



Střední průmyslová škola strojní  
a elektrotechnická a Vyšší odborná škola,  
Liberec 1, Masarykova 3

## TUXMAN - FPGA

Maturitní/Ročníková práce

Autor	<b>Martin Přivozník</b>
Obor	<b>Informační technologie</b>
Vedoucí práce	<b>Ing. Vladimír Prokeš</b>
Konzultant práce	<b>Ing. Petr Socha</b>
Školní rok	<b>2019/2020</b>

## **Anotace (Resumé)**

Tématem práce je návrh arkádové hry na RTL úrovni a její implementace na programovatelném hradlovém poli, tj. FPGA. Uživatelský vstup je zajištěn pomocí PS/2 klávesnice a výstup prostřednictvím VGA. K implementaci je použit jazyk VHDL.

## **Klíčová slova**

RTL, programovatelné hradlové pole, FPGA, PS/2, VGA, VHDL

## **Summary**

The topic of this thesis is designing an arcade game on an RTL level and its implementation on a programmable gate array, i.e. FPGA. User input is provided by PS/2 keyboard and output is shown using VGA. Language called VHDL is used for implementation.

## **Keywords**

RTL, programmable gate array, FPGA, PS/2, VGA, VHDL

## Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní/ročníkovou práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne 3. února 2020

---

Martin Přívozník

# Obsah

Úvod . . . . .	1
<b>1 Analýza</b>	<b>2</b>
1.1 Číslicový návrh . . . . .	2
1.1.1 Kombinační obvody . . . . .	2
1.1.2 Sekvenční obvody . . . . .	8
1.1.3 Synchronní a asynchronní návrh . . . . .	10
1.1.4 Jazyk VHDL . . . . .	10
1.2 Programovatelná hradlová pole FPGA . . . . .	10
1.2.1 Dostupné prostředky . . . . .	11
1.2.2 Logická syntéza . . . . .	11
1.2.3 Vývojová deska Digilent Basys 2 . . . . .	11
1.2.4 Prostředí Xilinx ISE . . . . .	11
1.3 Rozhraní . . . . .	11
1.3.1 7-segmentový displej . . . . .	11
1.3.2 PS/2 . . . . .	11
1.3.3 VGA . . . . .	11
1.4 Hra Pacman . . . . .	11
1.4.1 Princip . . . . .	12
1.4.2 Ovládání . . . . .	12
<b>2 Návrh hry</b>	<b>13</b>
2.1 Specifikace hry . . . . .	13
2.1.1 Cíl a chování hry . . . . .	13
2.1.2 Herní mapa . . . . .	13
2.1.3 Postavy a jejich chování . . . . .	13
2.2 Herní textury . . . . .	13
2.2.1 RGB . . . . .	13
2.2.2 Textury jako matice barev . . . . .	13
<b>3 Implementace</b>	<b>14</b>
3.1 Modul pro čtení z PS/2 klávesnice . . . . .	14
3.1.1 Zpracování vstupu . . . . .	14
3.1.2 Generování výstupu . . . . .	14
3.2 Modul pro výstup na VGA monitor . . . . .	14

3.2.1	Časování VGA . . . . .	15
3.2.2	Propsání textur na monitor . . . . .	15
3.3	Herní logika . . . . .	15
3.3.1	Herní mapa . . . . .	15
3.3.2	Ovládání postav . . . . .	15
3.3.3	Cíle hry . . . . .	15
<b>4</b>	<b>Testování</b>	<b>16</b>
	Závěr . . . . .	17

# Úvod

# Kapitola 1

## Analýza

Tato kapitola obsahuje stručný souhrn znalostí a informací potřebných pro následný návrh a implementaci. V sekci 1.1 je vysvětlen číslicový obvod a postup jeho návrhu. V sekci 1.2 stručně vysvětlují programovatelné hradlové pole a dále vybranou vývojovou desku. Sekce 1.3 se zabývá použitými komunikačními rozhraními, které zajišťují uživatelský vstup a výstup. Na závěr kapitoly, v sekci 1.4 vysvětlují princip a funkčnost hry, jenž je vzorem pro můj návrh.

### 1.1 Číslicový návrh

V této sekci se věnuji tomu, co je číslicový obvod a jak jej navrhnout jak ve schématu, tak v jazyce popisujícím hardware. V podsekci 1.1.1 rozeberu logické funkce, prostředky jejich popisu a realizace pomocí logických hradel. Podsekcí 1.1.2 je zaměřená na návrh sekvenčních obvodů a synchronních sekvenčních automatů (FSM), na což naváže podsekcí 1.1.3, ve které vysvětlují princip hodinových domén a plně sekvenčního návrhu. V podsekci 1.1.4 stručně ukážu, jak převést schéma číslicového obvodu do kódu v jazyce popisujícím hardware (Hardware Description Language, HDL), v mém případě do jazyka Very High Speed Integrated Circuit Hardware Description Language (VHDL).

#### 1.1.1 Kombinační obvody

##### Booleovská funkce

Booleovská funkce je funkce  $N$  vstupů a  $M$  výstupů nad množinou  $\{0,1\}$ . V případě, kdy má funkce více jak jeden výstup, lze ji rozdělit na  $M$  funkcí s jedním výstupem. Uvážíme-li Booleovu algebru, platí pro operace sčítání a násobení pravidla uvedená v tabulce 1.1. Operace se dvěma vstupními hodnotami nazýváme binární operace. Některé binární operace, přestože často používají stejná značení  $+$  a  $*$  jako v algebře reálných čísel, mají v Booleově algebře stejnou prioritu a jiný význam (žádná operace nemá přednost) [2]. Pro logický součet a logický součin platí základní pravidla v tabulce 1.2. Příkladem re-

$$\begin{array}{lll}
a + b = b + a & a * b = b * a & \text{(komutativita)} \\
a + (b + c) = (a + b) + c & a * (b * c) = (a * b) * c & \text{(asociativita)} \\
a + (b * c) = (a + b) * (a + c) & a * (b + c) = (a * b) + (a * c) & \text{(distributivita)} \\
a + 0 = a & a * 1 = a & \text{(neutralita 0 a 1)} \\
a + \bar{a} = 1 & a * \bar{a} = 0 & \text{(vlastnosti negace)}
\end{array}$$

Tabulka 1.1: Axiomy a vztahy Booleovy algebry.[1]

$$\begin{array}{lll}
\text{de Morgan} & \overline{(a + b)} = \bar{a} * \bar{b} & \overline{(a * b)} = \bar{a} + \bar{b} \\
\text{idempotence} & a + a = a & a * a = a
\end{array}$$

Tabulka 1.2: Základní pravidla Booleovy algebry.[2]

prezentace Booleovské funkce je pravdivostní tabulka 1.3, kde  $in_1$  a  $in_2$  jsou vstupní hodnoty a  $out$  je výstupní hodnota. Pravdivostní tabulka obsahuje vždy  $N^2$  řádků, aby reprezentovala výstupní hodnotu pro všechny možné kombinace vstupních hodnot. Další možností je Booleovská formule [2]. K vyjádření formule a k popisu booleovské funkce používáme nejčastěji základní funkce uvedené v tabulce 1.4

### Kombinační obvod

Kombinační logický obvod, je takový obvod, ve kterém jsou výstupní hodnoty dány pouze aktuální kombinací vstupních proměnných. Neobsahuje žádnou paměť předchozích stavů. Jedinou výjimkou je krátký časový interval, za který logický člen (AND, NAND, OR, NOR ...) vyhodnotí výstup na základě vstupních hodnot. Tento časový interval může být zanedbatelný v případě krátkých datových cest. V případě dlouhé datové cesty může být potřeba na tento časový interval brát zřetel a zvážit optimálnější řešení.

Platí, že u číslicových obvodů každá proměnná v operaci nabývá hodnotu jednoho tzv. bitu. Bit je základní jednotkou dat a může nabývat hodnot 0, nebo 1. Reprezentací číslicového obvodu je schéma číslicového obvodu, kde každá z funkcí je reprezentována tzv. schématickou značkou. Schématické značky mohou být různé, dokud z nich jasně vyplývá, jakou funkci zastupují. Schématické značky jsou propojené signály, které představují jednotlivé bity. Pro minimalizaci je možné několik signálů (bitů) zakreslit jediným konektorem, pokud je označený počtem bitů, které reprezentuje. Nejčastěji používané normy značení

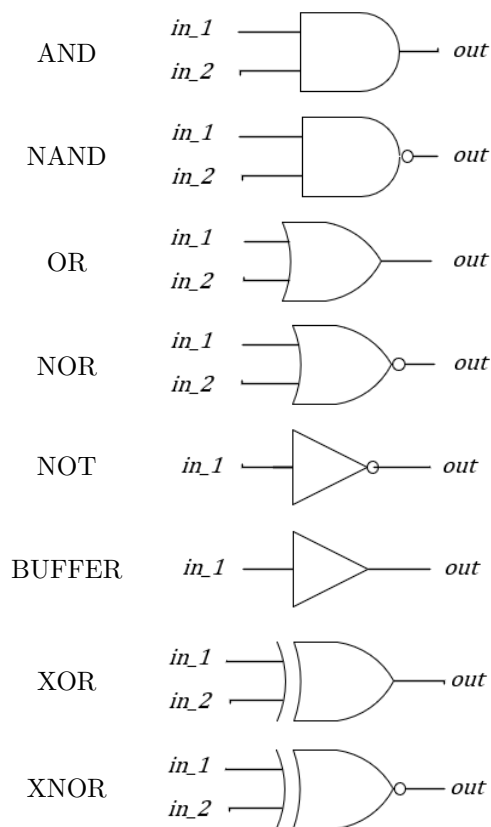
$in_1$	$in_2$	$out$
0	0	$f(0, 0)$
0	1	$f(0, 1)$
1	0	$f(1, 0)$
1	1	$f(1, 1)$

Tabulka 1.3: Pravdivostní tabulka.



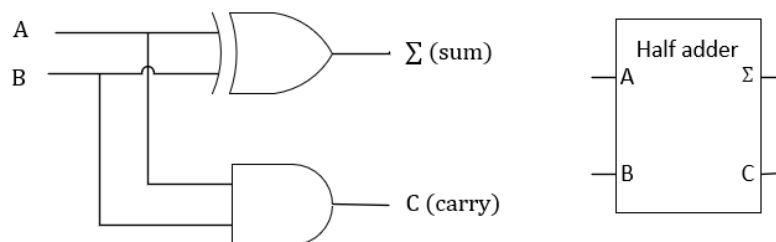
Název	Pravdivostní tabulka			Formule
AND (logický součin)	$in_1$	$in_2$	$out$	$out = in_1 * in_2$
	0	0	0	
	0	1	0	
	1	0	0	
	1	1	1	
NAND (negovaný logický součin)	$in_1$	$in_2$	$out$	$out = \overline{in_1 * in_2}$
	0	0	1	
	0	1	1	
	1	0	1	
	1	1	0	
OR (logický součet)	$in_1$	$in_2$	$out$	$out = in_1 + in_2$
	0	0	0	
	0	1	1	
	1	0	1	
	1	1	1	
NOR (negovaný logický součet)	$in_1$	$in_2$	$out$	$out = \overline{in_1 + in_2}$
	0	0	1	
	0	1	0	
	1	0	0	
	1	1	0	
NOT (logická negace)	$in_1$	$out$		$out = \overline{in_1}$
	0	1		
	1	0		
BUFFER (opakovač)	$in_1$	$out$		$out = in_1$
	0	0		
	1	1		
XOR (nonekvivalence)	$in_1$	$in_2$	$out$	$out = in_1 \oplus in_2$
	0	0	0	
	0	1	1	
	1	0	1	
	1	1	0	
XNOR (ekvivalence)	$in_1$	$in_2$	$out$	$out = \overline{in_1 \oplus in_2}$
	0	0	1	
	0	1	0	
	1	0	0	
	1	1	1	

Tabulka 1.4: Tabulka nepoužívanějších základních logických funkcí.



Tabulka 1.5: Schématické značky některých nejpoužívanějších základních logických funkcí (americká norma ANSI).

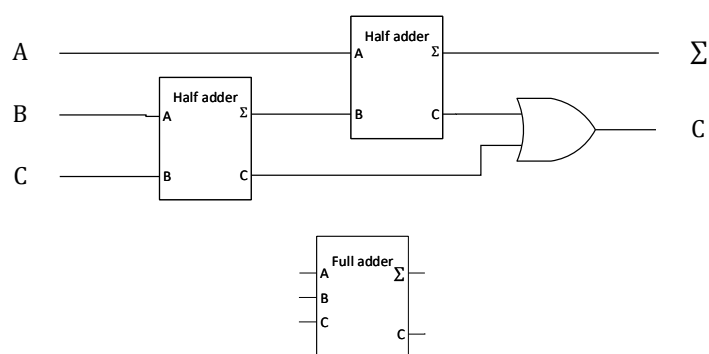
jsou evropská a americká. Příklady schématických značek pro nejpoužívanější logické funkce jsou uvedené v tabulce 1.5. Pro usnadnění práce můžeme využívat logických bloků, které plní danou funkci. Opět platí, že ze značení logických bloků ve schématu musí plně vyplívat, jakou funkci zastupují. Logický blok, který má definovanou funkci může být použit schématu. Při návrhu číslicových obvodů využíváme hierarchie, kde jsou pro každý logický blok popsány vstupy i výstupy a v případě, kdy se nejedná o základní logické bloky, tak je popsána i funkce (formule, pravdivostní tabulka, nebo schéma bloku) a vhodně přiřazena ke schématu. Jedním ze základních logických bloků je poloviční sčítačka, viz obrázek 1.1. Poloviční sčítačka umí sečíst dvě jednobitová čísla a vygenerovat bit do vyššího řádu (carry) podle pravdivostní tabulky 1.6. Zřetězením dvou polovičních sčítaček a přenesením pomocí funkce OR získáme poté kompletní sčítačku, viz obrázek 1.2.[2] Kompletní sčítačka umí sečíst jednobitová čísla, vygenerovat bit do vyššího řádu a přijmout bit z nižšího, sčítá tedy tři bity. Sčítá počet jedniček na vstupech.



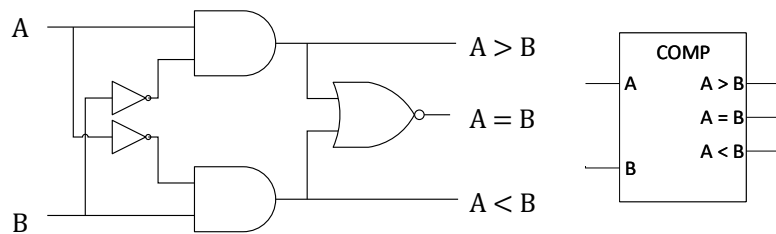
Obrázek 1.1: Schéma poloviční sčítačky a příklad jejího značení ve schématu.

$A$	$B$	$C$	$\Sigma$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

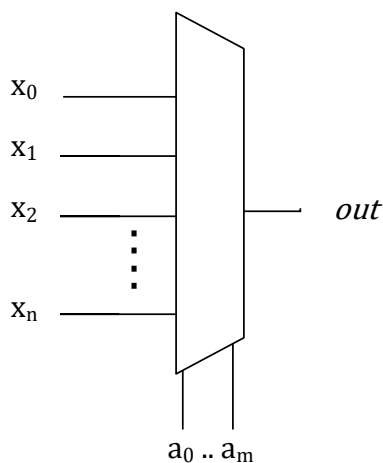
Tabulka 1.6: Pravdivostní tabulka poloviční sčítačky.



Obrázek 1.2: Schéma kompletní sčítačky a příklad jejího značení ve schématu.



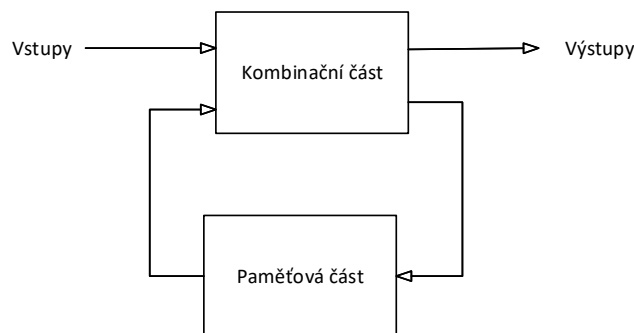
Obrázek 1.3: Schéma komparátoru a příklad jeho značení ve schématu.



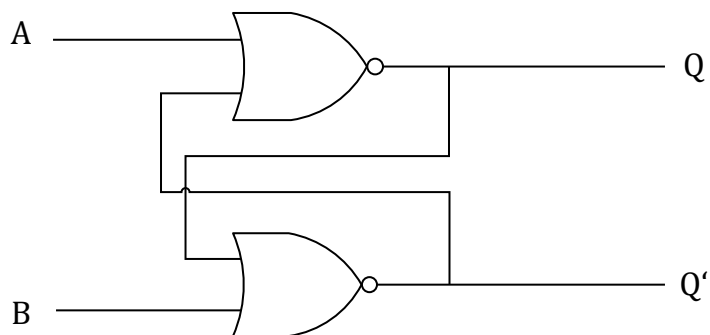
Obrázek 1.4: Schématická značka multiplexoru.

Pro porovnávání dvou hodnot a vyhodnocení jejich nerovnosti používáme tzv. komparátor, viz obrázek 1.3. Dalším důležitým základním logickým blokem je multiplexor, viz obrázek 1.4. Multiplexor na základě řídicího vstupu/řídicích vstupů ( $a$ ) přivádí na výstup ( $out$ ) jeden ze vstupních signálů ( $x$ ).

Číslicový obvod lze realizovat například z integrovaných obvodů, v nichž bývají hradla realizována z několika tranzistorů. Logické hodnoty představuje napětí přivedené na obvod. Logická 1 bývá reprezentována napětím kladným a logická 0 napětím nulovým.



Obrázek 1.5: Huffmanův model konečného automatu.



Obrázek 1.6: Schéma RS klopného obvodu s využitím logických hradel NOR.

### 1.1.2 Sekvenční obvody

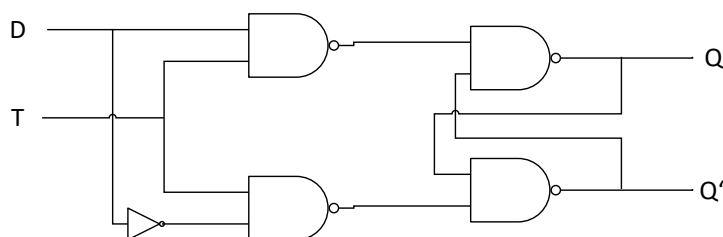
#### Sekvenční logický obvod

Sekvenční logický obvod je takový obvod, ve kterém výstupní hodnoty nejsou dány pouze aktuální kombinací vstupních proměnných, ale zároveň předchozím stavem. Sekvenční obvod si tedy musí zapamatovat předchozí hodnoty pomocí paměti, která bývá realizována pomocí zpětné vazby [2]. Sekvenční obvod se dělí na dvě části - kombinační a paměťovou, kde paměťová část je tvořena kombinačním obvodem, ve kterém bývá zavedena zpětná vazba a kombinační část bývá tvořena typickým kombinačním obvodem. Obecné schéma konečného automatu je na obrázku 1.5.

Příkladem kombinačního obvodu se zpětnou vazbou může být tzv. RS klopný obvod, viz obrázek 1.6. RS klopný obvod překlápí mezi dvěma mezními hodnotami. Pravdivostní tabulka RS klopného obvodu se může dělit na tři části - zakázaný stav (ZS), překlápěcí část a paměť, viz tabulka 1.7. Pokud se tedy

$A$	$B$	$Q$	$Q'$
0	0	ZS	
0	1	0	1
1	0	1	0
1	1	Paměť	

Tabulka 1.7: Pravdivostní tabulka RS klopného obvodu s využitím NOR.



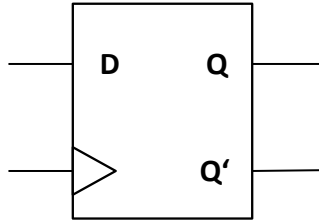
Obrázek 1.7: Schéma D klopného obvodu.

změní vstupní hodnoty z  $[0, 1]$ , nebo  $[1, 0]$  na  $[1, 1]$ , na výstupu bude předchozí hodnota, dokud neproběhne další změna vstupních hodnot. Častěji využívaným klopným obvodem je tzv. D klopný obvod, viz obrázek 1.7. Výhodou D klopného obvodu je fakt, že narozdíl od RS klopného obvodu nemá zakázaný stav, viz pravdivostní tabulka 1.8. Signál  $T$  u D klopného obvodu se dá považovat za tzv. povolovací signál. Pokud na  $T$  přijde sestupná hrana (změna stavu z logické 1 na logickou 0), tak na výstupu  $Q$  zůstane předchozí hodnota signálu  $D$ . Při náběžné hraně na  $T$  (změna stavu z logické 0 na logickou 1) se na výstup  $Q$  zapíše aktuální hodnota na signálu  $D$ . Příkladem značení D klopného obvodu je na obrázku 1.8. Vstupní signál označený trojúhelníkem je signál  $T$ . Trojúhelník mířící směrem do "krabíčky" značí, že obvod mění výstupní hodnotu v reakci na náběžnou hranu na vstupu s tímto trojúhelníkem. Pokud má číslicový obvod takový vstup, je zřejmé, že se jedná o sekvenční obvod.

Příkladem sekvenčního logického bloku je čítač. Čítač je takový sekvenční logický obvod, který uchovává v „paměti“ informaci o tom, kolikrát zaznamenal změnu stavu (dle návrhu náběžná, nebo sestupná hrana) na vstupním signálu.

$T$	$D$	$Q$	$Q'$
0	0	Paměť	
0	1		
1	0	0	1
1	1	1	0

Tabulka 1.8: Pravdivostní tabulka D klopného obvodu.



Obrázek 1.8: Příklad značení D klopného obvodu ve schématu.

### Časové průběhy

Časový průběh je grafické znázornění pravdivostní tabulky. Dá se popsat tzv. časovým diagramem. Časový diagram popisuje, jaké logické hodnoty nabývá daný signál v jaký čas. Příklad časového diagramu je na obrázku 1.9.

### Konečný automat

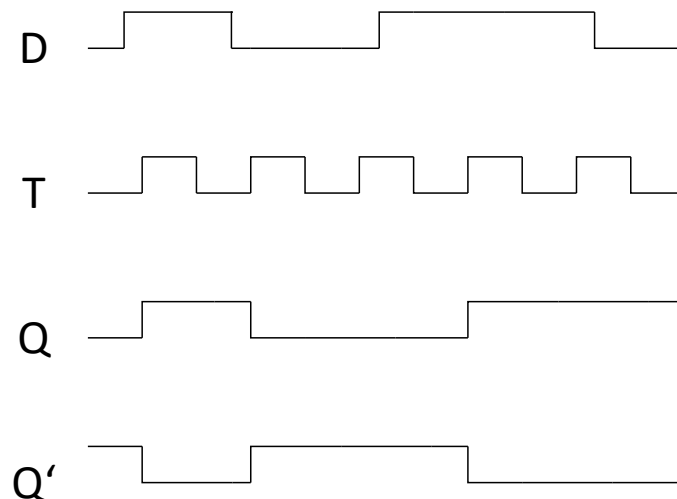
Konečný automat (angl. Finite State Machine - FSM) je matematickým modelem sekvenčního obvodu. Má definované stavy, mezi kterými za daných podmínek přepíná a mění svůj výstup. Variantu konečného automatu můžeme volit v případě, kdy se setkáváme s náročnějším sekvenčním obvodem. Konečný automat musí popisovat přechodové funkce (podmínky pro změnu stavu a do kterého) a výstupní funkce (jaký signál v danou chvíli udává na výstup). Příklady možných návrhů konečných automatů jsou Mealyho návrh a Moorův návrh. Mealyho návrh FSM

#### 1.1.3 Synchronní a asynchronní návrh

#### 1.1.4 Jazyk VHDL

### 1.2 Programovatelná hradlová pole FPGA

Tato sekce se zabývá tím, co jsou programovatelná hradlová pole (Field of programmable gate array, FPGA) a jak probíhá práce s těmito hradlovými poli na vývojové desce. V podsekcí 1.2.1 popisují části FPGA, které jsou použity pro implementaci číslicového obvodu. Podsekcí 1.2.2 stručně vysvětluje kroky nezbytné pro implementaci samotného číslicového obvodu na základě jeho popisu VHDL kódem. V podsekcích 1.2.3 a 1.2.4 je stručně popsána použitá vývojová deska a vývojové prostředí, které se váže k FPGA, které deska obsahuje.



Obrázek 1.9: Časový diagram D klopného obvodu.

### 1.2.1 Dostupné prostředky

### 1.2.2 Logická syntéza

### 1.2.3 Vývojová deska Digilent Basys 2

### 1.2.4 Prostředí Xilinx ISE

## 1.3 Rozhraní

Tato sekce popisuje komunikační rozhraní použité v návrhu a implementaci, která zajišťují uživatelský vstup a výstup. V podsekcí 1.3.1 vysvětlují rozhraní 7-segmentového displeje. Sekce 1.3.2 a 1.3.3 popisují protokoly PS/2 a VGA.

### 1.3.1 7-segmentový displej

### 1.3.2 PS/2

### 1.3.3 VGA

## 1.4 Hra Pacman

Tato sekce vysvětluje hru, již má můj návrh klonovat. Podsekcí 1.4.1 popisuje cíl a pravidla hry. V podsekcí 1.4.2 vysvětlují ovládání hry.



- 1.4.1 Princip
- 1.4.2 Ovládání

## Kapitola 2

# Návrh hry

V této kapitole se zabývám návrhem herní logiky, na základě které je navržen číslicový obvod pro finální implementaci. Sekce 2.1 rozebírá konkrétní prvky hry a návrhy pro jejich řešení. V sekci 2.2 vysvětluji, jak jsou navržené textury, které se zobrazují na výstup.

### 2.1 Specifikace hry

V této sekci rozebírám konkrétní prvky hry a vysvětluji návrhy pro jejich řešení. V podsekcí 2.1.1 popisují, jak má hra fungovat jako celek a jak dosáhnout cíle hry. Podsekcí 2.1.2 vysvětluje návrh herního pole, na kterém se hra odehrává a sekce 2.1.3 poté popisuje, jak se chovají jednotlivé postavy na navrženém herním poli.

#### 2.1.1 Cíl a chování hry

#### 2.1.2 Herní mapa

#### 2.1.3 Postavy a jejich chování

### 2.2 Herní textury

Tato sekce popisuje návrh textur, které se zobrazují na výstup. V podsekcí 2.2.1 vysvětluji způsob použití barev pro výstup. Sekce 2.2.2 se zabývá tím, v jaké podobě jsou textury, které se mají propisovat.

#### 2.2.1 RGB

#### 2.2.2 Textury jako matice barev

## Kapitola 3

# Implementace

Tato kapitola se zabývá návrhem číslicových obvodů pro jednotlivé logicky oddělené bloky na základě herní logiky a jejich implementací. Sekce 3.1 a 3.2 popisují číslicové obvody navržené pro čtení vstupních dat z klávesnice a generování výstupu na monitor. Sekce 3.3 poté vysvětluje vnitřní zapojení jednotlivých bloků zajišťujících logickou funkčnost hry. V sekci ?? je poté popsáno kompletní zapojení všech částí do funkčního celku.

### 3.1 Modul pro čtení z PS/2 klávesnice

V této sekci vysvětlují číslicový obvod, který jako celek zpracovává vstupní signál z klávesnice a generuje daný výstup. Podsekce 3.1.1 rozebírá číslicový obvod pro zpracování vstupního signálu a podsekce 3.1.2 řeší číslicový obvod pro generování výstupního signálu.

#### 3.1.1 Zpracování vstupu

#### 3.1.2 Generování výstupu

### 3.2 Modul pro výstup na VGA monitor

Tato sekce popisuje číslicový obvod, který zařizuje funkčnost monitoru a možnost propsání výstupu na něj. V podsekci 3.2.1 popisují obvod, který zajišťuje funkčnost a v podsekci 3.2.2 vysvětlují, jak propisují textury na monitor.

### **3.2.1 Časování VGA**

### **3.2.2 Propsání textur na monitor**

## **3.3 Herní logika**

V této sekci popisují číslicový obvod, který zajišťuje vnitřní funkcionalitu samotné hry. V podsekcí 3.3.1 je vysvětlený obvod, který řeší herní plochu, na které se postavy pohybují. Podsekcí 3.3.2 popisuje obvod ovládající postavy ve hře a v podsekcí 3.3.3 je vysvětlen obvod, který řeší splnění cílů hry.

### **3.3.1 Herní mapa**

### **3.3.2 Ovládání postav**

### **3.3.3 Cíle hry**

# Kapitola 4

## Testování

## Závěr

# Literatura

- [1] G. Boole, *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities.* Dover Publications, 1854.
- [2] doc. Ing. Hana Kubátová CSc., *Struktura a architektura počítačů s řešenými příklady*, 2nd ed. Thákurova 1, 160 41, Praha 6: České vysoké učení technické v Praze, 2018.