



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

ŘÍDÍCÍ OBVOD S ROZHRANÍM HDMI PRO MODULÁRNÍ LED DISPLEJE

DRIVER UTILIZING HDMI INTERFACE FOR MODULAR LED DISPLAYS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Bartek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vojtěch Dvořák

BRNO 2016



VYSOKÉ UČENÍ FAKULTA ELEKTROTECHNIKY
TECHNICKÉ A KOMUNIKAČNÍCH
V BRNĚ TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**
Ústav mikroelektroniky

Student: Bc. Tomáš Bartek

ID: 146781

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Řídící obvod s rozhraním HDMI pro modulární LED displeje

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte vlastnosti rozhraní HDMI a navrhnete obvod pro zobrazení přijatých dat na modulárních LED displejích. Vyberte vhodný obvod FPGA s ohledem na požadovanou rychlost a plochu a do něj implementujte přijímač dat z rozhraní HDMI. Řídící obvod FPGA bude schopen komunikovat s nadřazenou jednotkou pomocí standardu RS485, detekovat chybné LED diody v modulárních LED displejích a přijímat data ze senzorů teploty. Návrh v jazyce VHDL proveďte s ohledem na maximální univerzálnost kódu pro různé rozlišení obrazu a různou velikost LED displeje.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 8.2.2016

Termín odevzdání: 26.5.2016

Vedoucí práce: Ing. Vojtěch Dvořák

Konzultant diplomové práce:

doc. Ing. Lukáš Fajcik, Ph.D., předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá modernizací na poli informačních LED panelů. Jejím hlavním cílem je implementace vstupního HDMI rozhraní do obvodu FPGA, který řídí jednotlivé modulární LED displeje, ale rovněž implementace pokročilých funkcí jako komunikace s řídicí jednotkou, teplotní ochrana a detekce vadných LED.

KLÍČOVÁ SLOVA

Informační panely, HDMI, LED, VHDL, FPGA

ABSTRACT

This work deals with modernization of information LED panels. It mainly focuses on utilizing input HDMI interface into FPGA, which controls modular LED displays, but also on ensuring professional functions such as communication with control unit, thermal security and detection of faulty LED.

KEYWORDS

Information panels, HDMI, LED, VHDL, FPGA

BARTEK, T. *Řídicí obvod s rozhraním HDMI pro modulární LED displeje*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 58 s. Vedoucí práce Ing. Vojtěch Dvořák.



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Technická 12, CZ-61600 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře vybudované v rámci projektu CZ.1.05/2.1.00/03.0072 **Centrum senzorických, informačních a komunikačních systémů (SIX)** operačního programu Výzkum a vývoj pro inovace.



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Řídící obvod s rozhraním HDMI pro modulární LED displeje“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Vojtěchu Dvořákovi za odborné vedení, konzultace a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	10
1 Návrh konceptu řízení	11
2 HDMI rozhraní	13
2.1 Fyzická vrstva	13
2.1.1 HDMI kabel	13
2.1.2 HDMI konektor	14
2.1.3 Rozdílová signalizace	15
2.2 Signálová vrstva a kódování dat	16
2.2.1 Řídící perioda	16
2.2.2 Datová perioda	18
2.2.3 Obrazová perioda	19
2.3 Identifikační data	21
2.4 Další funkce HDMI rozhraní	23
3 Modulární LED displeje	24
3.1 Datové rozhraní	24
3.2 Další funkce LED modulů	25
4 Blokový návrh FPGA obvodu	27
4.1 HDMI rozhraní	28
4.2 Zpracování vstupních dat	31
4.3 Identifikační data	32
4.4 Pulsně šířková modulace	33
4.5 Paměť	35
4.5.1 Generování adresy pro čtení z paměti	35
4.6 Testovací obrazce	35
4.7 Generátor časových signálů	36
4.8 MODBUS protokol	36
4.8.1 Rozhraní RS485	37
4.8.2 Paměť protokolu	37
4.8.3 Řízení FPGA	37
4.9 Teplota	38
4.9.1 Řízení teplotních čidel	38
4.10 Detekce chybných LED	43
5 Výsledky návrhu FPGA obvodu	45

Závěr	47
Literatura	49
Seznam symbolů, veličin a zkratk	51
Seznam příloh	53

SEZNAM OBRÁZKŮ

1.1	Blokové schéma konceptu práce	11
1.2	Testovací kit	12
2.1	Vrstvový model standardu HDMI	13
2.2	Řez HDMI kabelem [5]	14
2.3	Srovnání různých typů konektorů HDMI	14
2.4	Architektura HDMI spojení [3]	15
2.5	Přehled rozložení režimů přenášených dat pomocí HDMI rozhraní [3]	16
2.6	Časová struktura datové periody [3]	18
2.7	Algoritmus enkodování obrazových dat [3]	20
2.8	Algoritmus dekódování obrazových dat [3]	21
2.9	Příklad základního bloku EDID dat	22
3.1	Schematické zapojení prototypu LED panelu	24
3.2	Vstupní konektor LED modulu	25
3.3	Časový diagram řízení LED modulu se zvýrazněným režimem detekce [10]	25
4.1	Blokové schéma návrhu FPGA obvodu	27
4.2	Blokové schéma HDMI komponenty v FPGA	28
4.3	Hodinové signály pro zpracování HDMI signálu	29
4.4	Blok ISERDES2 v obvodu Spartan 6 XC6SLX9 [18]	30
4.5	Princip vyhledávání rámců v sériovém toku	31
4.6	Stavový automat pro I ² C komunikaci	33
4.7	Stavový diagram pro generátor PWM	34
4.8	Formát datové zprávy MODBUS protokolu [6]	37
4.9	Centrální stavový automat pro měření teploty LED modulů	38
4.10	Inicializace 1-Wire senzorů na sběrnici	39
4.11	Čtení jednoho bitu z 1-Wire sběrnice	39
4.12	Odeslání 1-Wire příkazu, resp. ROM čísla senzoru, po sběrnici	40
4.13	Vyhledávání senzorů připojených do 1-Wire sběrnice	41
4.14	Spuštění konverze teploty a následné vyčtení ze senzorů po 1-Wire sběrnici	42
4.15	Přiřazení 1-Wire čidel ke konkrétním LED modulům	43
4.16	Detekce vadných LED	44

SEZNAM TABULEK

1.1	Vybrané parametry obvodu Spartan 6 XC6SLX9-3TQG144I	11
2.1	Variace datové hlavičky během řídicí periody [3]	17
2.2	Přiřazení kontrolních bitů do datových kanálů s TMDS kódováním [3]	17
2.3	Předpis enkodování kontrolních bitů [3]	18
4.1	Popis časových signálu v návrhu FPGA	36
5.1	Přehled využití FPGA pro různé rozlišení LED panelu	46
5.2	Přehled využití FPGA po implementaci pro řízení prototypu LED panelu	46

ÚVOD

Tato práce se věnuje oblasti informačních LED panelů, které jsou v dnešní době součástí každodenního života nespočtu lidí. Zejména se věnuje inovaci na poli ovládání těchto LED panelů z pohledu zdroje video signálu. V současné době se za tímto účelem využívají mikroprocesorové jednodeskové řídicí jednotky, jejichž výstupní video rozhraní je proměnlivé v závislosti na použitém mikroprocesoru a ve většině případů není možné přenášet výstupní video obraz na větší vzdálenosti.

Cílem této práce je vytvořit takový návrh, který bude využívat pro příjem obrazových dat standardizované HDMI rozhraní. HDMI rozhraní je v současnosti přítomno téměř na všech zařízeních generujících obraz, a tudíž bude možné řídit LED panel pomocí jednodeskových linuxových počítačů, osobních počítačů, tabletů i mobilních telefonů. Jedinou podmínkou je přítomnost výstupního HDMI rozhraní.

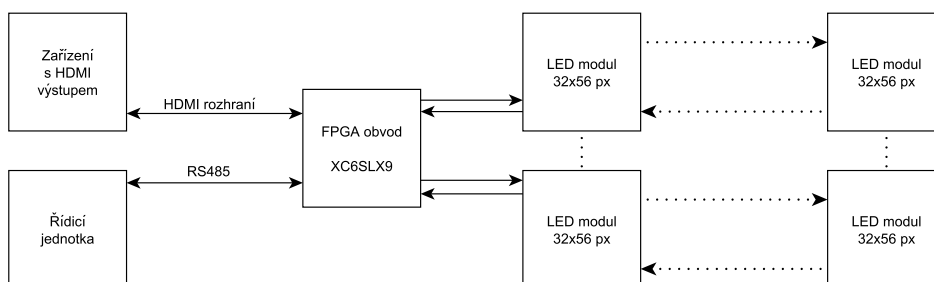
Práce si rovněž klade za cíl implementovat do návrhu pokročilé funkce pro řízení LED panelů. Patří sem schopnost detekce vadných LED na modulech, a to včetně jejich přesné pozice v rámci panelu. Dále je to komunikace FPGA obvodu s nadřízenou jednotkou a možnost vzdáleného řízení FPGA obvodu.

Profesionální LED moduly obsahují i teplotní čidla, a proto do návrhu bude zařazeno teplotní řízení a měření teploty.

Teoretická část práce se bude převážně věnovat HDMI rozhraní. Budou představeny jednotlivé vrstvy rozhraní a detailně rozebráno kódování, které standard HDMI používá. V praktické části bude podrobně představen návrh FPGA obvodu včetně dosažených výsledků.

1 NÁVRH KONCEPTU ŘÍZENÍ

Cílem této práce je navrhnout a implementovat nový způsob řízení LED modulů, který bude využit firmou ELEKON, s.r.o. Na obrázku č. 1.1 je zobrazeno blokové schéma nového konceptu řízení. Zásadní rozdíl oproti momentálně používanému systému řízení je v univerzálnosti jednotky, která generuje obrazová data.



Obr. 1.1: Blokové schéma konceptu práce

Předmětem této práce je pouze implementace pro FPGA obvod a nezabývá se ani konstrukcí LED modulů ani řídicí jednotkou.

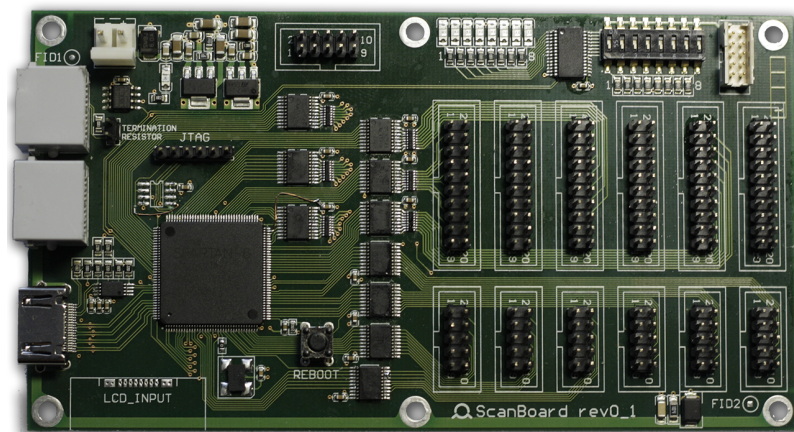
Pro implementaci návrhu byl vybrán obvod firmy XILINX, Inc. Spartan 6 XC6SLX9-3TQG144I. Přehled uvažovaných parametrů obvodu je uveden v tabulce.

Tab. 1.1: Vybrané parametry obvodu Spartan 6 XC6SLX9-3TQG144I

Parametr	Hodnota
počet ekvivalentních logických buněk	9152
velikost blokové paměti RAM	576 Kb
pouzdro	TQG144
počet I/O pinů	102
pracovní teploty	-40 – 100 °C
Bloky PLL	2

Podmínkou pro výběr FPGA obvodu byla nutnost vývodového pouzdra typu QFP. Důvodem byla výroba testovacího kitu firmou ELEKON, s.r.o., která v současnosti nemá technologii pro osazování pouzder typu BGA. Této podmínce vyhovuje pouze právě vybraný obvod.

Náhled na testovací kit je na obrázku č. 1.2.



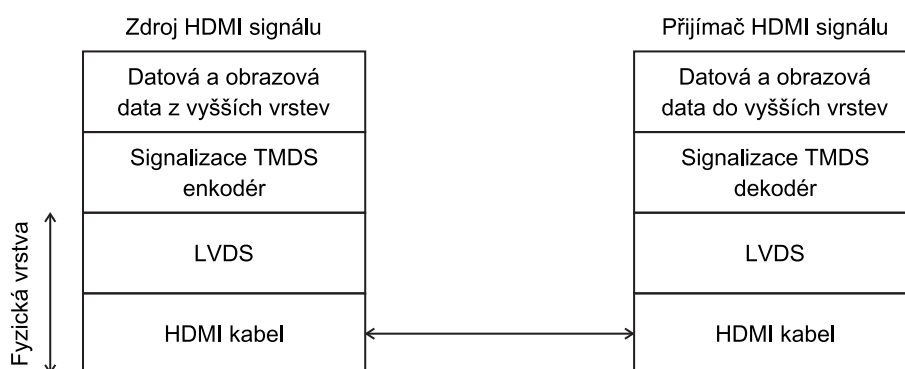
Obr. 1.2: Testovací kit

Jelikož se jedná o prototyp komerčního výrobku, není součástí této práce schéma zapojení testovacího kitu ani použitých LED modulů.

2 HDMI ROZHRAŇÍ

Standard HDMI patří v dnešní době mezi nejrozšířenější video rozhraní. Od verze HDMI 1.0 již bylo vydáno několik dalších, které postupně zvyšují univerzálnost jeho použití a přidávají další funkcionality. Tato práce se bude detailně zabývat verzí HDMI 1.3, která byla vydána v roce 2006.

HDMI rozhraní obsahuje, oproti svému předchůdci DVI, kanál CEC¹ a implementuje přenos až 8kanálového audio signálu. HDMI rozhraní je zpětně kompatibilní s DVI a je možné přes redukci zapojit zdroj DVI dat do HDMI přijímače.



Obr. 2.1: Vrstvový model standardu HDMI

Na popis HDMI rozhraní je aplikován vrstvý model a následující kapitoly budou odpovídat jednotlivým vrstvám v modelu.

2.1 Fyzická vrstva

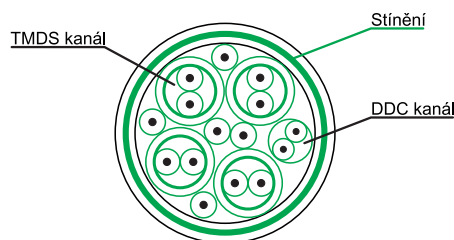
2.1.1 HDMI kabel

Nejnižší vrstvou HDMI rozhraní je samotné fyzické médium, které je tvořeno HDMI kabelem. Náhled na konstrukci HDMI kabelu je na obrázku č. 2.2.

Z obrázku je patrné, že HDMI kabel není konstrukčně příliš jednoduchý a obsahuje několikanásobné stínění. Kabely jsou certifikovány do různých kategorií:

- standardní HDMI kabel,
- standardní průmyslový HDMI kabel,
- standardní HDMI kabel s ethernetem,

¹Datový kanál sloužící k ovládní zobrazovací elektroniky skrze zdroj video signálu.



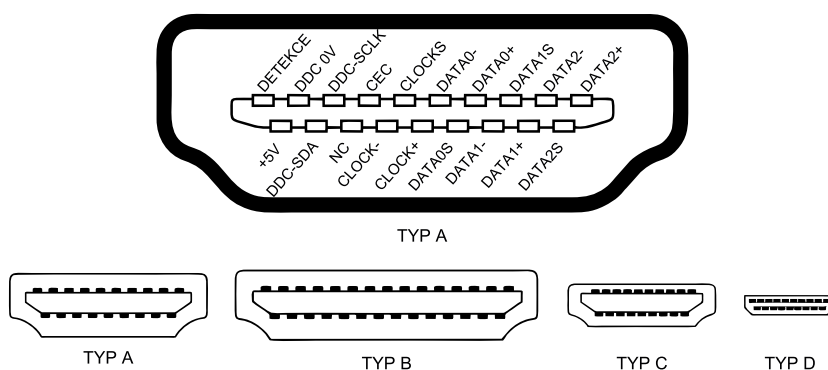
Obr. 2.2: Řez HDMI kabelem [5]

- vysokorychlostní HDMI kabel,
- vysokorychlostní HDMI kabel s ethernetem.

Všechny druhy kabelů se vyznačují poměrně vysokými pořizovacími náklady, které se odrážejí i do prodejní ceny.

2.1.2 HDMI konektor

V současnosti existuje několik různých konstrukčních provedení HDMI konektoru. Typy A, C a D disponují 19 pinů a jedná se pouze o jejich postupnou miniaturizaci. Typ B však obsahuje 29 pinů a má dvojnásobnou šířku pásma – jedná se o alternativu k DVI Dual konektoru. Srovnání HDMI konektorů je na obrázku č. 2.3.



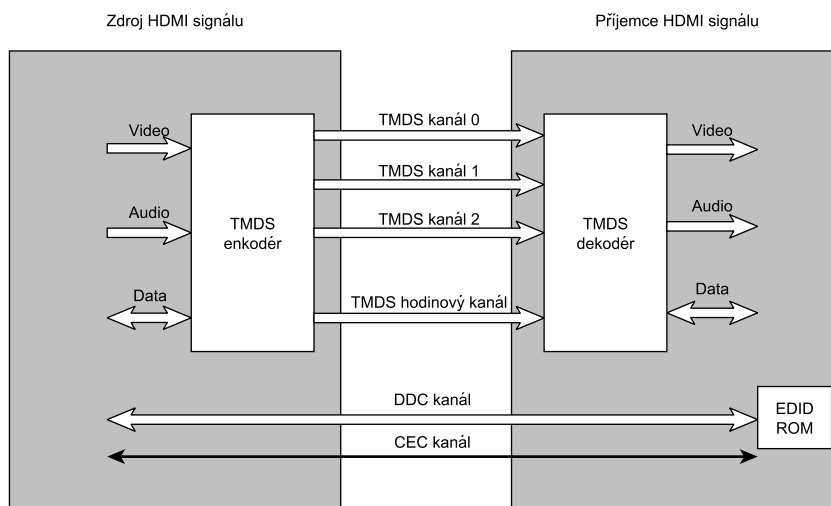
Obr. 2.3: Srovnání různých typů konektorů HDMI

Na obrázku č. 2.4 je zobrazena základní architektura HDMI spojení. HDMI rozhraní se dá rozdělit do několika dalších menších skupin:

- tři datové kanály s TMDS kódováním,
- kanál pro přenos hodinového signálu,
- DDC kanál – umožňuje identifikaci přijímače dat,
- CEC kanál – umožňuje řízení zařízení propojených k HDMI rozhraní,

- napájení – +5 V pro přijímač dat,
- detekce připojení – zabezpečuje funkci *plug & play* (ihned po připojení je možné zařízení používat).

Tato práce se nebude podrobně věnovat CEC kanálu, jelikož pro uvažovanou aplikaci není žádným způsobem zajímavý.



Obr. 2.4: Architektura HDMI spojení [3]

2.1.3 Rozdílová signalizace

Přenášení signálů po kanálech s TMDs kódováním je zabezpečeno pomocí LVDS standardu, který definuje elektrické charakteristiky pro rozdílovou sériovou komunikaci.

LVDS standard využívá pro spojení kroucené páry vodičů, a díky použití nízkonapěťových úrovní signálu umožňuje vysoké rychlosti přenosu dat. LVDS standard se vztahuje pouze na samotné fyzické médium, a je tedy využíván i v jiných aplikacích než je HDMI.

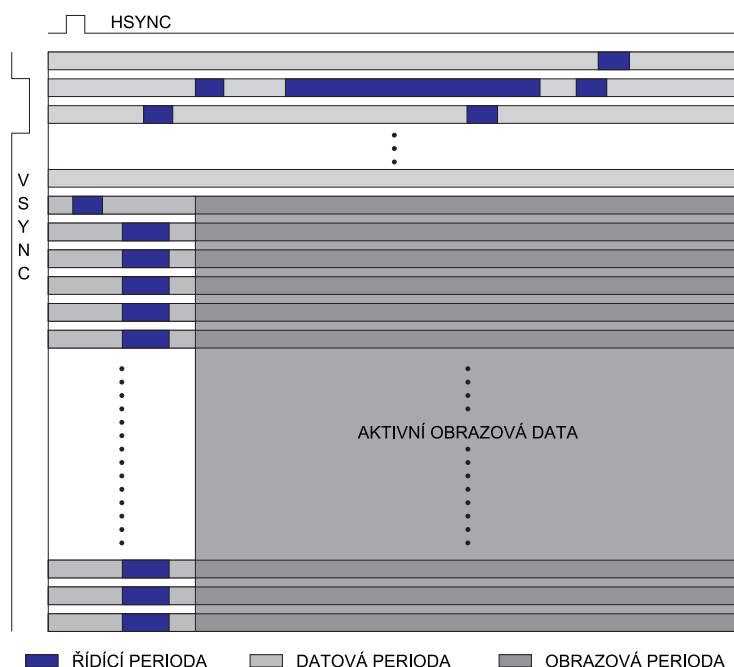
Principem LVDS je vysílání signálu jako rozdílu napětí mezi dvěma vodiči. V praxi to znamená, že zdroj signálu budí vodiče maximálně proudem 4,5 mA s požadovanou orientací. Na straně přijímače signálu jsou vodiče zakončeny 100 až 120ohmovým rezistorem, na kterém dochází k úbytku napětí maximálně 450 mV a orientaci napětí dle orientace proudu. To je interpretováno jako přenášená logická úroveň. Díky použití rozdílového napětí není přijímač ovlivněn aditivním rušením. [11]

2.2 Signálová vrstva a kódování dat

Další vrstvou HDMI rozhraní po LVDS signalizaci je signálová vrstva, která obsahuje TMDS signalizaci. Cílem TMDS signalizace je transformace vstupního n -bitového slova na 10bitový TMDS znak tak, aby výstup splňoval požadavek na minimální počet přechodů mezi log. 1 a log. 0 a stejnosměrná složka signálu v čase byla nula. HDMI standard v rámci komunikace zavádí tři různé režimy přenášených dat:

- řídicí perioda – *Control Period*,
- datová perioda – *Data Island Period*,
- obrazová perioda – *Video Data Period*.

Každý režim je enkodován, resp. dekódován, podle vlastních pravidel. Rozložení jednotlivých režimů v sériovém datovém toku je znázorněno na obrázku č. 2.5.



Obr. 2.5: Přehled rozložení režimů přenášených dat pomocí HDMI rozhraní [3]

2.2.1 Řídicí perioda

Řídicí perioda je využívána k přenášení hlavičky dat a rovněž ji přijímač využívá ke znakové synchronizaci. Během řídicí periody dochází v rámci TMDS k enkodování 2bitových slov na 10bitové v rámci každého ze tří kanálů. Celkem tedy dochází k přenosu 6 bitů: HSYNC, VSYNC, CTL0, CTL1, CTL2 a CTL3. Enkodovací algoritmus je definován tabulkou 2.3.

Tab. 2.1: Variace datové hlavičky během řídicí periody [3]

CTL0	CTL1	CTL2	CTL3	Datový režim
1	0	0	0	obrazová perioda
1	0	1	0	datová perioda

Datová hlavička

Datová hlavička je přenášena vždy před každou změnou režimu dat a určuje tedy, zda další datový rámec patří do režimu datové periody nebo obrazové periody. V tabulce č. 2.1 jsou uvedeny jediné dvě možné kombinace, kterých mohou kontrolní bity CTL0 až CTL3 nabývat. Zbývající bity HSYNC a VSYNC jsou na režimu dat, potažmo na kontrolních bitech, nezávislé a jsou dekodovány samostatně.

Přiřazení kontrolních bitů do jednotlivých kanálů s TMDS kódováním je znázorněno v tabulce č. 2.2.

Tab. 2.2: Přiřazení kontrolních bitů do datových kanálů s TMDS kódováním [3]

Datový kanál	D0	D1
0	HSYNC	VSYNC
1	CTL0	CTL1
2	CTL2	CTL3

Znaková synchronizace

Pro zabezpečení správnosti komunikace je nutné, aby přijímač byl schopen určit hranice jednotlivých TMDS znaků v sériovém datovém toku. To je zabezpečeno rovněž pomocí datové hlavičky. Zatímco TMDS signalizace se snaží během datového režimu a obrazového režimu zabezpečit co nejmenší počet přechodů mezi log. 1 a log. 0 (více v kapitolách 2.2.3 a 2.2.2), tak během přenášení hlavičky dat probíhá enkodování kontrolních bitů dle tabulky č. 2.3².

Z tabulky je zřejmé, že TMDS znak obsahuje 7 nebo 8 přechodů mezi log. 1 a log. 0. Díky tomu je umožněno příjemci dat detekovat tento kontrolní znak, interpretovat jej a nastavit správné hranice dalších TMDS znaků v sériovém datovém toku.

²Sloupce D1 a D0 odpovídají kontrolním bitům z tabulky č. 2.2.

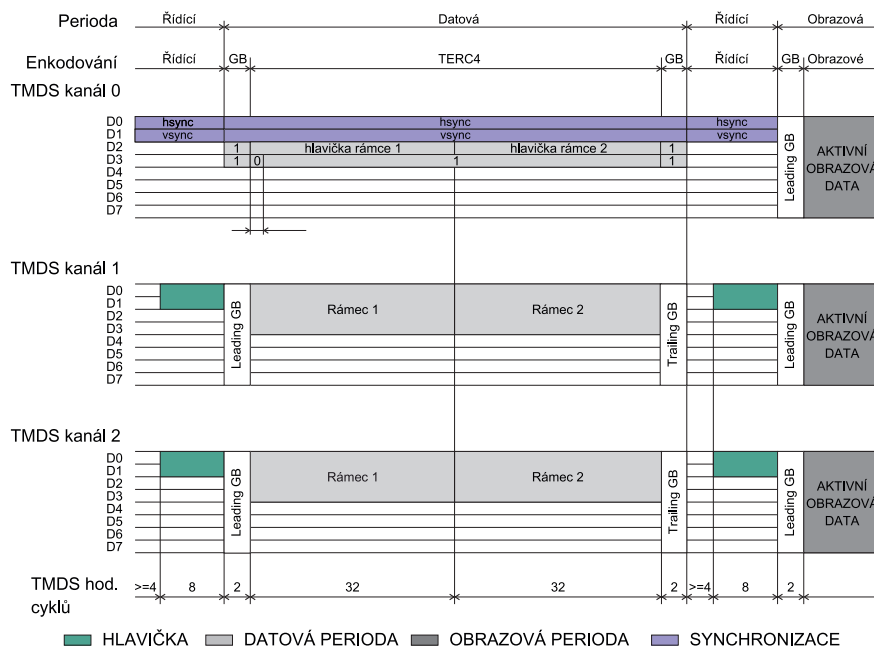
Tab. 2.3: Předpis enkodování kontrolních bitů [3]

D1	D0	TMDS znak
0	1	1101010100
0	0	0010101011
1	1	0101010100
1	0	1010101011

2.2.2 Datová perioda

Datová perioda je využívána k přenosu audio vzorků a dalších pomocných dat. Během datové periody dochází pomocí TMDS signalizace k enkodování čtyř datových bitů do 10bitového slova.

Každou datovou periodu předchází datová hlavička, tak jak je popsáno výše. Po datové hlavičce následuje *Leading Guard Band* a až poté jsou přenášena první data datové periody. Během každého TMDS znaku v datové periodě jsou v rámci kanálu 0 přenášeny signály HSYNC a VSYNC. Náhled na rozložení datové periody je na obrázku č. 2.6.



Obr. 2.6: Časová struktura datové periody [3]

Z obrázku je patrné, že v rámci kanálu 0 je ještě přenášena hlavička datové periody na pozici bitu 2. Kanály 1 a 2 současně přenášejí datové rámce odpovídající hlavičce z kanálu 0. Bezprostředně po těchto datech následuje *Trailing Guard Band*.

Pomocí hlavičky datové periody je možné deklarovat několik různých přenášených datových rámců, jako například audio vzorky, audio hodinový signál, gamut, kontrolní data a další.

Napříč všemi třemi datovými kanály jsou během datové periody TMDS znaky enkodovány pomocí TERC4 (TMDS kódování s redukcí chyb) kódování, které vybírá pouze taková 10bitová slova, která mají nejmenší pravděpodobnost vzájemného ovlivňování napříč všemi kanály. Díky tomu je značně potlačena četnost chyb na těchto kanálech. Jelikož enkodování, resp. dekódování, datové periody není předmětem této práce, přesahuje detailní popis TERC4 kódování rámec této práce.

2.2.3 Obrazová perioda

Obrazová perioda je využívána k přenášení informací o aktivních zobrazovaných pixelech. Každá obrazová perioda je rovněž předcházena datovou hlavičkou. Po datové hlavičce následuje *Video Leading Guard Band*, který má konstantní hodnotu a je definován v HDMI specifikacích. Poté jsou již přenášeny obrazové TMDS znaky, které již nejsou následovány *Trailing Guard Band* bity.

Během obrazové periody dochází k enkodování 8bitové informace do 10bitového TMDS znaku v rámci všech tří datových kanálů s TMDS kódováním. Výsledný TMDS znak jednak minimalizuje počet přechodů mezi log. 1 a log. 0, ale také se snaží zabezpečit co nejmenší stejnosměrnou složku přenášeného signálu v čase. Popis TMDS enkodování se dá rozdělit do dvou samostatných kroků.

První krok

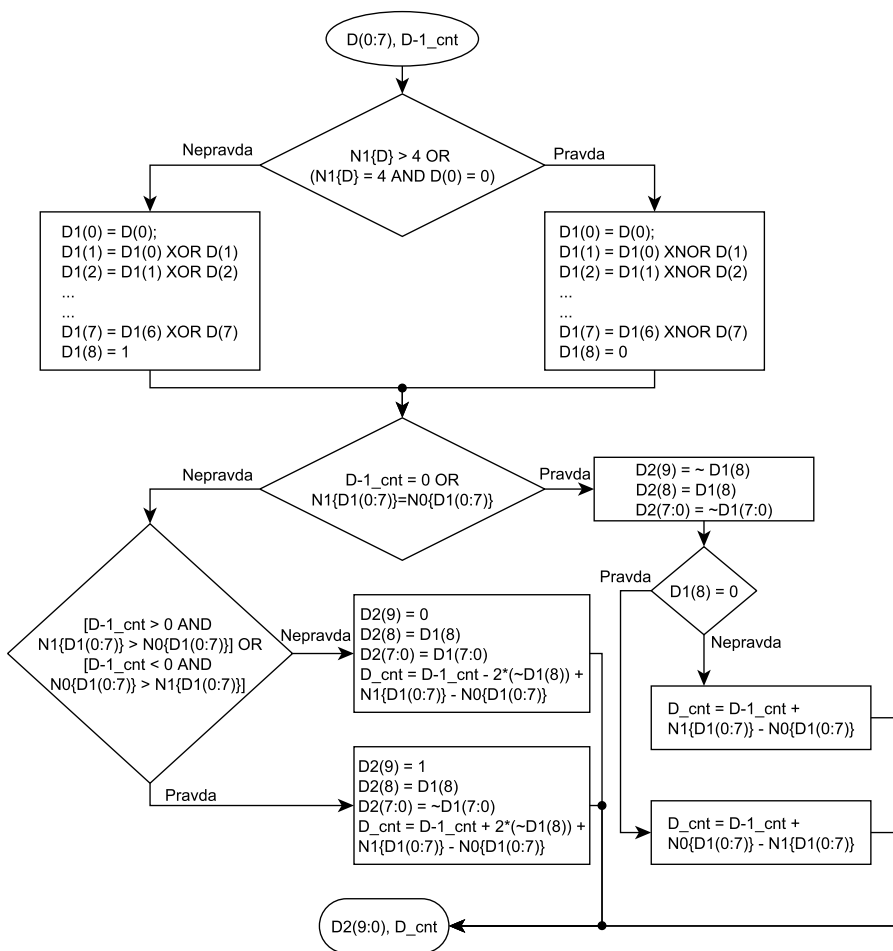
Prvním krokem je převedení vstupního 8bitového slova na 9bitové. Výstupní 9bitové slovo má shodný nejnižší bit se vstupním, další bity jsou výsledkem operace XOR, respektive XNOR. Jaká operace je vybrána, záleží na počtu log. 1 ve vstupním slově. Pokud je počet log. 1 vyšší než 4, je použita logická funkce XNOR. Pokud je naopak počet menší, je použita funkce XOR. Devátý bit výstupního slova reprezentuje, zdali byla použita funkce XNOR nebo XOR.

Druhý krok

Druhá fáze převodu vstupního slova je zaměřená na minimalizaci stejnosměrné složky výstupních TMDS znaků. To je prováděno pomocí selektivního negování vstupních 8bitových slov a deklarováno pomocí desátého bitu. Desátý bit jednoduše ve výsledku reprezentuje, zdali byla ($b_9 = 1$) negace provedena, či nebyla ($b_9 = 0$).

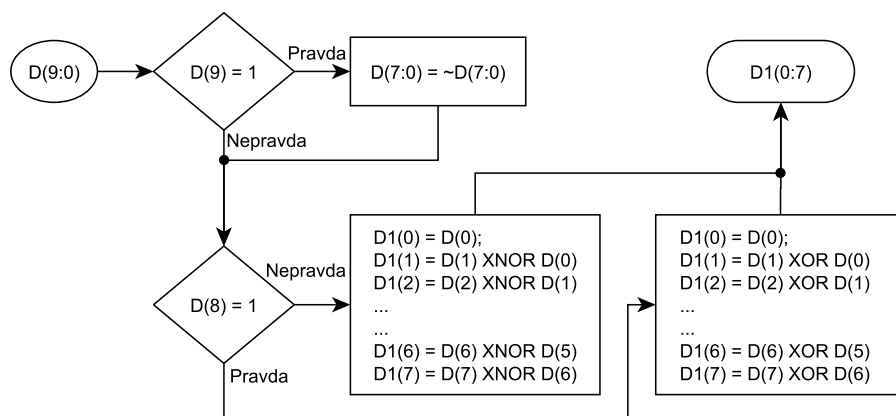
Pokud enkodovací algoritmus registruje na výstupním kanále velké množství log. 1 a současně převáděné slovo obsahuje více log. 1 než log. 0, dochází k inverzi.

Celý enkodovací algoritmus je zobrazen pomocí vývojového diagramu na obrázku č. 2.7.



Obr. 2.7: Algoritmus enkodování obrazových dat [3]

Dekódování TMDS obrazových dat je značně jednodušší než jejich enkodování. Celý proces je znázorněn na obrázku č. 2.8. Příchozí data jsou nejdříve na základě nejvyššího bitu znegována nebo zachována. Poté na základě devátého bitu je použita logická funkce XNOR nebo XOR a výstupem této operace jsou již přímo obrazová data.



Obr. 2.8: Algoritmus dekódování obrazových dat [3]

2.3 Identifikační data

HDMI rozhraní obsahuje kromě datových kanálů s TMDS kódováním také datový kanál pro výměnu informací zdroje s přijímačem. Tento kanál se nazývá DDC. Pomocí tohoto kanálu mohou být přenášena i data ve formátu EDID, který definuje standard organizace VESA [13]. Díky této komunikaci je umožněna automatická konfigurace na straně zdroje HDMI signálu podle možností a požadavků HDMI přijímače.

HDMI konektor obsahuje mj. pin pro detekci připojení přijímače. Jakmile zdroj signálu zaregistruje, že byl přijímač připojen, zahájí sériovou komunikaci s cílem získat data ve formátu EDID. Pokud tohoto není dosaženo, zdroj signálu nezačne vysílat HDMI data.

Základní blok EDID dat má velikost 128 bajtů a obsahuje tyto informace o přijímači:

- hlavička (neměnná),
- výrobce, model a rok výroby,
- EDID verze,
- základní informace o displeji – poměr stran, fyzická velikost,

- barevné charakteristiky,
- časování – detailně popsané nativní a další podporované,
- počet dalších EDID bloků,
- kontrolní bajt – takový, aby součet všech bajtů byl roven nule.

Verze EDID 1.3 dovoluje, pomocí bajtu definujícího počet dalších EDID bloků, navýšení až na 256 takovýchto EDID bloků, které mohou nést další informace o přijímači dat. Například konfigurace reproduktorů nebo další podporovaná časování.

```

addr || 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
=====
00h  || 00 FF FF FF FF FF FF 00 4C 62 AA AA 01 00 00 00
10h  || 06 1A 01 03 80 3C 22 78 2E 2F A5 A5 54 50 9E 27
20h  || 10 50 54 00 42 00 01 00 61 C0 01 00 01 00 01 00
30h  || 01 00 01 00 01 00 97 20 56 64 50 00 2D 30 14 32
40h  || A4 01 56 00 53 00 00 1E 00 00 00 FF 00 30 30 30
50h  || 30 30 30 30 30 30 30 30 30 31 00 00 00 FD 00 32
60h  || 4C 1E 53 15 00 0A 20 20 20 20 20 20 00 00 00 FC
70h  || 00 53 43 41 4E 42 4F 41 52 44 0A 20 20 20 00 5D

```

Obr. 2.9: Příklad základního bloku EDID dat

Jelikož cílem této práce je implementace přijímače HDMI dat, je nutné, aby návrh FPGA podporoval i tuto komunikaci. Struktura EDID je zobrazována jako seznam hexadecimálních bajtů seřazených v tabulce, kde první sloupec odpovídá adrese prvního bajtu v řádku. Na obrázku č. 2.9 je zobrazen základní blok EDID dat. V tomto EDID bloku jsou již zakomponovány všechny potřebné parametry pro tuto práci a přímo tato struktura bude použita při samotné implementaci do FPGA. Struktura byla vytvořena pomocí bezplatného programu *Deltacast E-EDID Editor* a definuje například:

- že se jedná o digitální přijímač dat,
- preferované rozlišení 1366x768 bodů,
- obnovovací frekvenci 70 Hz,
- datum výroby rok 2016,
- fyzické rozměry displeje 60x34 cm,
- název zařízení *SCANBOARD* a další.

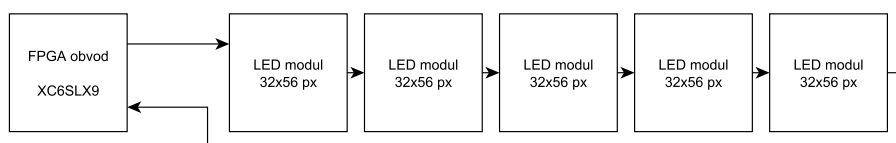
2.4 Další funkce HDMI rozhraní

Mezi další funkcionality HDMI rozhraní patří kanál CEC, který slouží k ovládání všech komponent zapojených do topologie HDMI. V praxi to znamená, že jsme schopni ovládat jediným ovladačem televizi, set-top box, DVD rekordér a další. Jedinou podmínkou je přítomnost HDMI rozhraní mezi zařízeními. Tato funkce není v návrhu uvažována.

Dále rozhraní obsahuje detekční HPD (detekce připojení zařízení) kanál. Díky tomuto kanálu je umožněno zdroji HDMI signálu rozeznat, zdali bylo připojeno zařízení schopné se identifikovat pomocí EDID. Zdroj signálu poté zahájí čtení EDID skrze DDC kanál.

3 MODULÁRNÍ LED DISPLEJE

Návrh FPGA bude probíhat pro řízení pěti monochromatických modulárních LED displejů s rozlišením 32x56 pixelů, které se mohou kaskádně zapojovat za sebe. Displeje disponují vstupním rozhraním, které není standardizováno a je popsáno níže. Celkový rozměr prototypu LED panelu je 160x56 pixelů.



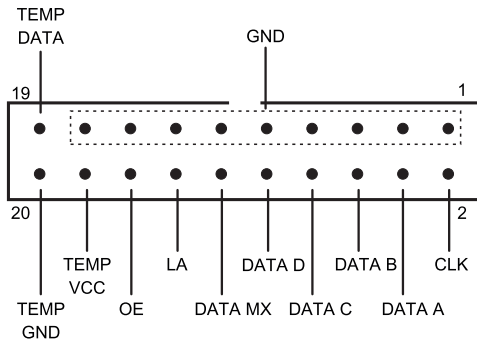
Obr. 3.1: Schematické zapojení prototypu LED panelu

3.1 Datové rozhraní

Na jednom modulárním LED displeji je osazeno 1792 LED. V případě, že by každá LED musela být napojena na vlastní výstup budiče proudu, bylo by použito 112 budičů každý s 16 výstupy. Z důvodu úspory jsou LED seskupeny do čtyř sekcí, v rámci kterých jsou spojeny anody LED a klesá počet budičů na 8. Pro řízené připojování společných anod obsahuje LED modul multiplexovací obvod MC74HC595D [8], který na základě svého datového vstupu (*DATA_MX*) připojuje společné anody LED na napájení. Díky tomu je při multiplexu 1:4 možné řídit celý displej

LED moduly dále obsahují budiče SCT2027 [10], pomocí kterých jsou řízeny jednotlivé LED. Jedná se o posuvný registr, který má jeden datový vstup *DATA_A*, resp. *DATA_B* až *DATA_D*, hodinový vstup *CLK*, jeden latch vstup *LA* a jeden zapínací vstup *OE*.

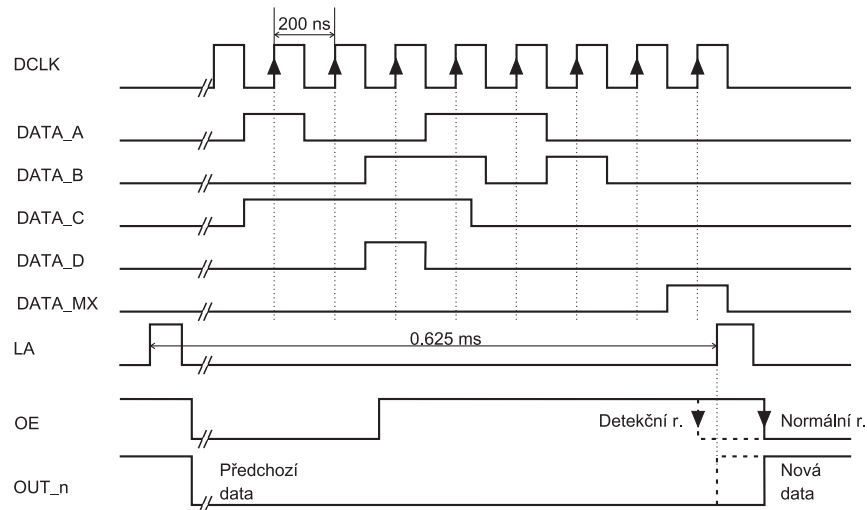
Časový diagram řídicích signálů je znázorněn na obrázku č. 3.3. Rozložení vstupního rozhraní je na obrázku č. 3.2. Pro vyšší spolehlivost spojení jsou mezi datovými vodiči zapojeny společné země.



Obr. 3.2: Vstupní konektor LED modulu

3.2 Další funkce LED modulů

Budiče osazené na LED modulech jsou schopny detekovat chybné LED. Pro detekci chyb je potřeba upravit pořadí řídicích signálů dle obrázku č. 3.3. V případě normálního režimu následuje *OE* impulz až po *LA*. V případě detekce je tomu naopak. Délka *LA* signálu je v detekčním režimu minimálně 200 ns. Detekce je realizována porovnáváním hodnot napětí na výstupech registrů obvodů SCT2027 s interní referencí, a pokud je napětí vyšší než referenční, obvod na příslušné pozici změní bit v posuvném registru. Díky tomu jsme schopni na výstupu registrů detekovat změnu oproti vstupním datům, a tedy detekovat chybu přerušování obvodu – vadnou LED.

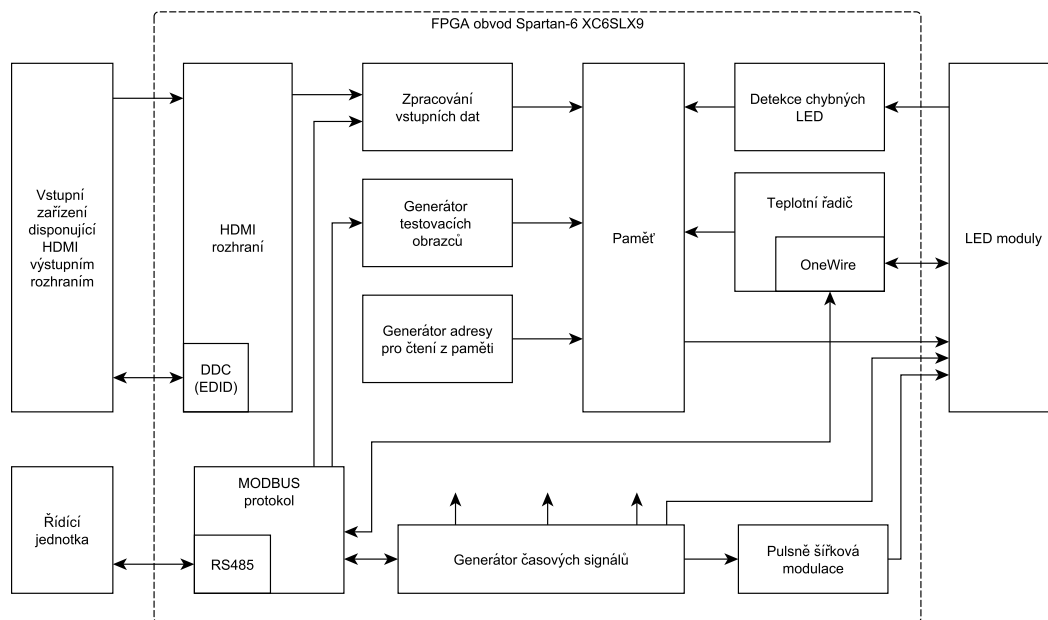


Obr. 3.3: Časový diagram řízení LED modulu se zvýrazněným režimem detekce [10]

Datové rozhraní dále obsahuje tři samostatné signály pro teplotní čidla. Jsou to dva piny pro napájení teplotních čidel *TEMP_VCC*, *TEMP_GND* a jeden datový *TEMP_DATA*. Na LED modulech jsou osazeny 1-Wire čidla od firmy Dallas Semiconductor DS18B20 [2], které umožňují měření teploty na každém modulu samostatně.

4 BLOKOVÝ NÁVRH FPGA OBVODU

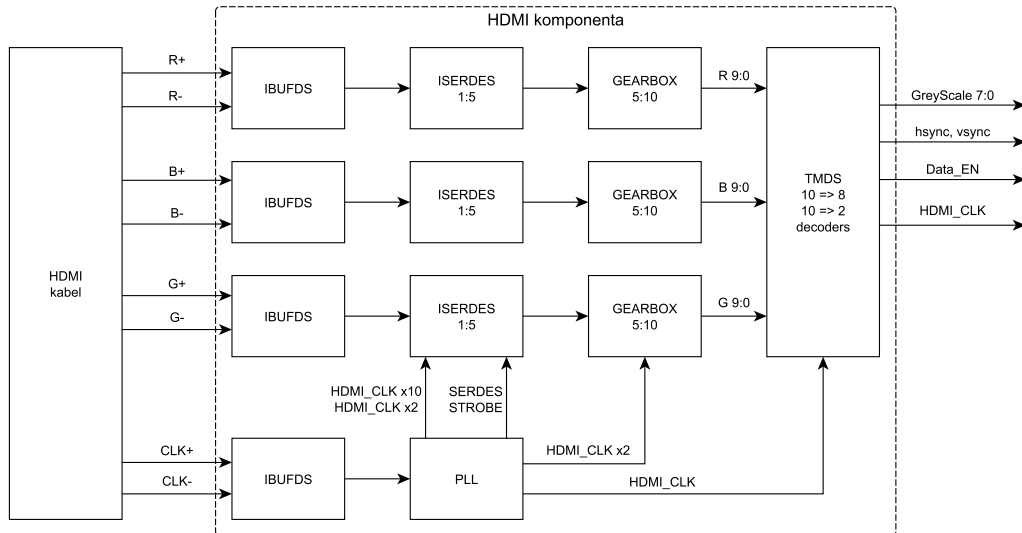
V této části práce jsou postupně popsány všechny bloky návrhu FPGA obvodu. Blokovaný návrh je zobrazen na obrázku č. 4.1.



Obr. 4.1: Blokové schéma návrhu FPGA obvodu

4.1 HDMI rozhraní

Klíčovým blokem celého návrhu je vstupní HDMI komponenta, která zpracovává vstupní HDMI signál a převádí ho na dekodovaná obrazová data. Blokové schéma vstupního rozhraní je možno vidět na obrázku č. 4.2.



Obr. 4.2: Blokové schéma HDMI komponenty v FPGA

Zpracování rozdílových signálů

Všechny vstupní signály z HDMI rozhraní jsou v první řadě zpracovány pomocí vstupního rozdílového bufferu *IBUFDS* [18], který je dostupný v FPGA obvodu na vybraných dvojicích vstupně-výstupních pinů a je možné ho v návrhu využít. Výstupem tohoto bufferu je již jednoduchý signál ve zvoleném napěťovém standardu.

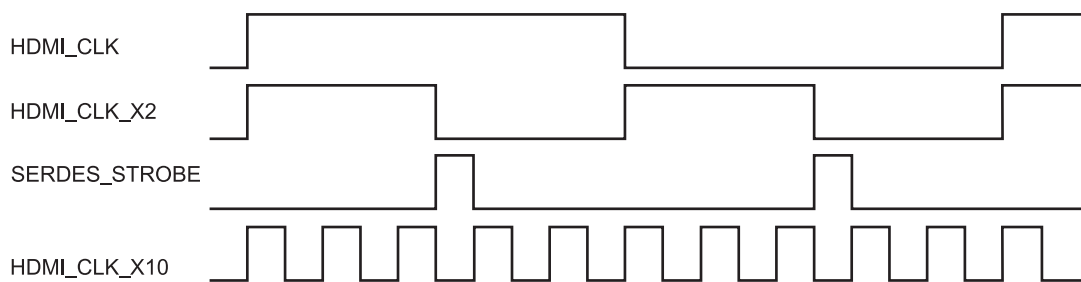
Zpracování hodinového signálu

Podstata přenášení obrazových dat pomocí TMDS kódování byla detailně popsána v kapitole 2.2. Z popisu vyplývá, že každé přenášené slovo je zakódováno pomocí 10bitového TMDS slova. Na to je potřeba myslet i v případě zpracování hodinového signálu, který je přenášen v samostatném HDMI kanálu. V tomto kanálu není uplatněno TMDS kódování a je pouze přenášen hodinový signál pomocí LVDS standardu. Kmitočet přenášeného hodinového signálu odpovídá kmitočtu HDMI signálu před TMDS kódováním. V rámci datových kanálů je ale přeneseno 10 bitů během jedné periody hodinového signálu, proto je nutné pro zpracování dat vygenerovat na straně FPGA desetinásobek přijímaného

kmitočtu hodinového signálu.

Tento požadavek je v FPGA obvodu realizován rovněž dedikovanými bloky. Prvním blokem je *PLL_BASE* [16], který generuje násobky hodinových signálů. Ty jsou následně přivedeny do jednoduchých *BUFG* bloků. Desetinásobek hodinového signálu je rovněž přiveden do bloku *BUFPLL*. Tento blok je vhodný ke zpracování velmi rychlých signálů a umožňuje generování řídicího signálu *SERDES_STROBE*, který je rovněž potřeba ke zpracování HDMI signálu.

Časový diagram všech signálů generovaných pomocí bloku PLL v návrhu FPGA obvodu je zobrazen na obrázku č. 4.3. Signál *HDMI_CLK* je jediný vstupní signál.

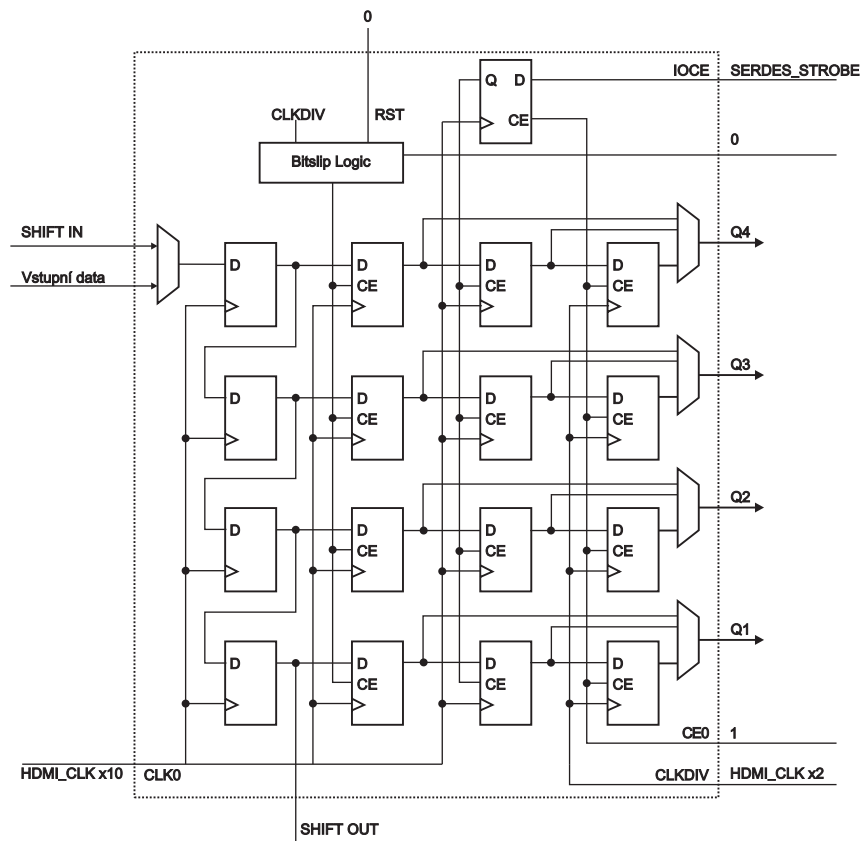


Obr. 4.3: Hodinové signály pro zpracování HDMI signálu

Zpracování vstupních dat

Serializace vstupních dat je zabezpečena pomocí dedikovaných vnitřních bloků *ISERDES2* [18]. Bloky *ISERDES2* slouží v základní konfiguraci k zachytávání sériových toků v poměru 1:2, 1:3 a 1:4. Pro příjem HDMI signálu by však bylo vhodné zachytávat sériová data v poměru 1:5. To je dosaženo pomocí zapojení dvou *ISERDES2* bloků za sebe v režimech *master* a *slave*.

Do bloků *ISERDES2* vstupuje sériový tok vstupních dat, hodinové signály *HDMI_CLK x10*, *HDMI_CLK x2* a *SERDES_STROBE*. Desetinásobek hodinového signálu je využit k zachytávání sériových dat na vstupu. Dvojnásobek hodinového signálu reprezentuje požadovaný výstupní kmitočet zachycených paralelních dat a jejich poměr určuje i poměr zachytávání dat. V tomto případě je to právě 1:5. Signál *SERDES_STROBE* zde slouží jako povolovací impuls a při jeho aktivaci jsou data přepsána na výstup bloku *ISERDES2*.



Obr. 4.4: Blok ISERDES2 v obvodu Spartan 6 XC6SLX9 [18]

Vstup *SHIFT_IN* a výstup *SHIFT_OUT* slouží k propojení dvou *ISERDES2* bloků v režimu *master* a *slave*. *Master* blok předává prostřednictvím výstupu *SHIFT_OUT* signál do vstupu *SHIFT_IN* bloku *slave*.

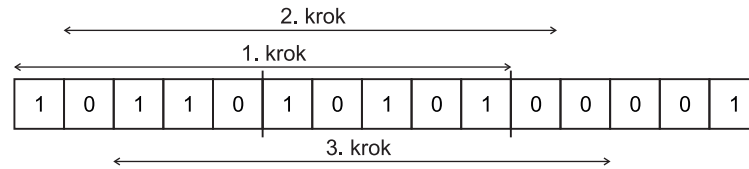
Vyhledávání rámců v sériovém toku

Z předchozího bloku vstupní serializace máme dostupné 5bitové vektory o kmitočtu *HDMI_CLK x2*. Úkolem bloku *GEARBOX* je uložení tří po sobě jdoucích 5bitových slov a generování 10bitových TMDS vektorů o kmitočtu *HDMI_CLK*.

V každém datovém kanálu jsou periodicky přenášeny kontrolní bity, které mají pevné a jedinečné 10bitové ekvivalenty (tabulka č. 2.3). Pokud nalezneme na výstupu bloku *GEARBOX* některá z těchto 10bitových slov, znamená to, že je 15bitový uchovávaný vektor dělen správně.

Za účelem automatické detekce těchto hranic je v FPGA obvodu implementovaný 15bitový čítač, který je nulován detekcí některého ze synchronizačních znaků. Pokud dojde k přetečení čítače, znamená to, že nebyl na výstupu detekován synchronizační

vektor a posune se pomocí signálu *framing* hranice dělení 15bitového vektoru uvnitř bloku *GEARBOX*. Takto se hranice dělení posouvá, než je zachyceno synchronizační slovo. Poté je již čítač periodicky nulován a nedochází k další změně signálu *framing*.



Obr. 4.5: Princip vyhledávání rámců v sériovém toku

Na obrázku č. 4.5 je zobrazen princip vyhledávání správných rámců v sériovém toku, kdy synchronizační slovo bylo nalezeno až ve třetím kroku posunu hranic.

Dekódování dat

Posledním blokem příjmu HDMI dat jsou TMDS dekodéry. V FPGA obvodu je implementován algoritmus z obrázku č. 2.8, který generuje na svém výstupu 8bitová obrazová data. Dále je implementován dekodér podle předpisu z tabulky č. 2.3, který generuje řídicí signály včetně *HSYNC* a *VSYNC*.

Jelikož FPGA obvod řídí monochromatické LED moduly, je výhodné převést výstupní 24bitový RGB signál na 8bitový se stupni šedé. V FPGA obvodu je implementován vztah:

$$Greyscale = \frac{RED + GREEN + BLUE}{4} \quad (4.1)$$

Součet je realizován 10bitovou sčítačkou a dělení odebráním dvou LSB z výsledku součtu.

Celý blok HDMI rozhraní po implementaci využívá 127 LUT, 57 Slices, 120 Slice registrů, jeden blok PLL a tři paměti LUTRAM (obsah posuvného registru v bloku *gearbox* je implementován jako LUTRAM).

4.2 Zpracování vstupních dat

Obrazová data jsou před uložením do paměti FPGA obvodu dekodována blokem pro zpracování vstupních dat. Jelikož obraz na LED panelu je pouze monochromatický, není potřeba uchovávat 8bitovou informaci o stupni šedé každého obrazového bodu, ale postačuje pouze jednobitová informace. Rovněž není

potřeba uchovávat celý přenášený obraz, ale pouze výsek, který je zobrazován na LED panelu.

Pro správnou orientaci v přijímaném obrazu jsou uvnitř bloku dva samostatné čítače: 11bitový pro čítání souřadnice X a 10bitový pro Y. Z bloku paměti MODBUS je přivedena počáteční hodnota X a Y, od které má být obraz ukládán. Velikost panelu je při provozu neměnná a je možné ji nastavit pouze změnou parametrů v samostatné knihovně, která je součástí návrhu FPGA. Z MODBUS paměti je rovněž přivedena prahová hodnota *data_th* pro dekodér vstupních dat a jednobitový signál *data_invert*, kterým je možno invertovat výstupní hodnotu.

Do hlavní paměti RAM je tedy ukládána jednobitová informace v závislosti na souřadnicích obrazového bodu, jeho číselné hodnotě a stavu signálu *data_invert*. Log. 1 v paměti reprezentuje rozsvícený bod na LED panelu. Adresa pro zápis bitu je generována v závislosti na hodnotách čítačů X a Y.

Blok po implementaci využívá 46 LUT, 39 Slices a 44 Slice registrů.

4.3 Identifikační data

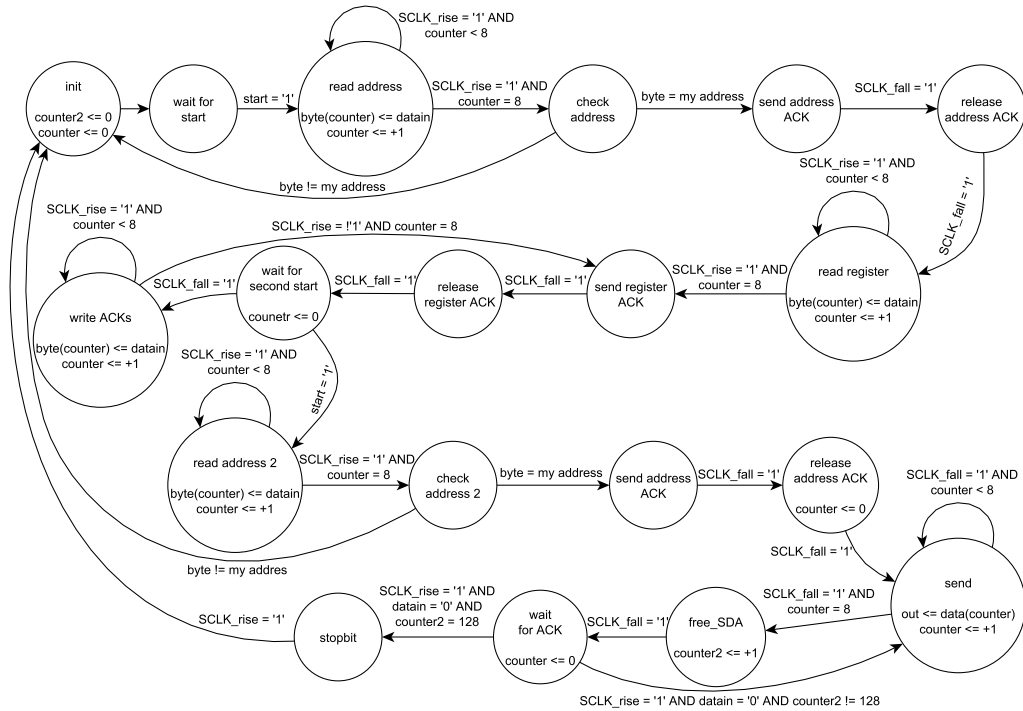
Formát a obsah identifikačních dat je podrobně popsán v kapitole 2.3. Předávání těchto dat je realizováno pomocí samostatného kanálu DDC v rámci HDMI rozhraní a komunikace je založena na standardu I²C.

V FPGA obvodu je komunikace implementována pomocí stavového automatu, který je upraven pouze k odesílání EDID dat. Diagram stavového automatu je na obrázku č. 4.6¹. Adresa, na které FPGA obvod komunikuje, je *A0h*.

Stavový automat čeká ve stavu *wait_for_start* na startovní stav sběrnice. Poté přechází do *read_address*, ve kterém přijme adresu, se kterou *master* chce komunikovat. Pokud se jedná o adresu *A0h*, potvrdí FPGA příjem zápisem *ACK* na sběrnici. Po potvrzení *master* odešle adresu registru, který chce zapsat. FPGA potvrdí příjem a ve smyčce potvrzuje hodnoty registrů, které chce *master* zapisovat. Přijaté hodnoty registrů však nejsou v FPGA obvodu zpracovány, a pouze se zahazují. FPGA obvod pouze čeká na zahájení další komunikace, ve které bude *master* chtít vyčítat data. Jakmile toto nastane, FPGA obvod postupně odešle EDID data a ukončí komunikaci.

Blok po implementaci využívá 75 LUT, 37 Slices a 35 Slice registrů.

¹Setrvání v pracovních stavech je znázorněno pomocí *else* smyček pouze ve vybraných případech. Pro zvýšení přehlednosti stavových diagramů, v této práci, nejsou *else* smyčky uvedeny všech stavech.



Obr. 4.6: Stavový automat pro I²C komunikaci

4.4 Pulsně šířková modulace

Intenzita jasu LED panelu je řízena pomocí pulsně šířkové modulace (dále jen PWM).

PWM je technika přenášení informace za pomoci nastavování různé délky aktivní doby (doba v log. 1) impulzu v rámci jeho definované maximální délky, kdy ke každé různě dlouhé aktivní době je přiřazena jiná informace. S touto technikou lze docílit přenesení vícecestavového signálu pomocí pouze dvoustavového nosiče [1].

V běžných aplikacích se jas LED řídí pomocí změny velikosti proudu, který jí protéká. Budící proud I_0 je však v budičích SCT2027 pevně nastaven pomocí R_{ext} a není možné jej měnit. Proto je v této aplikaci ke každé různé intenzitě jasu LED přiřazena různá aktivní doba pulzu OE signálu. Díky tomu je možné s prodlužující se aktivní dobou zvyšovat i dobu svícení LED a tím i jas LED modulu.

Zvolená obnovovací frekvence LED panelu je 400 Hz , a tedy maximální doba OE je:

$$t_{OE} = \frac{1}{400} = 2,5\text{ ms} \quad (4.2)$$

V případě použitých LED modulů však tento údaj neplatí. Z důvodu snížení počtu datových vodičů, které řídí moduly, dochází k multiplexování čtyř společných anod LED. Zároveň s tím dochází ke snížení počtu potřebných budičů LED. Ve výsledku to však i znamená, že je potřeba 4x větší kmitočet zobrazování jednotlivých multiplexovaných řádků. Se zvýšením kmitočtu se však zmenší i čas povolovacího pulzu OE :

$$t'_{OE} = \frac{t_{OE}}{4} = 0,625 \text{ ms} \quad (4.3)$$

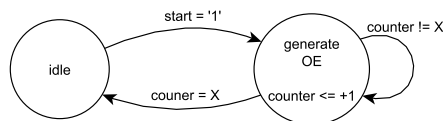
To znamená, že pro PWM je k dispozici impuls o délce $t = 0,625 \text{ ms}$.

Pro rozdělení času $0,925 \text{ ms}$ na 100 úrovní bude vhodné ke každé úrovni přiřadit počet náběžných hran CLK, které je potřeba načíst, abychom dosáhli požadovaného času. Jedna úroveň jasů představuje čas:

$$t_{level} = \frac{0,625 \text{ ms}}{100} = 6,25 \mu\text{s} \quad (4.4)$$

Pokud chceme vyjádřit tento čas v počtu náběžných hran hodinového signálu, musíme ho ještě vydělit dobou periody tohoto signálu:

$$N_{CLK} = \frac{6,25 \mu\text{s}}{20 \text{ ns}} = 312 \quad (4.5)$$



Obr. 4.7: Stavový diagram pro generátor PWM

Generátor PWM je v FPGA obvodu realizován pomocí stavového automatu. Náhled na stavový automat je na obrázku č. 4.7. Stavový automat čeká ve stavu *idle* na startovací impuls. Pokud je impuls zaznamenán, přejde do stavu *generate_OE*, ve kterém se přičítá hodnota čítače a rovněž je vypočtena koncová hodnota čítače:

$$X = brightness \cdot 312 \quad (4.6)$$

Signál *brightness* je 7bitové slovo z paměti MODBUS protokolu. Pokud je stavový automat ve stavu *generate_OE*, je na výstupu z bloku signál OE v log. 0. Jakmile čítač dosáhne hodnotu X , stavový automat se přepne zpět do stavu *idle* a signál OE je v log. 1.

Blok po implementaci využívá 25 LUT, 9 Slices a 18 Slice registrů a jeden blok násobičky DSP14A1.

4.5 Paměť

Vstupní data jsou do paměti ukládána na základě bloku zpracování vstupních dat. V FPGA obvodu je paměť obrazových dat implementována do blokové paměti RAM. Z důvodu vyčítání čtyř různých pozic v paměti zaujímá prototyp LED panelu o rozlišení 160x56 pixelů čtyři blokové paměti. Ty jsou konfigurovány jako 16384x1 bit. LED panel využívá pouze 8960 pozic paměti a v případě navýšení počtu LED modulů by do přesáhnutí hodnoty 16384 nenarůstal počet využitých blokových pamětí.

4.5.1 Generování adresy pro čtení z paměti

LED moduly, ze kterých je panel složen, nemají posuvné sériové registry, resp. budiče SCT2027, seřazeny lineárně. Z důvodu zjednodušení návrhu desky plošných spojů a pokrytí celého panelu pouze pomocí čtyř datových vodičů s multiplexem 1:4 jsou sériové registry zapojeny ve speciálním pořadí. V návrhu FPGA je generování adresy, ze které se mají data na výstup vyčítat, věnován samostatný blok. Ten obsahuje pevnou paměť ROM s předpisem pozic pro čtení jednoho datového vodiče. Z důvodu možné změny a pohodlnějšího napojení na návrh je tento předpis při implementaci načítán ze samostatného textového souboru. Ten je pro ukázkou přiložen v příloze č. P.3.

Paměť obrazu využívá čtyři RAMB16BWER bloky (dále pouze BRAM – Bloková paměť RAM).

4.6 Testovací obrazce

Za účelem jednoznačného otestování správné funkčnosti řízení LED panelu je do návrhu FPGA obvodu zařazen i testovací režim. Ten se dá vzdáleně zapnout skrze MODBUS protokol a je možné nastavit čtyři různé testovací obrazce. Jedná se o běžící kříž a běžící bod a jejich inverze. Dále je možné vzdáleně nastavit i rychlost testovacích obrazců.

Podoba obrazců není vybrána náhodně. Testování běžících jednopixelových linek je nejvíce náročné na správné časování signálů a pomáhá odhalovat chyby v návrhu.

Generátor testovacích obrazců je v FPGA obvodu implementován jako dva páry čítačů. Jedna dvojice reprezentuje X a Y pozici pixelu, který má být uložen

do paměti obrazových dat. Druhý pár představuje X a Y souřadnici rozsvíceného bodu na panelu. V závislosti na zvoleném testovacím obrazci je do paměti ukládána log. 1 reprezentující rozsvícený pixel.

Po implementaci využívá generátor 107 LUT, 33 Slices a 71 Slice registrů.

4.7 Generátor časových signálů

Pro správnou funkci a řízení LED modulů je potřeba v návrhu generovat několik časových signálů. Přehled všech je uveden v tabulce 4.1.

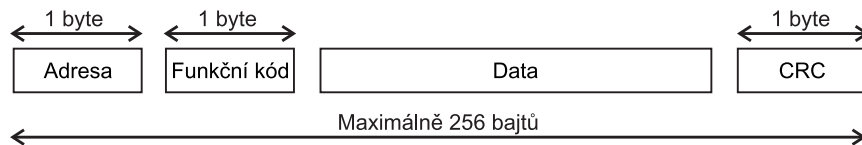
Tab. 4.1: Popis časových signálů v návrhu FPGA

Název signálu	Popis
t_period	změna multiplexovaného řádku na LED modulu
t_latch	přepsání dat na řídicích datových kanálech LED modulů
t_datastart	startovací impuls pro generování řídicích dat pro LED moduly
LA_normal	latch enable impuls – v LED modulech slouží k přepsání hodnoty na vstupu registrů na jejich výstupy
LA_detection	latch enable impuls v případě detekce
OE_normal	výstup bloku pulsně šířkové modulace
OE_detection	v případě detekce je výstupní signál OE logickým součtem OE_normal a OE_detection
DMX_normal	řídicí signál pro multiplexovací obvod v LED modulech
DCLK	hodinový signál o kmitočtu 5 MHz

Všechny časové signály kromě *OE_normal* jsou generovány na základě hlavního 17bitového čítače a po implementaci je využito 53 LUT, 20 Slices a 29 Slice registrů.

4.8 MODBUS protokol

Součástí návrhu FPGA obvodu je i vzdálené řízení FPGA obvodu. Za tímto účelem byl vybrán a implementován otevřený protokol MODBUS. Ten definuje datové zprávy pro komunikaci mezi serverem a klientem a umožňuje připojení více klientů.



Obr. 4.8: Formát datové zprávy MODBUS protokolu [6]

V návrhu byly implementovány pouze 3 potřebné funkční kódy: *03h* – čtení registrů, *10h* – zápis registrů a *14h* – čtení souboru. Čtení a zápis registrů slouží k vyčítání parametrů a řízení FPGA obvodu. Čtení souboru je implementováno za účelem předání přesných pozic chybných LED. Každý z funkčních kódů má nepatrně odlišnou skladbu datového rámce. Struktura jednotlivých rámců je zobrazena v příloze č. P.2.

4.8.1 Rozhraní RS485

Komunikaci MODBUS protokolu je možné implementovat na libovolnou fyzickou vrstvu. V tomto návrhu byl vybrán standard RS485. RS485 je standard pro sériovou komunikaci, který využívá ke komunikaci pouze dva vodiče a je možné připojit na sběrnici až 32 zařízení. Logické úrovně jsou reprezentovány pomocí rozdílového napětí mezi vodiči.

4.8.2 Paměť protokolu

Veškerá data přijatá MODBUS protokolem jsou ukládána do paměti v FPGA obvodu. Paměť je složena z 512 16bitových registrů. Detailní popis registrů je přiložen v příloze č. P.1.

Vzhledem k velikosti paměti je zvolena její implementace do blokové paměti RAM.

4.8.3 Řízení FPGA

Na základě registrů MODBUS protokolu jsou pro FPGA generovány řídicí signály. Toto je řešeno čítačem 0 až 511, který donekonečna prochází obsah paměti, a pokud právě kontrolovaný registr obsahuje řídicí informaci, tak je vygenerován příslušný řídicí signál. Jedná se například o jas LED panelu, příkaz ke spuštění detekce, teplotní limity, testovací obrazce a další. Kompletní přehled je v tabulce č. P.1.

Celý blok MODBUS protokolu po implementaci využívá 401 LUT, 183 Slices, 268 Slice registrů. Jedna paměť BRAM je využita pro ukládání přijaté zprávy a 2 paměti BRAM pro ukládání všech registrů protokolu MODBUS.

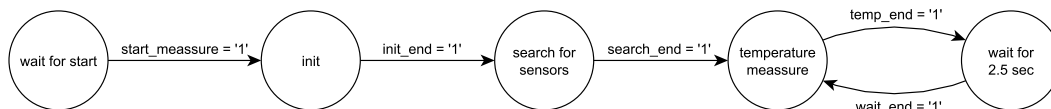
4.9 Teplota

Důležitou součástí návrhu FPGA obvodu je měření teploty a v závislosti na změřených teplotách jednoduché teplotní řízení. Pro měření teploty je na každém LED modulu osazen obvod DS18B20. Jedná se o programovatelný 9 až 12bitový digitální teploměr schopný komunikovat skrze 1-Wire sběrnici. Nejzajímavější vlastností 1-Wire sběrnice je, že pro obousměrnou komunikaci je potřeba pouze jeden datový signál. Samotné čidlo navíc potřebuje společnou zem a napájení. Je však možné používat čidlo i bez napájení pouze s datovým pinem a společnou zemí. V tomto režimu je čidlo napájeno z datového kanálu. Tento způsob však s sebou nese nutné zpomalení datových přenosů a další omezení.

Z důvodu přítomnosti pouze jednoho datového kanálu je standard komunikace poměrně rozsáhlý a k jeho implementaci v FPGA bylo nutné vytvořit velký stavový automat. Ten byl kvůli přehlednosti a univerzálnosti rozdělen do více menších.

4.9.1 Řízení teplotních čidel

Hlavní stavový automat je zobrazen na obrázku č. 4.9. Jedná se o nekonečnou smyčku měření teploty v intervalech 2,5 s, kterou předchází inicializace teplotních čidel a vyhledání unikátních identifikačních čísel čidel². Ve stavech *init*, *search_for_sensors* a *temperature_meassure* jsou spuštěny jiné stavové automaty a přechod do dalšího stavu je podmíněn jejich ukončením.

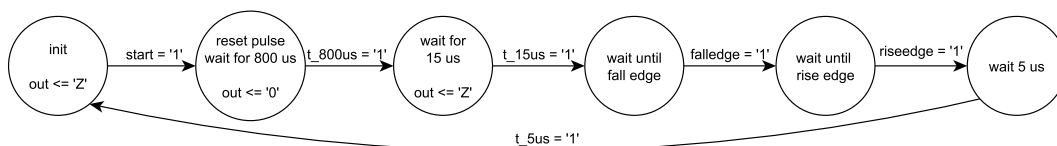


Obr. 4.9: Centrální stavový automat pro měření teploty LED modulů

²Více v této podkapitole v sekci *Vyhledávání teplotních čidel*.

Inicializace čidel

Základním stavem sběrnice je inicializace čidel. Toho je dosaženo pomocí vybuzení log. 0 na sběrnici po dobu $800\ \mu\text{s}$. Poté FPGA nastaví na výstup vysokou impedanci a čeká $15\ \mu\text{s}$. Díky PULLUP rezistorům je sběrnice zvednuta do stavu log. 1. Po uběhnutí časové prodlevy začne FPGA vzorkovat sběrnici za účelem detekce sestupné hrany. Pokud je tato hrana detekována, znamená to, že na sběrnici jsou přítomny 1-Wire čidla. V takovém případě je na sběrnici alespoň jedno funkční čidlo, které je připraveno komunikovat. Stavový automat vyčká na ukončení tohoto ohlašovacího impulsu, poté čeká $5\ \mu\text{s}$ a je ukončen.

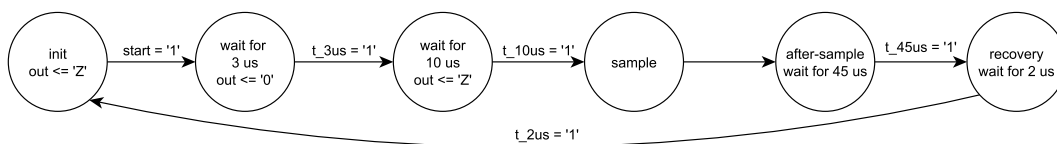


Obr. 4.10: Inicializace 1-Wire senzorů na sběrnici

Čtení ze sběrnice

V klidovém stavu je výstup FPGA nastaven do vysoké impedance. Čtení ze sběrnice probíhá po jednotlivých bitech. Pokud chce FPGA zahájit čtení bitu, nastaví výstup do log. 0 po dobu $3\ \mu\text{s}$. Po tomto signálu je výstup FPGA opět nastaven do stavu vysoké impedance a čidlo vybudí sběrnici do stavu log. 1 nebo log. 0. FPGA obvod čeká $10\ \mu\text{s}$, a poté navzorkuje hodnotu sběrnice a uloží ji jako příchozí bit.

Poté následuje další čekání na ukončení odesílání bitu a přechod do základního stavu.



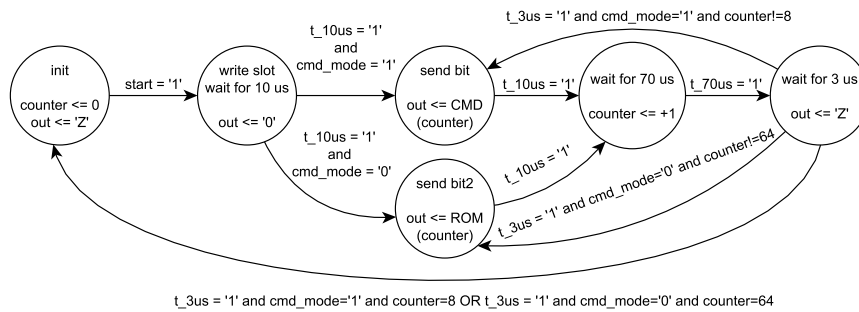
Obr. 4.11: Čtení jednoho bitu z 1-Wire sběrnice

Zápis na sběrnici

Zápis bitů na 1-Wire sběrnici probíhá podobným způsobem jako čtení bitů z čidel. Nejdříve je po dobu $10\ \mu\text{s}$ nastaven výstup FPGA do log. 0. Poté následuje vybuzení sběrnice požadovanou hodnotou log. 1 nebo log. 0. Dalším stavem je čekání $70\ \mu\text{s}$ během kterých 1-Wire čidlo navzorkuje hodnotu na sběrnici.

Po uplynutí časové prodlevy FPGA obvod nastaví výstup do vysoké impedance, která slouží k oddělení následujících bitů.

Stavový automat pro odeslání 8bitového příkazu, resp. 64bitového ROM čísla senzoru, je pouze modifikován k opakovanému odeslání 1bitové informace.

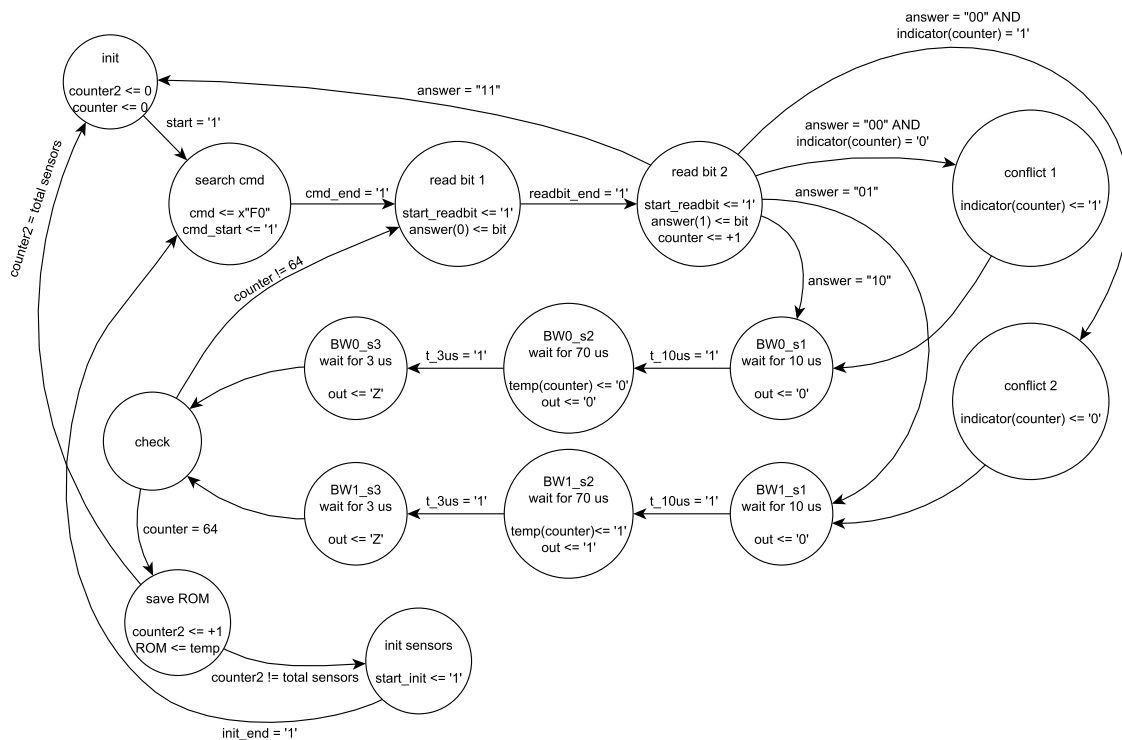


Obr. 4.12: Odeslání 1-Wire příkazu, resp. ROM čísla senzoru, po sběrnici

Vyhledání teplotních čidel

Každé teplotní čidlo obsahuje unikátní 64bitové identifikační číslo. Prvních 8 bitů je výrobní číslo modelu čidla. V případě DS18B20 je to 28h. Následujících 48 bitů je unikátní číslo. Posledních 8 bitů je CRC předchozích 8 bitů [2].

Pro komunikaci FPGA obvodu s více 1-Wire senzory, je nutné zjistit unikátní čísla všech čidel přítomných na sběrnici. Vyhledávací algoritmus je popsán pomocí stavového automatu na obrázku č. 4.13. Pro snadnější pochopení algoritmu si zde rozepíšeme jednoduchou modelovou situaci se třemi teplotními čidly.



Obr. 4.13: Vyhledávání senzorů připojených do 1-Wire sběrnice

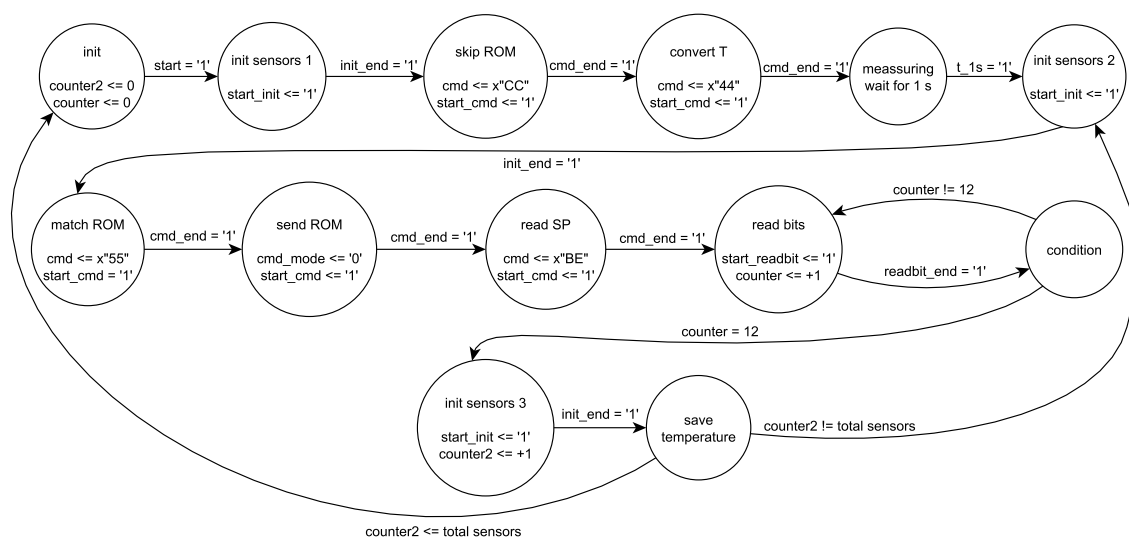
Teplotní čidla budou mít zjednodušená 3bitová identifikační čísla *001*, *010* a *100*.

1. Vyhledávání je spuštěno odesláním příkazu *Search ROM (F0h)*.
2. FPGA spustí čtení prvního bitu senzorů. Čidla na to reagují tak, že na staví na své výstupy první bity svých ROM. Výsledkem na sběrnici je logický součin všech výstupů. V našem případě log. 0.
3. FPGA spustí čtení dalšího bitu. Čidla nastaví na své výstupy negaci svého prvního bitu v ROM. Výsledkem je opět logický součin. V našem případě opět log. 0.
4. V tuto chvíli mohou existovat 4 různé kombinace prvních 2 vyčtených bitů:
 - 00: Na sběrnici jsou alespoň dvě čidla s různou hodnotou prvního bitu.
 - 01: Na sběrnici jsou připojena čidla s hodnotou prvního bitu 0.
 - 10: Na sběrnici jsou připojena čidla s hodnotou prvního bitu 1.
 - 11: Na sběrnici není přítomno žádné čidlo.
5. Jelikož výsledek byly dvě log. 0, zvolíme, že chceme pokračovat v čtení čidla s bitem 0 a uložíme příznak, že v příštím kole máme zvolit 1.
6. Zapiše na sběrnici log. 0. Díky tomu všechna čidla, která mají na první pozici log. 1, jsou odpojena a dále komunikují pouze čidla s hodnotou 0.
7. Kroky 2 až 6 opakujeme až do vyčtení celého identifikačního čidla senzoru.

Algoritmus opakujeme až do nalezení všech teplotních čidel na sběrnici s využitím získaných příznaků v kroku 5.

Měření teploty

Posledním stavovým automatem je samotné měření teploty. Měření je zahájeno inicializací senzorů. Poté FPGA odešle 1-Wire příkaz *Skip ROM (CCh)*, který zabezpečí, že budou data přijímat všechny senzory na sběrnici. Poté následuje příkaz *Convert T (44h)*, po kterém je ukončena komunikace a čidla zahájí převod teploty do své interní paměti.



Obr. 4.14: Spuštění konverze teploty a následné vyčtení ze senzorů po 1-Wire sběrnici

Po uplynutí 1 s FPGA opět inicializuje čidla a zahájí vyčítání teploty z konkrétního senzoru pomocí příkazu *Match ROM (55h)*, který je následován 64 bity identifikačního čísla. Po odeslání zvolené čidlo zapisuje na sběrnici 12bitovou informaci o změřené teplotě. FPGA obvod přeruší komunikaci inicializačním impulzem a uloží přijatou teplotu do paměti. Postup je opakován pro všechna teplotní čidla na sběrnici.

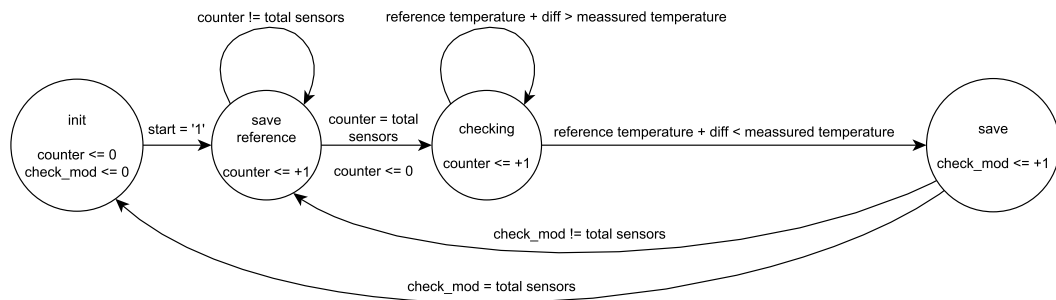
Teplotní ochrana

V návrhu FPGA obvodu je zabudována konfigurovatelná teplotní ochrana. Teplota LED modulů je periodicky měřena a ukládána. Pokud některý z LED modulů překročí horní teplotní hranici, je nastaven příkaz pomocí signálu *blackout*. Ten je napojen do vstupního dekodéru dat a zabezpečí, že dojde k vypnutí zobrazování

na příslušném LED modulu.

Jakmile je LED modul vypnut, dá se předpokládat, že jeho teplota začne klesat. Při poklesu teploty o konfigurovatelnou hodnotu je LED modul opět uveden do normálního provozu.

Jelikož z principu 1-Wire sběrnice není možné stanovit pořadí teplotních čidel, je nutné přiřadit senzory ke konkrétním LED modulům. Toho je dosaženo pomocí algoritmu popsáno ve stavovém automatu na obrázku č. 4.15. Algoritmus je založen na rozsvícení pouze jednoho LED modulu a kontrolování, které teplotní čidlo bude měřit zvýšenou teplotu.



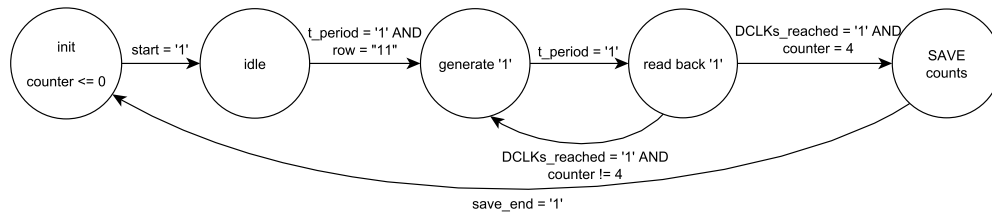
Obr. 4.15: Přiřazení 1-Wire čidel ke konkrétním LED modulům

Tento blok v FPGA využívá po implementaci 638 LUT, 234 Slices a 457 Slice registrů, 28 LUTRAM bloků a dvě paměti BRAM.

4.10 Detekce chybných LED

Detekce vadných LED je umožněna díky budičům SCT2027 a rovněž díky zpětnému připojení výstupu dat z LED modulů. Na obrázku č. 3.3 je znázorněno dosažení detekčního režimu.

Řízení detekce je provedeno pomocí stavového automatu na obrázku č. 4.16. Jakmile je automat spuštěn, čeká na časový impuls t_{period} společně s hodnotou $row = "11"$. Tato podmínka je důležitá kvůli přesnému určení pozice vadné LED. V následujícím stavu jsou generovány na datových výstupech FPGA do LED modulů log. 1, což zabezpečí rozsvícení, resp. otestování, všech LED.



Obr. 4.16: Detekce vadných LED

Ve stavu *read_back_1* je nejdříve pozastaveno odesílání hodinového signálu do LED modulů po dobu detekce chyb. Poté je hodinový signál obnoven a dochází ke čtení logických hodnot na všech čtyřech datových vstupech. Pokud FPGA vzorkuje log. 0, jedná se o reprezentaci chybné LED a je přičtena jednička do příslušného čítače.

Po vyčtení všech dat dojde k otestování, zdali byly zkontrolovány všechny multiplexované řádky a postup se případně opakuje. V posledním stavu *SAVE_counts* dochází k zápisu detekovaných počtů do paměti MODBUS protokolu.

Paralelně s detekcí počtu vadných LED probíhá i ukládání přesné pozice vadné LED ve vztahu ke konkrétnímu LED modulu. Toho je docíleno přičítáním hodnot do čítače zpětně přijatých bitů a podpůrnou kombinační logikou jsou ukládány pozice vadných LED do samostatné paměti. Každý LED modul má v paměti vyhrazeno místo pro 256 vadných LED, což je zhruba 15% z celkového počtu LED. Pokud by bylo vadných LED více, bude to již řešeno výměnou celého modulu.

Detekce chyb po implementaci využívá 552 LUT, 211 Slices a 409 Slice registrů a dvě paměti BRAM pro ukládání přesné pozice vadných LED.

5 VÝSLEDKY NÁVRHU FPGA OBVODU

V navržené konfiguraci a při řízení LED panelu s rozlišením 160x56 pixelů je v FPGA obvodu využito 44 % LUT, 62 % Slices, 14 % Slice registrů a 34 % BRAM16BWER pamětí. Maximální pracovní kmitočet návrhu je 118 MHz a maximální možný kmitočet vstupního HDMI signálu 136 MHz. Proudový odběr v normálním provozu je v této konfiguraci 290 mA při 5 V.

Teoretické maximální rozlišení, které je možné pomocí vypracovaného návrhu řídit, lze vypočítat z dostupných prostředků v FPGA. Nejvíce zde záleží na počtu bloků paměti *RAMB16BWER*. Vybraný FPGA obvod má těchto bloků 32, z čehož 7 je využito k podpůrným funkcím a 25 jich zbývá pro ukládání obrazu. Implementace obrazové paměti vyžaduje uspořádání čtyř bloků paměti paralelně vedle sebe, z čehož plyne dostupné místo pro ukládání obrazových dat jako 6 bloků po 16 Kb.

$$\text{Maximální rozlišení} = 6 \text{ bloků} \cdot (16 \cdot 1024 \text{ bitů}) = 98304 \quad (5.1)$$

Toto rozlišení odpovídá například rozměru panelu 300x300 pixelů. Detailní přehled využití FPGA pro různý počet LED modulů je uveden v tabulce č. 5.1.

Pokud by bylo potřeba řídit větší rozlišení než podporované, lze bez problému použít rozdělovač HDMI signálu a zapojit paralelně dvě desky s FPGA obvodem. Pomocí řídicí jednotky by se poté nastavila pro každý obvod FPGA jiná počáteční souřadnice zobrazování a výsledný obraz by byl složen ze dvou samostatných částí.

Tab. 5.1: Přehled využití FPGA pro různé rozlišení LED panelu

Počet LED modulů	Počet pixelů	Slice Reg	LUTs	RAMB16BWER
1	1792	14 %	49 %	34 %
2	3584	14 %	50 %	34 %
3	5376	14 %	50 %	34 %
4	7168	14 %	50 %	34 %
5	8960	14 %	51 %	34 %
6	10752	14 %	51 %	34 %
7	12544	14 %	51 %	34 %
8	14336	14 %	51 %	34 %
9	16128	14 %	51 %	34 %
10	17920	14 %	52 %	46 %
15	26880	14 %	52 %	46 %
20	35840	14 %	52 %	59 %
30	53760	14 %	52 %	71 %
40	71680	14 %	52 %	84 %
50	89600	14 %	52 %	96 %
54	96768	14 %	52 %	96 %

Tab. 5.2: Přehled využití FPGA po implementaci pro řízení prototypu LED panelu

Název bloku	LUT	Slices	Slice Reg	Další
HDMI rozhraní	127	57	120	1x PLL, 3x LUTRAM
Identifikační data	75	37	35	-
Pulsně šířková modulace	25	9	18	-
Paměť	-	-	-	4x RAMB16BWER
Testovací obrazce	107	33	71	-
Generátor časových impulzů	53	20	29	-
MODBUS protokol	401	183	268	3x RAMB16BWER
Teplota	638	234	457	28x LUTRAM, 2x RAMB16BWER
Detekce vadných LED	522	211	409	2x RAMB16BWER
Zpracování vstupních dat	46	39	44	-

ZÁVĚR

Cílem této práce bylo navrhnout, implementovat a otestovat řízení informačních LED panelů za použití HDMI rozhraní. Současně s tímto byl kladen důraz na pokročilé funkce jako schopnost komunikace s nadřizovanou jednotkou, měření teploty LED modulů, základní teplotní řízení, detekci vadných LED a univerzální generický VHDL popis.

V rámci práce byl vybrán FPGA obvod firmy XILINX, Inc. Spartan 6 XC6SLX9 v průmyslové variantě. Návrh FPGA obvodu byl realizován a otestován na prototypu LED panelu s rozlišením 160x56 obrazových bodů.

Nejdůležitějším blokem návrhu byla implementace vstupního HDMI rozhraní. V teoretické části práce byl vytvořen detailní popis HDMI kódování. V návaznosti na to byl navržen VHDL popis HDMI komponenty, která je schopna přijímat a dekodovat příchozí HDMI signál. Pro zabezpečení co největší kompatibility byla do návrhu FPGA zařazena i komunikace v rámci DDC kanálu uvnitř HDMI rozhraní. FPGA obvod se identifikuje jako digitální zobrazovací zařízení a vyžádá si generování obrazu s rozlišením 1366x768 obrazových bodů.

Komunikace FPGA obvodu s řídicí jednotkou je zabezpečena pomocí MODBUS protokolu, který využívá jako fyzickou vrstvu standard RS485. Řídicí jednotka je schopna nastavovat parametry zobrazování, spouštět detekci vadných LED a mnoho dalšího. Rovněž je schopna skrze MODBUS protokol vyčítat stavové informace o LED panelu. Jsou to například teploty LED modulů, chybové hlášení, vadné LED a podobně.

Maximální velikost rozlišení, kterou je FPGA obvod schopen řídit, je 98304 obrazových bodů. Hodnota je přímo úměrná počtu dedikovaných blokových pamětí uvnitř FPGA obvodu. Obraz na LED panelu je generován s obnovovacím kmitočtem 400 Hz. Zobrazení je stabilní a neproblikávají žádné obrazové body. Ověření probíhalo pomocí zobrazování testovacích obrazců, které jsou náročné na přesné časování a jsou schopny odhalit i malé nepřesnosti.

Velkým blokem návrhu bylo měření teploty a základní teplotní řízení. FPGA obvod je schopen komunikovat s 1-Wire teplotními čidly DS18B20 od firmy Dallas Semiconductor, a v případě překročení maximální povolené teploty vyřadí zobrazování na konkrétním LED modulu.

Zdrojové kódy VHDL návrhu obsahují uživatelskou knihovnu, která slouží k nastavení základních parametrů pro syntézu a implementaci. Je zde možné nastavit počet LED modulů zapojených v řadě za sebou, rozměry jednotlivých LED modulů. Dále také obnovovací frekvenci obrazu na LED panelu, či přenosovou rychlost rozhraní RS485. Všechny tyto parametry jsou konfigurovatelné pouze v rámci VHDL kódu a při provozu již neměnné.

Knihovna je pomocí proměnných napojena na celý návrh a při syntéze jsou signály a časové konstanty syntetizovány genericky.

Dále knihovna obsahuje inicializační data pro paměť MODBUS protokolu jako jas LED, počáteční souřadnice zobrazování, teplotní limity a další. Tyto parametry je již možné vzdáleně měnit skrze MODBUS protokol.

LITERATURA

- [1] BARR, Michael. *Introduction to Pulse Width Modulation* [online]. 2007 [cit. 2015-12-10]. Dostupné z: <http://www.barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>
- [2] DALLAS SEMICONDUCTOR. *Programmable Resolution 1-wire Digital Thermometer* [online]. 2005 [cit. 2015-12-5]. Dostupné z: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Temp/DS18B20.pdf>.
- [3] HDMI Licensing, LLC. *HDMI Specification Version 1.3a* [online]. 2006-11-12 [cit. 2015-12-5]. Dostupné z: <http://www.microprocessor.org/HDMISpecification13a.pdf>
- [4] HUANG, Dai a Rick BALLANTYNE. *Synthesizable FPGA Interface for Retrieving ROM Number from 1-Wire Devices* [online]. 2001-05-08 [cit. 2015-12-5]. Dostupné z: http://www.xilinx.com/support/documentation/application_notes/xapp198.pdf.
- [5] MidSouth Wire & Cable Co. *Product page* [online]. 2007 [cit. 2015-12-13]. Dostupné z: <https://www.midsouthelectronics.com/HDMI350BJ.htm>
- [6] MODBUS-IDA *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b* [online]. 2006-12-28 [cit. 2015-12-5]. Dostupné z: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf.
- [7] NXP Semiconductors. *I²C-bus specification and user manual* [online]. 2011-04-04 [cit. 2016-05-11]. Dostupné z: http://www.nxp.com/documents/user_manual/UM10204.pdf
- [8] ON Semiconductor. *MC74HC595A Datasheet* [online]. 2016-01 [cit. 2016-05-18]. Dostupné z: http://www.onsemi.com/pub_link/Collateral/MC74HC595A-D.PDF1.
- [9] RONEŠOVÁ, Andrea. *Přehled protokolu MODBUS* [online]. 2015-05 [cit. 2015-12-5]. Dostupné z: <http://home.zcu.cz/~ronesova/bast1/files/modbus.pdf>.
- [10] StarChips Technology. *16-bit Constant-Current LED Driver with ModelessTM Error Detection* [online]. 2013-06 [cit. 2016-5-10]. Dostupné z: http://www.starchips.com.tw/pdf/datasheet/SCT2027V01_03.pdf.
- [11] Texas Instruments. *An Overview of LVDS Technology* [online]. 1998 [cit. 2015-12-13]. Dostupné z: <http://www.ti.com/lit/an/snla165/snla165.pdf>

- [12] TORRES, Gabriel. *Inside HDMI* [online]. 2006-02-13 [cit. 2015-12-5]. Dostupné z: <http://www.hardwaresecrets.com/inside-hdmi-high-definition-multimedia-interface/2/>
- [13] Video Electronics Standards Association. *VESA ENHANCED EXTENDED DISPLAY IDENTIFICATION DATA STANDARD, Release A, Rev. 2* [online]. 2006-11-25 [cit. 2015-12-5]. Dostupné z: ftp://ftp.cis.nctu.edu.tw/pub/csie/Software/X11/private/VeSaSpEcS/VESA_Document_Center_Monitor_Interface/EEDIDrAr2.pdf
- [14] XILINX, Inc. *Implementing a TMDS Video Interface in the Spartan-6 FPGA* [online]. 2010-12-13 [cit. 2015-12-5]. Dostupné z: http://www.xilinx.com/support/documentation/application_notes/xapp495_S6TMDS_Video_Interface.pdf
- [15] XILINX, Inc. *Spartan-6 FPGA Block RAM Resources* [online]. 2011-07-08 [cit. 2016-05-11]. Dostupné z: http://www.xilinx.com/support/documentation/user_guides/ug383.pdf.
- [16] XILINX, Inc. *Spartan-6 FPGA Clocking Resources* [online]. 2016-06-19 [cit. 2016-05-11]. Dostupné z: http://www.xilinx.com/support/documentation/user_guides/ug382.pdf
- [17] XILINX, Inc. *Spartan-6 FPGA DSP48A1 Slice* [online]. 2014-05-29 [cit. 2016-05-12]. Dostupné z: http://www.xilinx.com/support/documentation/user_guides/ug389.pdf.
- [18] XILINX, Inc. *Spartan-6 FPGA SelectIO Resources* [online]. 2016-11-21 [cit. 2016-05-11]. Dostupné z: http://www.xilinx.com/support/documentation/user_guides/ug381.pdf

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

BRAM	Bloková paměť RAM
CEC	Consumer Electronics Control – Řízení zákaznických zařízení
DDC	Display Data Channel – Datový kanál pro komunikaci zařízení se zdrojem HDMI dat
DVI	Digital Visual Interface – Digitální video rozhraní
EDID	Extended Display Identification Data - Rozšířená displejová identifikační data
FPGA	Field-programmable Gate Array - Programovatelné hradlové pole
HDMI	High-Definition Multi-media Interface – Rozhraní pro přenos video a audio informací ve vysokém rozlišení
HPD	Hot Plug Detect – Detekce připojení zařízení
LED	Light Emiting Diode – Záření emitující dioda
LUT	Look Up Table – Náhledová tabulka
LUTRAM	Look Up Table Random Access Memory – Paměť realizována pomocí LUT
LVDS	Low-voltage Differential Signaling – Nízkonapěťové rozdílové kódování
RAM	Random Access Memory – Paměť s přímým přístupem
PLL	Phase Locked Loop – Fázový závěs
TERC4	TMDS Error Reduction Coding – TMDS kódování s redukcí chyb
TMDS	Transition Minimized Differential Signalling – Kódování s minimálním počtem logických přechodů
VESA	Video Electronics Standards Association – Uskupení výrobců a vývojářů grafických karet a zobrazovacích zařízení
VHDL	Very High Speed Integrated Circuits Hardware Description Language – Jazyk pro popis velmi rychlých integrovaných obvodů
XNOR	Exclusive nor – Negovaný exkluzivní logický součet

XOR Exclusive or – Exkluzivní logický součet

SEZNAM PŘÍLOH

P.1 MODBUS protokol	54
P.2 Přehled datových rámců implementovaných funkcí MODBUS protokolu	55
P.3 Předpis pro čtení z paměti RAM	57
P.4 Stromová struktura zdrojových souborů	58

PŘÍLOHY

P.1 MODBUS protokol

Tab. P.1: Náhled na paměť MODBUS protokolu

Adresa [dec]	Adresa [hex]	Název	Počet 16b reg	Formát dat	Jedn.	Nás.	Popis	Režim
0	000	STATE	1	uint16	-	-	0x0001-Alarm; 0x0002-přehřátí; 0x0004-chyba HDMI; 0x0008-probíhá kalibrace 1-Wire senzorů; 0x0016-probíhá LED Error detekce; 0x0032-testovací režim	R
1	001	CMD	1	uint16	-	-	1 - vynucení LED testu 2 - kalibrace 1-Wire senzorů	W
2	002	BRIGHTNESS	1	uint16	%	-	Jas LED - 0..100	R/W
3	003	TEST_PATTERN	1	uint16	-	-	1/0 - ON/OFF testovací obrazec	R/W
4	004	TEST_PATTERN_MODE	1	uint16	-	-	0..3 Režim testovacího obrazce	R/W
5	005	TEST_PATTERN_SPEED	1	uint16	ns	4 us	Rychlost testovacího režimu	R/W
6	006	DATA_INVERT	1	uint16	-	-	1/0 - ON/OFF inverze vstupních dat	R/W
7	007	DATA_TRESHOLD	1	uint16	-	-	0..255 - Rozhodovací práh pro zobrazení pixelu na panelu.	R/W
8	008	POS_X	1	int16	px	-	Offset tabule - X	R/W
9	009	POS_Y	1	int16	px	-	Offset tabule - Y	R/W
10	00A	TEMP_LIMIT1	1	int16	°C	-	Limit teploty 1 - nouzové vypnutí modulu při přehřátí	R/W
11	00B	TEMP_LIMIT2	1	int16	°C	-	Limit teploty 2 - "hystereze"opětovného zapnutí modulu po přehřátí	R/W
12	00C	TEMP_LIMIT3	1	int16	°C	-	Limit teploty 3 - požadované oteplení LED modulu při 1-Wire kalibraci	R/W
13	00D	TEMP_LIMIT4	1	int16	°C	-	Limit teploty 4 - bude upřesněno	R/W
14	00E	TEMP_LIMIT5	1	int16	°C	-	Limit teploty 5 - bude upřesněno	R/W
15	00F	TEMP_LIMIT6	1	int16	°C	-	Limit teploty 6 - bude upřesněno	R/W
16	010	TEMP_LIMIT7	1	int16	°C	-	Limit teploty 7 - bude upřesněno	R/W
17	011	REZERVA	15	-	-	-	-	-
32	020	LED_CNT	2	int32	-	-	Celkový počet LED	R
34	022	LED_FLT_CNT	2	int32	-	-	Celkový počet vadných LED	R
36	024	LED_FLT_HASH	1	uint16	-	-	HASH vadných LED	R
37	025	SIZE_X	1	int16	px	-	Velikost tabule - šířka	R
38	026	SIZE_Y	1	int16	px	-	Velikost tabule - výška	R
39	027	MODULE_CNT	1	int16	-	-	Počet LED modulů	R
40	028	PANEL_REFRESH	1	int16	Hz	-	Obnovovací frekvence panelu	R
41	029	1-WIRE_SENSORS	1	int16	-	-	Počet detekovaných 1-wire čidel	R
42	02A	REZERVA	6	-	-	-	-	-
48	030	MODULE_TEMP	64	int16	°C	16	Teplota jednotlivých LED modulů	R
112	070	MODULE_FLT_CNT	64	uint16	-	-	Počet vadných LED na modulech	R
176	0B0	MODULE_0	4	module	-	-	Informace o modulu 0	R
180	0B4	MODULE_1	4	module	-	-	Informace o modulu 1	R
184	0B8	MODULE_2-63	248	module	-	-	Informace o modulu 2 až 63	R
	ADR+0	MODULE_POS_X	1	int16	px	-	Pozice modulu X Relativně k tabuli	R
	ADR+1	MODULE_POS_Y	1	int16	px	-	Pozice modulu Y Relativně k tabuli	R
	ADR+2	MODULE_SIZE_X	1	int16	px	-	Velikost modulu - šířka	R
	ADR+3	MODULE_SIZE_Y	1	int16	px	-	Velikost modulu - výška	R
432	1B0	REZERVA	79	-	-	-	-	-

P.2 Přehled datových rámců implementovaných funkcí MODBUS protokolu

Tab. P.2: Čtení registrů

Požadavek	funkční kód	1 bajt	03h
	počáteční adresa	2 bajty	0000h až FFFFh
	počet registrů	2 bajty	1 až 125
Odpověď	funkční kód	1 bajt	03h
	počet bajtů	1 bajt	2*N
	hodnoty registrů	N *2 bajty	data

Tab. P.3: Zápis registrů

Požadavek	funkční kód	1 bajt	10h
	počáteční adresa	2 bajty	0000h až FFFFh
	počet registrů	2 bajty	1 až 123
	počet bajtů	1 bajt	2*N
	hodnoty registrů	N *2 bajtů	data
Odpověď	funkční kód	1 bajt	10h
	počáteční adresa	2 bajty	0000h až FFFFh
	počet registrů	2 bajty	1 až 123

Tab. P.4: Čtení souboru

Požadavek	funkční kód	1 bajt	14h
	počet bajtů	1 bajt	7 až 245
	typ souboru	1 bajt	06h
	číslo souboru	2 bajty	1 až 65535
	číslo záznamu	2 bajty	1 až 9999
	délka záznamu	2 bajty	N
Odpověď	funkční kód	1 bajt	14h
	délka celé odpovědi	1 bajt	7 až 245
	délka odpovědi souboru	1 bajt	5 až 245
	typ souboru	1 bajt	06h
	soubor	N *2 bajtů	data

P.3 Předpis pro čtení z paměti RAM

F8F F8E F8D F8C F8B F8A F89 F88 F87 F86 F85 F84 F83 F82 F81 F80
D0F D0E D0D D0C D0B D0A D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
A8F A8E A8D A8C A8B A8A A89 A88 A87 A86 A85 A84 A83 A82 A81 A80
80F 80E 80D 80C 80B 80A 809 808 807 806 805 804 803 802 801 800
58F 58E 58D 58C 58B 58A 589 588 587 586 585 584 583 582 581 580
30F 30E 30D 30C 30B 30A 309 308 307 306 305 304 303 302 301 300
08F 08E 08D 08C 08B 08A 089 088 087 086 085 084 083 082 081 080
F6F F6E F6D F6C F6B F6A F69 F68 F67 F66 F65 F64 F63 F62 F61 F60
CEF CEE CED CEC CEB CEA CE9 CE8 CE7 CE6 CE5 CE4 CE3 CE2 CE1 CE0
A6F A6E A6D A6C A6B A6A A69 A68 A67 A66 A65 A64 A63 A62 A61 A60
7EF 7EE 7ED 7EC 7EB 7EA 7E9 7E8 7E7 7E6 7E5 7E4 7E3 7E2 7E1 7E0
56F 56E 56D 56C 56B 56A 569 568 567 566 565 564 563 562 561 560
2EF 2EE 2ED 2EC 2EB 2EA 2E9 2E8 2E7 2E6 2E5 2E4 2E3 2E2 2E1 2E0
06F 06E 06D 06C 06B 06A 069 068 067 066 065 064 063 062 061 060
F4F F4E F4D F4C F4B F4A F49 F48 F47 F46 F45 F44 F43 F42 F41 F40
CCF CCE CCD CCC CCB CCA CC9 CC8 CC7 CC6 CC5 CC4 CC3 CC2 CC1 CC0
A4F A4E A4D A4C A4B A4A A49 A48 A47 A46 A45 A44 A43 A42 A41 A40
7CF 7CE 7CD 7CC 7CB 7CA 7C9 7C8 7C7 7C6 7C5 7C4 7C3 7C2 7C1 7C0
54F 54E 54D 54C 54B 54A 549 548 547 546 545 544 543 542 541 540
2CF 2CE 2CD 2CC 2CB 2CA 2C9 2C8 2C7 2C6 2C5 2C4 2C3 2C2 2C1 2C0
04F 04E 04D 04C 04B 04A 049 048 047 046 045 044 043 042 041 040
F2F F2E F2D F2C F2B F2A F29 F28 F27 F26 F25 F24 F23 F22 F21 F20
CAF CAE CAD CAC CAB CAA CA9 CA8 CA7 CA6 CA5 CA4 CA3 CA2 CA1 CA0
A2F A2E A2D A2C A2B A2A A29 A28 A27 A26 A25 A24 A23 A22 A21 A20
7AF 7AE 7AD 7AC 7AB 7AA 7A9 7A8 7A7 7A6 7A5 7A4 7A3 7A2 7A1 7A0
52F 52E 52D 52C 52B 52A 529 528 527 526 525 524 523 522 521 520
2AF 2AE 2AD 2AC 2AB 2AA 2A9 2A8 2A7 2A6 2A5 2A4 2A3 2A2 2A1 2A0
02F 02E 02D 02C 02B 02A 029 028 027 026 025 024 023 022 021 020
F0F F0E F0D F0C F0B F0A F09 F08 F07 F06 F05 F04 F03 F02 F01 F00
C8F C8E C8D C8C C8B C8A C89 C88 C87 C86 C85 C84 C83 C82 C81 C80
A0F A0E A0D A0C A0B A0A A09 A08 A07 A06 A05 A04 A03 A02 A01 A00
78F 78E 78D 78C 78B 78A 789 788 787 786 785 784 783 782 781 780
50F 50E 50D 50C 50B 50A 509 508 507 506 505 504 503 502 501 500
28F 28E 28D 28C 28B 28A 289 288 287 286 285 284 283 282 281 280
00F 00E 00D 00C 00B 00A 009 008 007 006 005 004 003 002 001 000

P.4 Stromová struktura zdrojových souborů

