| 3. Foto dokumentující činnost | 6 |
| 4. Schéma zapojení | 8 |
| 5. Metrika kódu | 9 |
| 6. Vyhodnocení | 9 |

1. Popis zvoleného problému

Cílem tohoto projektu je vytvořit aplikaci pro FPGA ve VHDL Předmětem mého řešení je implementace hodin ve VHDL využívající VGA. Tento projekt přímo vychází z příkladu z přednášek. Hodiny budou realizovány za pomoci blokové grafiky, jejíž zobrazování bude realizováno přes paměť programu ROM, která bude obsahovat definice jednotlivých číslíc využívané pro zobrazení času. Hodiny bude také možné nastavovat z klávesnice přípravku FITkit, nebo přes příkazovou řádku obsluhuje aplikace QDevkit.

2. Popis navrženého řešení

1. Specifikace časové jednotky

Jelikož pro řešení projektu, je nezbytné nějakým způsobem měřit čas, je třeba specifikovat jednotku, ve kterých bude řešení ideální, jelikož předmětem projektu jsou hodiny, tak bude ideální sekunda. Tuto jednotku vytvoříme ze signálu CLK, kdy se budeme snažit přiblížit hodnotě 1Hz, který zhruba odpovídá jedné sekundě. V našem případě je tato funkčnost řešena čítačem typu integer, který dělí vstupní frekvenci hodin právě na 1Hz, tím dojde ke korektnímu určení základní časové jednotky pro hodiny.

2. Realizace času

Časová jednotka je již specifikovaná, nyní bude třeba změřit násobky základní časové jednotky, tj. minuta, hodina apod. Takovou realizaci implementujeme za pomoci registrů. Bude třeba řídit každou číslici zvlášť, proto je třeba naimplementovat registr pro sekundy, čítající jak sekundy samotné, tak desítky sekund, jakmile čítač načte 59 sekundu dojde k vygenerování povolovacího signálu ke spuštění čítače minut. Obdobný postup bude aplikován i u desítek minut a hodin, či desítek hodin a poté reset. Dále je potřeba ošetřit čítač hodnot, jelikož minuta nemá 60 sekund, nýbrž 59 a ta jako kdyby 60 sekunda je ta „nultá“. tj. hodiny nikdy namísto desítek nebude mít číslo 6. Toto ošetření se aplikuje i na minuty, hodiny apod. U řádů hodin je třeba ještě uvažovat, že hodiny sice mají 24 hodin, ale počítáme do 23:59:59 a poté 00:00:00. Tady se situace trochu komplikuje, jelikož u hodin je možné počítat až do hodnoty 9 pokud desítky hodin jsou 0 nebo 1, pokud by byli totiž 2 tak u hodin může nastat maximální hodnota pouze 3. Toto ošetření bude nutné také aplikovat v editačním režimu, aby nedošlo k nastavení hodnoty, jejíž hodnota pozbývá smysl.

Hodnoty které aktualizuje registr, jsou posílány na rozhraní VGA řadiče, který tyto hodnoty zobrazí na výstup.

3. Rom paměť

Aby bylo možné čísla zobrazovat na VGA port, bude třeba číslíce někam uložit. K takovému účelu nám bude sloužit paměť programu ROM. Která se skládá z dvourozměrného pole hodnot s výčtem jednotlivých hodnot. Pro implementaci byly zvoleny hned dvě ROM paměti, jedna pro samotné číslíce a ta druhá pouze pro oddělovač sekund, minut a hodin, aby se vše zjednodušilo.

Blokové schéma pamětí pro zobrazení číslic viz obrázek níže.

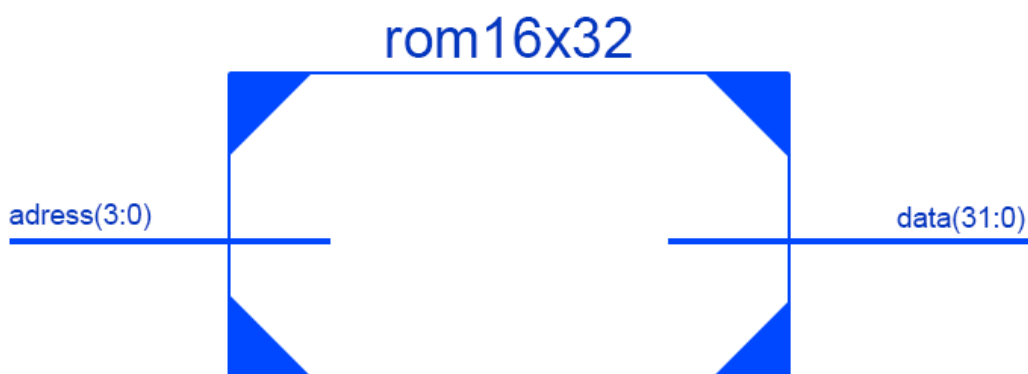
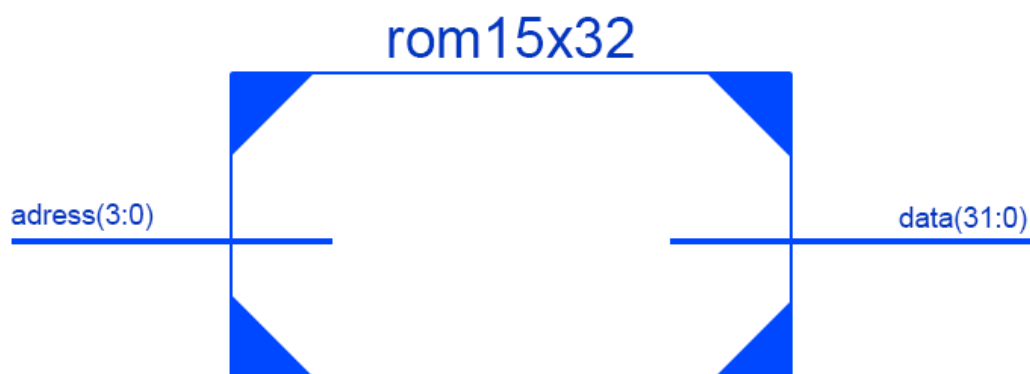


Schéma paměti použité pro oddělovač.



4. Řadič VGA

Stará se o zobrazení načtených hodnot v ROM na obrazovku. Aby bylo možné vůbec možné hodnoty posílat na výstup, je třeba VGA řadič nejprve nastavit, tj. kolik řádků se bude vykreslovat, rozlišení výstupního obrazu, dále horizontální a vertikální synchronizaci apod. v našem případě byl zvolen mód 640x480px k němuž je nutné taktování na 25mHz a obnovovací frekvence 60Hz.

Pro zobrazování jednotlivých číslic z paměti ROM je využito 2x drsnější zobrazení. Barvy jsou zakódovány na dvou bitech a za pomoci multiplexoru posílány na řadič VGA. O pozici tj. o umístění dané číslice na VGA displeji se stará opět multiplexor.

5. Řízení z klávesnice

Řízení hodin realizujeme za pomoci keyboard kontroléru, který snímá maticovou klávesnici. Jelikož kontrolér nám interpretuje jednotlivé číslice a znaky z klávesnice za pomoci 16-bitové sběrnice, musíme jednotlivé číslice tj. bity nastavené do log.1 převést na odpovídající číselnou hodnotu, jaká byla stisknuta na klávesnici. Toho docílíme pomocí multiplexoru, který vždy na výstup nastaví signál typu integer s odpovídající hodnotou dané číslice.

Aby bylo možné naše hodiny z klávesnice nastavovat, musí být nastaven signál přerušení di log.1, který se vygeneruje jakmile dojde ke stisknutí hvězdičky (*) na třetím bitu sběrnice maticové klávesnice. Pokud dojde k takovému přerušení, dojde k resetování hodin do počátečního stavu tj. 00:00:00 a dojde ke změně barvy aktuálně editované číslice. Jakmile je daná číslice editovaná můžeme se přesunout na další číslici pomocí stisku tlačítek A tj. doprava nebo tlačítkem B doleva.

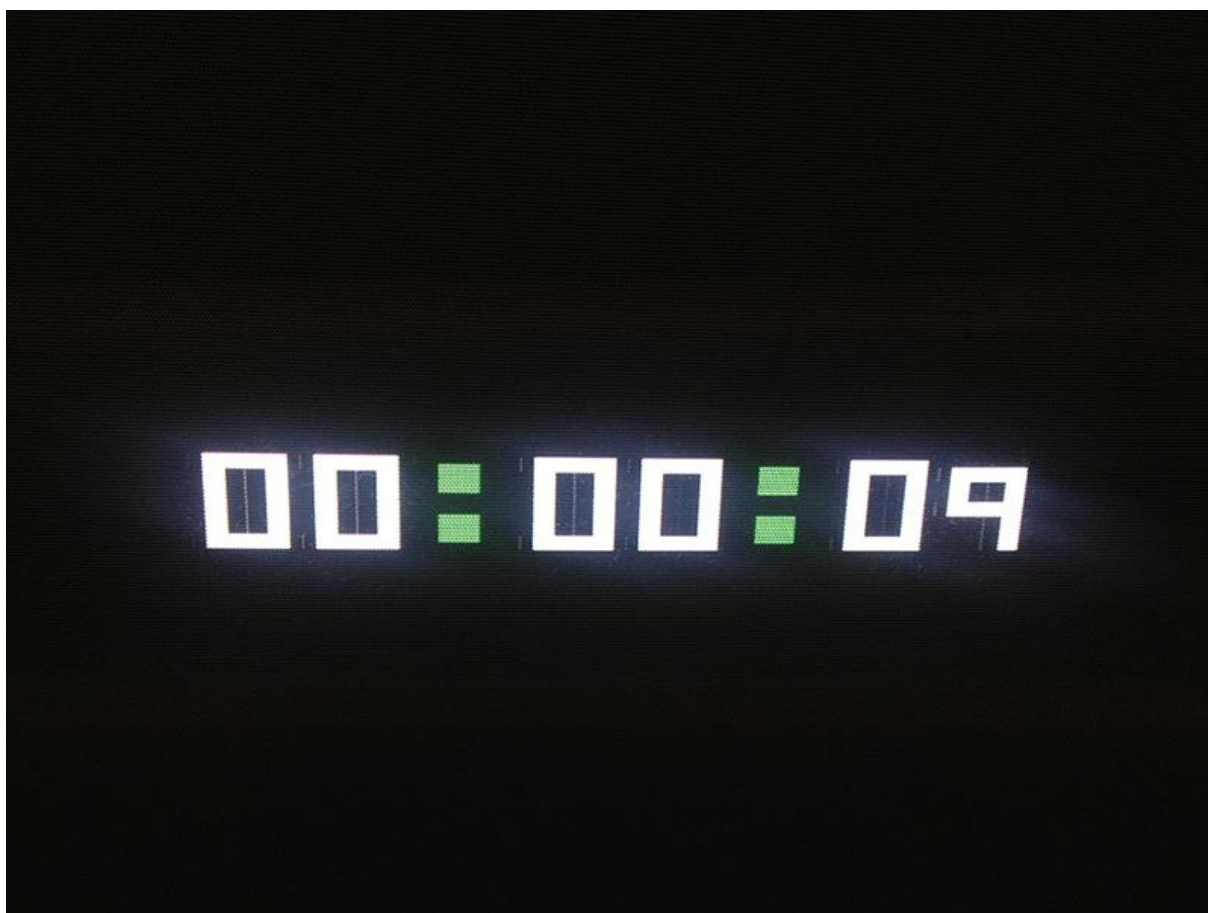
Takto je možné editovat všechny číslice hodin a po nastavení požadovaného času se čeká na potvrzovací signál opět z klávesnice. V případě vygenerování povolovacího signálu se hodiny spustí s nastavenými hodnotami.

6. Ovládání/Souhrn

Jak již bylo zmíněno, ovládání hodin je realizován skrze klávesnici přípravku FITkit. V případě, editace číslic, je třeba se dostat do editačního režimu hodin. Do editace se dostaneme skrze hvězdičku po stisku na klávesnici, po stisku hvězdičky zároveň dojde k asynchronnímu resetu všech číslic a dojde k podsvícení aktuálně editované číslice. Jakmile jsme editovali číslici, posun na další/zpět provedeme pomocí tlačítek A/B, kde A slouží pro posun doprava na displeji a B pro posun doleva. V případě že jsme s editací hotovy a chceme hodiny spustit, stačí zmáčknout klávesu křížek (#). Hodiny je možné resetovat stejným tlačítkem i v editačním režimu aniž by se spustily (v našem případě hvězdička).

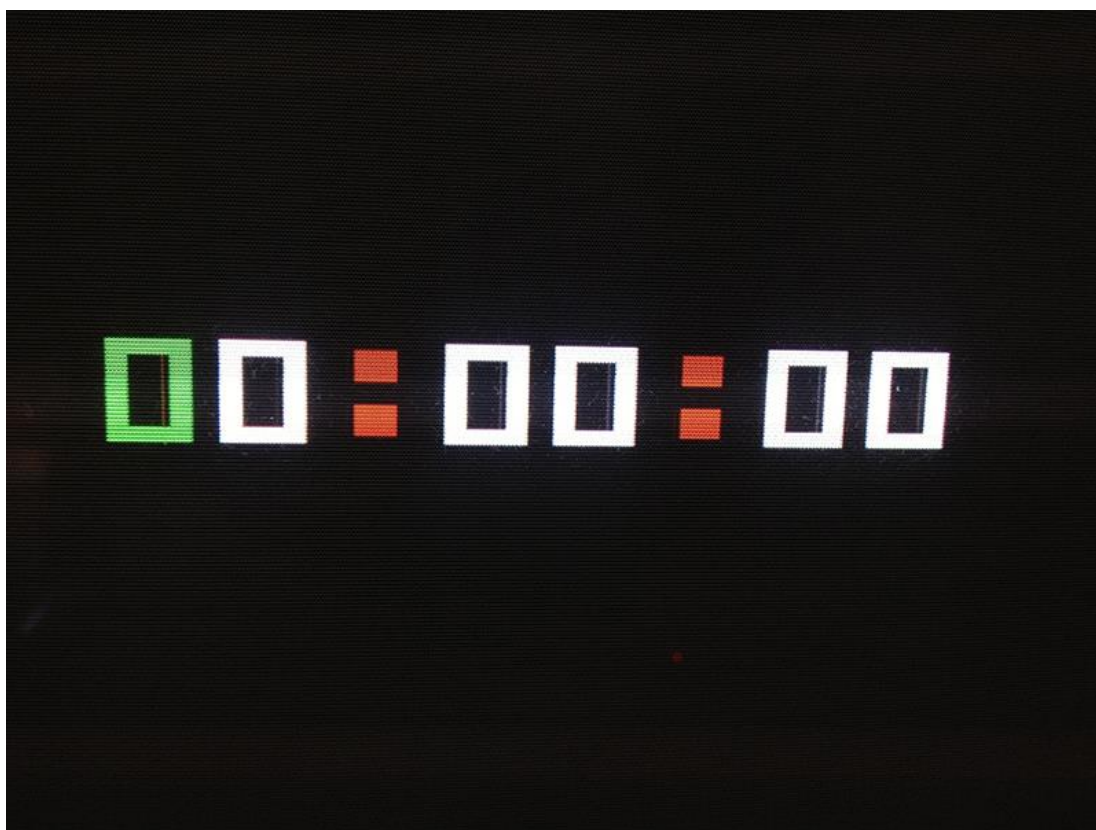
3. Foto dokumentující činnost

Při spuštění hodin aplikací Qdevkit dojde ihned ke spuštění času. Pro názornost rozlišujeme editační režim od běhu hodin zbarvením oddělovače, pokud je zbarvený do zelena znamená to, že hodiny jsou aktivní, při stisku klávesy * se zbarví do červena a vyznačí se editovaná číslice, což hned indikuje uživateli, že hodiny edituje.

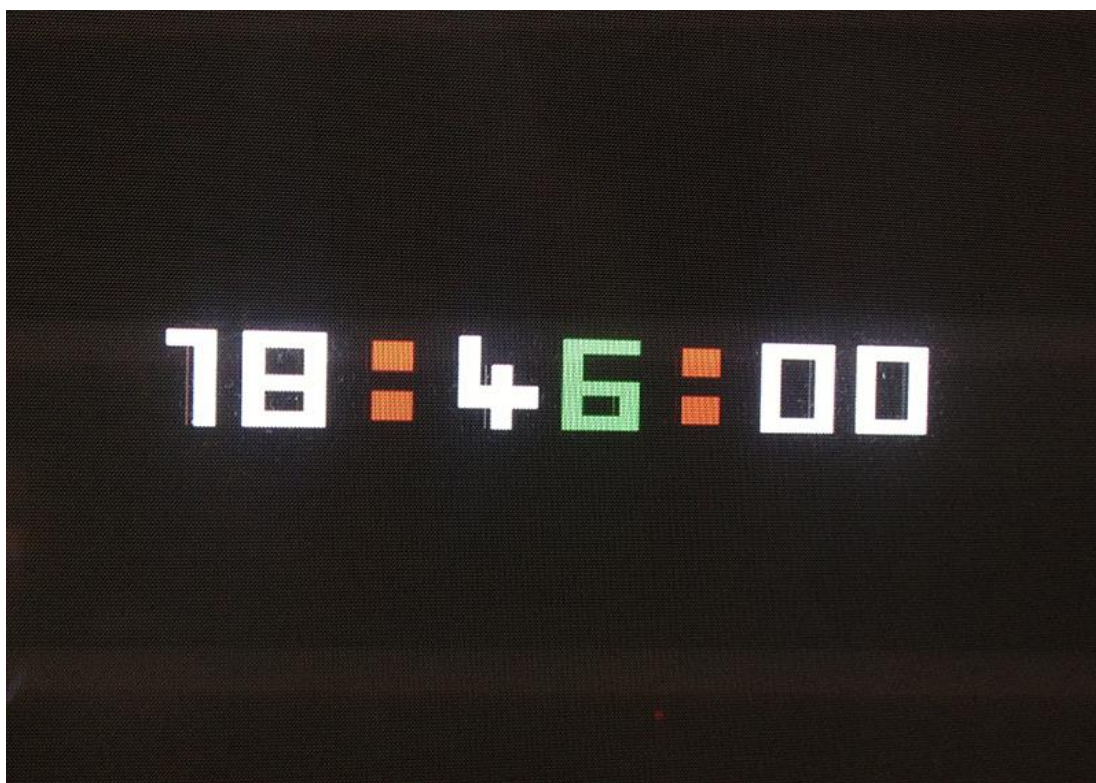


Obr.1 – Ukázka hodin.

Jak již bylo zmíněno výše, editační režim vypadá následovně:



Obr.2 – Editační režim.



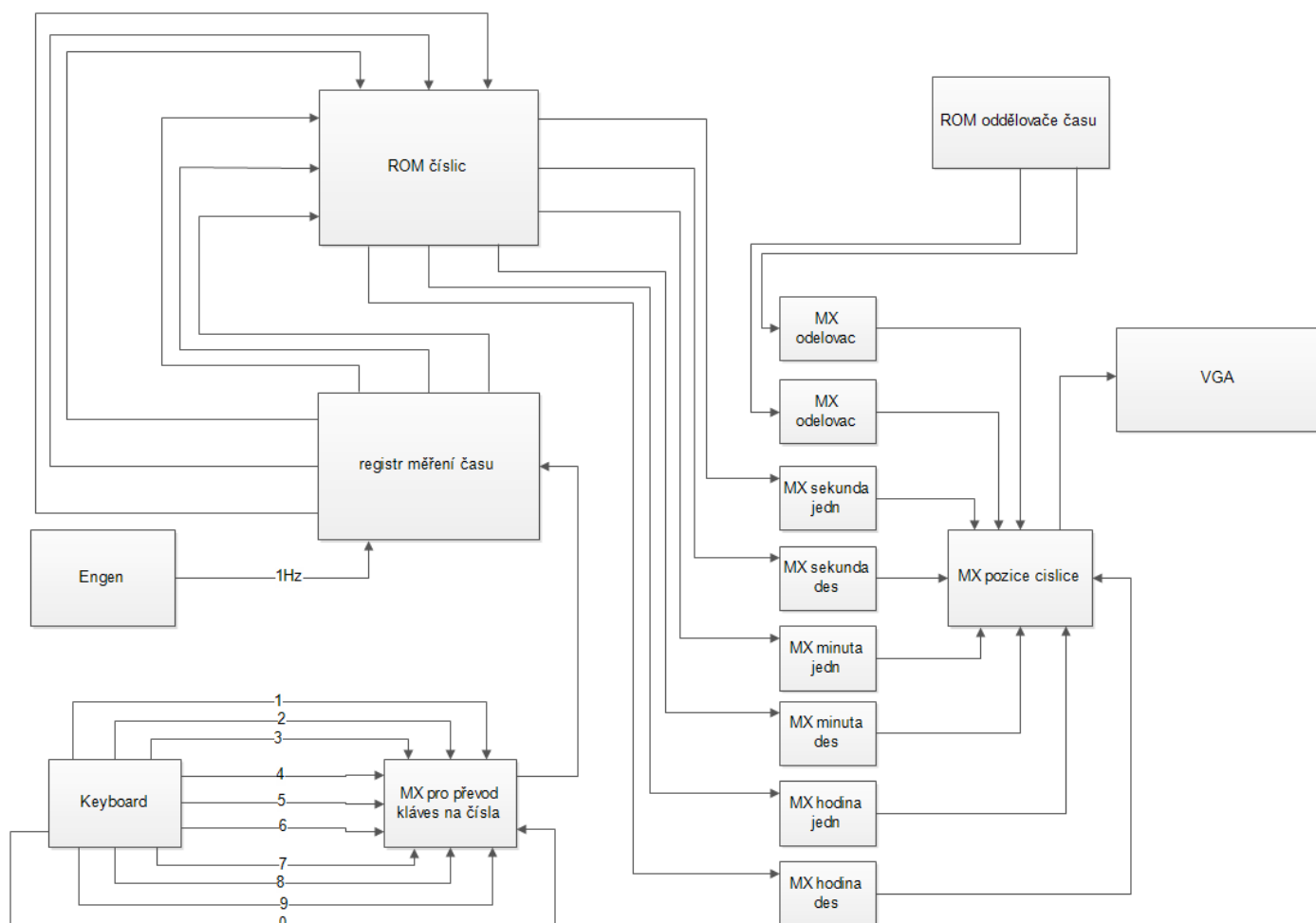
Obr.3 – Ukázka funkčnosti editačního režimu.

4. Schéma zapojení

Základním stavebním prvkem je registr engen, který dělí frekvence na 1hz ~ 1s. Registr pro měření času jej poté využívá jako nutnou podmínku pro čítání hodnot jednotlivých signálů, které řídí hodiny. Na tento registr je ještě napojen keyboard, tj. při vstupu z klávesnice dojde k postupnému přenastavení řádů sekund, minut a hodin. Registr pro měření času těmito signály řídí výběr dané číslice z paměti ROM, kde jsou číslice předdefinovány. Dochází k načítání jednotlivých číslic z paměti ROM na 2 bitovém signálu a za pomoci multiplexorů předávány na výstup VGA.

Pro implementaci je zapotřebí 6 multiplexorů pro každou číslici zvlášť, tj. řád jednotek sekund, řád desítek sekund apod. a dále multiplexor pro oddělovač hodin, v našem případě to je klasicky dvojtečka. Výstup těchto multiplexorů je napojen na další multiplexor pro korektní určení sloupce a řádku na displeji, kde se má daná číslice/oddělovač zobrazit.

Popis zapojení viz obrázek níže:



5. Metrika kódu

Celkový počet fpga souborů

- 2

Zdrojové soubory

- engen.vhd – 36 řádků.
- top.vhd – 604 řádků.

6. Vyhodnocení

Implementace hodin ve VHDL vycházela z podkladů dodávané z přednášek, které umožnily velice efektivně pracovat ve VHDL a pochopit a naučit se základní principy pro práci s řadičem VGA či keyboard kontrolérem. Díky tomu byla implementace zadaného projektu na přijatelné úrovni. Pro jednoduchost byla ROM rozdělená na paměť samotných číslic a paměť pouze pro dvojtečku jako oddělovač řádu hodin, minut a sekund což velice zjednodušilo implementaci.

Během implementace jsem narazil na řadu problémů, co se týče synchronizace klávesnice. Tyto problémy jsem ale postupem času odstranil a přišel i s daleko efektivnějším návrhem celého problému. Po delším zvažování jsem se ale nakonec rozhodl již své řešení neměnit. Při testování hodin jsem nenarazil na žádnou zásadní chybu, a tudíž považuji implementaci hodin za korektní.