



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**JEDNODUCHÁ VIDEOHRA NA BÁZI REKONFIGURO-
VATELNÉ PLATFORMY FPGA**

SIMPLE VIDEO GAME BASED ON RECONFIGURABLE FPGA PLATFORM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ADAM KUBA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2020

Zadání bakalářské práce



Student: **Kuba Adam**
Program: Informační technologie
Název: **Jednoduchá videohra na bázi rekonfigurovatelné platformy FPGA**
Simple Video Game Based on Reconfigurable FPGA Platform
Kategorie: Vestavěné systémy

Zadání:

1. Seznamte se s vývojovým kitem na bázi obvodu FPGA dle pokynů vedoucího a jazykem VHDL pro návrh číslicových obvodů.
2. Prostudujte existující projekty pro zvolenou platformu. Pozornost věnujte zejména možnostem obsluhy rozhraní HDMI a připojení klávesnice či myši.
3. Po dohodě s vedoucím zvolte vhodný typ hry. Identifikujte její části vhodné pro realizaci v MCU a FPGA na cílové platformě. Na úrovni blokových diagramů demonstруйте funkční závislosti těchto částí.
4. Proveďte implementaci zvolené hry na úrovni firmware pro MCU a dále vytvořte v jazyce VHDL potřebné hardwarové prvky umístěné do obvodu FPGA.
5. Implementaci videohry doplňte o možnost využití ovládacích prvků typu klávesnice nebo myš, které budou připojené ke zvolené vývojové platformě za použití vhodného rozhraní.
6. Proveďte zhodnocení dosažených výsledků a diskutujte možnosti dalšího rozšíření projektu.

Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 28. května 2020
Datum schválení: 25. října 2019

Abstrakt

Cílem této práce je vytvoření videohry. Cílová platforma je vývojový kit Minerva, který je osazen mikrokontrolérem Kinetis K60 a programovatelným hradlovým polem Spartan 6 od firmy Xilinx. Videohra obsahuje grafický výstup pomocí HDMI, k tomu využívá obvod TFP410, který se stará o generování obrazu. Vstup od uživatele je realizován pomocí klávesnice připojené na pinech kitu. Videohra má sloužit jako ukázková aplikace pro výukový kit.

Abstract

The aim of this thesis is to create video game. The target platform in this thesis is educational kit Minerva, which is equipped by microcontroller Kinetis K60 and Field-programmable gate array Spartan 6 by company Xilinx. The video game includes graphics output via HDMI, using the TFP410 circuit, which takes care of image generation. User input is realized using a keyboard connected to the pins of the kit. The video game is intended to serve as a sample application for a educational kit.

Klíčová slova

Videohra, FPGA, MCU, HDMI

Keywords

Video game, FPGA, MCU, HDMI

Citace

KUBA, Adam. *Jednoduchá videohra na bázi rekonfigurovatelné platformy FPGA*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Jednoduchá videohra na bázi rekonfigurovatelné platformy FPGA

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Adam Kuba
27. května 2020

Poděkování

Chtěl bych poděkovat panu Ing. Václavu Šimkovi za pomoc při řešení problémů.

Obsah

1	Úvod	3
2	Platforma	4
2.1	Výukový kit Minerva	4
2.2	Mikrokontrolér Kinetis K60	5
2.3	FPGA Xilinx Spartan 6	5
2.4	Vývojové prostředky pro platformu	6
2.4.1	Vývojové prostředky pro MCU	6
2.4.2	Vývojové prostředky pro FPGA	7
3	HDMI	8
3.1	Vývoj rozhraní	8
3.2	Vlastnosti HDMI	8
3.2.1	Kompatibilita HDMI a DVI	8
3.2.2	Připojení	9
3.3	Signály a kódování	9
4	Vstupní zařízení	11
4.1	Klávesnice	11
4.1.1	Vývoj klávesnic	11
4.1.2	Technologie klávesnic	11
4.2	Myš	13
5	Reproduktor	14
5.1	Historie	14
5.2	Princip reproduktorů s pohyblivou cívkou	14
5.3	Typy reproduktorů podle frekvence	15
6	Systémové sběrnice	16
6.1	FlexBus	16
6.2	I2C	17
7	Videohry	19
7.1	Historie	19
7.2	Inspirace pro práci	20
8	Řadič rozhraní HDMI	21
8.1	TFP410	21
8.2	Princip řadiče	21

9	Návrh hry	22
9.1	Návrh hry	22
9.2	Schéma propojení jednotlivých prvků	22
9.3	Uchovávání dat o hře v MCU	23
10	Implementace	24
10.1	Pohyb a kolize	24
10.2	Vykreslování	24
10.3	Vstup a výstup	25
10.4	Problémy s vývojovými prostředími	25
11	Možnosti dalšího vývoje	26
12	Závěr	27
	Literatura	28
A	Obsah CD	29

Kapitola 1

Úvod

Výuková platforma Minerva kit slouží studentům k praktické realizaci hardwarových i softwarových projektů. Minerva obsahuje mikrokontrolér, programovatelné hradlové pole (FPGA) a řadu periférií. Mezi periferie obsažené na platformě patří například HDMI, audio kodek, čtečka paměťových karet, AD převodník.

Cílem práce je vytvoření videohry pro výukový kit Minerva. Videohra má sloužit jako ukázka co lze na kitu vytvořit a jakým způsobem se dají využívat jednotlivé periferie platformy. Videohra bude nejspíše sloužit dalším studentům jako ukázkový projekt. Bude kladena snaha o jednoduché zavádění projektu na platformu a dobrou dokumentaci jednotlivých částí kódu.

V následujících kapitolách se seznámíme s výukovým kitem Minerva, hardwarem, který se na ní nachází i jakými prostředky se dá programovat. Blíže se podíváme na propojení mikrokontroléru a FPGA pomocí sběrnice FlexBus. Dále se zaměříme na princip fungování HDMI a jeho řadiče TI TFP410 a to jakým způsobem funguje komunikace mezi ním a FPGA. Podíváme se na hardware, kterým se dají přenášet informace od uživatele do kontrolérů.

Kapitola 2

Platforma

V této kapitole se podíváme na platformu, pro kterou byla vyvíjena videohra. Seznámíme se s platformou jako celkem i podrobněji s jejími jednotlivými perifériemi.

2.1 Výukový kit Minerva

Výukový kit Minerva je samostatný hardware o velikosti 8x13 cm. Minerva byla vytvořena zaměstnanci Fakulty informačních technologií Vysokého učení technického. Vývoj začal v roce 2012. Minerva vznikala jako výkonnější, nástupce již používaných FITkitů verze 1.0, 1.2, 2.0. [1]

Minerva obsahuje mikrokontrolér (MCU) od firmy Freescale Semiconductor, která byla později koupena firmou NXP Semiconductors. Mikrokontrolér je z řady K60, jedná se o 32-bitový jednojádrový ARM MK60DN512ZVMD10. Tento procesor je postavený na technologii ARM Cortex-M4, což je procesor pracující až na frekvenci 100 MHz a má malou spotřebu energie. Má přístup k 512KB FLASH paměti a 128KB RAM paměti.

Na Minervě je ještě jeden mikrokontrolér řady MC9S08. Ten obsahuje pouze 8-bitový procesor s jádrem S08 pracujícím na frekvenci 48 MHz. Tento mikrokontrolér slouží k ladění a debugování ostatních modulů na kitu.

Mikrokontrolér obsahuje hodně modulů, což pomáhá jeho rozšíření. Mezi nejdůležitější moduly patří ethernet (připojení do sítě LAN), USB, UART (jednotka pro asynchronní sériovou komunikaci), I2C (modul pro sériovou komunikaci typu master-slave), FlexBus (slouží k rychlé komunikaci na desce kitu), SDHC (modul pro práci s SD kartou), ADC (analogově-digitální převodník) a DAC (digitální-analogově převodník).

Na Minervě je stěžejní, tak jako na jejích předchůdcích FPGA (Field-Programmable Gate Array), neboli programovatelné hradlové pole. FPGA na kitu je XC6SLX9 řady Spartan 6 od firmy Xilinx. K FPGA je připojena rychlá paměť DDR2. Zároveň jsou do FPGA připojeny sběrnice, jako je například Flexbus pro rychlou komunikaci s MCU, dále sběrnice pro komunikaci s HDMI řadičem, pro přenos dat na audio kodek a další.

Pro výukový kit existují i hardwarová rozšíření RF 2.4 GHz pro bezdrátovou komunikaci a Ethernet modul pro další ethernetové rozhraní kompatibilní s 10Mbit/100Mbit. Minerva na rozdíl od jiných komerčně prodávaných výrobků podobného typu jako je například Raspberry PI neobsahuje operační systém.

2.2 Mikrokontrolér Kinetis K60

Mikrokontrolér MK60DN512ZVMD10 [4] z řady Kinetis K60 je založen na jádře ARM Cortex-M4. Řada K60 se vyznačuje tím, že má nízkou spotřebu energie, přístup pomocí USB, obsahuje i rozhraní pro ethernet a dále má šifrování a detekci manipulace. Mikrokontrolér na kitu Minerva má integrované řadiče pro sériovou komunikaci pomocí rozhraní SPI, I2C a UART. Dále disponuje komunikačními rozhraními CAN (provozní sběrnice) a SDHC, sloužícím pro přístup k SD kartě. Může také zpracovávat analogové signály pomocí dvou převodníků šestnácti bitového ADC a dvanácti bitového DAC. Mikrokontrolér umožňuje použití časovačů jako jsou Watchdog, FlexTimer a RTC. Mikrokontrolér obsahuje i generátor náhodných čísel. Tyto čísla jsou brána z bitového posunu hodinového signálu a jsou statisticky náhodná.

Pro úsporu potřebné energie využívá mikrokontrolér systém Clock Gating, to znamená že modul, který nemá zpřístupněné hodiny je vypnutý, tím pádem nespotebovává energii. Při každém restartu jsou všechny hodiny vypnuté a pustí se jen ty pro které se nastaví bity v registrech SCGCx. Pokud se přistupuje k vypnutému modulu je program ukončen.

Mikrokontrolér má přístup k 512KB FLASH paměti, která je rozdělena na dvě části, protože mikrokontrolér umožňuje vykonávání programu přímo z paměti. Aby nedošlo k poškození paměti při vykonávání programu je zde modul Memory Protection Unit (MPU), který kontroluje všechny přístupy do paměti. Dále se zde nachází interní paměť RAM s velikostí 128KB. Tato paměť obsahuje zavaděče, pomocné proměnné a prostor pro výměnu dat s rozhraními.

Jádro ARM Cortex-M4 je založeno na harvardské architektuře, která se vyznačuje tím, že odděluje paměť programu od jeho dat. Tímto způsobem se dá dosáhnout zrychlení provádění instrukcí. Cortex-M4 je založen na architektuře Armv7-M a pracuje až na frekvenci 100MHz. Jádro umožňuje využívání MAC (Multiply-Accumulate operation) a SIMD (Single instruction Multiple Data) instrukcí. MAC instrukce vezme dvě proměnné ty vynásobí a posléze přičte k třetí proměnné, tyto operace provede v jednom taktu.

Jádro obsahuje Debug Access Port (DAP) pro podporu ladění za běhu. DAP je jedno externí rozhraní, které přenáší přístup k celému systému. DAP umožňuje přístup k registrům a paměti externímu ladicímu nástroji (OSBDM) pomocí rozhraní JTAG a SWD. Pomocí Debug Access Port lze spustit program v ladícím režimu, při kterém lze sledovat vykonávání jednotlivých instrukcí, zápisu do paměti atd.

2.3 FPGA Xilinx Spartan 6

Na kitu Minerva se nachází programovatelné hradlové pole od firmy Xilinx řady Spartan 6.[11] Přesně jde o typ XC6SLX9, tento typ disponuje 1430 slices. Každý tento prvek obsahuje čtyři LUT (Look-up table) a osm klopných obvodů typu flip-flop. Jejich kombinací lze dosáhnout parametrů 90Kb distribuované RAM spolu s 11440 bistabilních klopných obvodů. FPGA obsahuje 32 bloků dedikované RAM, které mají každá velikost 18Kb. Programovatelné hradlové pole také obsahuje 16 DPS Slice, které obsahují násobičku, sčítačku a akumulární registr. DPS jsou cílené na zpracovávání signálů.

Jednotlivé bloky RAM paměti mohou být konfigurovány, jako dvě nezávislé paměti o velikosti 9Kb, nebo jako jedna paměť o 18Kb. Každá RAM může být adresovatelná pomocí dvou portů, ale lze zvolit nastavení pro adresaci pouze jedním portem. Každý blok RAM obsahuje výstupní registr pro zvýšení rychlosti postupného čtení. Celý obsah RAM lze inicializovat nebo vymazat pomocí konfiguračního bitstreamu.

Operace pro čtení a zápis do paměti jsou synchronní. Paměť RAM má dva porty, které mohou být nastaveny každý na jinou operaci, ale přistupují k jedné sdílené paměti, proto jsou dvě možnosti co bude na výstupu když jeden zapisuje. Jedna možnost je, že na výstupu je přesně to co je v paměti to znamená, že to mohou být nekonzistentní data. Druhá možnost je, že na výstupu jsou data, která jsou přepisována, to znamená data před začátkem zápisu.

Pomocí programu Xilinx CORE Generator lze vygenerovat konfiguraci portů, modulů, synchronní FIFO a konvertory pro šířku dat. Možnosti nastavení jednotlivých portů u bloků RAM jsou:

- Single Port
- Simple Dual Port
- True Dual Port

V režimu Simple Dual Port je jeden port nastaven pro čtení a druhý pro zápis. U tohoto typu může datová sběrnice dosahovat šířky 36b, kdežto u single port má šířku pouze 18b. True Dual Port může být nastaven pouze u RAM, které jsou nastaveny jako jedna paměť o velikosti 18Kb. Tento režim umožňuje nastavení obou portů, pro čtení i zápis. Proto když přistupují na stejnou adresu a jeden z nich zapisuje mohou být data nekonzistentní.

Naprogramovat FPGA na Minervě jde dvěma způsoby. První z nich je, že se bitstream nahraje do flash paměti. Protože FPGA je na kit nastaveno tak, že po zapnutí si vždy vezme konfiguraci z této paměti. Takže při změně flash paměti se změní i FPGA obvod. Druhý způsob je pomocí rozhraní JTAG jehož piny jsou vyvedeny na desku kitu. K těmto portům lze připojit Platofm Cable USB 2 od firmy Xilinx a pomocí něj nahrávat do FPGA konfiguraci.

Paměť flash je připojena k programovatelnému hradlovému poli pomocí rozhraní SPI. Velikost paměti je 512KB. Z této velikosti je navíc dostupná jen část, a proto se do paměti dá uložit pouze jedna konfigurace. [2]

2.4 Vývojové prostředky pro platformu

V této části se podíváme na vývojové prostředky jak pro mikroprocesor, tak pro programovatelné hradlové pole.

2.4.1 Vývojové prostředky pro MCU

Firma NXP dodává vývojové prostředí pro své Kinetis MCU. Jedná se o integrované vývojové prostředí Kinetis Design Studio.[10] Je založeno na open source programech Eclipse, GNU Compiler Collection (GCC), GNU Debugger (GDB) a dalších. Základ prostředí je postaven na Eclipse Luna 4.4, proto může jet na operačních systémech Windows, Linux (Ubuntu, RedHat, Centos 7) i Mac OS X. KDS je nástupce prostředí CodeWarrior, který byl vyvíjen od roku 1993, nahradil ho v roce 2014.

Na kitu Minerva je komunikace mezi MCU a vývojovým prostředím pomocí OSBDM, které je založené na mikrokontroléru HCS08. Tento kontrolér je připojen k MCU na ladících portech a k počítači pomocí USB. Ladění lze provádět pomocí různých rozhraní například SEGGER J-Link, PandE Multilink, Command Line (CL) debugging with GDB and OpenOCD TCL.

V KDS se pro programování MCU využívá jazyk C a C++, nemají žádné omezení délky kódu, tyto kódy se před nahráváním překládají pomocí GCC.

V době psaní práce se KDS přestalo vyvíjet, ale pořád je udržované. Protože společnost NXP začala s vývojem nového prostředí MCUXpresso IDE využívající novější verzi prostředí Eclipse.

2.4.2 Vývojové prostředky pro FPGA

Pro vývoj na programovatelné hradlové pole lze využít vývojové prostředí Xilinx ISE.[12] Je problém, že společnost Xilinx ukončila vývoj tohoto prostředí a zaměřila se na vývoj prostředí Vivado Design Suite. Ovšem Vivado nepodporuje FPGA Spartan 6, proto je potřeba sehnat prostředí Xilinx ISE verze 14.7, jeho poslední verze z roku 2013. Toto ISE (Integrated Synthesis Environmen) lze stáhnout na virtuálním počítači pro Virtual Box. Instalovat ho přímo na počítač lze také, ale je to zbytečně komplikované, oproti pouhému stažení virtuálního počítače, který stačí spustit a vše je nastaveno a funguje.

V Xilinx ISE se programují FPGA pomocí jazyka VHDL. Ten se posléze syntetizuje a nahrává do FPGA. Pro nahrávání lze použít program iMPACT, ten je na virtuálním počítači od Xilinxu také. IMPACT vyhledá kabel Platform Cable USB 2, pomocí kterého nahraje přes rozhraní JTAG konfiguraci do FPGA. Součástí Xilinx ISE je i simulátor ISIM a ModelISim, pomocí kterých lze simulovat chování naprogramovaného obvodu.

Kapitola 3

HDMI

V této kapitole se podíváme na rozhraní HDMI (High-Definition Multimedia Interface) a jeho vývoj, vlastnosti.[7]

3.1 Vývoj rozhraní

Vývoj rozhraní HDMI začal na jaře roku 2002 s cílem vytvoření rozhraní na přenos videa a audia, které by bylo zpětně kompatibilní s DVI. DVI bylo v té době nejrozšířenější rozhraní pro přenos obrazu na HD televize. Proto se vývoj HDMI verze 1.0 zaměřil na vylepšení DVI-HDTV. Pokrokem bylo udělán menšího konektoru a přidání podpory přenosu zvuku. Tato verze byla představena v prosinci roku 2002. Od půlky roku 2003, různé firmy podílející se na vývoji, začaly otevírat ATC (Authorized Testing Center). ATC slouží k testování produktů s HDMI rozhraním. HDMI se rychle rozšířilo, protože začalo poskytovat větší rozlišení a obnovovací frekvenci než jeho konkurence, především rozhraní SCART. Poslední verzí HDMI je 2.1.

3.2 Vlastnosti HDMI

HDMI slouží především k přenosu obrazu a zvuku, tento přenos je pouze jednosměrný od vysílače k přijímači. Všechna data se přes HDMI přenášejí nekomprimovaná. Rozhraní je kompatibilní s DVI, lze převádět z HDMI do DVI pomocí redukce. HDMI podporuje velké množství rozlišení. Verze 1.4 podporuje 4K rozlišení a verze 2.1 až 10K s tím, že podporuje 8K na obnovovací frekvenci 60Hz. Pro takto velký přenos dat (49.6Gbit/s) je potřeba začít používat kompresi DSC (Display Stream Compression). HDMI má přístup k EDID (Extended Display Identification Data), pomocí kterého získává podrobné informace o cílovém zařízení. Od verze 1.4 je možnost přenosu internetu pomocí ethernet channel s propustností 100Mbit/s.

3.2.1 Kompatibilita HDMI a DVI

HDMI je zpětně kompatibilní s DVI. Nevyžaduje žádnou konverzi signálů, stačí použití redukce nebo nesymetrického kabelu. Při tomto propojení obraz neztrácí kvalitu, ale nelze takto přenášet audio. Proto se musí zvuk řešit pomocí přídavných kabelů. Konverze z DVI do HDMI může fungovat stejným způsobem, jako z HDMI do DVI, je zde ovšem problém, pokud se do vysílání DVI přidá zvuk může dojít k potížím a obraz se nemusí zobrazovat.

3.2.2 Připojení

Kabely, používané na propojování zařízení přes HDMI, jsou tvořeny čtyřmi stíněnými kroucenými dvoulinkami a dalšími sedmi vodiči. Maximální délka kabelu není standardem definována, protože závisí na výrobě jednotlivých kabelů. Ovšem kabely delší jak 13m mají problém se získáním certifikátu. Delší kabely mají problém se stabilitou projevující se blikáním obrazu. Některé firmy vytváří aktivní zesilovače a opakovače pro propojování více kabelů dohromady. Pomocí aktivních HDMI kabelů lze dosáhnout až 30 metrů. Nebo pomocí převodu na kabel pro ethernet lze dosáhnout 250 metrů, a při převodu na optický kabel až 300 metrů.

Ve specifikaci verze 1.4 vznikl standard pro označování kabelů jako Standard nebo High Speed a kabely s Ethernetem nebo bez něj. Kabely Standard přenášejí maximálně 1080i, kdežto High Speed i 4K s frekvencí 30Hz. Ve specifikaci z roku 2015 pro HDMI 2.0 je další kabel označován jako Premium High Speed pro přenos až 18Gbit/s. V nejnovější verzi 2.1 je další kabel pojmenován jako Ultra High Speed umožňující přenos 4K na frekvenci 120Hz a 8K na frekvenci 60Hz.

HDMI má celkem pět typů konektorů pojmenovaných od A do E. Typy A a B jsou definovány ve specifikaci HDMI 1.0, typ C přišel s verzí 1.3 a typy D a E ve verzi 1.4.

Typ A má 19 pinů a má tím pádem přenosovou šířku pro ultra HD a 4k rozlišení. Tento typ je kompatibilní se single link DVI-D. Typ B je větší a má 29 pinů měl by sloužit k přenosu pro velké rozlišení, jako například WQUXGA (3840x2400) a je kompatibilní s dual link DVI-D. Typ B se nepoužívá protože s příchodem standardu 1.3 se zvýšila rychlost pro single link, nicméně pořád zůstává ve standardu.

Typ C neboli Mini-HDMI konektor je menší než typ A, ale má také 19 pinů. Tento typ má trochu jiné zapojení pinů než typ A, lze ho však použít jako adaptér pro jejich propojení. Tento typ se začal více používat u přenosných zařízení, jako tablety, kamery a navigace.

Typ D je ještě menší než typ C nazývá se Micro-HDMI a je velikostí podobný micro-USB. Má také 19 pinů, ale jejich zapojení se liší od typů A i C.

Typ E má systém (Automotive Connection System) proti vypadání a vibrování, zároveň má obal proti zanesení nečistotami. Pojistka na konektoru se podobá té používané na RJ-45 pro připojení ethernetu.

3.3 Signály a kódování

HDMI data se přenášejí pomocí packetů. Existuje množství typů packetů patří mezi ně obrazové, zvukové, ethernetové a další.

Pro přenos obrazových dat se používají tři kanály pro data a jeden hodinový signál. V každém datovém kanálu se přenášejí data v barvě modrá na kanálu 0, zelená na kanálu 1 a červená na kanálu 2. Hodinový signál má frekvenci, stejnou jako pixel rate přenášeného obrazu. S každým cyklem hodinového signálu se přeneše na každém datovém kanálu deset bitů. Tyto bity jsou kódovány pomocí různých technik.

Data, která přenášíme jsou tři typů. První jsou kontrolní data ty jsou dvou bitová, další jsou data packetu ty jsou čtyř bitová a pak jsou samotná video data ty mají délku osm bitů. Kontrolní a packetová data jsou kódovány pomocí převodových tabulek. Každé kombinaci bitů odpovídá desetibitové slovo. Kódování obrazových dat probíhá ve dvou fázích. V první fázi se osmi bitová data kódují na devíti bitová. Vezme se nejméně významný bit ten se zachová. Potom se udělá nad ostatními bity XOR a XNOR. Z těchto operací se vybere ta, ve které je ve výsledných datech méně změn z nuly na jedničku a opačně. Potom se přidá ze

osmibitové slovo, které prošlo touto operací jeden bit určující, která z logických funkcí byla provedena, nula pro XNOR a jednička pro XOR. V druhé fázi kódování se provádí inverze. Proveďte se pouze pokud je v datech více jedniček jak nul. Ovšem inverze se provádí pouze s osmi bity video dat a ne u indikačního bitu pro operace z první fáze kódování. Pokud byla inverze provedena nastaví se desátý bit na jedničku.

Komunikace tedy probíhá ve třech režimech. Hlavní z nich je pro přenos obrazových dat. Druhý režim je přenos informačních packetů o obrazových nebo jiných typech packetů. Tento režim se ve specifikaci nachází pod pojmem data island. Během tohoto režimu se posílají i HSYNC a VSYNC signály. Poslední režim je řídicí, zde se také přenáší HSYNC a VSYNC. Data v tomto režimu se přenáší vždy mezi dvěma nestejnými packety, to znamená mezi obrazovými daty a informačními daty.

Kapitola 4

Vstupní zařízení

Vstupní zařízení je hardware, který poskytuje data a kontrolní signály pro počítače, nebo zařízení, která je vyžadují. Mezi vstupní zařízení patří klávesnice, myš a joystick, které slouží k ovládání, zároveň sem patří i různé senzory, skenery, čtečky čárových kódů a další.

Vstupní zařízení mohou být zařazeny do tří skupin. První skupina jsou zařízení s jednotným vstupem, neboli vstupuje pouze audio, video atd. Další skupinou jsou zařízení lišící se typem vstupu, jestli je vstup nepřetržitý (myš) nebo oddělený (klávesnice). Poslední skupina je dělení podle počtu vstupních dimenzí (myš má dvě dimenze, snímače pohybu pro virtuální realitu mají šest dimenzí).

4.1 Klávesnice

Počítačová klávesnice je zařízení podobné psacím strojům. Klávesnice se staly hlavní vstupní metodou pro počítače po zastarání děrných štítků. Klávesnice detekují stisknutí jednotlivých kláves a předávají tyto informace řídicímu systému. Dají se také použít pro posílání signálu operačnímu systému počítače.

4.1.1 Vývoj klávesnic

První použití klávesnice u počítače přišlo v roce 1964, když Bell Labs a M.I.T. vytvořili MULTICS. Koncem sedmdesátých let minulého století se začala vyskytovat klávesnice u skoro všech počítačů, kvůli její jednoduchosti použití. První klávesnice byly mechanické a dlouho se vyráběly, byly těžké a většinou neměly ani žádné pouzdro. Takže pro první osobní počítače se začaly předělávat elektronické psací stroje, tak aby fungovali s nimi. Až koncem sedmdesátých let firmy Apple, Radio Shack a Commodore začaly se sériovou výrobou klávesnic. Teprve v devadesátých letech se nahradila mechanická tlačítka membránovými. Tímto se klávesnice staly tišší, lehčí a byly daleko levnější na výrobu.[5]

4.1.2 Technologie klávesnic

Klávesnice se skládají ze tří podstatných částí, jsou to kontrolní procesor, jednotlivé klávesy a připojovací technika.

Jednotlivé klávesy jsou zapojeny do tlačítkové matice. Tato matice je reprezentována, jako obvod s horizontálními a vertikálními vodiči, které jsou propojeny přes tlačítka. Takže když je tlačítko sepnuto, tak se spojí obvod a lze podle toho kudy může procházet proud

určit které tlačítko bylo zmáčknuto. Klávesnice využívají pro tlačítka různé technologie, ke kterým se dostaneme později.

Kontrolní procesor dělá to, že převádí stisk tlačítka na sken kódy, kterým rozumí počítače. Procesory jsou zapojeny do tlačítkové matice, tak že je proud postupně pouštěn do sloupců a postupně přijímán z řádků. Tímto způsobem lze určit, která klávesa byla stisknuta. Toto je náchylné ke zachytávání i krátkých sepnutí ("bouncing"), která uživatel nechtěl. Proto se v procesoru agregují signály a po určitém čase se vybere jedno potvrzené stisknutí.

Klávesnice se připojují k zařízení různými způsoby, první z nich je pomocí kabelu a druhý pomocí bezdrátových technologií. Dříve se pro kabelové připojení používaly konektory AT, PS/2 v dnešní době se ale nahrazují USB konektory. Bezdrátové technologie pro připojení klávesnice jsou nejčastěji pomocí Bluetooth nebo infračerveného záření. Nevýhodou bezdrátových klávesnic je možnost odposlechu komunikace.

V této části se podíváme na některé technologie pro určení stisku tlačítka.

Membránové klávesnice

Membránové klávesnice fungují na principu tří vrstev membrán. Horní vrstva má na horní straně vytištěný znak a na spodní straně vodivý pruh. Pod tou je vrstva, která ji odděluje od nejspodnější vrstvy, tak, aby se nedotýkala a nevytvářely elektrický kontakt. Spodní vrstva má vodivý pruh kolmo k horní vrstvě dohromady tvořící mřížku. Pokud dojde ke zmáčknutí horní vrstvy, vodivý pruh z horní vrstvy propojí obvod v dolní vrstvě a procesor klávesnice pozná, že bylo tlačítko stisknuto. Toto je jeden z nejrozšířenějších typů klávesnic u počítačů. Mají na horní membránu přidělaná plastová tlačítka, pomocí kterých se spojují membrány.

Mechanické klávesnice

U mechanických klávesnic se každá klávesa skládá z celého tlačítka. Každé tlačítko se skládá z pouzdra, pružiny a svorky. Mechanické klávesnice mají tři typy tlačítek jsou to lineární, hmatový a klikací. Výhodou tohoto typu je poměrně dlouhá životnost.

Kapacitní klávesnice

U tohoto typu klávesnice se stisknutím tlačítka změní kapacita kondenzátoru. Pro každou klávesu jsou dva kondenzátory typu D. Při stisknutí klávesy se změní impedance kondenzátorů, toto je lehce detekovatelné. Protože zde nedochází k elektrickému kontaktu, tak zde není potřeba řešit "bouncing". Detekují i to velmi jemné stisknutí klávesy uživatelem, proto lze na těchto klávesnicích psát rychleji. U tohoto typu klávesnic lze nastavovat citlivost pro jednotlivé klávesy samostatně pokud to výrobce podporuje.

Hall-effect klávesnice

Tyto klávesnice fungují na magnetismu. V jednotlivých klávesách jsou umístěny permanentní magnety. Pod klávesami jsou umístěny Hallovi sondy, které reagují na změnu magnetického pole vysláním elektrického signálu. Spínání je bezkontaktní. Toto jsou nejvíce spolehlivé klávesnice používané v jaderných elektrárnách, v kokpitech letadel atd.

4.2 Myš

Počítačová myš je ukazovací zařízení zachytávající pohyb ve dvou osách relativně k povrchu. Toto slouží většinou k pohybu ukazatele na display. Myši disponují ještě dvěma tlačítky a kolečkem pro scrollování. Jsou dva základní typy myši mechanické a optické.

Mechanické myši mají více způsobů zachytávání pohybu, ale nejrozšířenější jsou takzvané kuličkové myši. Tyto myši jsou mechanické, protože obsahují pohyblivé části. Pohybem myši po povrchu se otáčí kulička uvnitř. Z této kuličky se převádí pohyb na kolečka s vyříznutými dírami, přes které svítí diody na světelné senzory. Tyto senzory shromažďují světelné záblesky a převádějí je na vektory pro horizontální a vertikální pohyb. Xerox byla jedna z prvních firem zabývajících se výrobou tohoto typu myši.

Optické myši mají zaměřený zdroj světla na povrch, po kterém se myš pohybuje. Pomocí světelných detektorů, jako například foto-diody, detekují pohyb relativně k povrchu. První typy myši detekovaly pohyb pouze na specifických podložkách. Tyto podložky měly vždy mřížku, pomocí které se dal detekovat relativní pohyb. Novější typy pracují na různých površích, které jsou matné a nemají tak dobrou odrazivost jako například sklo. Specifické myši fungují i na lesklých površích, protože používají mikroskopii temného pole, ale tyto myši nejsou příliš časté a jsou výrazně dražší než obyčejné myši. Jako zdroj světla se u optických myši využívají dva typy zdrojů. První jsou LED používají většinou červenou barvu světla. Druhým typem zdroje světla je laserové, tyto používají infra laserové diody.

Rychlost pohybu myši se měří v jednotkách Mickeys za sekundu. V dnešní době se od této jednotky upouští a používá se DPI, neboli o kolik pixelů se kurzor pohne za palec.

Myši se připojují k počítačům, podobně jako klávesnice dříve pomocí PS/2 nyní více pomocí USB a bezdrátově.

Kapitola 5

Reproduktor

V této kapitole se podíváme na reproduktory. Reproduktor je zařízení, které převádí elektrický audio signál na zvuk. Nejrozšířenější reproduktory jsou s pohyblivou cívkou. Reproduktory se liší tím jakou frekvenci zvuku umí produkovat, podle toho se řadí do kategorií: subwoofer (hodně nízká frekvence), woofer (nízká frekvence), středové (střední frekvence), výšky (vysoké frekvence) a super výšky (nejvyšší slyšitelné frekvence).[3]

5.1 Historie

První reproduktor byl použit v telefonu od Johanna Reize z roku 1861, ten byl schopen produkovat čistý tón a tlumenou řeč. Další reproduktor v roce 1876 použil Alexandr Graham Bell ve svém telefonu. Tento reproduktor už byl schopen produkovat srozumitelnou řeč. Thomas Edison si v té době nechal patentovat systém používající stlačený vzduch pro zvýšení hlasitosti. V roce 1898 vynalezl Oliver Lodge reproduktor s pohyblivou cívkou (dynamický reproduktor). Tento princip se používá v reproduktorech dodnes. Cívka se používala jako elektromagnet, protože permanentní magnety s požadovanou velikostí a silou byly drahé. Protože pro elektromagnet je potřeba střídavý proud, jeho frekvence ovlivňovala výstupní zvuk tím, že do něj přidávala šum. V roce 1930 byl představen první reproduktor se stálým magnetem, ale kvůli jeho váze a velikosti se nerozšířil. Jeho rozvoj nastal až po druhé světové válce s rozšířením AlNiCo (Aluminium Nickel Cobalt) magnetů. Ve třicátých letech minulého století začala výroba reproduktorových soustav využívající více jednotlivých reproduktorů k vytvoření jednoho zvuku o více frekvencích.

5.2 Princip reproduktorů s pohyblivou cívkou

Reproduktory s pohyblivou cívkou převádějí elektrickou energii na mechanickou energii, přesněji na pohyb. Reproduktor se skládá z magnetu, koše, membrány (většinou kuželovitého tvaru) a cívky. Cívka je umístěná v magnetickém poli permanentního magnetu. Je zde umístěná ve válci, kde venkovní strana je magnet a jejím středem je protažena další část magnetu. Venkovní strana má severní polaritu a část uvnitř cívky má jižní. Když na cívku přivedeme elektrický proud, který má být převeden na zvuk, tak se cívka v magnetickém poli pohne nahoru nebo dolů podle toho jakým směrem jí prochází proud. Proud je většinou kontrolovaný pomocí akustického zesilovače. Pohyb cívky způsobí i pohyb membrány, která rozvibruje vzduch, což člověk vnímá jako zvuk.

5.3 Typy reproduktorů podle frekvence

Reproduktory se dělí z hlediska frekvence na několik typů. Zvukovou frekvenci ovlivňuje to v jaké konstrukci je reproduktor zabudován. Některé specifické reproduktory se zaměřují pouze na rozsah několika set Hz, jako například u telefonního reproduktoru staří 3-5 kHz.

Širokopásmové

Univerzální reproduktory jsou určené převážně pro spotřební elektroniku. Většinou mají frekvenční rozsah 55-13500 Hz.

Basové

Tyto se dělí do dalších dvou kategorií. Subwoofery s rozsahem 20-200Hz a woofery s frekvencí 40-500 Hz. Jedná se o velké reproduktory s výkyvy membrány až 5 mm. Průměr reproduktoru bývá mezi 15-60 cm. Součástí je otvor, aby mohl přes cívku proudit vzduch.

Středové

Jejich obvyklý rozsah je 250-2000 Hz. U těchto reproduktorů je kladen důraz na snížení zkreslení, protože je lidský sluch v těchto frekvencích nejcitlivější.

Výškové

Rozsah výškových reproduktorů je okolo 2-20 kHz. Toto je nejvyšší slyšitelná frekvence. Speciální typy mohou dosahovat až 100 kHz. Výškové reproduktory nemají membránu tvaru kuželu, ale mají tvar kulového vrchlíku.

Kapitola 6

Systemové sběrnice

V této kapitole se podíváme na dvě sběrnice používané při implementaci práce. První je FlexBus sloužící ke komunikaci mezi MCU a FPGA. Druhou je sběrnice I2C, pomocí které se přenáší konfigurace mezi FPGA a řadičem HDMI.

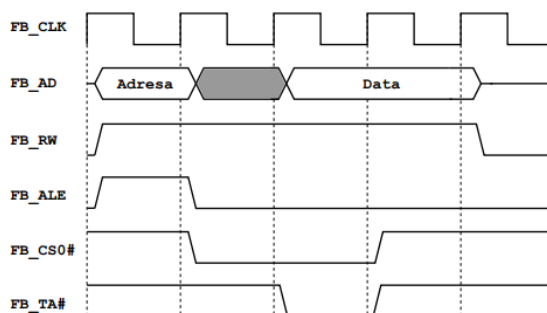
6.1 FlexBus

Mikrokontrolér Kinetis K60 disponuje rozhraním FlexBus.[2] FlexBus slouží pro komunikaci s externími obvody jako jsou paměti ROM, FLASH, programovatelná hradlová pole, nebo podobné obvody, které využívají komunikaci na podobném principu. FlexBus je paralelní synchronní sběrnice, která dovoluje konfiguraci parametrů a tím umožňuje přizpůsobení pro řadu externích zařízení. Jedná se o master-slave typ komunikace, MCU je vždy master, to znamená že vždy inicializuje komunikaci. Na kitu Minerva je sběrnice použita pro propojení MCU a FPGA.

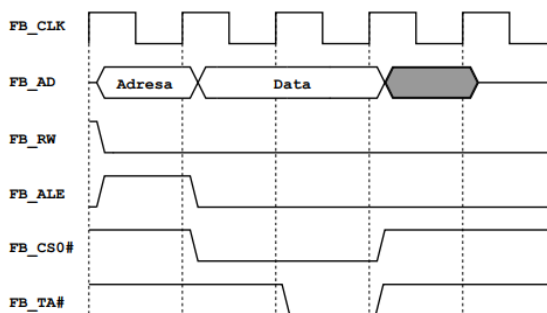
Adresovou a datovou šířku lze zvolit 8, 16 nebo 32 bitů. Předání adresy a dat je multiplexováno, protože adresa i data sdílí stejné signály FB_AD. Signály potřebné pro komunikaci jsou FB_CLK (Clock, hodiny), FB_RW (Read/Write, určení jestli se čte nebo zapisuje), FB_ALE (Address Latch Enabled, potvrzení platnosti nastavené adresy), FB_CS (Chip Select, výběr slave zařízení) a FB_TA (Transaction Acknowledged, dokončení přenosu).

Na obrázku 6.1 je ukázka průběhu čtení z externího zařízení. Synchronizaci komunikace zařizuje signál FB_CLK, který je generovaný na straně MCU. Při zahájení komunikace se nastaví adresa na portu FB_AD a signál na FB_ALE, ten informuje protější stranu o nastavení adresy. Signál FB_RW se nastaví na logickou jedničku, toto signalizuje, že se bude jednat o čtení. Počet taktů vystavení adresy, neboli jak dlouho bude adresa na sběrnici než začne přenos dat, je nastavitelný parametr. Po uplynutí této doby se v dalším taktu nastaví signál FB_CS0#, na logickou nulu. Ten se používá pro vybrání adresované periferie a vymezení datové části přenosu. Čtené zařízení nastaví data na port FB_AD a nastaví signál FB_TA#. V dalším taktu jsou data přečtena a posléze přenos ukončen nastavením FB_SC0# na logickou jedničku.

Zápis probíhá podobně jako čtení a jeho ukázka je na obrázku 6.2. Hlavním rozdílem je signál FB_RW, který je nastaven na logickou nulu pro zápis z MCU. Přijímací strana po přečtení dat nastavuje FB_TA# na logickou nulu pro potvrzení. Po potvrzení je obsah dat na portu nedefinovaný.



Obrázek 6.1: Operace čtení pomocí sběrnice FlexBus. [2]



Obrázek 6.2: Operace zápis pomocí sběrnice FlexBus. [2]

6.2 I2C

Sběrnice I2C je multi-masterová synchronní sériová half-duplexová (data lze přenášet pouze jedním směrem). [9] Multi-masterová znamená, že má více zařízení, která mohou posílat příkazy. Pro komunikaci používá dva signály SCL (hodinový signál) a SDA (data a adresy). Oba jsou zapojeny jako otevřený kolektor, to znamená že vodič je připojen přes pull-up rezistor. Když je sběrnice volná, tak jsou oba signály nastaveny na logickou jedničku. Počet připojených zařízení je v podstatě neomezený (vzhledem k elektrickým vlastnostem), omezuje ho pouze podporování adresovatelnosti 128 zařízení.

Zařízení připojené na sběrnici se dělí na master a slave. Master je v danou dobu aktivní pouze jeden, stará se o zahájení a ukončení komunikace, generuje hodinový signál. Slave je zařízení, které je adresováno masterem. Adresa je osmi bitová z toho sedm bitů je adresa a jeden bit určuje jestli se bude číst ze zařízení slave nebo do něj zapisovat.

Při přenosu se s každým pulzem hodin přenesou jeden bit. SDA se mění pouze když je SCL v logické nule, po celou dobu když je SCL v logické jedničce jsou data platná.

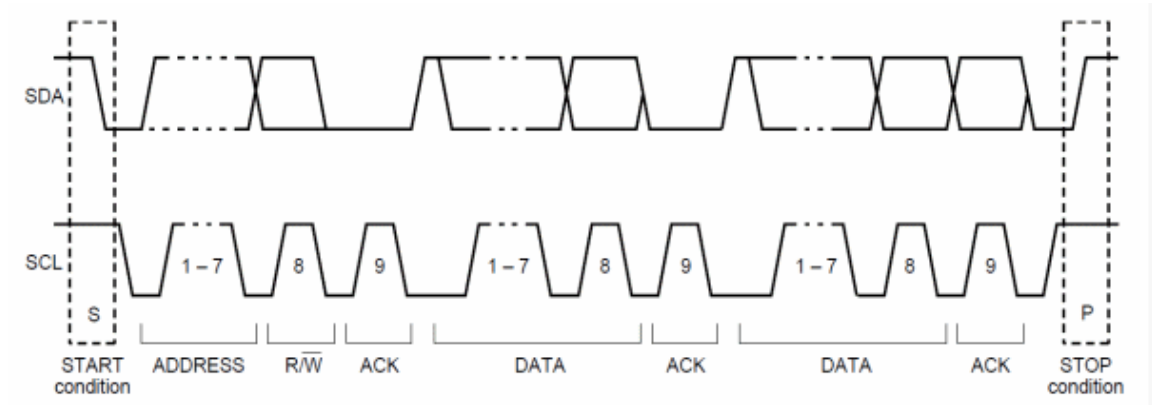
Komunikace na sběrnici začíná při poslání START podmínky a končí při poslání STOP podmínky. Tuto operaci lze provést pouze když je SCL v logické jedničce. START je přechod na SDA z logické jedničky na nulu a STOP je opačně. Po zahájení START je sběrnice aktivní a nemůže komunikaci začít jiný master.

Data se na sběrnici přenášejí po 8 bitech a každý bite musí být následován potvrzovacím bitem.

Komunikace probíhá zahájením START podmínky, jak je vidět na obrázku 6.3. Posléze se posílá adresa zařízení, pokud bylo nalezeno zařízení odešle potvrzení v ACK bitu. Po

obdržení ACK může začít přenos dat, při čtení posílá po každém bytu ACK master při zápisu slave. Po obdržení posledního bitu se provede podmínka STOP neboli, se nastaví signál SDA na logickou jedničku, když je SCL v logické jedničce. Pokud vysílač neobdrží ACK signál ukončí komunikaci pomocí signálu STOP.

Komunikaci může zahájit zařízení kdykoliv, pokud na sběrnici neprobíhá komunikace. Komunikace se řídí pomocí detekce kolizí. Pokud zařízení master vysílá data tak je zároveň kontroluje. Pokud vysílají dvě zařízení jednu logickou jedničku a druhé logickou nulu, tak na sběrnici bude logická nula, a zařízení které detekuje že data neodpovídají tomu co vysílá okamžitě ukončí komunikaci.



Obrázek 6.3: Průběh komunikace na sběrnici I2C.[9]

Kapitola 7

Videohry

V této kapitole se zaměříme trochu na historii videoher a to z jakých jsem si vzal při vypracovávání práce příklad.

7.1 Historie

První videohry začali vyvíjet počítačovní inženýři již v padesátých letech minulého století. V šedesátých letech na M.I.T. profesori a studenti hráli hry, které ovládali pomocí dřevných štítků. Později se začali ovládat pomocí tlačítek a grafické rozhraní se zobrazovalo na oscilátorech. V roce 1962 na M.I.T. vytvořili hru Spacewar!, jedná se o hru pro dva hráče, kteří se snaží zničit jeden druhému vesmírnou loď a zároveň manévrují v gravitačním poli hvězdy. Tato hra byla přidána na seznam deseti nejdůležitějších her historie, tento seznam vytvořila Knihovna Kongresu Spojených Států.

V sedmdesátých letech začal rozvoj videoher. Začaly se vytvářet arkády, které rozšířily popularitu videoher. Jedním z prvních kdo začal s výrobou arkád byla firma Atari. Vytvořili hru Pong, která simuluje stolní tenis. V této době se začaly rozvíjet i domácí herní konzole. Firma Magnavox a jejich série konzol Odyssey se stala hitem v půlce sedmdesátých let. S příchodem vyšších programovacích jazyků BASIC a C se začaly rozvíjet i videohry na počítače, protože jejich vývoj se stal lehčí, než byl v jazycích jako FORTRAN a COBOL. Většina počítačových her se vyvíjela na Apple II a Commodore PET.

V osmdesátých letech se více firem snažilo navázat na úspěch. Firmy Commodore, Sinclair, Atari začaly vytvářet počítače, to vedlo i k rozšíření her. Firma Apple přišla s počítačem Macintosh, který sice nepodporoval barevné rozhraní, ale měl grafický operační systém. To lákalo některé tvůrce her. Barva se vrátila s příchodem Mac II. V té době se také začalo rozvíjet hraní her po síti.

V devadesátých letech přišly přenosné konzole od firmy Nintendo, jako Game Boy. Na počítačích začaly být oblíbené strategie v reálném čase jako Duna 2, udávající standard pro většinu podobných her. Začal rozvoj hraní her přes Internet, jako first-person shooter game Quake (navazující na hru Doom), která byla jedna z prvních.

S přelomem tisíciletí přišlo hraní na mobilních telefonech, jako například u Nokii 6110 hra had. S postupem technologie u počítačů i konzol začal velký rozvoj her všech druhů, ten pokračuje dodnes.

7.2 Inspirace pro práci

Pro práci jsem měl ze začátku více návrhů, jaký typ hry by se dal udělat, kvůli tomu, že jsem netušil jak přesně bude fungovat rychlost a plynulost ovládání a vykreslování. Jedním společným cílem bylo, že hra bude ve dvou rozměrech.

Prvním typem hry byla skákačka jako je například Mario. V těchto hrách jde o to, že hráč se pomocí pár typů pohybů, kde většinou jde o pohyb dopředu dozadu a skok, dostat přes 2D prostředí. V této mapě se vyskytují různé překážky a nepřátelé. Pokud do hráče narazí nepřítel nebo spadne z plošiny, po kterých se pohybuje, začíná kolo znovu.

Dalším typem hry bylo něco podobného arkádové hře Space Invaders. V této hře jde o to, že hráč ovládá vesmírnou loď, která se pohybuje po spodním okraji obrazovky. Může s ní pohybovat pouze doprava a doleva. Z vrchního okraje obrazovky létají nepřátelé, které musí hráč sestřelit než se k němu dostanou nebo ho trefí vlastní střelbou.

Další hrou, na kterou jsem se díval byla další arkádová hra Pac-Man. V ní jde o to, že hráč musí běhat bludištěm a sbírat body a nesmí se ho dotknout nepřátelští duchové.

Kapitola 8

Řadič rozhraní HDMI

Pro vytvoření práce jsem využil řadič rozhraní HDMI vytvořený v práci pana Marka [8]. Tento řadič využívá obvod TFP410 pro generování HDMI signálů.

8.1 TFP410

TFP410 je obvod od Texas Instruments. Slouží ke generování DVI/HDMI obrazového výstupu. Obvod podporuje rozlišení od VGA do WUXGA. Dá se ovládat dvěma způsoby, pomocí konfiguračních pinů nebo pomocí I2C. Obvod poslouchá na I2C adrese 0x7E.

Obvod přijímá obrazová data o šířce 24 bitů, 8 bitů na barvu. Data kóduje a serializuje pro přenos přes kroucené dvojlinky. Může přijímat i data o šířce 12 bitů pro dosažení požadované rychlosti. Přijímá je na nástupní i sestupní hraně hodinového signálu.

8.2 Princip řadiče

Řadič umožňuje MCU vykreslování obrazu přes HDMI, pro tuto funkci využívá FPGA, které je připojené jak k MCU tak k obvodu TFP410. Programovatelné hradlové pole je připojeno k MCU pomocí FlexBus, přes kterou se posílají obrazová data. FPGA je k obvodu TFP410 připojeno pomocí dvou kanálů, prvním je I2C (konfigurace) a druhým jsou RGB data.

Základní adresa pro Flexbus je 0x60000100. FPGA je konfigurované na datovou šířku 16 bitů. Na adrese 0x60001040 lze nastavit rozlišení v ose x a na adrese 0x60001042 v ose y, rozlišení lze nastavit i přímo při syntetizaci řadiče ve VHDL pro FPGA. Adresa 0x60001044 umožňuje povolení nebo zakázání vykreslování.

Řadič umožňuje maximální frekvenci hodinového signálu 115 MHz. Proto řadič podporuje řadu rozlišení od 640x480 do 1280x1024, ale už nepodporuje FullHD, ten vyžaduje frekvenci 148,5 MHz.

V práci pana Marka je zmíněno, že narazil na problémy s vykreslováním na některých monitorech. Já jsem na tento problém nenarazil u normálních monitorů, pouze u jedné televize BenQ se nepodařilo zprovoznit přenos.

Kapitola 9

Návrh hry

9.1 Návrh hry

Prvním návrhem hry byla skákačka, podobná jako Mario. Nešlo o nic složitého v podstatě jen šlo pouze o pokus s plynulým pohybem postavy. S tím se vyskytla spousta problémů. Hlavním problémem byla rychlost překreslování postavy při pohybu, byl trhaný a postava se nedala úplně plynule ovládat. Takže od hry v reálném čase jsem upustil.

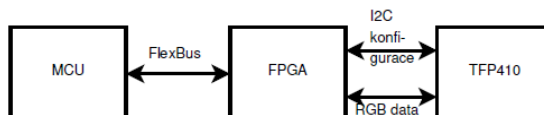
Hra implementovaná v této práci je tahová strategie, kde hráč ovládá vesmírnou loď. Tou se snaží projít bludiště na co nejmenší počet tahů. Aby to nebylo tak lehké vyskytují se v bludišti nepřátelé (létaající talíře), do kterých když nabourá, nebo nepřítel do hráče, tak se restartuje úroveň. Hráč může ničit nepřátele, které ma před sebou pomocí raket, výstřel se také počítá do tahů. Vždy se bude střídat pohyb hráče a pohyb nepřátel. Takže hráč může zůstat stát a nechá v pohybu nepřátele, kteří se bude snažit letět do zdi, to ovšem nepůjde, ale bude se to počítat jako hráčovy tahy a nepřátelé se budou hýbat.

Pro ovládání hry jsem zvolil membránovou klávesnici 4x4 tlačítka připojenou přes 8 pinů k výukovému kitu Minerva. Hra půjde ovládat i pomocí tlačítek přímo na kitu. Toto řešení bude pro testovací účely a pro možnost ovládání, když uživatel nebude mít přístup k tomuto typu klávesnice. Ke kitu Minerva půjde připojit i reproduktor, také pomocí pinů, pro zvukový výstup hry.

9.2 Schéma propojení jednotlivých prvků

Vnitřní propojení

Schéma propojení MCU s HDMI pomocí FPGA je na obrázku 9.1. K tomuto propojení je využit řadič HDMI z práce pana Marka.



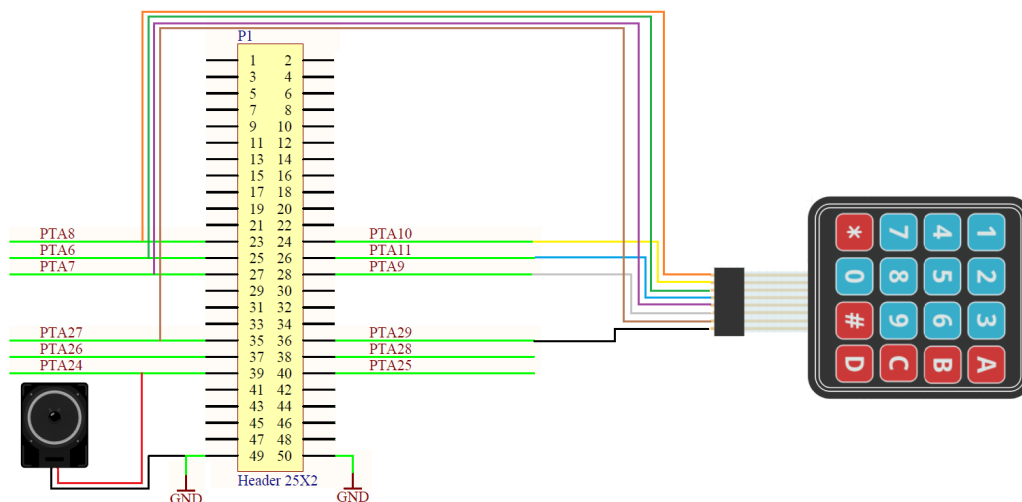
Obrázek 9.1: Propojení řadiče HDMI.

Připojení externího hardwaru

Externí hardwarové prvky jsou připojeny ke kitu Minerva pomocí pinového pole P1. Jsou připojeny k MCU pomocí port A.

Klávesnice je připojena na piny od 23 do 28 a piny 35, 36 celkem tedy na osmi pinech. Tyto piny odpovídají jednotlivým částem portu A přesně to lze nalézt ve schématu zapojení kitu Minerva 9.1. Čtyři tyto piny jsou pro řádky a zbylé čtyři pro sloupce.

Reproduktor je potřeba připojit na dva piny. Jeden z nich je možné připojit na pin 39 odpovídající 24 na portu A. Druhý musí být připojen na zem, a to na jeden z pinů 49 nebo 50.



Obrázek 9.2: Připojení klávesnice a reproduktoru ke kitu Minerva.

9.3 Uchovávání dat o hře v MCU

V MCU je potřeba uchovávat data o hře. Jsou zde dva typy dat, ke kterým musí být přístup. Data dynamická, která se budou měnit v průběhu hry, a data statická, která budou pořád stejná v průběhu hry. Mezi statická data patří celá mapa bludiště, která obsahuje nejen místa stěn, ale i startovní a cílovou pozici hráče, pozice nepřátel a kterým směrem začnou s pohybem. Mezi dynamická data patří aktuální stav hráče, létajících talířů, raket, laseru, explozí. Do těchto dat patří také výřez mapy, obsahující pouze zdi bludiště, který je aktuálně vykreslován.

Mapy se uchovávají ve dvourozměrných polích obsahujících čísla, které každé představuje jiný prvek hry. Šířka dynamické mapy je menší než šířka mapy statické. Dynamická mapa má šířku pouze tolik bloků kolik je vykreslováno a její obsah se přepisuje při pohybu hráče směrem doprava. Obě mapy mají stejnou výšku, a to tu která se vykresluje, to znamená, že když se hráč pohybuje nahoru a dolů tak se obsah dynamické mapy nemění.

Ostatní data potřebná pro chod hry jsou uložena ve strukturách. Všechny struktury, až na strukturu hráče, obsahují odkaz na další strukturu svého typu protože se tyto prvky mohou vyskytovat ve hře na více místech. Všechny tyto struktury musí obsahovat pozici na mapě a některé, ty co se pohybují nezávisle na hráčových tazích, musí mít i směr, kterým se budou hýbat.

Kapitola 10

Implementace

V této kapitole se podrobněji podíváme na implementaci, některých významnějších částí práce.

10.1 Pohyb a kolize

Jedno pole na mapě se skládá ze 3x3 pixelů. Vesmírná loď hráče zabírá 2x2 bloky mapy, protože jednotlivé pixely jsou velké, aby bylo poznat co daný objekt představuje.

Aby byl pohyb plynulý, je potřeba posunovat prvky po pixelech a ne po blocích mapy. K tomu je potřeba přidat do struktur pohyblivých objektů číslo pro určování ve kterém kroku pohybu se nacházejí. Pomocí tohoto offsetu v mapě lze určit přesnou polohu kam vykreslovat objekty.

Směr pohybu pro hráčovu vesmírnou loď přichází od uživatele, vždy pro jeden tah. Ostatní prvky mají daný směr pohybu. Nepřátelské lodě mají zadaný výchozí směr v mapě. Tyto lodě jsou v mapě definovány pomocí čtyř čísel. Vždy se jedná o tuto loď, ale každé číslo znamená jiný výchozí směr. Pokud narazí loď do zdi tak invertuje svůj směr, takže když letěla nahoru poletí dolů a analogicky. Rakety, které střílí hráč, mají vždy směr doprava.

Kolize se řeší při pohybu jednotlivých objektů. Vždy se kontroluje objekt, kterým hýbeme, se všemi ostatními objekty ve hře a ještě se kontroluje jestli směrem, kam se chce pohybovat není stěna bludiště.

Pokud hráč chce pohnout lodí doprava, tak se pohybuje mapa okolo lodi. Hráč vždy začíná na x-ové souřadnici jedna pokud se pohne doprava a pak doleva, tak za ním není stěna a ocitne se na pozici nula, posléze se může posunout opět doprava na jedničku. Při dalších posunech doprava se bude opět pohybovat mapa. Takže hráč se pohybuje pouze v těchto dvou pozicích. Toto mi přišlo jako nejlepší způsob pro posouvání v mapě. Pro to, aby se mapa pohybovala správně je v programu globální proměnná sloužící pro offset mapy. Od tohoto offsetu se potom kopíruje do dynamické mapy ze statické.

10.2 Vykreslování

Vykreslování prohýbá pomocí funkce, které předáme v parametrech souřadnice a barvu. Funkce tyto informace dále pošle přes FlexBus do FPGA, které je dále pošle do HDMI. Jedno pole na mapě se skládá z 3x3 pixelů.

Vykreslování bylo původně implementováno způsobem překreslování celé plochy. Toto fungovalo tak, že se vždy vykreslilo vše na obrazovku a když se mělo něco pohnout tak se

vše smazalo. Tím že se na všechny pixely vykreslila černá barva, a potom se znovu vykreslily všechny prvky hry. Toto sebou přineslo problém s tím, že obraz problikával a špatně se na něj dívalo tak od této varianty bylo upuštěno.

Řešení tohoto problému je takové, že se překresluje pouze ten prvek co se pohnul. Prvky se postupně pohybují vždy o jeden pixel, takže na pohyb o pole v mapě je potřeba ho překreslit třikrát, toto simuluje plynulost pohybu. Protože prvek se vždy překreslí na nové souřadnice je potřeba přebarvit původní pixely co už k prvku nepatří na barvu pozadí. Neboli je zbytečné překreslovat celý původní prvek na barvu pozadí, když se pak jeho většina zase bude znovu měnit.

Posunování mapy je řešeno tak, že se postupně prochází vykreslovaná mapa a porovnává se, zda je blok v x-ové ose jiný než ten, který je aktuálně vybrán. Pokud jsou stejné tak se nic nemění. Pokud jsou různé tak se překreslí tři pixely vpravo (mají stejnou souřadnici x a všechny možnosti souřadnice y v bloku) aby odpovídaly bloku který se na tuto pozici posunuje.

10.3 Vstup a výstup

Pro vstup ovládání jsou dvě možnosti. Je potřeba si vybrat, kterou možnost bude uživatel využívat při překladač kódu pro MCU. První možnost jsou tlačítka na kitu Minerva a druhá je klávesnice připojená na piny. Je potřeba si vybrat, protože když se vybere jedna a použije druhá může dojít k nestabilitě kódu.

Tlačítka na kitu se obsluhují pomocí volání přerušení. Kit zachytí přerušení na portu pro tlačítka a zavolá funkci na obsluhu přerušení. V této funkci se zjistí, které tlačítko bylo zmáčknuto a podle toho se zavolá funkce na ovládání hry.

Klávesnice funguje na principu skenování portů, ke kterým je připojena. Protože k určení, toho které tlačítko bylo zmáčknuté, jsou potřeba posílat impulzy do jednotlivých pinů řádků a zjišťovat jestli přijde z pinů sloupců odezva. Klávesnice je připojená jak je uvedeno v sekci kapitoly Návrh 9.2.

Výstup na reproduktor je implementován jako funkce, které se předá tón, z množiny definovaných, který má přehrát. Tato funkce potom generuje impulzy na příslušný port kitu. V podstatě se jedná o funkci, která invertuje bit portu a dělá to v určitých časových intervalech, aby to odpovídalo danému tónu. Tuto funkci je možné volat i když není připojen reproduktor a není dobré ji oddělovat z kódu kde je, protože by to narušilo časování vykreslování.

10.4 Problémy s vývojovými prostředími

Problém s vývojovým prostředím Xilinx ISE verze 14.7, je že pro jeho získání je potřeba zaregistrovat se na stránkách nxp.com. A potom najít verzi ISE 14.7, která by se dala lehce stáhnout, protože stránky pořád nutí do stahování studia Vivado. Toto je způsobeno tím, že Xilinx ISE se už nevyvíjí, ale pro práci s kitem je potřeba, poněvadž je to poslední verze podporující FPGA Spartan 6.

V KDS při vývoji pro MCU jsem narazil na problém s importy jednotlivých částí projektu. Prostředí psalo, že dané soubory nemůže nalézt, i když importy byly správně. Toto, jak jsem zjistil bylo tím, že určité složky projektu nebyly přidány do cesty importů. Nastavení propojování, jak jsem později zjistil je v projekt properties pod částí C/C++ Build Settings a zde v C Compiler Includes, kde je potřeba přidat všechny složky projektu.

Kapitola 11

Možnosti dalšího vývoje

Na výukovém kitu Minerva jsou další moduly, které by se daly využít k vylepšení videohry. Mezi takového moduly patří například audiokodek, který je vyveden na konektor typu jack. Pro audiokodek je vytvořená knihovna pro kit Minervu.[6] Tato knihovna umožňuje přehrávání audio souborů z SD karty pomocí MCU a FPGA. Ovšem je zde problém s touto knihovnou při využívání s HDMI, protože obě využívají FlexBus, ale každá má jinou šířku přenášených dat. Řadič HDMI využívá 16 bitů a audiokodek 32 bitů, takže je zde problém s adresací. Podařilo se mi udělat VDHL kód, který správně přijímal HDMI a audio, ale musely být kompilovány v MCU samostatně, nefungovaly v jednom programu. Toto by šlo nejspíše vyřešit přepsáním celé komunikace FlexBus pro řadič HDMI, aby také využíval 32 bitů.

Kapitola 12

Závěr

V práci se mi povedlo implementovat videohru pro platformu Minerva. Videohra vytvořená v práci je tahová strategie, kde hráč ovládá vesmírnou loď, kterou se snaží proletět bludištěm, s nepřátelskými loděmi nepřáteli, na co nejmenší počet tahů. Na Minervě se pro práci využívá mikrokontrolér Kinetis K60 a programovatelné hradlové pole Xilinx Spartan 6, vzájemně propojené pomocí sběrnice FlexBus. V rámci práce jsem prostudoval a využil knihovnu pro práci s HDMI. Pomocí, které se posílají data a konfigurace mezi FPGA a čipem TFP410. Zobrazování videohry funguje na většině monitorů, problém jsem měl pouze u televizoru BenQ. Vstup od uživatele je realizovaný pomocí klávesnice připojené na pinech výukového kitu Minerva. Dalším typem výstupu videohry je zvukový, prostřednictvím připojení reproduktoru k Minervě, také pomocí pinů. V práci byl kladen důraz na dokumentaci kódu, aby se dal používat jako zdroj inspirace pro další práci s kitem Minerva.

Literatura

- [1] *Minerva Kit*. [Online, navštíveno 23.05.2020]. Dostupné z: <https://minerva.php5.cz/uvod/o-projektu.php>.
- [2] BUCHTA, P. *Application Development Framework for the ARM Platform, diplomová práce*. Brno, FIT VUT v Brně, 2015.
- [3] EARGLE, J. *Loudspeaker Handbook*. ACM SIGCOMM Computer Communication Review, 2003. ISBN 978-1-4757-5678-4.
- [4] FREESCALE. *K60 Sub-Family Reference Manual*. [Online, navštíveno 23.05.2020]. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/K60P144M100SF2V2RM.pdf>.
- [5] GARLING, C. *Past is prototype: The evolution of the computer keyboard*. [Online, navštíveno 23.05.2020]. Dostupné z: <https://www.computerworld.com/article/2493059/past-is-prototype--the-evolution-of-the-computer-keyboard.html>.
- [6] HARTMANN, J. *Knihovna pro využití audiokodeku SGTL5000 na výukovém kitu Minerva*. Brno, FIT VUT v Brně, 2017.
- [7] HDMI LICENSING ADMINISTRATOR, I. *HDMI Specification*. [Online, navštíveno 23.05.2020]. Dostupné z: <https://www.hdmi.org/spec/index>.
- [8] MAREK, J. *Využití rozhraní HDMI na výukovém kitu Minerva, bakalářská práce*. Brno, FIT VUT v Brně, 2017.
- [9] NXP. *I2C Bus Specification*. [Online, navštíveno 23.05.2020]. Dostupné z: <https://i2c.info/i2c-bus-specification>.
- [10] NXP. *KDS_IDE: Design Studio Integrated Development Environment IDE*. [Online, navštíveno 23.05.2020]. Dostupné z: https://www.nxp.com/design/designs/design-studio-integrated-development-environment-ide:KDS_IDE.
- [11] XILINX. *Spartan-6 Family Overview*. [Online, navštíveno 23.05.2020]. Dostupné z: https://www.xilinx.com/support/documentation/data_sheets/ds160.pdf.
- [12] XILINX. *Xilinx ISE Design Suite*. [Online, navštíveno 23.05.2020]. Dostupné z: <https://www.xilinx.com/products/design-tools/ise-design-suite.html>.

Příloha A

Obsah CD

bp_xkubaa00.pdf	Text bakalářské práce.
kod_text	Složka se zdrojovými soubory technické zprávy.
kod_fpga	Složka s projekty pro prostředí Xilinx ISE.
kod_mcu	Složka s projekty pro prostředí KDS.
obvodove_schema.png	Obrázek připojení externího hardwaru.

Tabulka A.1: Obsah přiloženého CD.