



# 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

## APLIKACE FPGA V ŘÍZENÍ MATICOVÝCH DISPLEJŮ LED

FPGA APPLICATION IN LED MATRIX DISPLAY CONTROLLING

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Pavel Šunka

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Lukáš Fucik, Ph.D.

BRNO 2018



# Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**  
Ústav mikroelektroniky

**Student:** Bc. Pavel Šunka

**ID:** 154892

**Ročník:** 2

**Akademický rok:** 2017/18

## NÁZEV TÉMATU:

### Aplikace FPGA v řízení maticových displejů LED

#### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte funkční vzorek modulu s obvodem FPGA pro řízení monochromatických, barevných a kombinovaných maticových displejů LED. Prostudujte topologii LED matice, systémy statického a dynamického řízení maticových zobrazovačů LED. Seznamte se s barevným modelem. Prostudujte a navrhněte obvodové řešení modulu. Implementujte vstup dat přes datový proud obrazových dat přenášený periodicky přes rozhraní SPI. Navrhněte způsob řízení jasu LED, identifikaci rozměrů panelu. Porovnejte aktuální nabídku a možnosti obvodů FPGA od předních výrobců, vyberte vhodný typ pro funkční vzorek a sestavte projekt v jazyce VHDL. Vytvořte schéma zapojení a návrh desky plošných spojů části řídicího modulu pro začlenění FPGA. Realizujte funkční vzorek na modulu Adafruit RGB LED 64 x 32 bodů a ověřte jeho provoz.

#### DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 22.5.2018

**Vedoucí práce:** doc. Ing. Lukáš Fucík, Ph.D.

**Konzultant:**

**doc. Ing. Lukáš Fucík, 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 dává náhled do problematiky řízení plně barevných RGB LED panelů o větším počtu řádků a sloupců. Zabývá se komunikačními protokoly pro přenos obrazu a výměnu informací. Zaměřuje se implementaci FPGA do řídicí struktury RGB LED panelů. V první polovině se práce zaměřuje na prohloubení teoretických znalostí ohledně technologií LED, jejich řízení a zobrazování barvy. Dále řeší komunikaci s nadřazenými prvky. Poslední část teorie se věnuje FPGA obvodům. Druhá část práce popisuje praktický návrh FPGA obvodu od komunikace UART a SPI přes ukládání dat z komunikace až po samotné řízení panelu.

## **Klíčová slova**

RGB, LED, panel, FPGA, BCM

## **Abstract**

The thesis gives an overview of the issue of control of full-color RGB LED panels with a large number of rows and columns. It deals with communication protocols for image transfer and information exchange. It focuses on implementing the FPGA into the RGB LED control structure. In the first half, the thesis focuses on deepening the theoretical knowledge about LED technologies, their control and color display. It further solves communication with superior elements. The last part of the theory is dedicated to FPGA circuits. The second part of the thesis describes the practical design of the FPGA circuit from UART and SPI communication through data storage from the communication to the panel control itself.

## **Keywords**

RGB, LED, panel, FPGA, BCM

### **Bibliografická citace:**

ŠUNKA, P. Aplikace FPGA v řízení maticových displejů LED. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 54 s. Vedoucí diplomové práce doc. Ing. Lukáš Fucik, Ph.D..

## **Prohlášení**

Prohlašuji, že svou diplomovou práci na téma Aplikace FPGA v řízení maticových displejů LED 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 jsem si 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.

V Brně dne: 22. května 2018

.....

podpis autora

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.

## **Poděkování**

Děkuji vedoucímu diplomové práce doc. Ing. Lukáši Fajčíkovi, Ph.D. a odbornému vedoucímu Ing. Zdeňku Kakáčovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále děkuji firmě BUSE s.r.o., pro kterou byla diplomová práce tvořena, za materiální a finanční záštitu projektu.

V Brně dne: 22. května 2018

.....

podpis autora

# Obsah

1	LED technologie.....	1
1.1	Způsoby napájení LED [3]. .....	1
1.1.1	Stejnoseměrné (statické) řízení LED .....	2
1.1.2	Impulsní (dynamické) buzení .....	3
1.2	Řízení jasu LED.....	4
1.1.1	PWM.....	4
1.1.2	BCM.....	5
1.2	Zobrazení pomocí LED.....	6
1.2.1	Monochromatické zobrazení.....	7
1.2.2	Semi barevné zobrazení .....	8
1.2.3	Plně barevné zobrazení .....	8
2	Barva.....	9
2.1	Subtraktivní míchání barev .....	9
2.2	Aditivní míchání barev.....	10
2.3	RGB model.....	10
1.1.3	Barevná hloubka .....	11
1.1.4	Gamma korekce .....	12
3	Led matice.....	13
3.1	Řízení řádků .....	14
3.2	Řízení sloupců.....	14
3.3	Identifikace rozměrů panelu.....	15
4	Řídící obvody.....	16
4.1	Způsoby přenosu zobrazení na informační panel.....	16
4.2	Komunikace s nadřazeným prostředkem .....	16
4.2.1	UART.....	16
4.2.2	Ethernet.....	17
4.2.3	HDMI.....	19
4.3	Vnitřní komunikace.....	24
4.3.1	Paralelní komunikace.....	24
4.3.2	USART .....	24
4.3.3	SPI.....	25

4.3.4	I <sup>2</sup> C .....	25
5	Uspořádání řídicí části .....	26
5.1	MCU.....	26
5.2	FPGA.....	26
5.2.1	Komunikace s CPU.....	27
5.2.2	Obrazová paměť.....	27
5.2.3	Generování dat pro displej.....	28
6	FPGA .....	33
6.1	Vývoj a vlastnosti.....	33
6.1.1	Základní architektura FPGA.....	33
6.1.2	Další buňky FPGA.....	35
6.1.3	Výběr FPGA:.....	35
6.1.4	Vývojová deska 10M08 .....	36
7	Realizace projektu.....	38
7.1	PLL.....	38
7.2	Komunikace .....	39
7.2.1	SPI.....	39
7.2.2	MIKOMP .....	41
7.3	Obrazová paměť .....	46
7.3.1	Přepínač pamětí.....	46
7.4	Ovládání panelu .....	46
7.4.1	Testování velikosti panelu .....	47
7.4.2	Gamma korekce .....	49
7.4.3	Dynamická BCM .....	49
7.4.4	Řízení jasu.....	50
8	Výsledky projektu.....	51
8.1	Implementace na FPGA .....	52
9	Závěr .....	54

## Seznam obrázků

Obr. 1: Zapojení diody s rezistorem a určení pracovního bodu na VA charakteristice. [3] .....	2
Obr. 2: Voltampérová charakteristika diody. [4] .....	4
Obr. 3: PWM průběhy. ....	5
Obr. 4: BCM průběh jasu s binárním zápisem 1010.....	5
Obr. 5: Sedmi, čtrnácti a šestnácti segmentový displej. [5].....	6
Obr. 6: Zapojení maticového LED zobrazovače 8 x 8. [6].....	7
Obr. 7: Subtraktivní míchání barev. [7] .....	9
Obr. 8: Aditivní míchání barev. [7] .....	10
Obr. 9: RGB model. [7] .....	11
Obr. 10: Závislost výstupní intenzity na vstupní intenzitě a gamma korekce. ....	12
Obr. 11: Zapojení monochromatického displeje od firmy Adafruit. [9] .....	13
Obr. 12: Obvodový součet NOR. [32] .....	15
Obr. 13: Odeslání jednoho slova protokolem UART. ....	17
Obr. 14: Rozložení vodičů ethernetu v konektoru RJ-45. [10].....	18
Obr. 15: Ethernetový rámec. [14] .....	19
Obr. 16: Průřez vodiče a architektura fyzické vrstvy HDMI. [17] .....	20
Obr. 17: HDMI konektor typu A. [19].....	20
Obr. 18: Datová perioda HDMI přenosu. [20].....	21
Obr. 19: Odesílání jednotlivých pixelů v obrazovém režimu TDMS. [22] .....	22
Obr. 20: EDID formát. [22] .....	23
Obr. 21: MIKOMP protokol. ....	24
Obr. 22: Časový průběh I2C komunikace. [23].....	25
Obr. 23: Řízení jasu pomocí BCM a PWM jednobarevného panelu bez odstínu. ...	28
Obr. 24: Řízení intenzity barvy a jasu pomocí BCM a PWM. ....	29
Obr. 25: Řízení intenzity barvy a posuvného jasu pomocí BCM a PWM.....	30
Obr. 26: Posuvná BCM s posuvným jasem. ....	30
Obr. 27: Posuvná BCM s posuvným jasem. ....	31
Obr. 28: Vývojový diagram dynamického řízení BCM. ....	32
Obr. 29: Základní konstrukce FPGA. [26] .....	33

Obr. 30: Vstupně výstupní buňka. [27].....	34
Obr. 31: 10M08 Evalution kit. [31] .....	37
Obr. 32: Blokové schéma řídicího obvodu .....	38
Obr. 33:Karnaughova mapa a výsledná minimalizovaná funkce pro datový signál CLK. ....	40
Obr. 34:Karnaughova mapa a výsledná minimalizovaná funkce pro datový signál MOSI. ....	40
Obr. 35: Stavový automat pro řízení příjmu dat po UART. ....	42
Obr. 36: Stavový automat pro řízení příjmu dat komponenty mikomp.....	43
Obr. 37: Stavový automat pro řízení odesílání dat komponenty mikomp. ....	44
Obr. 38: Stavový automat pro nadřizenou komponentu mikomp_top.....	45
Obr. 39: Schéma zapojení řadiče MBI5024.[33] .....	47
Obr. 40: Vývojový diagram dynamického řízení BCM. ....	48
Obr. 41: Schéma zapojení vstupní části dynamické BCM. ....	49
Obr. 42: Schéma zapojení vstupní části dynamické BCM. ....	50
Obr. 43: Simulace komunikace testovacího řetězce po UART. ....	51
Obr. 44: Nasnímkovaná komunikace testovacího řetězce po UART logickým analyzátozem. ....	51
Obr. 45: Simulace signálů řídicích panel.....	52
Obr. 46: Průběh řídicích signálů panelu zachycený logickým analyzátozem. ....	52

## Seznam tabulek

Tab. 1: Vyzařované spektrum LED diod podle složení polovodiče. [1] .....	1
Tab. 2: Barevná (bitová) hloubka. ....	11
Tab. 3: Srovnání rychlostí a dosahu kabelu ethernetu. [12] .....	18
Tab. 4: Kódování jednotlivých kombinací řídicích dat. [21].....	21
Tab. 5:Kódování jednotlivých kombinací řídicích dat. [21].....	22
Tab. 6: Výhody a nevýhody FPGA. [25].....	27
Tab. 7: Srovnání FPGA prvků s bočními vývody. [29] [30] .....	36
Tab. 8: Popis jednotlivých stavů stavového automatu pro UART. ....	42
Tab. 9: Popis jednotlivých stavů pro řízení příjmu dat komponenty mikomp. ....	43
Tab. 10: Popis jednotlivých stavů stavového automatu pro UART. ....	44
Tab. 11: Povelý pro MIKOMP. ....	44
Tab. 12: Popis jednotlivých stavů stavového automatu pro UART. ....	45
Tab. 13: Tabulka využití FPGA.....	53

# Úvod

Zobrazovací grafické prvky se používají od nepaměti. Nejdříve se používaly mechanické prvky, jako například různé tabule a poutače. S příchodem elektřiny byly tyto prvky elektromechanické, například elektromechanicky měněné znaky v informačních cedulích na vlakových nádražích. Poslední v linii vývoje jsou zobrazovací prvky elektronické. Mezi tyto technologie patří hlavně elektromechanické bistabilní zobrazovací body (DOT), displeje z tekutých krystalů (LCD), světlo emitující diody (LED), organické světlo emitující diody (OLED) a elektronický inkoust (e-ink). Tyto technologie jsou široce aplikovány pro zobrazování dat v informačních panelech. Informační panely se používají v různých odvětvích byznysu. Slouží například pro reklamní účely, ať už jako blikající LED panel s nápisem otevřeno, dovezený z Číny, nebo velký billboard, zabírající plochu několika pater budovy, ve světových metropolích. Další velkou skupinou jsou panely provozní, které zobrazují výrobní data, sportovní výsledky, technologické veličiny nebo číselný přístupový systém na úradech. Z provozních panelů můžeme vyčlenit informační panely v dopravě. Této skupině informačních panelů se věnuje celá následující práce. V dopravě se zobrazovací zařízení používají převážně pro vizualizaci cíle trasy, názvu zastávek, čísla spojů a času. Práce se zabývá LED panely, a jejich řízením pomocí FPGA. Důrazněji se zaměřuje na barevné panely, které v dopravních prostředcích postupně nahrazují monochromatické. V jednotlivých kapitolách je popsána práce s barvou, LED technologie se zaměřením na panely, dále komunikace řídicích obvodů a dělení úkolů mezi nimi. Následující kapitoly se věnují FPGA a samotnému popisu realizace projektu. V závěru je shrnutí dosažených výsledků.

# 1 LED TECHNOLOGIE

Light Emitting diode, v překladu světlo vyzařující dioda, je dioda se speciálním P-N přechodem. Při průchodu proudem diodou v propustném směru dochází na přechodu k tzv. elektroluminiscenci. Volný elektron a díra se společně rekombinují a jejich energie se dle zákona o zachování energie přemění na foton. LED dioda tak vyzařuje nekoherentní světlo, které má úzké spektrum. Díky různému chemickému složení polovodičových materiálů jsme schopni pokrýt barevné spektrum světla od ultrafialového až po infračervené. Potřebnou složku barvy, vlnovou délku světla, jsme schopni dostat různými metodami, které se samozřejmě liší materiálovou a technologickou náročností, parametry, účinností, životností atd. Různé polovodičové materiály mají odlišnou velikost napětí v propustném směru. Maximální napětí v závěrném směru je u většiny LED kolem pěti voltů. Tab. 1 uvádí základní parametry několika LED.

**Tab. 1: Vyzařované spektrum LED diod podle složení polovodiče. [1]**

<b>Materiál</b>	<b>Vlnová délka [nm]</b>	<b>Barva</b>	<b>Napětí v propustném směru [V]</b>
GaAsP	850-940	Infračervená	1,2
GaAsP	630-660	Červená	1,8
GaAsP:N	585-595	Žlutá	2,2
AlGaP	550-570	Zelená	3,5
SiC	340-505	Modrá	3,6

V některých případech je výhodnější použít luminofor, který změní vlnovou délku světla. Hojně se této metody využívá třeba při vytváření bílého světla. Na LED vyzařující modré světlo se nanese vrstva luminoforu. Luminofor ozářený modrým světlem emituje světlo bílé. Dalším principem, jak vytvořit bílé světlo je smíchat základní barvy – červenou, modrou a zelenou (RGB složky) ve vhodném poměru. Díky míchání červené, zelené a modré je možné dosáhnout libovolného odstínu.

Na horní část LED je nutno dodat kulový vrchlík z epoxidové pryskyřice nebo akrylátového polyesteru. Tím se zabrání odrazu excitovaného světla od vzduchu zpět do materiálu diody. [2] Tento jev by snižoval účinnost a procentuálně zvyšoval sekundární absorpci fotonů.

## 1.1 Způsoby napájení LED [3].

LED jsou proudově buzené elektronické součástky emitující světlo, jejich charakteristické hodnoty určuje základní parametr – provozní proud v propustném směru. Proto je minimální požadavek kladený na obvod ovládní LED, schopnost dodávat proud

nezbytný pro vytvoření požadované intenzity světla. Nicméně kvůli rozložení (proměnlivosti) charakteristik během výrobního procesu LED, jsou pro elektrické parametry také stanoveny minimální a maximální hodnoty. Přestože se jedná o stejný produkt, bude se napětí a charakteristiky v propustném směru lišit v uvedeném rozsahu. Pokud se například přivede stejné napětí na různé vzorky jednoho výrobku, nemusí být velikost protékajícího proudu přesně stejná. Přirozeně, když se liší proud, liší se také intenzita produkovaného světla.

Pro ovládání LED je možné používat různé metody a konfigurace řídicích obvodů. Každé řešení má své výhody a nedostatky z pohledu požadovaných vlastností.

Metoda buzení LED může být stejnosměrná (statická), u které nastavený proud protéká diodou trvale nebo impulsní (dynamická), u které LED periodicky bliká. Přestože se obě metody rozsvícení LED zásadně liší, mohou se stejnosměrná i dynamická metoda pozorovateli jevit stejným výsledkem, pokud je interval mezi záblesky LED dostatečně krátký, aby blikání nedokázalo rozlišit lidské oko.

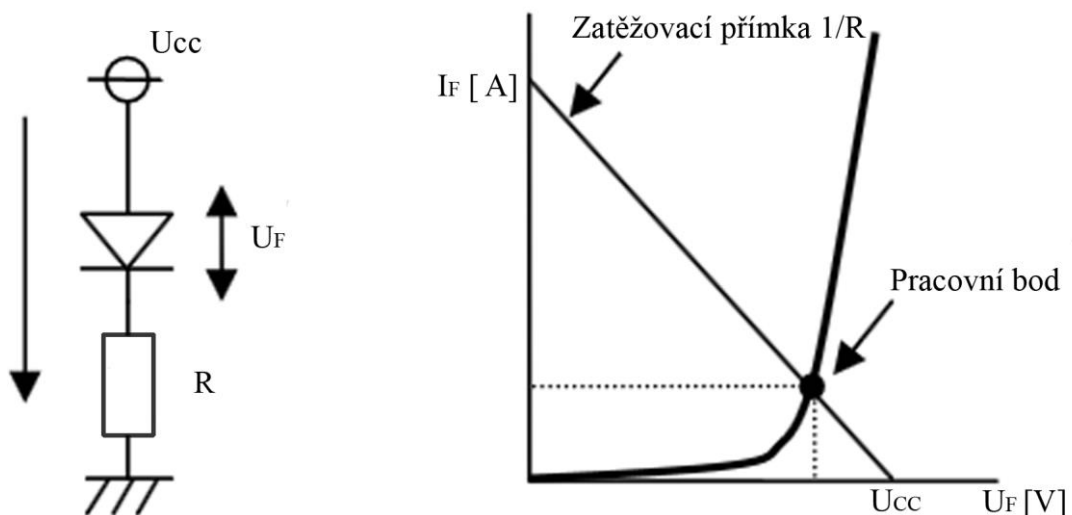
Stejnosemerna metoda buzení LED se běžně používá pro aplikace, jako jsou indikátory schopné poskytovat vysoké úrovně intenzity světla. Výhodou je trvalý svit bez nežádoucího blikání.

Impulsní metoda se většinou používá pro displeje s LED uspořádanými do matic a u světelných zdrojů s ovládáním jasu. Výhodou této metody je nízká cena, protože obvod lze realizovat pouze několika tranzistory.

### 1.1.1 Stejnosemerna (statické) řízení LED

#### Obvod napájení s konstantním napětím

Metoda používá rezistor pro omezení proudu. Proud protékající LED je určen podle obr. 1 a následující rovnice (1) [3].



Obr. 1: Zapojení diody s rezistorem a určení pracovního bodu na VA charakteristice. [3]

$$U_{CC} = U_F + R * I_F \quad [V]. \quad (1)$$

Podle známého napájecího napětí  $U_{CC}$  a proudu v propustném směru  $I_F$  můžeme podle technických údajů výrobce zjistit napětí v propustném směru  $U_F$  a určit hodnotu odporu omezujícího rezistoru  $R$ .

Hodnotu omezovacího rezistoru lze také získat z charakteristik závislosti  $I_F$ - $U_F$ . Bod nastavený podle proudu  $I_F$  na křivce je pracovní bod a přímka vedoucí z  $U_{CC}$  přes pracovní bod je zatěžovací přímka. Hodnota odporu se získá odečtením směrnice přímky  $1/R$ . Opačný postup je možné použít pro výpočet hodnoty proudu  $I_F$  z nastaveného odporu  $R$ . V praxi je třeba zvážit způsoby potlačení rozptylu  $I_F$ - $U_F$  charakteristik. Pokud je pracovní bod nastaven pro standardní charakteristiky, bude skutečný proud kolísat mezi  $I_{Fmax}$  a  $I_{Fmin}$  díky rozptylu charakteristik uvedených v technických údajích. Tyto změny lze eliminovat větším napájecím napětím nebo větším odporem.

Stejně pravidlo platí pro změny proudu  $I_F$  vlivem teploty. Takže pokud v obvodu konstantního napětí nelze zvýšit napájecí napětí nebo je v obvodu vzhledem k velikosti napájecího napětí sériově zapojeno velké množství LED, nelze pro odpor  $R$  získat dostatečně velkou hodnotu. Je nutné brát v úvahu reálné odchylky odezvy na změny teploty a kolísání napětí  $U_F$ .

### **Obvod napájení s konstantním proudem**

Pro získání stabilního proudu  $I_F$  při daném rozptylu  $U_F$  a změně teploty je nezbytný napájecí obvod se zdrojem konstantního proudu. Tato metoda má výhodu v mnohem větší odolnosti vůči změnám napájecího napětí.

Zatímco obvody napájení s konstantním proudem vynikají, pokud jde o kvalitu buzení LED, jsou méně vhodné z pohledu ceny a zastavěného prostoru.  $U_{CE}$  musí být nejméně o dva až tři volty vyšší, kvůli buzení tranzistorů a úbytku napětí (jeden až dva volty na  $R_E$ ), proto je další nevýhodou větší spotřeba energie. Při vícenásobném zapojení LED, je jednou z možností vytvořit zdroj konstantního proudu pro každé sériové zapojení. Tato metoda je vhodná pro aplikace, kde není omezující cena a zastavěný prostor.

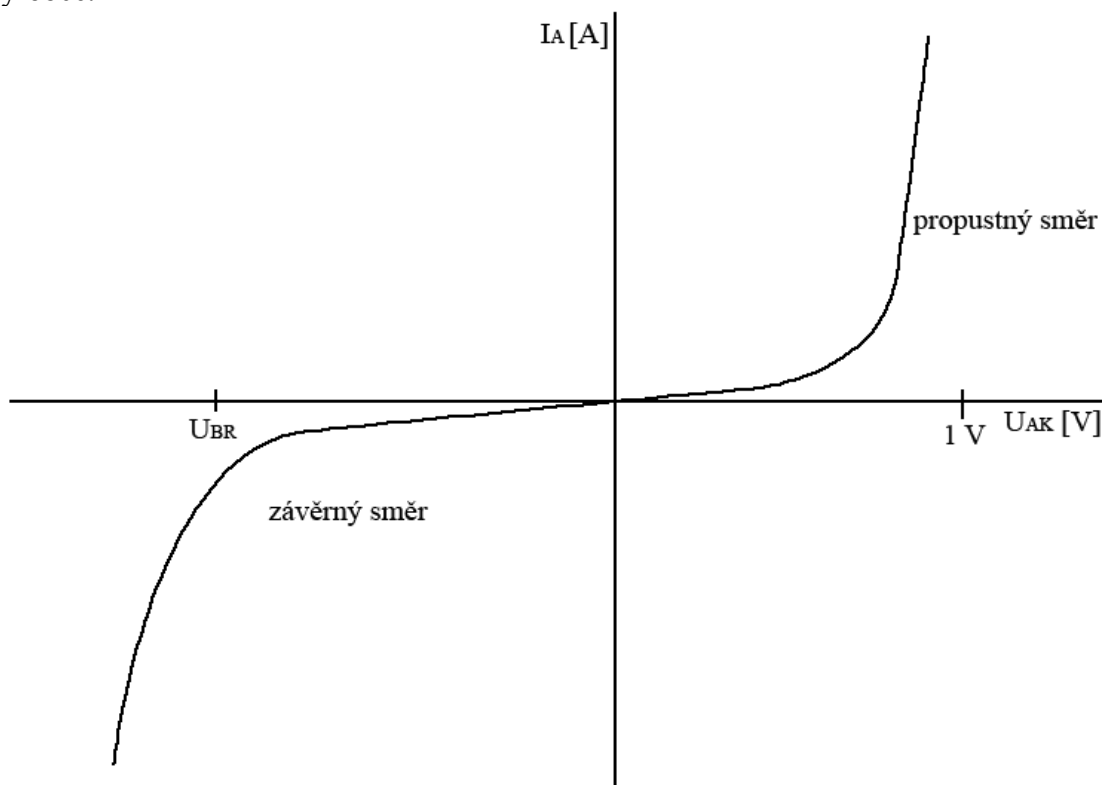
### **1.1.2 Impulsní (dynamické) buzení**

Impulsní budiče LED běžně používají bipolární tranzistor, jednoduchý spínací obvod založený na technologii MOS a elektronické obvody se zdrojem konstantního proudu, které jsou kompatibilní s různými hodnotami napětí v propustném směru. V současné době se používají integrované obvody speciálně určené pro LED osvětlení.

Pokud LED bliká na dostatečně vysoké frekvenci, jeví se její svit stejně, jako u stejnosměrného řízení (když pozorovatel stojí na místě), impulsní řízení nezpůsobí žádný problém. Pokud však frekvence blikání klesne pod určitou mez, projeví se u LED viditelné záblesky. Záblesky se obvykle projevují, pokud frekvence klesne pod padesát hertzů, proto se běžně vyžaduje frekvence spínání 70 Hertz nebo vyšší. Pokud se však pozorovatel pohybuje, případně panel vidí periferně, nebo když je LED sledována mechanickým snímačem, může být blikání patrné i při libovolně vysoké frekvenci. Proto se frekvence spínání musí přizpůsobit konkrétní aplikaci.

## 1.2 Řízení jasu LED

LED diody mohou být napájeny dvěma způsoby. Buď zdrojem konstantního proudu, nebo z napěťového zdroje s předřadným rezistorem. Na rezistoru se samozřejmě ztrácí určitý výkon. Napěťový zdroj a hodnotu odporu rezistoru je tedy potřeba zvolit s ohledem na tuto skutečnost. VA charakteristika na obr. 2 ukazuje fakt, že procházející proud roste exponenciálně vzhledem k přiloženému napětí v propustném směru diody. To opět ukazuje nevýhodu napěťového zdroje, kde malá změna napětí může znamenat velkou změnu proudu a v krajním případě následný průraz diody. To může být velký problém s přihlédnutím na rozptýl VA charakteristik diod i u stejných produktů od jednoho výrobce.



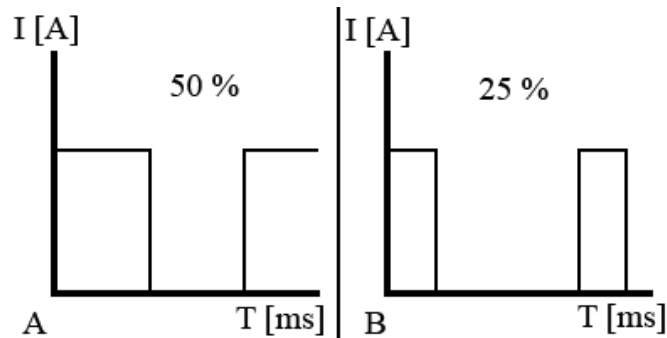
Obr. 2: Voltampérová charakteristika diody. [4]

Analogově lze tedy jas řídit pomocí napěťového zdroje a předřadného odporu, resp. potenciometru, který mění proud protékající diodou. Nebo přímo nastavením proudového zdroje.

### 1.1.1 PWM

Pulzně šířková modulace je nejvyužívanější prostředek pro řízení jasu LED diod. Kouzlo metody tkví v řízení jasu pomocí velikosti impulzu, kdy buď diodou neteče žádný proud (při logické nule), nebo diodou teče pracovní proud (logická jednička). Poměr šířky jednotlivých částí impulzů (neboli střída) nám potom udává jas diody. Lidské oko vnímá jas LED diody, jako střední hodnotu svícení.

Na obr. 3 jsou uvedeny dva příklady PWM průběhů. Průběh A, se střídou padesát procent, který lidské oko vnímá přibližně jako poloviční jas diody. Naopak průběh B, se střídou dvacet pět procent dává méně jasný výsledek a lidské oko jej vnímá přibližně jako čtvrtinu jasu.

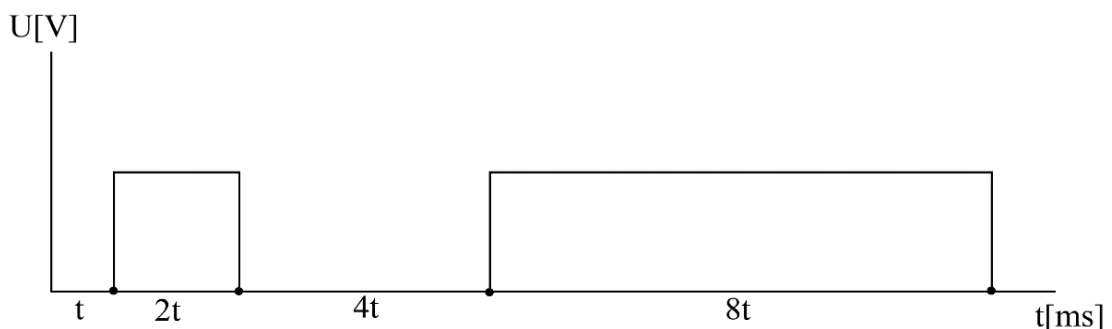


**Obr. 3: PWM průběhy.**

Řízení jasu však má frekvenční omezení. Minimální obnovovací frekvence je padesát hertz. Při nižší frekvenci lidské oko zaznamená blikání diody. Při vyšší frekvenci oko vnímá diskretní průběh jako spojitý. Druhé omezení je maximální pracovní frekvence. Pokud je pracovní frekvence stejná, jako obnovovací, například padesát hertz, není možné řídit jas, ale jen nastavit maximální svit nebo to, zda nesvítí vůbec. Se zvyšující se pracovní frekvencí se samozřejmě zvyšuje i hloubka rozlišení stupňů jasu. Pracovní frekvence je však shora omezena rychlostí řídicích prvků diody, jako jsou drivery, multiplexory či diskretní spínací tranzistory.

### 1.1.2 BCM

Binární kódová modulace vychází z binárního zápisu čísel. Pro  $n$ -bitový jas je generováno  $n$  časových úseků o různých šířkách. První časový úsek odpovídá LSB (nejméně významnému bitu) a další časový úsek je dvakrát větší než ten předešlý. Tedy v případě logické jedničky na místě po LSB bude vygenerován impuls o šířce dvojnásobku základního časového úseku. Na obr. 4 lze vidět BCM pro binární zápis hodnoty jasu 1010.

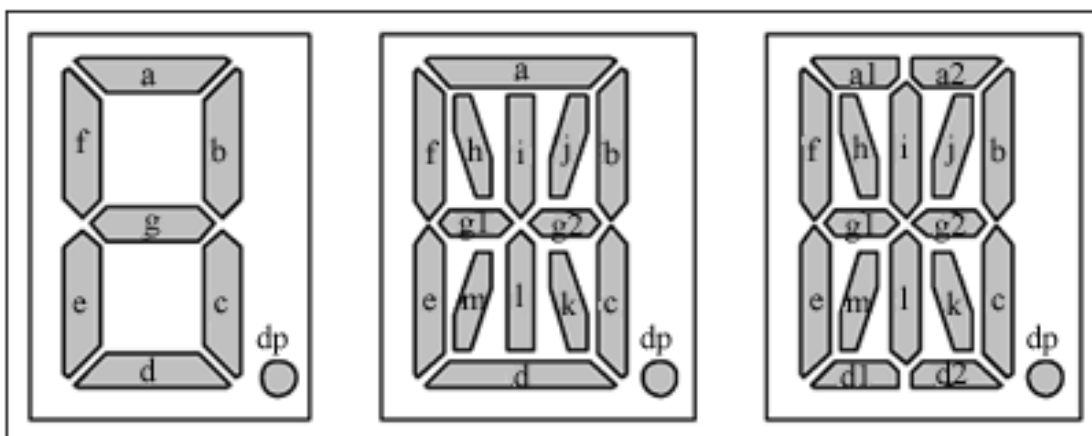


**Obr. 4: BCM průběh jasu s binárním zápisem 1010.**

Také u BCM musí platit minimální obnovovací frekvence padesát hertz. Jak jde vidět, při řízení jednobarevných LED se nevyplatí používat BCM modulaci, jelikož tvoří více užších napěťových pulzů. Její význam však je objasněn v nadcházejících kapitolách při řízení barevných LED diod.

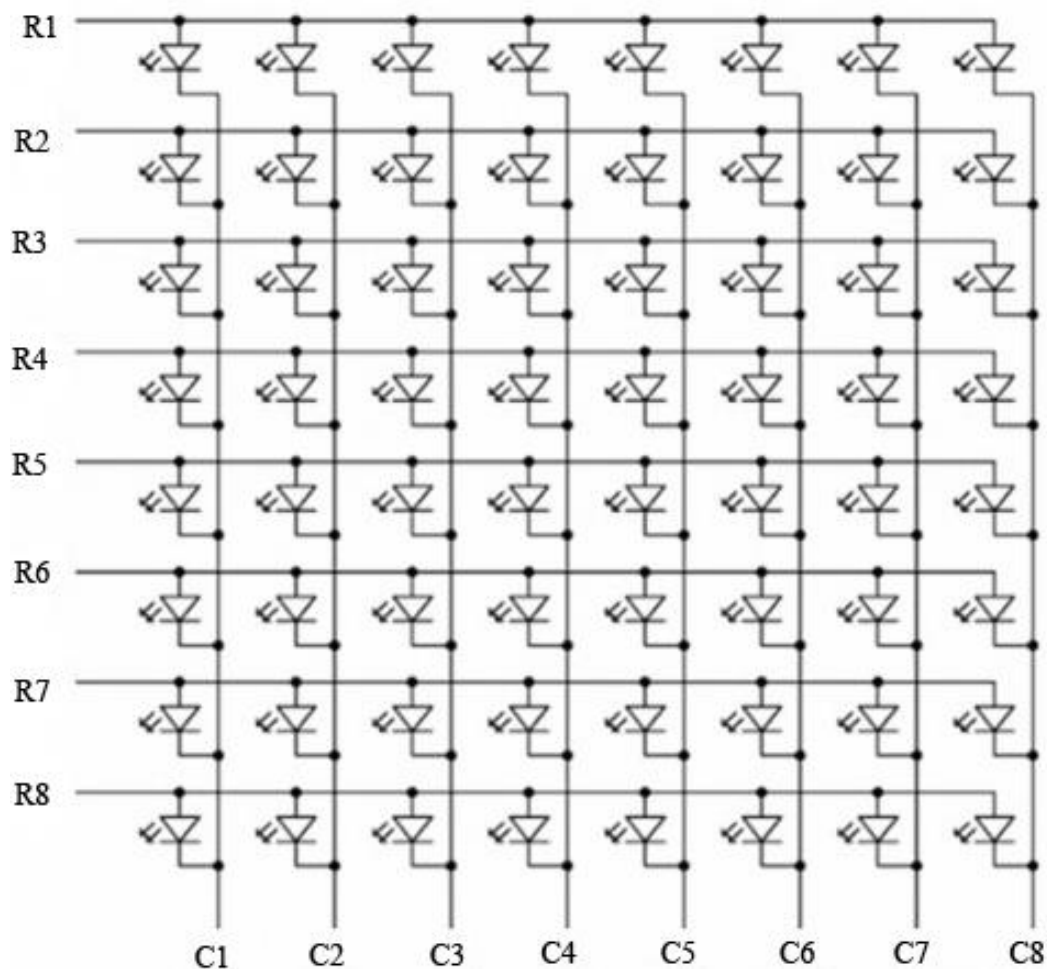
## 1.2 Zobrazení pomocí LED

Nejjednodušší LED alfanumerické zobrazovače jsou segmentové. Takové displeje mají buď sedm, čtrnáct nebo šestnáct segmentů viz obr. 5. Všechny segmenty mají společnou anodu, která je připojena na kladné napájecí napětí, nebo společnou katodu, připojenou na zem. Druhá strana diod v displeji se používá ke spínání (rozsvícení) jednotlivých segmentů. Ke spínání segmentů se používají různé elektronické obvody – nejčastěji mikrokontrolér. U zobrazovaných znaků se mohou objevovat doplňkové zobrazovače (jejich mechanické uspořádání je mimo uvedený počet segmentů), například tečka nebo třeba dvojtečka u segmentového displeje pro zobrazení časových údajů. Pro zobrazení více znaků se používá více segmentových znaků umístěných vedle sebe. Obvykle mají společné řídicí vodiče a různé způsoby připojení společných vývodů k napájecímu napětí, kterým se přepínají jednotlivé segmentové znaky.



**Obr. 5: Sedmi, čtrnácti a šestnácti segmentový displej. [5]**

Ostatní LED zobrazovače umožňují zobrazování nejen znaků, ale i jiných grafických obrazců. Buď se jedná o předem definované obrazce a dioda slouží jako zdroj světla, které těmito obrazci prochází, a tak je zvýrazňuje. Jedná se o různé kontrolky, podsvícené piktogramy, jednotlivé znaky nebo celá textová pole. Samostatnou a poměrně rozsáhlou skupinu tvoří maticové bodové LED displeje. Led diody jsou uspořádány do matice, tedy zarovnány do řádku a do sloupce obr. 6. Pro zmenšení počtu řídicích pinů jsou diody propojeny v řádku a ve sloupci. V daném časovém okamžiku je vždy sepnut pouze jeden řádek matice a sloupcovými vodiči se na konkrétním řádku nastaví svícení jednotlivých LED.



Obr. 6: Zapojení maticového LED zobrazovače 8 x 8. [6]

### 1.2.1 Monochromatické zobrazení

Při monochromatickém zobrazení používá celý zobrazovač jedinou barvu LED. Jednobarevné zobrazovače jsou nejvyužívanější LED zobrazovače kvůli jejich ceně, ve které se odráží samozřejmě nejsnadnější řízení a nejmenší nároky na kvantitu a kvalitu součástek tvořících řídicí diody. Nejpoužívanější barvy LED jsou zelená, jantarově žlutá a červená barva, ale komerční prostor samozřejmě nabízí výběr z celé škály barev. LED panely mohou být složeny z různě barevných bloků.

## 1.2.2 Semi barevné zobrazení

Využití dvou barev dává nové možnosti pro zajímavější a přehlednější zobrazení grafických dat. Používají se k tomu dvoubarevné LED diody, které mají ve většině případů LED se zelenou a červenou barvou, zapouzdřené do jediné součástky. Toto uspořádání umožňuje zobrazení znaků ve třech barvách – zelené, červené a žluté (oranžové), která vznikne současným rozsvícením obou barev. Dalšího druhu semi barevného zobrazení lze dosáhnout použitím tříbarevných diod v maticovém panelu, kde se při prvním kroku časového multiplexu jedna barva zobrazuje text/obrazec a ve druhém kroku barva pozadí. Při řízení je nutné postupně poslat do budičů LED dvojnásobný počet dat. První část dat pro text/obrazec a druhá část dat pro pozadí. Díky tříbarevným LED lze dosáhnout větší variabilitu barvy textu a pozadí. Oproti plně barevnému zobrazení má toto řešení daleko menší nárok na výpočetní čas řídicího prvku, kterým v tomto případě může být mikrokontroler.

## 1.2.3 Plně barevné zobrazení

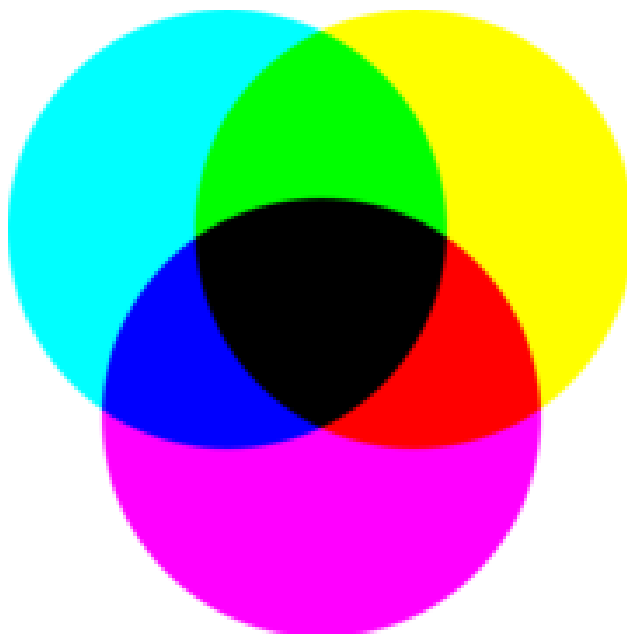
Plně barevné zobrazení vyžaduje zapojení tří barevných LED, které realizují červenou, modrou a zelenou složku barevného spektra. V minulosti byly využívány pro barevné zobrazení tři diskrétní LED diody umístěné vedle sebe nebo do trojúhelníku. V dnešní době jsou již všechny tři RGB barvy integrovány do jedné součástky. Pro řízení panelu je ve většině případů nutno použít rychlejší součástky i výkonnější mikroprocesory např. řadu ARM, nebo obvody, které zpracovávají úkoly paralelně, jako jsou například FPGA, ASIC a SoC. Při postupném zpracování dat má MCU problém se zvládnutím odesílání všech dat do budičů LED a současné komunikace s nadřazenými systémy. Díky rychlejším součástkám, velikosti paměti a větší pracovní frekvenci se dosahuje větší barevná hloubka i větší plocha zobrazení informačního panelu. Často se tyto časově kritické rutiny realizují systémem, který tvoří několik mikrokontrolérů.

## 2 BARVA

Barva je vlastnost látky, která závisí na mnoha faktorech. Může vzniknout vyzářením fotonů o určité vlnové délce, nebo také odrazem záření od látky. Vnímání barvy různými živočichy je různé. I mezi lidmi je vnímání barvy částečně subjektivní, což dokázalo nespočet studií. Lidské oko má tři druhy čípků, které zajišťují vnímání barev. Mají různé množství zastoupení v oku. Červených je šedesát čtyři procent, zelených třicet dva procent a modrých čtyři procenta [6]. Z toho důvodu je lidské oko nejméně citlivé na modrou barvu. Základní míchání barev může být subtraktivní a aditivní.

### 2.1 Subtraktivní míchání barev

Metoda, při které se jednotlivé složky barvy odečítají – viz obr. 7. Používá se pro barevný model CMY a CMYK. Základní barvy, které se vzájemně odečítají, jsou azurová (Cyan), purpurová (Magenta) a žlutá (Yellow). V případě modelu CMYK se ještě používá černá (black). Systém CMY, resp. CMYK, je rozšířený barevný model v tiskařství. Při překrytí dvou barev vznikají základní barvy RGB a průnikem všech tří barev je dosaženo černé. Z důvodu šetření barev je právě využíván model CMYK, kdy tmavé barvy a stupně šedi vytváří černý tiskový cartridge. Na papír se nanášejí barvy jednotlivě, tedy nejsou smíchány, ale tisknou se velmi blízko sebe. Oko tak není schopno rozeznat jednotlivé barevné body, které splývají a tvoří daný odstín barvy. [7]



Obr. 7: Subtraktivní míchání barev. [7]

## 2.2 Aditivní míchání barev

Aditivní míchání je přesný opak subtraktivního principu. Barvy se při svém průniku sčítají – viz obr. 8. Míchání je postaveno na použití světlených zdrojů a vnímání součtu složek světla lidským okem.

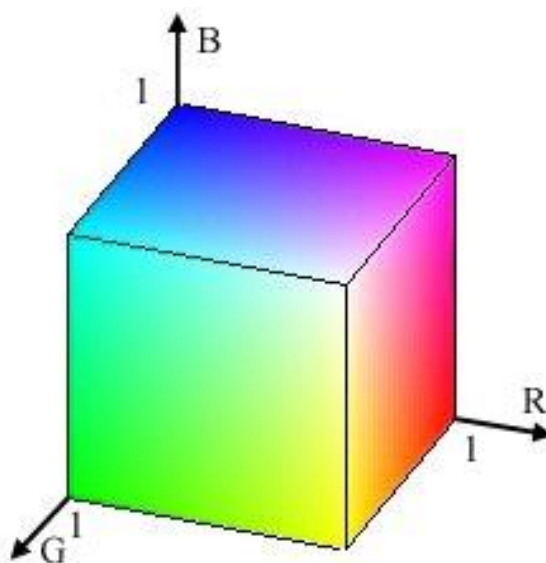


Obr. 8: Aditivní míchání barev. [7]

Modelů pro aditivní míchání je celá řada a výběr modelu záleží na aplikaci [7]:  
**YIQ** – model pro přenos TV signálu v Japonsku, Severní a Střední Americe,  
**PAL, SECAM** – přenos TV signálu v Evropě a Asii,  
**HSL/HSV/HSB** – pro práci s grafikou,  
**Munsell** – užití v medicíně, psychologii a průmyslu,  
**CIE** – měření odrazu a přenosu světla objekty, počítačová grafika,  
**RGB** – nepoužívanější model zaměřený na výstupní zařízení.

## 2.3 RGB model

RGB systém je složen ze základních barev pro aditivní míchání. Procentuální zastoupení jednotlivých složek, tedy červené, zelené a modré určuje konečný odstín barvy. Jsou-li zastoupeny všechny složky stejně, vzniká bílé světlo či stupně šedi. Model se dá představit, jako krychle na jejíž základních osách leží základní barvy a krychle má velikosti hran rovny jedné, resp. sto procent. Na obr. 9 je přesně taková krychle a jde z ní například vyčíst již zmíněný fakt: použijeme-li všechny tři složky se sto procentní intenzitou, vznikne nám bílé světlo.



Obr. 9: RGB model. [7]

### 1.1.3 Barevná hloubka

Pro potřeby číslicové techniky se hodnoty úrovní barev jednotlivých složek předávají v bitové reprezentaci. Podle počtu bitů vyhrazených pro základní barvy se určuje barevná hloubka. Používá-li se například k reprezentaci barev jen jeden bit, je tedy hloubka jednobitová. To znamená, že pro reprezentaci všech barev stačí jen log. nula, která reprezentuje černou barvu a log. jedničky, reprezentující barvu bílou. Čím je vyhrazeno více bitů pro jednu barvu, tím existuje více procentuálních diskretních úrovní na osách a vzniká tak větší množství barevných kombinací. V tab. 2 lze vidět bitová reprezentace barev a počet kombinací barev.

Tab. 2: Barevná (bitová) hloubka.

Barevná hloubka [bit]	R	G	B	Počet barev	Název
8	3	3	2	256	
15	5	5	5	32 768	Low Color
16	5	6	5	65 536	High Color
24	8	8	8	16 77 216	True Color

Při bitové hloubce šestnáct je možné použít model RGBA. Složka A zastupuje průhlednost. Pro 16 bitů je RGBA složena z hodnot buď 5551, nebo 4444. Průhlednost – alfa kanál vyjadřuje vztah jednotlivého barevného bodu vůči pozadí. V případě jednoho bitu pro kanál znamená log. nula neprůhledný a log. jedničky zcela průhledný bod. Má-li alfa kanál více bitů, vyjadřují bity procentuální průhlednost barevného bodu vůči pozadí. Tato práce pracuje s hloubkou šestnáct bitů bez alfa kanálu.

### 1.1.4 Gamma korekce

Lidské oko vnímá barvy trochu odlišněji, než je výstupní lineární intenzita světla. Aby byly barvy zobrazovány co nejreálněji pro lidské oko, zavádí se tzv. gamma korekce. Její vzorec [8] je:

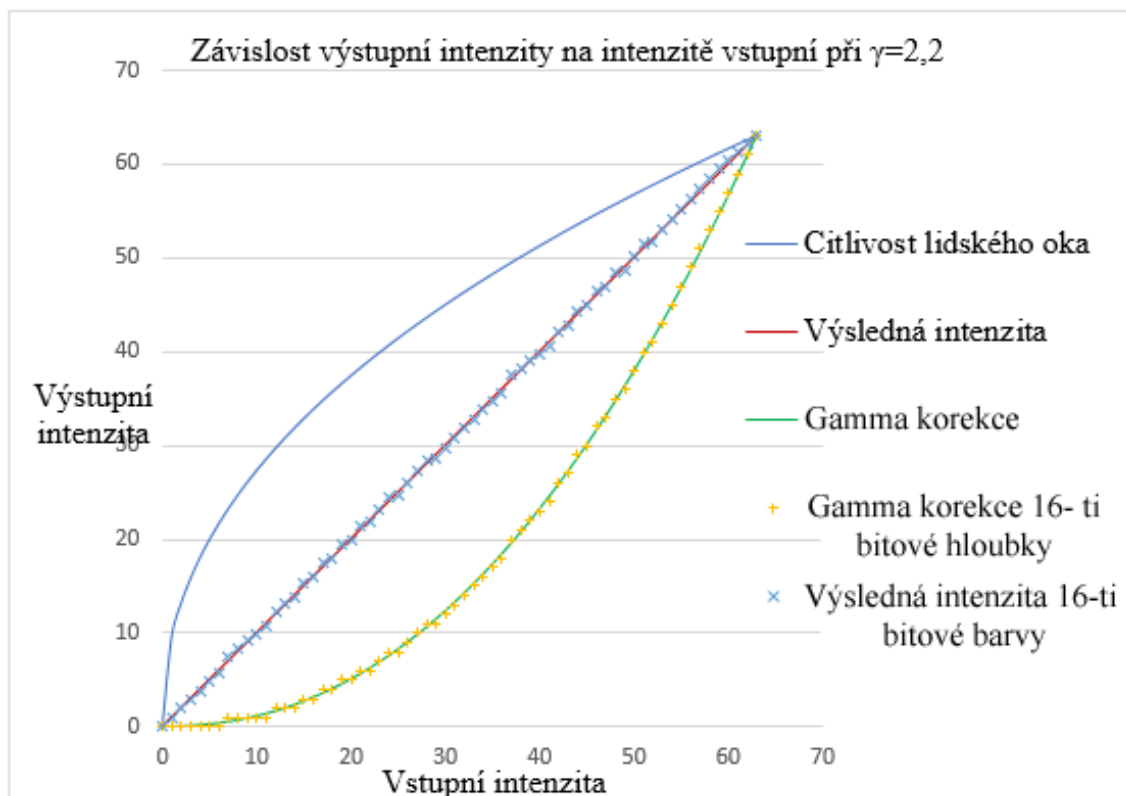
$$I_{vyst} = I_{vst}^{\gamma}, \quad (2)$$

kde  $I_{vyst}$  je výstupní proud,  $I_{vst}$  je vstupní proud,  $\gamma$  je konstanta s velikostí 1- 2,2. Aby byl vzorec použitelný v číslicové technice, tak pro každou složku barvy se zjistí její výstupní bitový zápis:

$$b_{vyst} = b_{max} * \left(\frac{b_{vst}}{b_{max}}\right)^{\gamma}, \quad (3)$$

kde  $b_{vyst}$  je výstupní bajtový zápis složky,  $b_{max}$  maximální hodnota n-bitů a  $b_{vst}$  vstupní bitový zápis barevné složky.

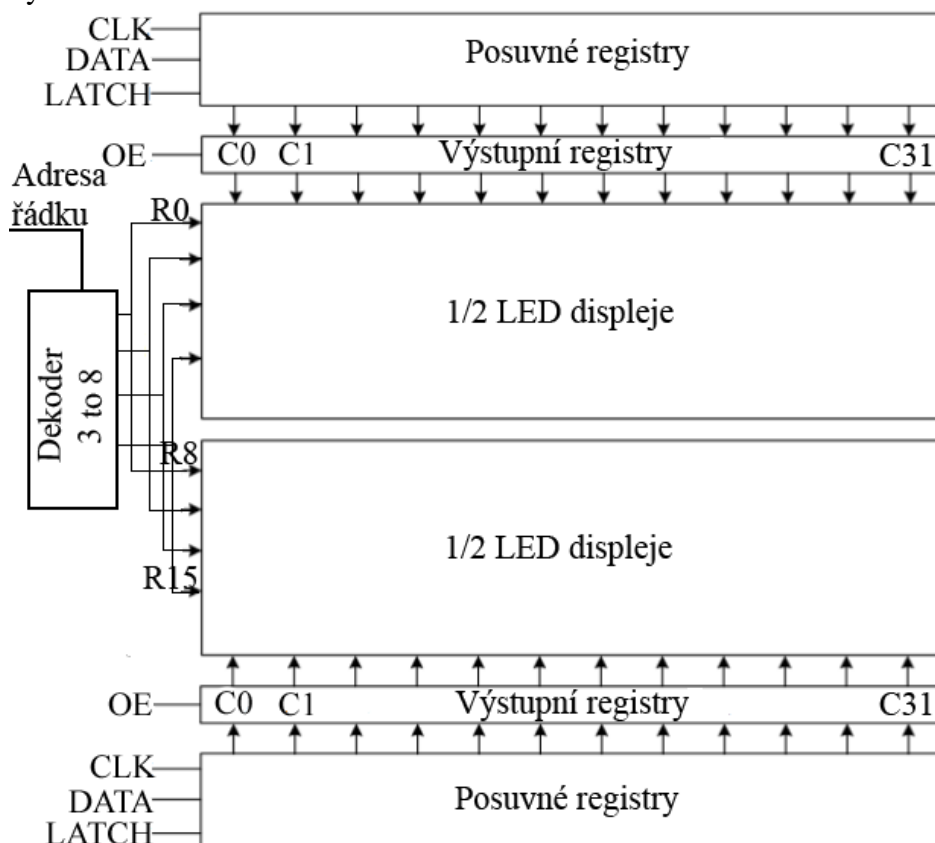
Grafické znázornění vnímání oka, výsledné intenzity i vstupní intenzity a odchylky šesti bitové barevné složce jde vidět na obr. 10. Aby šla vidět odchylka, která vznikne gamma korekcí, jsou voleny hodnoty os v bitovém zápisu. Maximální hodnota je šedesát tři, což odpovídá stoprocentní intenzitě. Na obrázku jde vidět, že odchylky od ideální výsledné intenzity jsou zcela zanedbatelné.



Obr. 10: Závislost výstupní intenzity na vstupní intenzitě a gamma korekce.

### 3 LED MATICE

Jak již bylo zmíněno v první kapitole, pro grafické zobrazení se velmi často využívá LED matice. Pokud se nejedná o malou matici s několika diodami, kde každé diodě připadá jeden řídicí pin a spínací prvek, jsou diody uspořádány do řádků a sloupců viz obr. 6. Řízení takto propojené struktury je složitější. Při malých počtech řádků a sloupců je možné matici řídit jen pomocí spínačů, obvykle tranzistorových, které jsou ovládány I/O piny MCU, případně FPGA. Při větším počtu LED nebo při aplikaci barevných LED diod již jsou matice řízeny přes multiplexory, resp. dekodéry, a drivery. Typické zapojení tvoří řádky, které se postupně spínají v časovém multiplexu a sloupce ovládané drivery typu posuvný registr. V případě plnobarevného panelu je použito třikrát více driverů. Ve speciálních aplikacích lze počet driverů snížit až o dvě třetiny pomocí postupného spínání barev. Toto řešení je však na úkor intenzity světla. Při použití driverů typu posuvný registr závisí pořadí odesílaných dat na vzájemném propojení driverů. Způsobů plnění posuvných registrů je několik. Odzadu, odpředu (v případě multiplexu řádků), shora, ze spodu (v případě sloupcového multiplexu), nebo různě po blocích při fragmentaci panelů na bloky.



Obr. 11: Zapojení monochromatického displeje od firmy Adafruit. [9]

Blokové schéma zapojení maticového LED modulu od firmy Adafruit industries demonstruje nejběžnější řízení většího počtu LED diod. V tomto případě pětiset dvanácti LED uspořádaných do šestnácti řádků a třiceti dvou sloupců, viz obr. 11. Na obrázku je

vidět rozložení matice barevných LED do dvou bloků. První blok je pro řádky nula až sedm a druhý blok je pro řádky osm až patnáct. Přepínání řádků nula až sedm, resp. osm až patnáct je řízeno BCD dekodérem. Řízení řádků obstarává v každém bloku několik driverů s posuvnými registry a paměťovým blokem. Podle obrázku se oba dva bloky plní po řádcích shora dolů a bity posuvných registrů od posledního sloupce.

### 3.1 Řízení řádků

Pro časový multiplex jedné strany matice, obvykle řádků, se využívá kódování one hot. Což znamená, že vždy je sepnutý jen jeden řádek v bloku. To dává samozřejmě omezení počtu současně sepnutých řádků. S počtem zvyšujících se řádků klesá intenzita svitu jednotlivých LED. Maximálně se spíná šestnáct řádků v bloku, intenzita LED je pak o patnáct šestnáctin snížena vůči trvalému sepnutí. Malá intenzita může být kritickou veličinou při použití panelu např. ve venkovním prostředí, kdy okolní osvětlení sníží viditelnost panelu. Řešením tohoto problému může být zvýšení proudu, který protéká LED. Obvykle je maximální hodnota pětkrát až šestkrát vyšší než nominální. V aplikacích optimalizovaných na intenzitu svícení se používá časový multiplex se čtyřmi kroky. Běžně se volí multiplex s osmi kroky. Pro potřeby spínání vždy jednoho řádku slouží několik součástek, které musí mít výstupy uzpůsobeny na proud, který protéká všemi současně rozsvícenými diodami. Nejvíce se využívají BCD dekodéry, například tři vstupy na osm výstupů, dva na čtyři, čtyři na šestnáct, které binární kód převedou na kód one hot (jedna z N). Další variantou je opět dekodér, ale se sériovým vstupem, který potřebuje jen dva řídicí piny (data a latch). Může ušetřit jeden, resp. dva piny na úkor výpočetního výkonu řídicí jednotky. Poslední využívanou součástí je čítač s výstupním registrem, modifikovaný na kód one hot, když čítač načítá určitou hodnotu. Taková součástka musí mít zpětnou vazbu na řídicí systém, jinak hrozí vznik chyby v důsledku zobrazování špatného řádku.

### 3.2 Řízení sloupců

Řízení sloupců probíhá převážně pomocí LED driverů. Což jsou součástky, které se skládají ze tří hlavních částí. První je posuvný registr, druhá část je výstupní registr a třetí obvod napájení LED viz obr. 11. Posuvné registry mají dva řídicí vodiče:

1. **CLK** vodič-při nástupní/sestupní hraně se posouvají data v registru,
2. Datový vodič.

Výstupní registr má dva zbylé vodiče:

3. **OE** – output enable, povoluje/zakazuje výstupní registr,
4. **LATCH** – provádí přepis dat z posuvného registru do paměti výstupního registru.

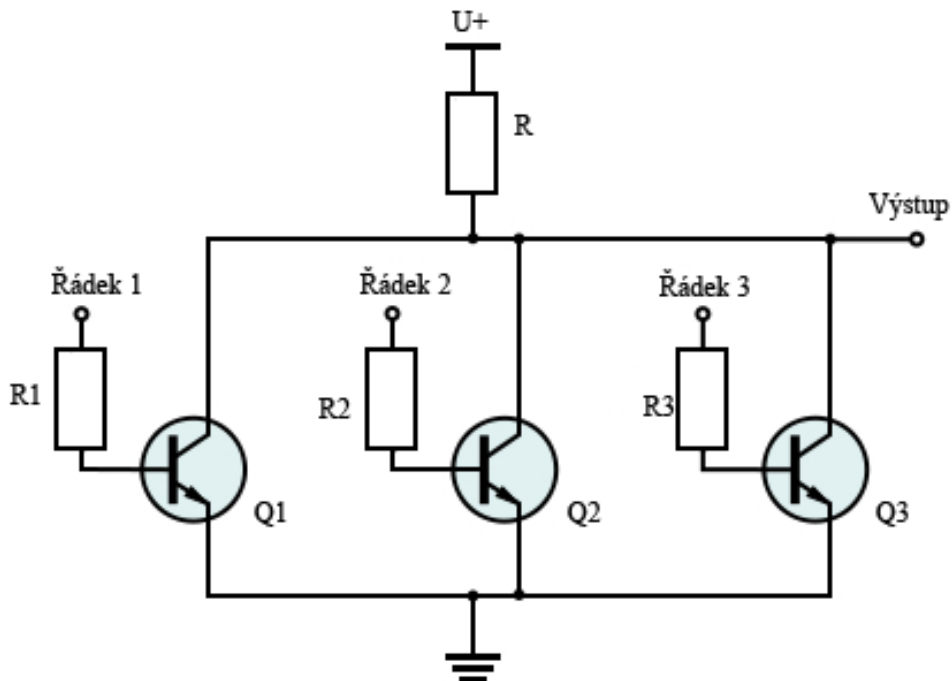
U LED driverů hraje velkou roli maximální frekvence plnění posuvných registrů.

### 3.3 Identifikace rozměrů panelu

Identifikace rozměrů panelu je nutná ve chvíli, kdy se pro jeden řídicí prvek používají různé velikosti panelů. Řídicí prvek k tomu musí být uzpůsoben. V praxi to znamená mít v sobě testovací funkci, která zjišťuje velikosti panelu a také mít řídicí strukturu uzpůsobenou proměnlivé velikosti. Tedy řídicí prvek uzpůsoben řízení největšího rozměru panelu a případně menší velikosti dát nadřazené jednotce vědět o menší velikosti a také v čítacích registrech snížit číslo čítaných řádků a sloupců. Pro samotné testování je potřeba přivést výstup z LED driveru a také z výstupů řádků.

Za předpokladu konstantního počtu sloupců ve všech řádcích je možné sledovat jen jeden blok posuvných registrů. Jednoduše datový výstup posledního posuvného registru přivedeme na vstupní pin testovacího prvku. Poté stačí jen poslat do driverů testovací řetězec, který je čten z výstupu posledního registru. Zároveň čítač sloupců počítá počet hran odeslaných do driveru, a tedy počet dat, resp. sloupců. Testovací řetězec může mít libovolnou podobu, ale doporučuje se nejdříve odeslat nulová data, což znamená samé nuly do maximálního počtu sloupců a následně testovací řetězec. Nejjednodušší je poslat za nulovými tady jednu jedničku. Což ale při určitém procentu chybovosti může znamenat špatně interpretovanou velikost. Tedy je lepší řetězec zkomplikovat třeba na 1011, to už záleží na návrháři. Jak prvotní nulování dat, tak komplikovanější řetězec mají za úkol správný odhad velikosti sloupců.

Při zjišťování počtu řádků je samozřejmě vše složitější. Je nutné při testování monitorovat jednotlivé řádky. Z důvodu velkého počtu řádků však není možné všechny výstupy z řádků přivést na piny nadřazeného prostředku. Pro sloučení výstupu z řádků se hodí více vstupové hradlo OR případně NOR. Tady se naskytá mnoho příležitostí od DDL přes TTL až po MOS hradla. Hojně se tu využívá tzv. obvodového součtu. Na můžeme vidět negaci odporového součtu, tedy hradlo NOR. V praxi se používají převážně MOS tranzistory typu N, jelikož minimálně ovlivňují sepnutý panel.



Obr. 12: Obvodový součet NOR. [32]

## 4 ŘÍDÍCÍ OBVODY

Řízení LED panelu lze rozdělit na vnější a vnitřní. Vnější řízení je obvykle z počítače, ať už klasického nebo palubního. Ten ale není tématem práce. Vnitřní řízení obstarává MCU nebo FPGA obvod, případně kombinace obou.

### 4.1 Způsoby přenosu zobrazení na informační panel

Informace, která se má zobrazit na LED panelu se přenáší z nadřazeného prostředku pomocí různých komunikačních rozhraní a protokolů. Přenášená data se transformují na tvar, který jednoznačně určí, na kterém místě a jaká dioda bude svítit. Přenos informací mezi počítačem a informačním panelem je možné třemi základními způsoby. Případně se používá jejich kombinace.

1. **Kódové řízení:** předem domluvenými kódovými povely na obou stranách komunikace. Pro jednotlivé kódy je v databázi panelu uložen konkrétní obsah zobrazení. Databáze umožňuje také vytváření autonomních cyklů zobrazení. Povely se hodí při vybarvování většího obrazce, pro adresaci předem uložené grafiky a textů nebo pro různé dynamické změny obrazu. Hlavní výhodou tohoto řízení je přenos krátkých zpráv pro realizaci složitých zobrazení. Nevýhodou je nutnost interní databáze v panelu a její aktualizace.
2. **Textové řízení:** je výhodné pro psaní textů. Přičemž grafické fonty písma musí být uloženy ve vnitřní paměti panelu. Pomocí řídicí posloupnosti ASCII znaků poté probíhá jejich adresování v paměti, vyčítání a následně ukládání do pracovní paměti pro zobrazení.
3. **Grafické řízení:** má nejmenší nároky na řídicí systém panelu. Komunikačním protokolem jsou do panelu odesílány data o grafice, tedy přímo obrazce, které se mají pozorovat na panelu. Tyto obrazce jsou zpracovány a hned nahrány do pracovní paměti zobrazení. K tomuto účelu nejlépe slouží BMP formát, který má 16 - ti bitovou hloubku.

### 4.2 Komunikace s nadřazeným prostředkem

Pro komunikaci s nadřazenými prostředky je nutno použít spolehlivá hardwarová rozhraní a protokoly, které omezují ztrátu informace při přenosu na několik jednotek až desítek metrů. Například v tramvajích je hlavní řídicí jednotka panelů umístěna u řidiče, ale informační panely jsou rozmístěny po celé délce dopravního prostředku.

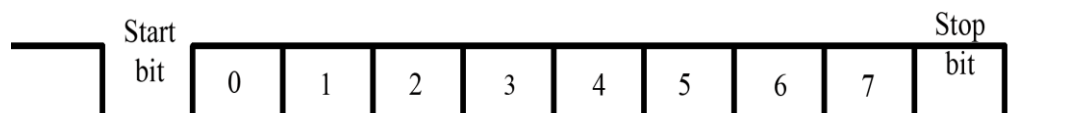
#### 4.2.1 UART

Asynchronní sériová komunikace, která probíhá po dvou vodičích. Tx-odesílací pin, Rx-přijímací pin. Třetí vodič je společná zem, vůči které jsou posílány signály. Komunikaci řídí nadřazený prvek Master a na druhé straně je podřízený prvek Slave. Oba komunikující systémy musí mít předem dohodnutou přijímací/vysílací frekvenci

(přenosovou rychlost). Ta se udává v Baud, což udává maximální počet změn signálu za jednu sekundu. Pro sériovou komunikaci je ekvivalentní jednotka b/s (bit za sekundu), počet bitů přenesených za vteřinu. Dalším důležitým faktorem, který oba dva komunikující systémy musí znát, je přesný průběh komunikace. UART komunikace má volitelné parametry přenosu dat:

1. délka přenášeného slova: pět až osm bitů,
2. **paritní bit**: lichá/sudá parita, žádný paritní bit, mark a space,
3. **stop bit**: jeden, jeden a půl a dva stop bity.

Nejpoužívanější nastavení parametrů je osmi bitové slovo bez parity s jedním stop bitem, tak jak jde vidět na obr. 13.



**Obr. 13: Odeslání jednoho slova protokolem UART.**

Synchronizace datového přenosu probíhá přes vzorkování na základě start bitu. Ten je vždy nastaven do logické nuly. Pro příjem slova se čeká, až je přijímací pin nastaven druhou stranou do logické nuly. Jakmile se tak stane, dojde k odečtení poloviny bitového intervalu a podle této hodnoty probíhá následné vzorkování přijímaného slova. Slovo je odesíláno od nejnižšího bitu tedy LSB. Za slovem je stop bit, který je vždy logická jednička.

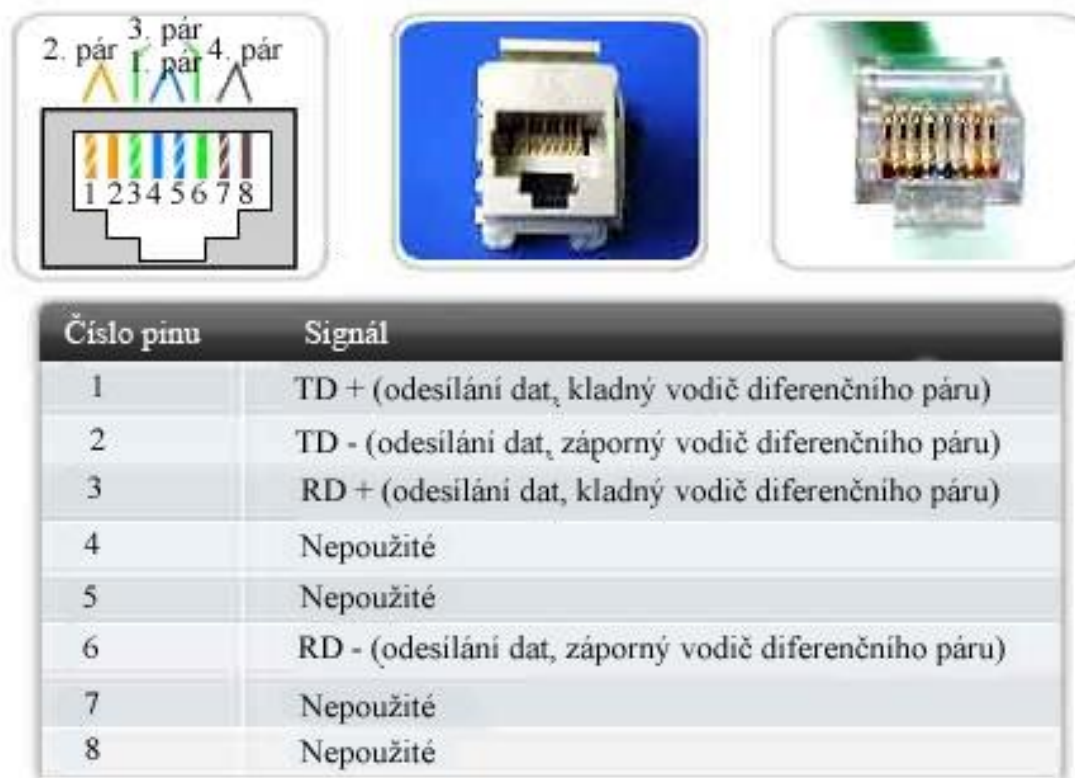
Pro správnou komunikaci musí být dodržena všechna výše popsaná pravidla. Další podmínky určuje komunikační protokol, který je již libovolný a závisí na jednotlivých aplikacích a potřebách uživatele.

## 4.2.2 Ethernet

Jedná se o síťový komunikační kanál, který je z velké části standardizovaný normou IEEE 802.3. Jako přenosová média se používají koaxiální kabely, kroucené dvojlinky a optická vlákna. Z počátku byla komunikace po Ethernetu pouze polo duplexní, což znamená, že měla jen jeden vodič pro příjem i odesílání dat. Duplexní komunikace mohla probíhat až po implementaci více kroucených dvojlinek viz obr. 14, nebo v případě optických vláken zavedením více vidových optických vláken, či rozdílné vlnové délky vysílaných paprsků.

Jak jde vidět na obr. 14 komunikace na kroucené dvojlince probíhá po diferenčním páru. Signál je vzorkován určitou rychlostí a zjišťují se změny signálu při vzorkovacím čase. Přejde-li napětí ze záporného na kladné, je přechod interpretován, jako log. jedna. Je-li přechod opačný, tedy z kladného napětí na záporné interpretace je log. nula. [11].

### RJ-45 Konektor



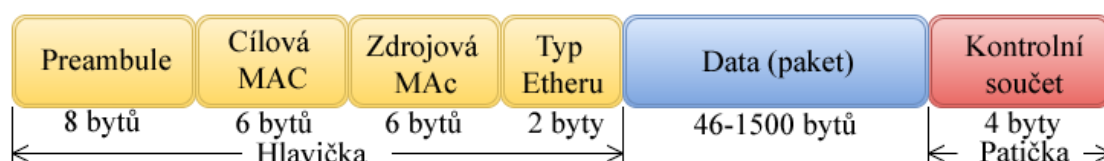
**Obr. 14: Rozložení vodičů ethernetu v konektoru RJ-45. [10]**

Od počátků Ethernetu, kdy byla rychlost deset Mbps, se rychlost zvýšila až na tisícinásobek. V následující tabulce je příklad rychlostí a maximálních délek komunikačních médií při jejich rozdílných kabelech.

**Tab. 3: Srovnání rychlostí a dosahu kabelu ethernetu. [12]**

Přenosová rychlost [Mbit/s]	Typ ethernetu	Kabel	Maximální délka kabelu [m]
10	10Base-5	Koaxiální	500
10	100Base-Tx	Kroucená dvojlinka	100
100	100Base-Tx	Kroucená dvojlinka	100
100	100Base-Fx	Optické vlákno	400
1 000	1000Base-T	Kroucená dvojlinka	100
1 000	1000Base-CX4	Optické vlákno	2 000
10 000	10GBase-LX4	Kroucená dvojlinka	100
10 000	10GBase-LX4	Optické vlákno	300

Komunikace by nemohla fungovat bez protokolů, tedy pravidel, podle kterých probíhá komunikace. Protokolů je celá řada např. IP, TCP, HTTP nebo ethernet. Posledně zmíněný protokol je kupříkladu využíván pro komunikaci v lokálních sítích. Jako adresace, pro který přístroj je zpráva určena slouží MAC adresa, která je pro každý přístroj na světě jedinečná. MAC adresa příjemce i odesílatele je odeslána v tzv. rámcích. Takový rámec je na obr. 15. Hlavním cílem osmibajtové preambule je informovat ostatní zařízení, že se začíná posílat ethernetový rámec. Preambule začíná jedničkou a potom se střídá sekvence deset. Poslední dva bity jsou 0b11. Typ ethernetu může být použit na zapsání, jaké jsou v paketu data, nebo je-li hodnota menší než tisíc pět set, značí velikost paketu. Kontrolní součet (FCS-Frame Check Sequence) říká, zda byl rámec odeslán správně nebo s chybou. [13]



Obr. 15: Ethernetový rámec. [14]

### 4.2.3 HDMI

High Definition Multimedia Interface je nejvíce využívané digitální audio-vizuální rozhraní, které umožňuje přenos nekomprimovaného zvuku a obrazu. HDMI je schopno přenášet obraz s různým rozlišením, od malého až po velmi kvalitní. Od jeho vzniku v roce dva tisíce dva bylo vydáno mnoho verzí do kterých byly přidány nové funkce a vylepšení. Rozhraní má několik vrstev.

#### Fyzická vrstva

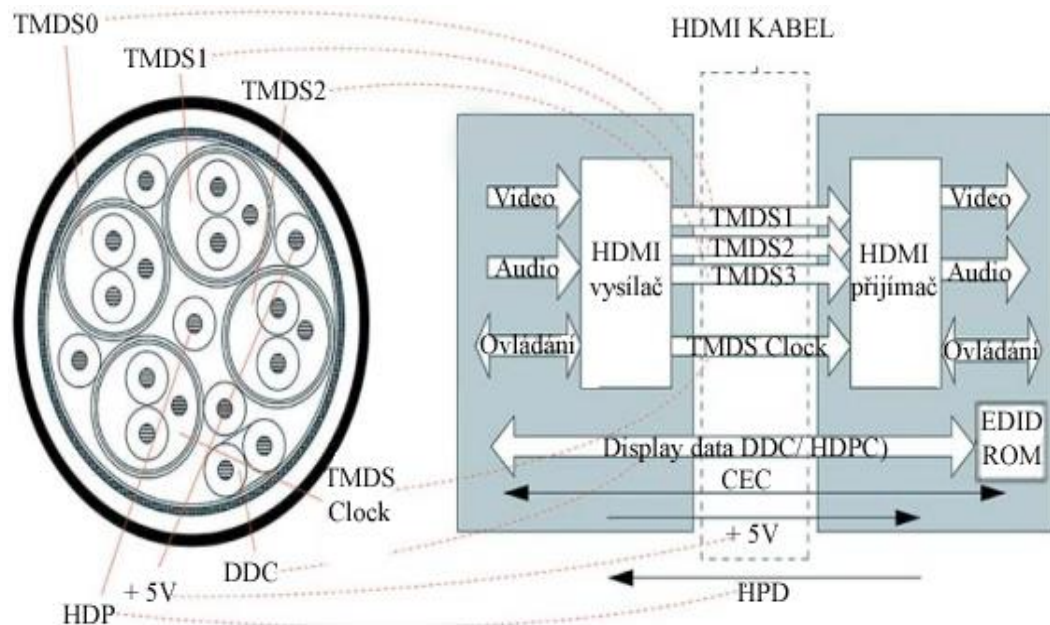
Fyzická vrstva se stará o propojení mezi vysílačem a přijímačem. Zahrnuje kabel, konektor a samozřejmě zásuvku. Kromě těchto hmatatelných věcí obsahuje i LVDS standard, který se stará o správnou a rychlou komunikaci. Vysoká rychlost bez rušení se dosahuje, podobně jako u ethernetu, kroucenými diferenčními vodiči. Na nich vznikají malé úbytky napětí max. dvacet milivoltů, které jsou na straně přijímače rozeznány a je jim přiřazena příslušná logická hodnota [15]. Díky možnosti malých úbytků napětí není HDMI přenos jen energeticky úsporný, ale také velmi odolný. Díky diferenčním párům je možné komunikovat vysokou rychlostí.

Kabelů existuje celá řada pro běžné užití nebo průmyslové. Od standardu 1.3 se dělí na dvě základní kategorie:

**kategorie 1** – pro pásmo do 75 MHz (1080i/720p),

**kategorie 2** – pro pásmo do 340 MHz (1600p). [16]

Na obr. 16 je možné vidět průřez HDMI kabelem bez ethernetu a celková architektura fyzické vrstvy.



**Obr. 16: Průřez vodiče a architektura fyzické vrstvy HDMI. [17]**

Pro různá zařízení a různé formáty jsou definovány čtyři různé konektory. Typ A má devatenácti pinový konektor, který je nejvíce používaný. Typ B má konektor s dvaceti devíti piny a poskytuje dvojnásobnou šířku pásma. Typy C a D jsou opět devatenácti pinové a jsou určeny pro menší přenosná zařízení. [19] Na obr. 17 je znázorněno rozložení pinů.

**Piny 1-9** jsou tři datové páry a k nim příslušné tři piny zajišťující stínění.

**Piny 10-12** mají na starost dva vodiče hodinového signálu a třetí je stínění.

**Pin 13** přenáší CEC informaci.

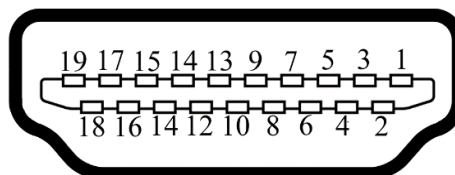
**Pin 14** je nevyužitý.

**Pin 15 a 16** je datový pár DDC užívaný na identifikační kanál mezi zařízeními.

**Pin 17** je připojený na zem.

**Pin 18** je napájení pět voltů.

**Pin 19** je detekce připojení. [18]

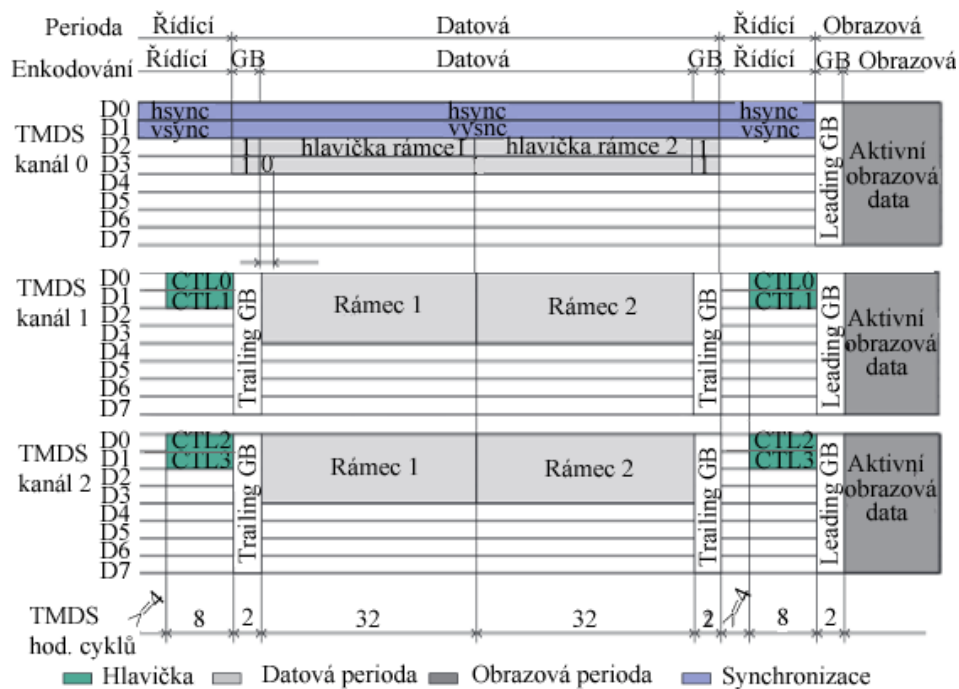


**Obr. 17: HDMI konektor typu A. [19]**

### TMDS signalizace

Signalizace slouží ke kódování/dekódování HDMI dat do TMDS znaků. TMDS znaky jsou deseti bitové. Kódování má za úkol minimalizovat počet přechodů mezi logickými

úrovněmi a zároveň zajistit, aby stejnosměrná složka byla v čase nulová. Posílaná data mají jeden řídicí režim a dva režimy pro posílání dat. Je to režim obrazový a datový, ve kterém se přenáší audio, gamut (dosažitelný rozsah barev), kontrolní data atd. Pro potřeby této práce není datový režim důležitý. Mezi jednotlivými režimy se vkládají tzv. guard gapy, což jsou mezery oddělující různé režimy. To je patrné na obr. 18, kde je zobrazena posloupnost jednotlivých režimů v jedné datové periodě.



**Obr. 18: Datová perioda HDMI přenosu. [20]**

### Řídicí režim

Řídicí režim má dvě základní funkce. První funkcí je určit, zda po řídicím bloku následují datová či obrazová data, k tomu slouží čtyři bity CTL0-CTL4. CTL0 a CTL1, které jsou přenášeny na datovém kanálu TDM1. Na TDM2 se přenášejí zbylé dva bity. Další funkcí je přenos synchronizačních impulzů Hsync a Vsync, které se přenášejí na TDM0. V tab. 4 jsou naznačeny kombinace určující režim následující po řídicím.

**Tab. 4: Kódování jednotlivých kombinací řídicích dat. [21]**

CTL0	CTL1	CTL2	CTL3	Následující režim
1	0	0	0	obrazový
1	0	1	0	datový

Kvůli potřebě přijímače bezchybně rozlišit jednotlivé hranice znaků na sériové lince, jsou řídicí data speciálně kódována. Při kódování se nevyužívá běžného TDMS standardu, tedy co nejméně přechodů, ale volí se tu mezi sedmi a osmi přechody logických úrovní. [21] V tab. 4 jsou uvedeny jednotlivé kombinace a jejich výsledný kód.

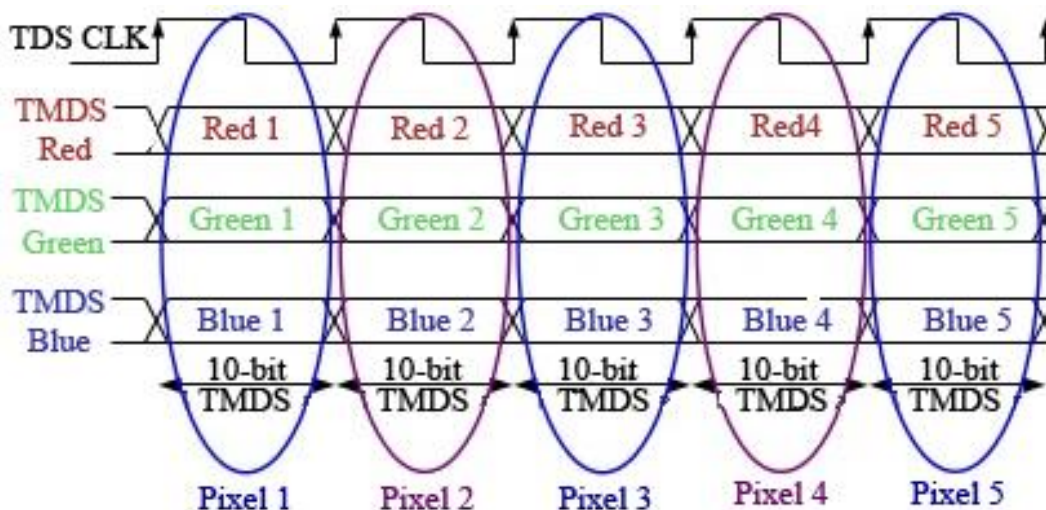
**Tab. 5:Kódování jednotlivých kombinací řídicích dat. [21]**

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

### Obrazový režim

V obrazovém režimu jsou samozřejmě posílána data potřebná pro obraz. Ta se kódují, podle již zmíněných pravidel. Nejmenší počet přechodů se dosahuje pomocí operace XOR, nebo XNOR mezi bity. O příslušné operaci vypovídá devátý bit. Je-li v log. jedničce, byla použita operace XOR. Určení, jaká bude použita operace závisí na počtu logických jedniček v osmibitovém slově. Je-li počet log. jedniček větší než čtyři, nebo roven čtyřem a zároveň nultý bit v nule, použije se operace XNOR. Operace je prováděna vždy mezi příslušným vstupním bitem a předešlým výstupním bitem. Jen nultý výstupní bit zůstává stejný, jako bit vstupní.

Druhé pravidlo, minimální stejnosměrná složka, je zajištěno bitovou negací. Podle počtu logických jedniček a disparity (počet jedniček a nul v předešlém signálu) se určí, zda bude negace použita či nikoli. O použití negace informuje poslední desátý bit. Logická jednička značí použití negace.



**Obr. 19: Odesílání jednotlivých pixelů v obrazovém režimu TDMS. [22]**

Pro přijímač HDMI signálu jsou důležité tyto dva poslední bity a podle jejich stavu se deseti bitový signál opět dekóduje zpět na osmibitový, který dává informaci o jedné složce barvy. Jak jde vidět na obr. 19, v jednom hodinovém taktu TDMS je nahrán jeden pixel. Z čehož vyplývá nutnost použít vzorkovací signál desetkrát vyšší než frekvence

TDMS. Obr. 19 zároveň znázorňuje fakt, že je odesílána dvaceti čtyř bitová RGB barva. Odesílání pixelů probíhá po řádcích zleva. To znamená, že se do matice nejprve odesílá nultý řádek a nultý sloupec a dále se postupuje po sloupcích. Mezi řádky je vždy odesílána datová perioda a samozřejmě řídicí signály.

### Identifikace přístroje

Pro identifikaci připojení přijímače k vysílači slouží pin detekce. K odesílání informací o přijímači slouží DDC kanál. Cílem tohoto kanálu je zajistit bezobslužné připojení přijímače k vysílači. Nejpoužívanější formát dat pro HDMI je EDID, který posílá dostačující informace o přijímači, a tak se může vysílač nastavit a odesílat správná data. Formát EDID má velikost sto dvacet osm bajtů. [22] Obsah EDID formátu viz obr. 20.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	00	FF	FF	FF	FF	FF	FF	00	05	D7	02	61	00	00	00	00
10	14	17	01	03	80	3D	00	78	06	EE	91	A3	54	4C	99	26
20	0F	50	54	FF	EF	80	81	00	81	40	81	80	81	C0	90	40
30	95	00	A9	40	B3	00	02	3A	80	18	71	38	2D	40	58	2C
40	45	00	F4	36	11	00	00	1E	28	3C	80	A0	70	B0	23	40
50	30	20	63	00	F4	36	11	00	00	1A	00	00	00	FD	00	17
60	78	0F	64	11	04	11	00	00	F8	38	00	01	00	00	00	FC
70	00	4D	49	44	52	41	20	53	65	72	69	65	73	0A	01	C6
80	02	03	2D	70	57	90	1F	20	21	22	05	14	04	13	3C	3D
90	3E	11	12	02	03	15	16	06	07	29	2F	01	23	09	1F	04
A0	6C	03	0C	00	00	00	00	21	C0	0D	0D	16	16	3B	3D	00
B0	A0	80	80	21	40	30	20	35	00	B1	64	00	00	00	1A	00
C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	37

POVINNÉ

VOLITELNÉ

Obr. 20: EDID formát. [22]

V dnešní době existuje celá řada programů, které jsou schopny na základně vstupních údajů vygenerovat EDID blok. Což značně ulehčí práci, jelikož není nutné kvůli vytvoření bloku, proniknout do hloubky tohoto formátu.

## 4.3 Vnitřní komunikace

Vnitřní komunikace v panelu probíhá mezi řídicími obvody LED matice, jako jsou posuvné registry a přepínače časového multiplexu a nadřazeným čipem. Další možností je komunikace mezi dvěma a více čipy na desce. Více čipů na desce se objevuje především ve velkých panelech. Takové systémy používají uspořádání, kde každý čip řídí část panelu. Vícečipová architektura se využívá i při aplikaci nových technologií. Což je příklad i této práce. O ovládání jednotlivých LED se stará FPGA, které časově zvládne urdit barevný panel, ale o specifické komunikace mezi nadřazeným prvkem, případně o správu fontů písma a databází se stará MCU část, kterou již má firma navrhnutou. Tímto způsobem lze samozřejmě urychlit vývoj. Uvedená koncepce vyžaduje řešení komunikace mezi jednotlivými čipy. Komunikace se dá rozdělit na sériovou a paralelní.

### 4.3.1 Paralelní komunikace

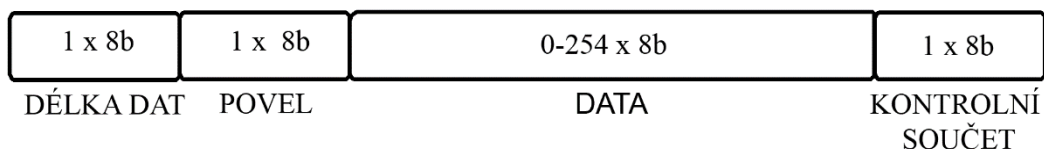
Při paralelní komunikaci jsou přenášena data po více vodičích zároveň. Díky čemuž se zvyšuje počet přenesených dat za jednotku času. Paralelní komunikace jsou například ISA, ATA, SCSI, PCI, FBS, GPIB a další. Nevýhodou samozřejmě je větší počet vodičů, a tedy i zabraných pinů. Z toho důvodu tato práce řeší komunikaci mezi čipy sériově.

### 4.3.2 USART

Universal Synchronous Asynchronous Receiver Transmitter představuje rozšíření klasické UART komunikace. Stále je zachována možnost komunikace asynchronní, jak bylo napsáno v odstavci 4.2.1. Navíc je možné komunikovat synchronně. Jeden komunikační kanál je nahrazen vodičem s hodinovým signálem. Po datovém vodiči jsou posílána data oběma směry (polo duplexní přenos). Pro komunikaci jsou používány různé protokoly. V této práci je použit MIKOMP.

#### MIKOMP

Protokol používá pro přenos osmi bitovou délku slova bez parity s jedním stop bitem. Protokol vždy začíná velikostí řetězce, následuje jednobitový povel a samotná data. Vše uzavírá osmi bitový kontrolní součet viz obr. 21. Kontrolní součet je výsledkem součtu dvojkových doplňků povelu a dat. Velikost řetězce udává délku dat s povelu. Podle druhu povelu a aplikace je někdy stejným způsobem vyžadována odpověď.



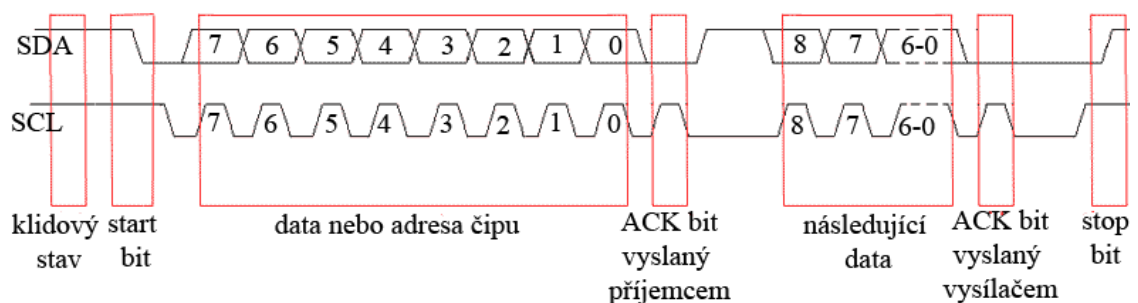
Obr. 21: MIKOMP protokol.

### 4.3.3 SPI

Rozhraní pro sériové periferie (SPI) má dva datové vodiče MOSI (Master Out, Slave In) a MISO (Master IN, Slave Out), jeden hodinový signál, který vysílá zařízení Master a jeden CS (Chip Select) signál, kterým master vybírá slave zařízení pro komunikaci. Pokud je třeba se zařízením komunikovat, přivede na jeho vstup CS log. nule. Díky CS může master komunikovat s různými zařízeními typu slave na společných vodičích MISO, MOSI a CLK. Čtení a odesílání dat může být prováděno na různých hranách hodinového signálu i při různých přechodech hodinového signálu.

### 4.3.4 I<sup>2</sup>C

Dvouvodičová synchronní sériová komunikace. Alternativní typ sběrnice má označení TWI, které není chráněnou značkou. Jedná se o polo duplexní sběrnici, která je synchronizována hodinovým signálem. Ten se spíná jen při komunikaci. Aby mohlo být k masteru připojeno více zařízení typu slave, má každé zařízení originální sedmibitovou nebo desetibitovou adresu. Na obr. 22 je znázorněn průběh komunikace. Nejprve se začíná start bitem (stav, kdy je hodinový signál v high a datový se mění na low). Dále je odeslána adresa od MSB a vysílač čeká na potvrzující ACK bit. Přejde-li nula, znamená to, že data byla přenesena správně a po přechodu SDA do H při SCL v L úrovni, může pokračovat odesílání dat. V případě, že přijímač odpoví logickou jedničkou, znamená to chybu komunikace a vysílač odešle stop bit. Stop bit je odeslán i v případě, že vysílač chce komunikaci ukončit. Změna dat probíhá při SCL=0 a v SCL=1 jsou data čtena.



Obr. 22: Časový průběh I2C komunikace. [23]

## 5 USPOŘÁDÁNÍ ŘÍDÍCÍ ČÁSTI

Jak již bylo zmíněno, pro vývoj řízení LED panelu je v určitých situacích lepší použití FPGA pouze pro časově náročné funkce a zbývající část, která obstarává např. specifickou komunikaci s nadřazeným prostředkem, je výhodnější realizovat pomocí MCU.

### 5.1 MCU

Jednočipový mikrokontrolér obsahující celou řadu obvodů. Má čtyři hlavní části:

- **CPU** – centrální procesorová jednotka systému,
- **ROM** – paměť pro software
- **RAM** – paměť pro proměnné a zásobník,
- **I/O piny a periferie** – zajišťují styk s okolními systémy.

systém, watchdogy, převodníky, komunikační rozhraní (UART, SPI, I2C, CAN, USB atd.). [24] MCU zpracovává data v čase synchronně. Tzn. v jeden časový okamžik je výpočetní paměť věnována jedné úloze. Jakmile úloha skončí, paměť je uvolněna pro další úlohu čekající v řadě. Což je výhodnější pro větší počet malých úkonů, které není nutné provádět současně.

MCU bude proto použit, jako prostředník mezi nadřazeným zařízením a FPGA. Jeho funkcionality se tedy soustřeďuje na příjem instrukcí a obrazových dat od nadřazeného prostředku, jejich zpracování a odesílání sestavených obrazců do FPGA. Další funkcí MCU budou výpočty a příjem dat z měřicích obvodů, například měření jasu, teploty a napájení a odesílání dat potřebných pro úpravu jasu do FPGA.

### 5.2 FPGA

Čip s programovatelnými hradlovými poli je velice flexibilní nástroj pro řízení časově náročných obvodů. Jeho největší výhodou je paralelní zpracování jednotlivých úkonů. Maximální pracovní frekvence je omezena fyzickou strukturou na čipu a dosahuje stovek MHz až jednotek GHz. Reálná frekvence je však omezena vlastním návrhem. Omezení frekvence určuje nejdelší kombinační cesta obvodu. Existují však návrhářské metody, kterými jde pracovní frekvence zvýšit. Samozřejmě obvykle na úkor vyšší spotřeby a většího počtu buněk. I když se jedná o FPGA, jako nástroj vhodný pro každou aplikaci, musí se návrhář zamyslet, zda nepoužít třeba levnější MCU, a to hlavně v řešeních, kde jsou data zpracovávána sériově, a ještě v řádech jednotek a desítek MHz. Další výhody a nevýhody FPGA vůči MCU znázorňuje tab. 6.

**Tab. 6: Výhody a nevýhody FPGA. [25]**

Výhody	Nevýhody
100 % uživatelský návrh řešení	Znalost speciálních nástrojů
Vysoký výpočetní výkon	Časová náročnost návrhu řešení
Paralelní zpracování	Zdlouhavá verifikace
Obvykle nižší spotřeba	Drahé vývojové nástroje
Obvykle nižší cena řešení	Vyšší cena vývoje
Kompaktní řešení	Absence floatových jednotek
Množství pinů	Nižší výkon soft jader
Rychlé interní paměti	Nižší výkon hard jader než u klasických řešení
Podpora DSP operací	Složitější řešení napájení
Clock management	
SoftCore procesory	
Integrovaná hard jádra procesorů	

FPGA bude mít v systému na starosti tři hlavní úkoly. Jejich implementací se zabývá praktická část práce.

### 5.2.1 Komunikace s CPU

Příjem obrazových informací, který bude probíhat periodicky po sériovém synchronním rozhraní typu SPI. Šestnáctibitové slovo vyhovuje pro odesílání hodnot barevných složek jednotlivých pixelů, které mají rovněž stejnou bitovou délku. Přenosová rychlost bude přizpůsobena možnostem CPU a požadavkům na periodu odesílání.

Příjem informace o jasu bude sériovou linku typu UART komunikačním protokolem MIKOMP. Informace o jasu bude odesílána s určitou periodou.

### 5.2.2 Obrazová paměť

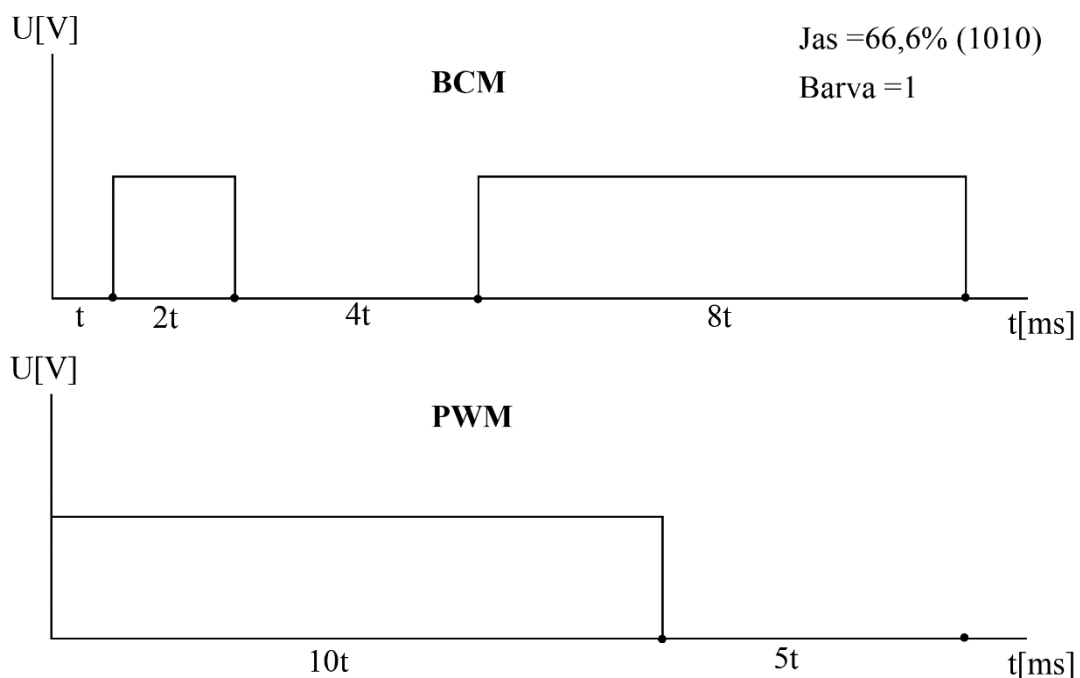
Obrazová paměť slouží pro uložení stávajícího a následujícího obrazce a jejich čtení. Pro obrazce o určité velikosti je využívána hlavně vnitřní RAM paměť FPGA. V případě velkých rozměrů LED panelu nebo velké barevné hloubky, kde je přesáhnuta kapacita interní paměti, musí být používány rychlé externí paměťové prvky (komunikace v řádu desítek MHz) typu RAM. Opět je nutno řešit ekonomiku celé věci. Tedy zda koupit čip s větší RAM pamětí za více peněz, nebo na paměť na čipu nehledět a koupit raději externí paměť. Na což se nabalují další náklady, jako je montáž paměti, zvětšení plochy desky, nutnost řešit složitější komunikaci a synchronizaci dat, větší spotřeba i snížená odolnost proti EMC. Všechny parametry je nutné dát na pomyslnou misku vah a vybrat vhodnou variantu.

Jak již bylo zmíněno, je potřeba vytvořit dva bloky RAM paměti o stejné velikosti. Z jedné paměti se čte aktuální obrazec a do druhé je zapisován nový snímek. Jakmile skončí načítání obrazce, zajistí časový multiplex záměnu funkce obou pamětí. Pro RGB LED panely od firmy Adafruit industries, nebo jim podobné, které jsou rozděleny na dva bloky se hodí využít dvouportovou RAM paměť, pokud je na daném čipu obsažena.

### 5.2.3 Generování dat pro displej

Pro správné generování dat na displej je nutné velkou řadu obvodů. Prvně to jsou čítače řádků a sloupců, ze kterých se získává adresa pro vyčítání dat z RAM. Dále je to obvod pro řízení barevné hloubky a obvod pro ovládání jasů. Pokud by se řešila jen jedna složka, tedy buď barevná hloubka nebo jen jas, je lepší pro řízení použít pulzně šířkovou modulaci. Což je příklad monochromatického panelu bez odstínů barvy.

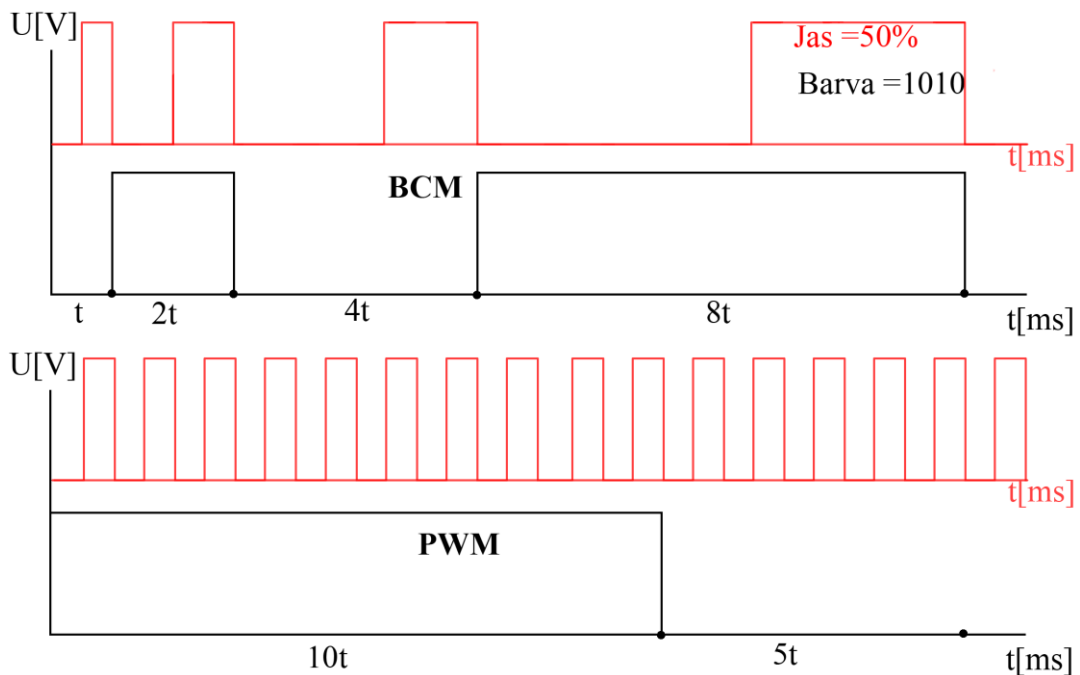
U monochromatického panelu bez odstínu barvy je řízen jen jas. Jak jde vidět na obr. 23, tak v případě PWM vznikne jen jeden napěťový pulz pro celý jas. V případě BCM můžou vzniknout jeden nebo dva pulzy různé šířky. Samozřejmě platí pravidlo, že čím užší pulz, tím větší vyzařování rušení a zahřívání diod, resp. celého systému. diody více zahřívají. Proto je lepší u monochromatických panelů bez odstínu barvy používat pulzně šířkovou modulaci.



**Obr. 23: Řízení jasů pomocí BCM a PWM jednobarevného panelu bez odstínů.**

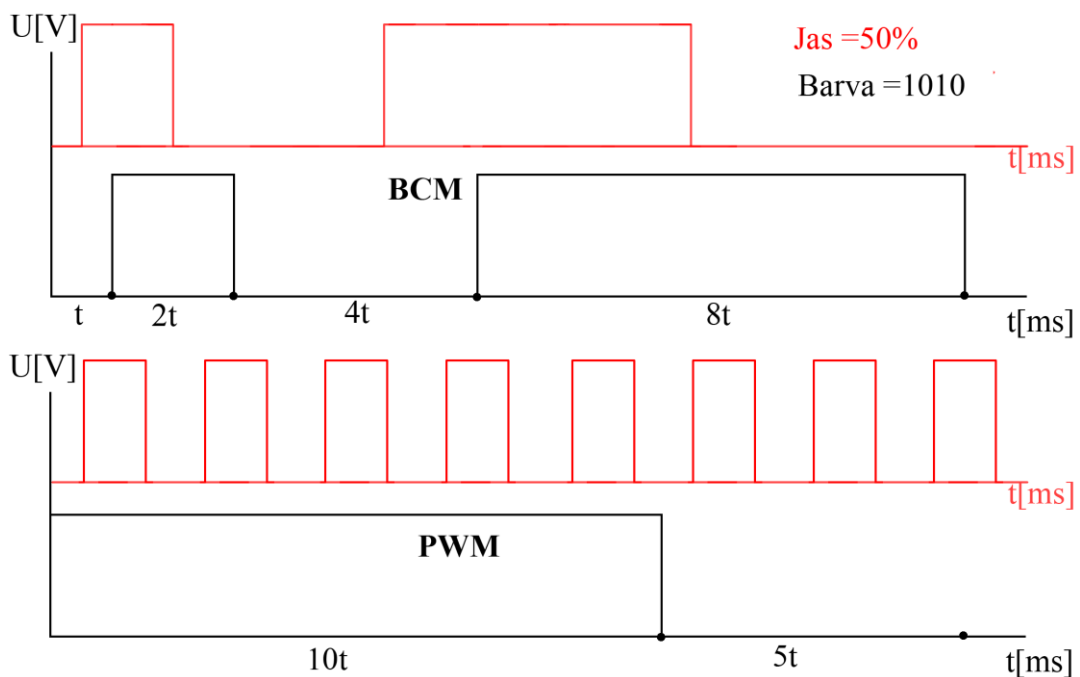
Jinak tomu však je, pokud do hry vstoupí ještě další veličina – barevná hloubka. V praxi se obvykle řeší nejdřív barevná hloubka, tedy zda bude spínána pomocí BCM nebo PWM a následně jas, který je řešen závisle na barevné hloubce samotným vývodem, OUTPUT ENABLE. OE bývá negovaný, což znamená, že při log. jedničce je obvod vypnut a LED nesvítí. Na obr. 24 je kombinace barevné hloubky a jasů. Jak je vidět, tak při PWM řízení jasů musí udělat stejný počet pulzů, jako je počet odstínů barev jedné barevné složky (pro čtyř bitovou barvu patnáct pulzů). Zato při BCM je potřeba udělat

jen stejný počet, jako je barevná hloubka (pouze čtyři pulzy). Což je z hlediska EMC a ztrát výhodnější.



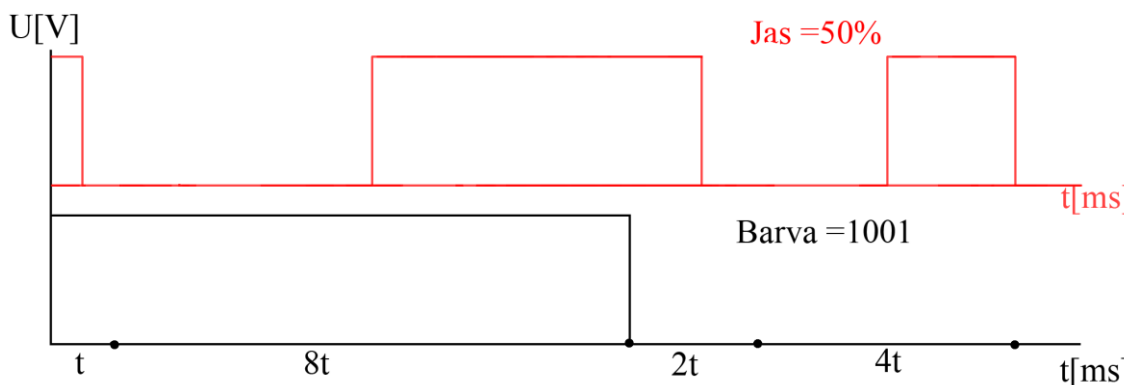
**Obr. 24: Řízení intenzity barvy a jasu pomocí BCM a PWM.**

Optimalizací obou modulací lze zmenšit počet pulzů na polovinu a zvýšit jejich šířky. Jedná se o přemísťující jas. Kdy jednou je OE na počátku úseku barvy a další úsek na konci viz obr. 25. Je však milné se domnívat, že na hardwarovém výstupu budou všechny napěťové pulzy dvojnásobné. Při změně logických hodnot barevné hloubky může nejmenší napěťový pulz být opět o velikosti jasové složky nejmenšího časového úseku, tedy poloviční pulzu ( $1t$ ). Na počet změn logických hodnot barevné intenzity je daleko lepší PWM, jelikož u ní je pro jednu složku barvy maximálně jedna změna logické hodnoty. U BCM jich může stejně jako počet bitů barevné hloubky. Tento fakt však nestačí na to, aby PWM byla výhodnější než BCM. Jelikož stále má mnohem více pulzů než BCM. Dalším faktem je, že u PWM nastane přesný počet polovičních pulzů, a to je počet složek barev (tři pro RGB) vynásobený počtem LED diod. U BCM obvykle polovičních pulzů nastane víc, však nejmenších polovičních pulzů, pro úsek  $1t$  nastane maximálně stejně jako u PWM. Ale obvykle jich bývá méně.



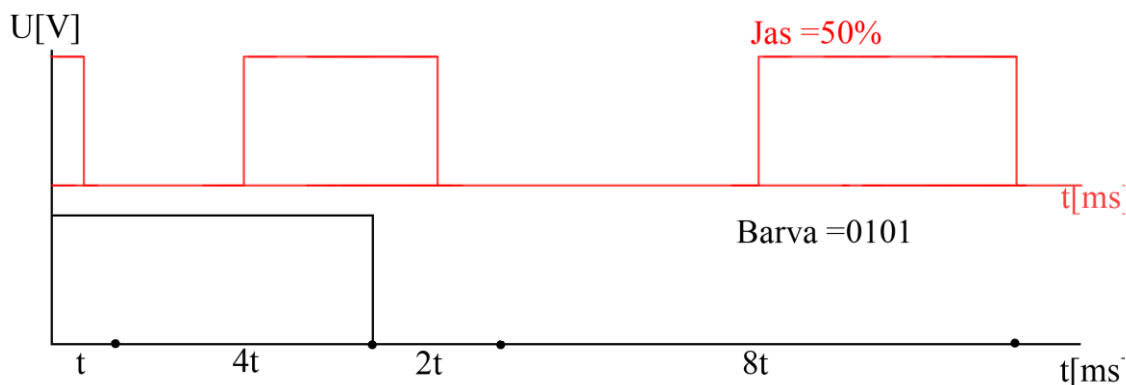
**Obr. 25: Řízení intenzity barvy a posuvného jasu pomocí BCM a PWM.**

Samozejmě, že tato optimalizace může pomoci k lepší EMC a menším tepelným ztrátám, avšak s optimalizacemi lze dále pokračovat přesouváním samotných časových BCM úseků do nejvýhodnějších dvojic tak, aby byl napěťový pulz co nejširší. Tedy k napěťovému pulzu o časové základně  $1t$  přiřadit největší napěťový pulz, v tomto případě  $8t$ . Jediným úskalím je správné umístění jasových pulzů, které musí být po stranách vybraných dvojic. Tedy vznikne v případě  $8t=1$  a  $1t=1$  napěťový pulz o velikosti  $9t$  zmenšený o devíti násobek jasové složky. Na obr. 26 jde vidět případ, kdy výsledný pulz je o velikosti  $9t$  zmenšený o devítinásobek jasové složky.



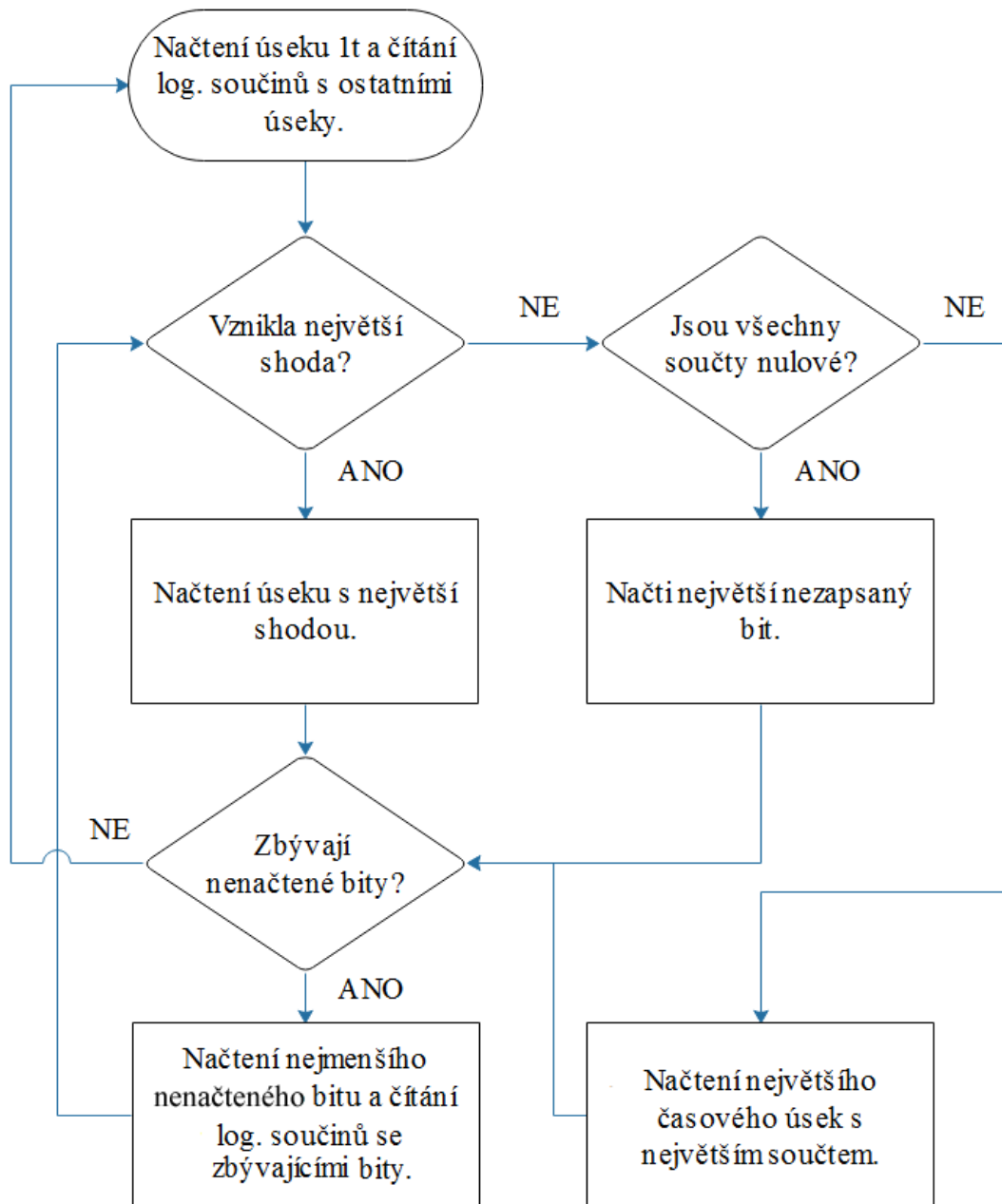
**Obr. 26: Posuvná BCM s posuvným jase.**

Z obrázku jde vidět, že pro zobrazení barvy 1001 o polovičním jasu je potřeba jediný pulz o velikosti  $4,5t$ . Pro srovnání při klasické BCM by vznikly dva pulzy. Jeden o velikosti  $0,5t$  a druhý o velikosti  $4t$ . A v klasické PWM by vzniklo devět pulzů se šířkou  $0,5t$ . Úskalí posuvné BCM spočívá v různorodosti barev. Pokud např. bude potřeba zobrazit barvu 0101 posuvná BCM nevytvoří žádný efekt vůči klasické binárně kódové modulaci. Nejvýhodnější by bylo za časový úsek  $1t$  přiřadit úsek  $4t$  viz obr. 27.



**Obr. 27: Posuvná BCM s posuvným jasem.**

Přesně tahle úvaha vede k dynamické binární kódové modulaci. V dynamické BCM jde o hledání nejvýhodnějších dvojic úseků. Lépe řečeno je vybírán úsek s největšími shodami logických jedniček v jednotlivých bitech barvy. Příkladem může být spínání dvou RGB LED diod s hodnotami jednotlivých složek barvy ( $R1=1001$ ,  $G1=0101$ ,  $B1=1110$ ,  $R2=0111$ ,  $G2=1000$  a  $B2=0000$ ). Pro tyto složky je zobrazen nejmenší, nultý, bit (časový úsek  $1t$ ) a hledá se k němu nejvýhodnější časový úsek. Složky barvy s  $LSB=0$  nehrají v hledání žádnou roli. Tedy zůstaly složky 1001, 0101 a 0111. Jasně jde vidět, že nejvýhodnější pro spárování je druhý bit (úsek  $4t$ ). Pro šestnácti bitovou barevnou hloubku, kdy jsou v konečném důsledku hodnoty všech složek šesti bitové, je samozřejmě výsledná rozhodovací logika složitější, viz vývojový diagram na obr. 28.



**Obr. 28: Vývojový diagram dynamického řízení BCM.**

Pro správné vnímání barev na panelu musí být, před dynamickou BCM, barevné složky upraveny pomocí převodní tabulky gama korekce. Dále se musí pro zachycení odeslaných dat, v posuvných registrech, do výstupních registrů správně řídit signál LATCH.

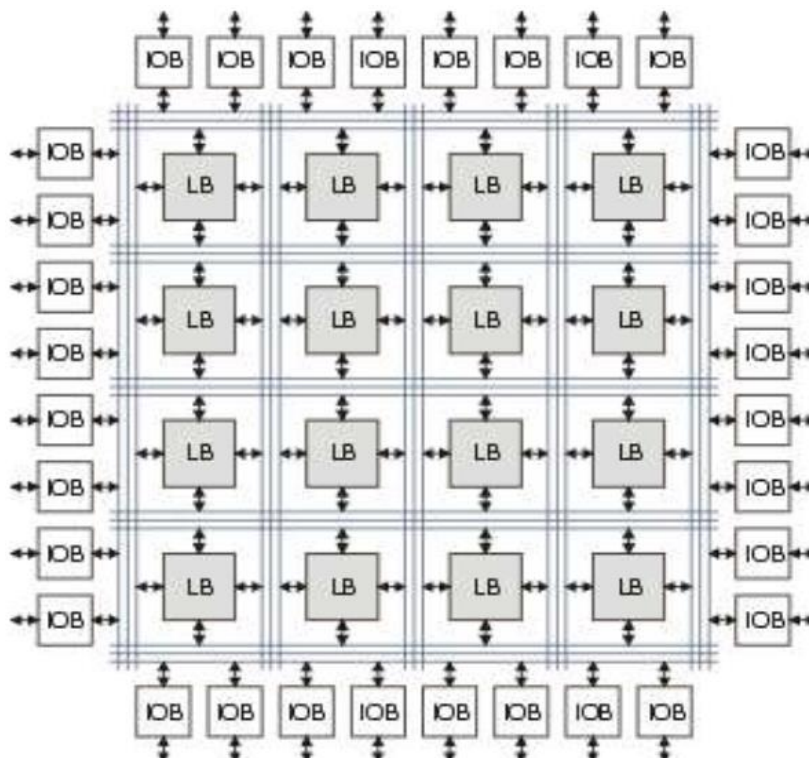
## 6 FPGA

### 6.1 Vývoj a vlastnosti

První programovatelná hradlová pole byla vyrobena v polovině osmdesátých let minulého století firmou Signetics Corporation, pod názvem FPLA, další vývoj na sebe nenechal dlouho čekat, a tak byly do obvodů přidány sekvenční prvky a zařízení dostalo název PLA. Dále firma Lattice Semiconductor vyrábí mazatelné obvody GAL. Velkým milníkem je vytvoření konfigurovatelné tabulky LUT (Look Up Table), která nahrazuje klasickou síť hradel AND a OR. Ta je využívána i v moderních FPGA a CPLD obvodech. Popis obvodů probíhá pomocí jazyků Verilog a VHDL, které se používají vedle SystemVerilogu a C++ k behaviorálnímu popisu obvodu. [26]

#### 6.1.1 Základní architektura FPGA

Základní myšlenka architektury FPGA u výrobců Altera nebo Xilinx je stejná. Vstupně výstupní buňky na pinech (IOB), jsou sítí cest napojeny do matice logických buněk (CBL u Xilinx a LBA pro Alteru). Mezi buňkami opět existuje síť propojovacích cest a uzlů viz obr. 29.

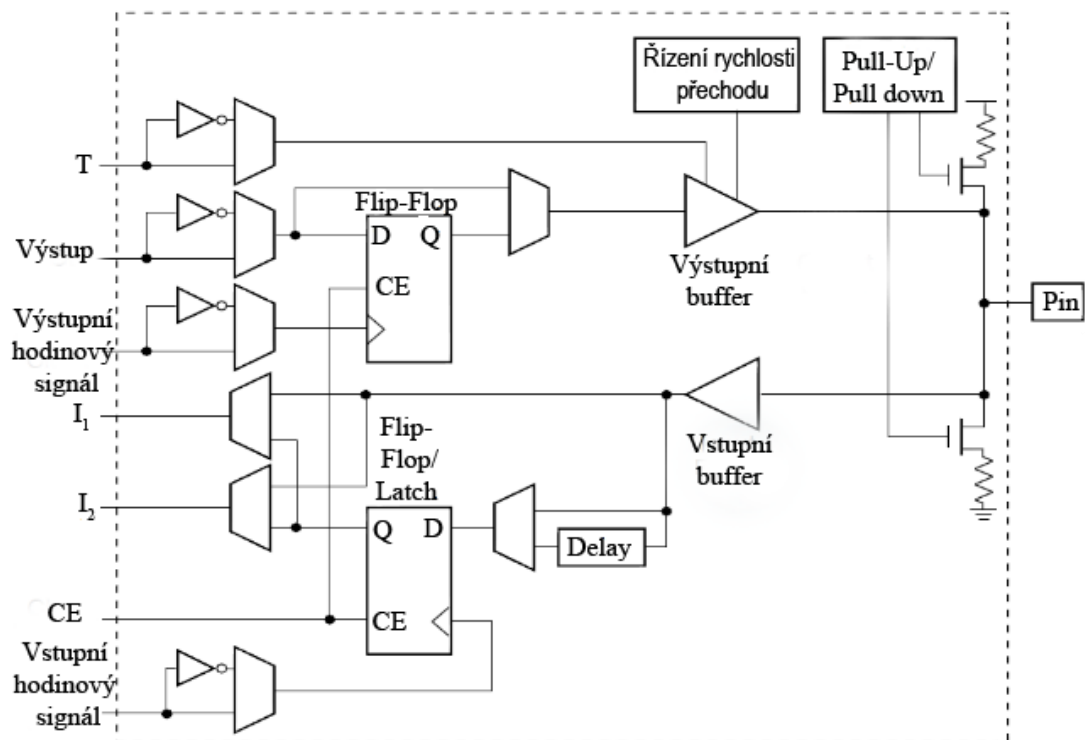


Obr. 29: Základní konstrukce FPGA. [26]

## IO buňka

Na vstupně výstupní buňku, tedy rozhraní mezi čipem a okolními obvody jsou kladeny vysoké požadavky [27]:

- podpora více logických standardů,
- podpora diferenčních vstupů,
- podpora pro synchronizaci paralelních sběrnic,
- podpora rychlého přenosu dat,
- integrované zakončovací rezistory,
- ESD ochrana, pull-up/down rezistory, ...



Obr. 30: Vstupně výstupní buňka. [27]

Tyto požadavky však nemusí splňovat všechny piny. Proto jsou v FPGA vytvořené pinové banky, tedy piny se stejnými vlastnostmi. Můžeme tedy najít speciální piny pro hodinový signál, pro rychlé přenosy, diferenční vstupy atd. Na obr jde vidět základní struktura IOB. Jsou na ní vstupní/ výstupní zesilovače (OUTBUF, INBUF), třístavový výstup, programovatelný vstupní invertor, podpora diferenčních párů, vstupní/ výstupní klopný obvod, pull-up/ pull-down rezistor a zpožďovací člen. [28]

## Logická buňka

Logická buňka se dělí na kombinační a sekvenční část. Kombinační část je založena na LUT (Look-Up-Table). LUT – náhledová tabulka, která je schopná ze vstupů vytvořit jakoukoli kombinační funkci. Počet vstupů závisí na jednotlivých obvodech např. šest vstupů má FPGA Spartan-6 a čtyři vstupy Spartan-3. Kromě kombinačního využití lze LUT využít jako distribuovaná paměť RAM/ROM nebo posuvný registr.

Sekvenční část buňky tvoří klopné obvody typu D. Kromě datového vstupního vodiče ještě mají vstupy pro hodinový signál (CK), povolení hodin(CE), synchronní a asynchronní reset (SRST/ARST). Lze také nastavit různé parametry, jako je třeba typ hrany překlápění (HIGH, LOW).

## 6.1.2 Další buňky FPGA

V FPGA samozřejmě nejsou jen výše zmíněné základní buňky. Nedílnou součástí je bloková paměť RAM. Výhodou je rychlost a synchronizovanost s jádrem. Tyto paměti lze různě štěpit na menší bloky o různém počtu a velikosti ukládaných datových slov – bajtů. Poměrně novou buňkou v obvodech FPGA je paměť FLASH, jejíž hlavní úkol je konfigurace čipu. Případně se dá její část využít uživatelem, jako klasická paměť. FPGA obsahuje také další obvody nazývané IP core:

- komunikační rozhraní,
- mikroprocesory,
- DSP buňky pro matematické operace,
- A/D převodníky,
- bloky pro úpravu hodinového signálu,
- ...

## 6.1.3 Výběr FPGA:

Výběr probíhal u produktů dvou předních výrobců FPGA – Intel (Altera) a Xilinx. Hlavními požadavky byly:

- pouzdro s bočními vývody (QFP, TQFP),
- RAM paměť pro panel 256 x 32 (tedy  $16 \times 2 \times 256 \times 32 = 262\,144$  bitů),
- IP core pro management hodin,
- maximální frekvence 50 MHz a vyšší,
- provozní napájecí napětí 3,3 V,
- minimálně 20 I/O pinů.

**Tab. 7: Srovnání FPGA prvků s bočními vývody. [29] [30]**

FPGA	Výrobce	Počet makrobuněk	RAM paměť [Kbit]	IP core hodin	Max. frekvence [MHz]	Počet I/O pinů	*Cena [Kč]
Spartan6 XC6SLX4	Xilinx	3840	570	DCL, PLL	375	102	291
Spartan6 XC6SLX9	Xilinx	9151	589	DCM, PLL	375	102	364
MAX 10 10M08SAE	Altera	8000	387	PLL	450	101	373
Cyclone EP1C3T	Altera	2910	599	PLL	275	56	593

\*Cena je uvedena bez DPH.

Z těchto čtyř čipů byl vybrán MAX 10 od Altery. Přestože všechny čtyři obvody měly dostačující parametry a výrobce Xilinx má i nižší cenu, tak velká výhoda MAX 10 je v jeho konfigurační paměti integrované přímo na čipu. U ostatních je nezbytná investice do konfigurační paměti FLASH mimo čip, což zvyšuje náklady a plochu zastavěného prostoru na DPS. Neoddiskutovatelná je také vyšší pravděpodobnost chyby způsobená dalším prvkem v obvodu. Pro vývoj aplikace a její otestování byla zakoupena deska 10M08 Evaluation kit.

### 6.1.4 Vývojová deska 10M08

Deska je osazena již zmíněným FPGA obvodem 10M08SAE144C8GES. Dále obsahuje:

- JTAG konektor,
- napájecí mini USB konektor
- 50 MHz oscilátor,
- 40 I/O pinů,
- analogových pinů,
- 14 Arduino pinů,
- 8 analogových pinů,
- 6 LED,
- 2 tlačítka – rekonfigurační a resetovací,
- 6 přepínačů.

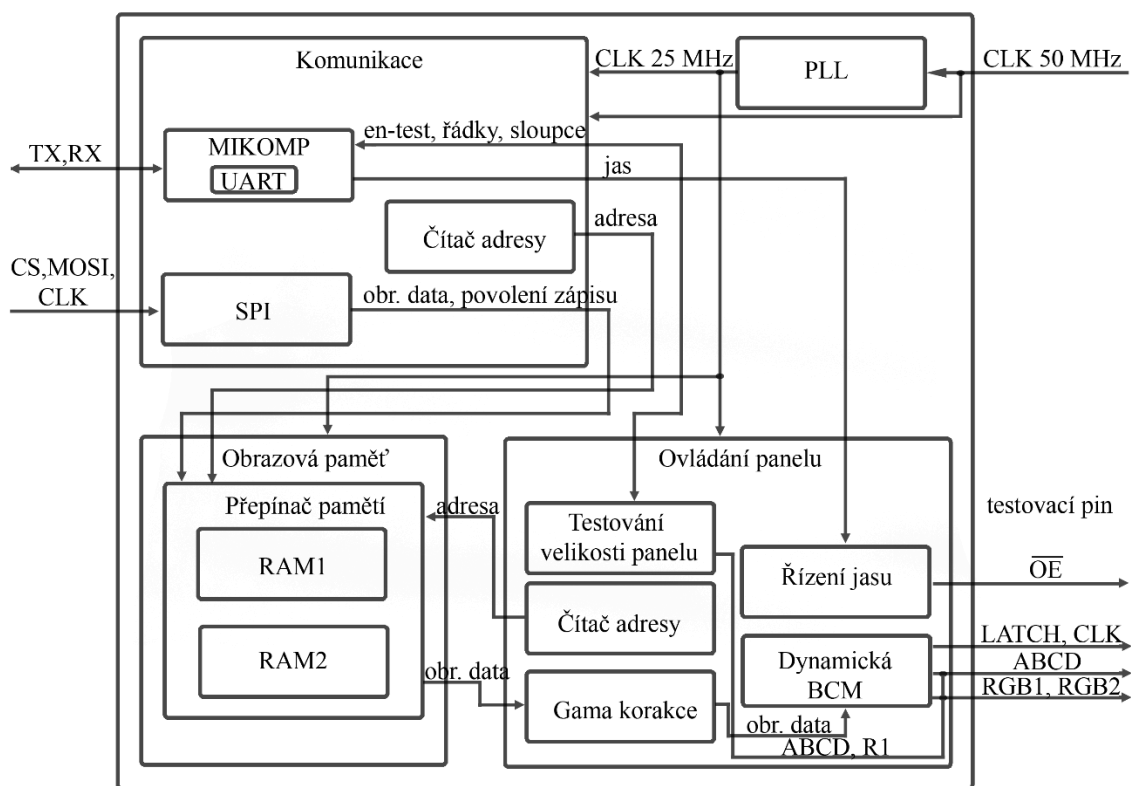


Obr. 31: 10M08 Evaluation kit. [31]

## 7 REALIZACE PROJEKTU

Pro realizaci projektu poslouží zkušební RGB panel o velikosti 255 řádků a 32 sloupců. Panel má stejnou topologii, jako již zmíněný panel od firmy Adafruit industries viz kapitola obr. 32. Jen má trojnásobný počet driverů (pro každou barvu zvlášť). O odesílání dat a komunikaci s FPGA se stará STM 32 nucleo. Vnitřní struktura FPGA viz obr. 32 se skládá ze čtyř bloků:

1. PLL,
2. komunikace,
3. obrazová paměť,
4. ovládání panelu.



Obr. 32: Blokové schéma řídicího obvodu

### 7.1 PLL

Pro dělení kmitočtu dvěma je použit fázový závěs. Do čipu je z vývojové desky posílána frekvence padesát megahertz a pro řízení panelu je potřeba dvacet pět megahertz.

## 7.2 Komunikace

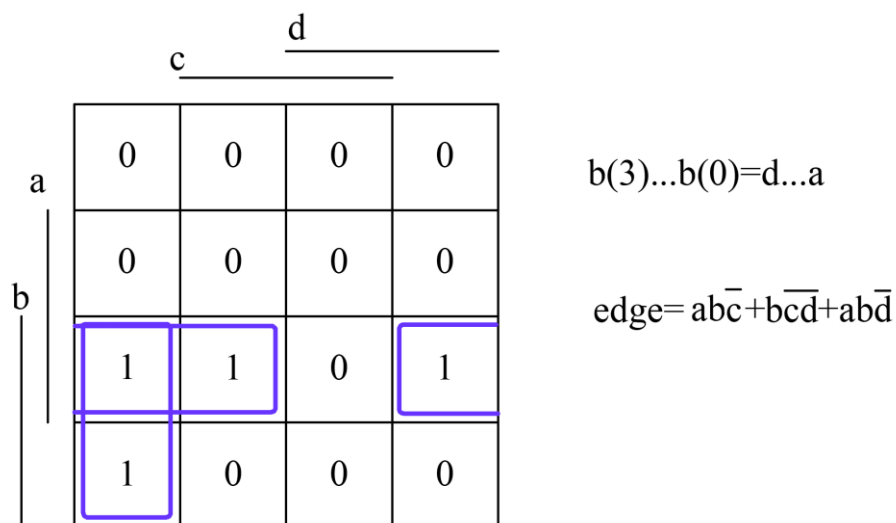
Komunikace mezi FPGA a MCU je obstarávána dvěma na sobě nezávislými komunikačními proudy. Po rozhraní SPI bude FPGA přijímat obrazová data a UART bude využíván pro oboustrannou komunikaci, která přenáší údaje zajišťující řízení jasu a identifikaci velikosti panelu. Kromě těchto dvou komponent je pro potřeby zápisu ještě vytvořen čítač adresy. Jedná se o klasický čítač s povolením čítání při příjmu celého šestnácti bitového slova přes SPI. Čítač se resetuje na konci příjmu jednoho obrazce.

### 7.2.1 SPI

SPI komunikace je navržena, jako jednosměrná. Linka MISO je trvale připojeno na zem. Po MOSI vodiči jsou odesílány jednotlivé obrazové bajty. Komunikace probíhá klasicky, kdy po sepnutí CS do log. nuly je při nástupné hraně hodinového signálu dvanáct megahertz čten jeden bit dat od MSB po LSB. Obrazová data jsou odesílána od vrchního levého rohu směrem doprava a postupně po řádcích dolů. To znamená, že první bod obrazu bude odeslán pro nultý řádek a nultý sloupec. Druhý bod pro nultý řádek a první sloupec. Celý obrazec, dvě stě padesát šest řádků krát třicet dva sloupců bajtů, je odeslán v jedné zprávě.

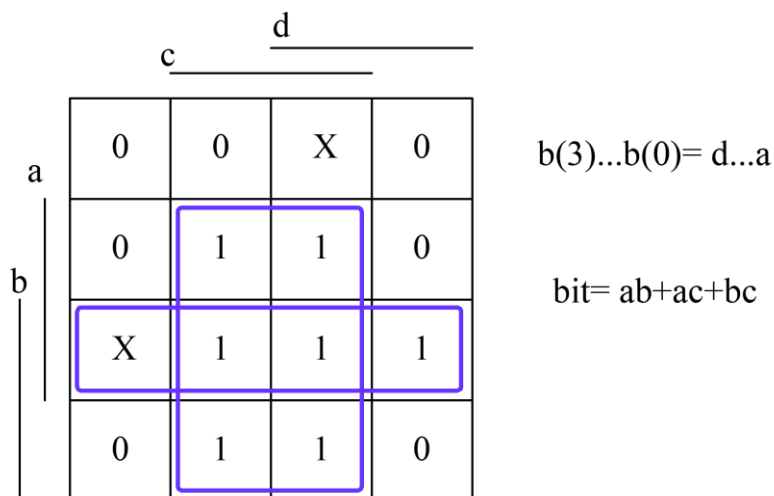
Samotná implementace příjmu dat je vcelku jednoduchá. Problémy v komunikaci by mohl dělat nesynchronní příjem dat s hodinovým signálem FPGA. V případě změny dat přesně ve chvíli vzorkování signálu, by mohl mít za následek metastabilitu a špatnou interpretaci čtených dat. To by mohlo být kritické hlavně při metastabilitě hodinového a datového signálu. Proto na příjem obou dvou signálů jsou vloženy dva za sebou jdoucí klopné obvody. Za klopnými obvody jsou umístěny čtyřbitové posuvné registry na vzorkování signálu frekvencí padesát megahertz. Z pohledu hardwaru jde tedy o šestibitový posuvný registr s vyvedenými čtyřmi horními bity. Díky čtyřnásobné vzorkovací frekvenci můžeme jeden bit navzorkovat čtyřikrát, a tak eliminovat chybu vzniklou rušením.

Zjištění nástupné hrany hodinového signálu poté probíhá jednoduše vyhodnocením všech čtyř výstupů posuvného registru. V ideálním případě je v posuvném registru hodnota 0011, ale může se stát, že rušením vznikne chyba. Vznikne-li jedna, nebo dvě okrajové chyby, kombinační výstupní logika je odhalí a vyhodnotí nástupnou hranu správně. Výsledná kombinační funkce se určí podle Karnaughovy mapy na obr. 33.



**Obr. 33:**Karnaughova mapa a výsledná minimalizovaná funkce pro datový signál CLK.

Vyhodnocení příchozích dat z MOSI opět eliminuje vzniklou jednu chybu nebo dvě okrajové chyby. Logika prioritně vyhodnocuje vnitřní bity (1. a 2. bit), jelikož je vyšší pravděpodobnost ustáleného signálu blíž středu čteného bitu. Opět Karnaughovou mapu a výslednou funkci lze vidět na obr. 34.



**Obr. 34:**Karnaughova mapa a výsledná minimalizovaná funkce pro datový signál MOSI.

Správná data se následně zapisují do šestnácti bitové paměti. Zároveň se zápisem také čítač čítá do šestnácti. Jakmile načítá, odesílá data do nadřazeného bloku.

## 7.2.2 MIKOMP

Pro přenos zpráv po sériovém rozhraní UART se používá protokol MICOMP (název je zkratka slov mikroprocesorová komunikace). Protokol MICOMP určuje komunikační rámec a uspořádání zpráv umístěných na rozhraní UART.

### UART

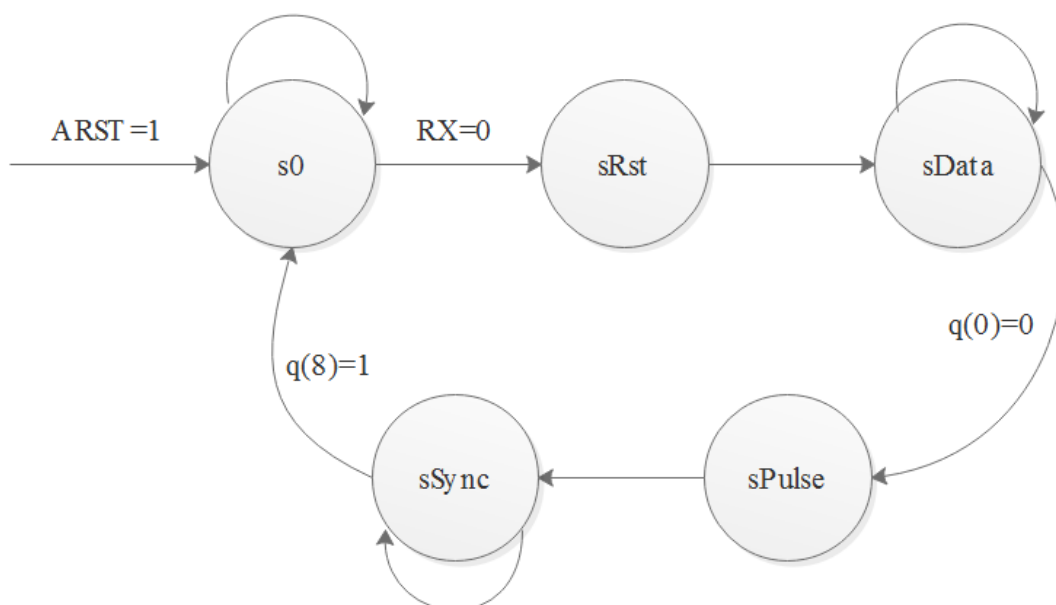
Pro komunikaci přes UART je nutné znát rychlost přenosu. Přenosovou rychlost může mít Slave pevně implementovanou v sobě. Je možná také varianta, že přenosovou rychlost zná jen Master a na počátku komunikace odešle synchronizační bajt, ze kterého může Slave určit přenosovou rychlost. Pro potřeby této práce však je přenosová rychlost stanovena 9600 Baud. Jelikož je komunikace velmi pomalá, není nutné používat obvody pro odstranění metastability. Vznikne-li metastabilita při přechodu do start bitu, bude v nejhorším případě impulz z děličky kmitočtu o jeden takt hodin opožděný, což nemá žádný vliv na správné čtení dat.

Pro implementaci je nutná dělička kmitočtu podle vzorce:

$$\text{Dělicí konstanta} = \frac{f_{\text{pracovní}}}{f_{\text{přenosová}}} \quad (4)$$

Pro pracovní frekvenci dvacet pět megahertz a přenosovou frekvenci devět tisíc šest set hertz je zaokrouhlená dělicí konstanta 2604.

Dělička frekvence, resp. čítač náběžných hran musí být navrhnutý tak, že při resetu se jeho hodnota nastaví do poloviny pracovního kmitočtu. Díky tomu při příchodu synchronizačního pulzu je čítač resetován a načítá tak polovinu pulzu. Díky tomu jsou následující data čtena v polovině své délky. Kromě čítače ještě příjem dat obsahuje devíti bitový posuvný registr plněný z vrchu a resetovaný stejně jako čítač. Jen při resetu jsou do registrů vloženy samé jedničky. Díky tomu a devátému bitu je možné eliminovat čítač příchozích bitů, případně zmenšit počet stavů ve stavovém automatu. Posledně zmíněný stavový automat typu Moore řídí celý příjem dat a vypadá následovně.



**Obr. 35: Stavový automat pro řízení příjmu dat po UART.**

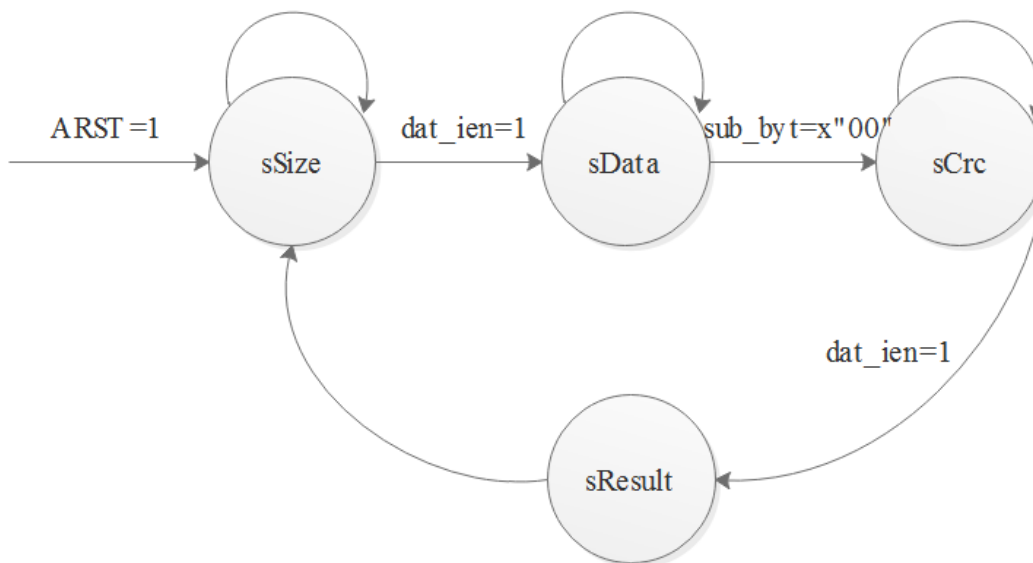
**Tab. 8: Popis jednotlivých stavů stavového automatu pro UART.**

Stav	Popis stavu
s0	Stav, který je načten po asynchronním resetu. Dále se čeká na start bit, aby mohl přejít do dalšího stavu.
sRst	Jednotaktový stav, který má za následek poslat resetovací pulz do děličky kmitočtu a posuvného registru.
sData	Tento stav čeká až se načte všech 8 bitů. Při posunutí start bitu na nulovou pozici registru se stavový automat překlápí dále.
sPulse	Opět jednotaktový pulz, při němž se odešle informace MIKOMPu, že je všech 8 bitů v posuvném registru.
sSync	Stav čekající na stop bit. Případně má-li MSB hodnotu 1, přechází do s0 bez čekání na stop bit. Což nemá na funkci UART žádný vliv.

Odesílání dat z UART bloku je snazší. K němu je potřeba jen dělička kmitočtu, čítač odeslaných bitů a posuvný deseti bitový posuvný registr. Do něhož se při potřebě odeslání nahrají na první 2 bity 01 a dál data. Postupně je vše při pulzu z děličky odesíláno. Čítač odeslaných bitů se stará o to, aby MIKOMP věděl, že všechny bity jsou odeslány. Pro samotný protokol MICOMP je v programu hlavní komponenta mikomp a poté nastavbová komponenta mikomp\_top. Mikomp zajišťuje správný příjem rámců, tedy velikost zprávy, kód povelu, data a kontrolní součet. Mikomp\_top má na starosti povel, které jsou definovány pro komunikaci, zpracování těchto povelů a odesílání odpovědi. Pro účely DP je dostačující velikost dat tři bajty.

Komponenta mikomp je opět rozdělena na 2 části. První část řeší příjem a druhá odesílání dat. Pro příjem dat je potřeba funkce odečítání přijatých bajtů a sčítání kontrolního součtu. Ta se dá udělat buď pomocí binárního sčítání dvojkového doplňku, nebo sčítáním všech přijatých bajtů. Tedy jak velikosti zprávy, tak kontrolního součtu.

Výsledek musí dát na posledních 8 bitech hodnotu 0. Celý příjem je opět řízen stavovým automatem typu Mealy viz obr. 36.

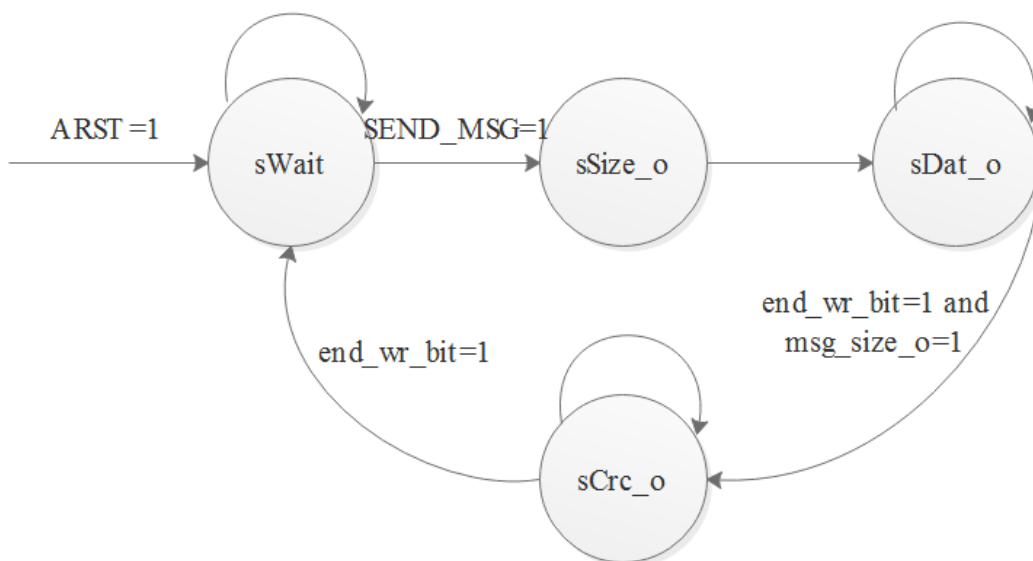


**Obr. 36: Stavový automat pro řízení příjmu dat komponenty mikomp.**

**Tab. 9: Popis jednotlivých stavů pro řízení příjmu dat komponenty mikomp.**

Stav	Popis stavu
sSize	Stav, který je načten po asynchronním resetu. Při příjmu prvního bajtu dat, velikosti zprávy, nastaví velikost do odčítačky.
sData	V tomto stavu jsou posílána data do nadřazeného prvku, je povolena odčítačka dat a sčítačka kontrolního součtu. Vše se děje při příjmu nových dat.
sCrc	Čeká na poslední bajt-kontrolní součet.
sResult	Jednotaktový stav, který řeší správný příjem dat, tedy odesílá informaci, zda byl kontrolní součet na konci nulový, nebo nebyl. Zároveň je mazána paměť kontrolního součtu.

Odesílání dat přes protokol MIKOMP vyžaduje trochu více prvků. První z nich je opět odčítačka bajtů, další je paměť zprávy, tedy tři osmi bitové paměťové bloky řazené, jako posuvný registr. Další komponenta je pro kontrolní součet. Tato komponenta nejdříve udělá z bajtu dvojkový doplněk následně jej sečte s výstupem osmi bitové paměti a do té samé paměti nahraje nový součet. Opět jsou tyto všechny komponenty řízené stavovým automatem typu Mealy.



**Obr. 37: Stavový automat pro řízení odesílání dat komponenty mikomp.**

**Tab. 10: Popis jednotlivých stavů pro odesílání dat z komponenty mikomp.**

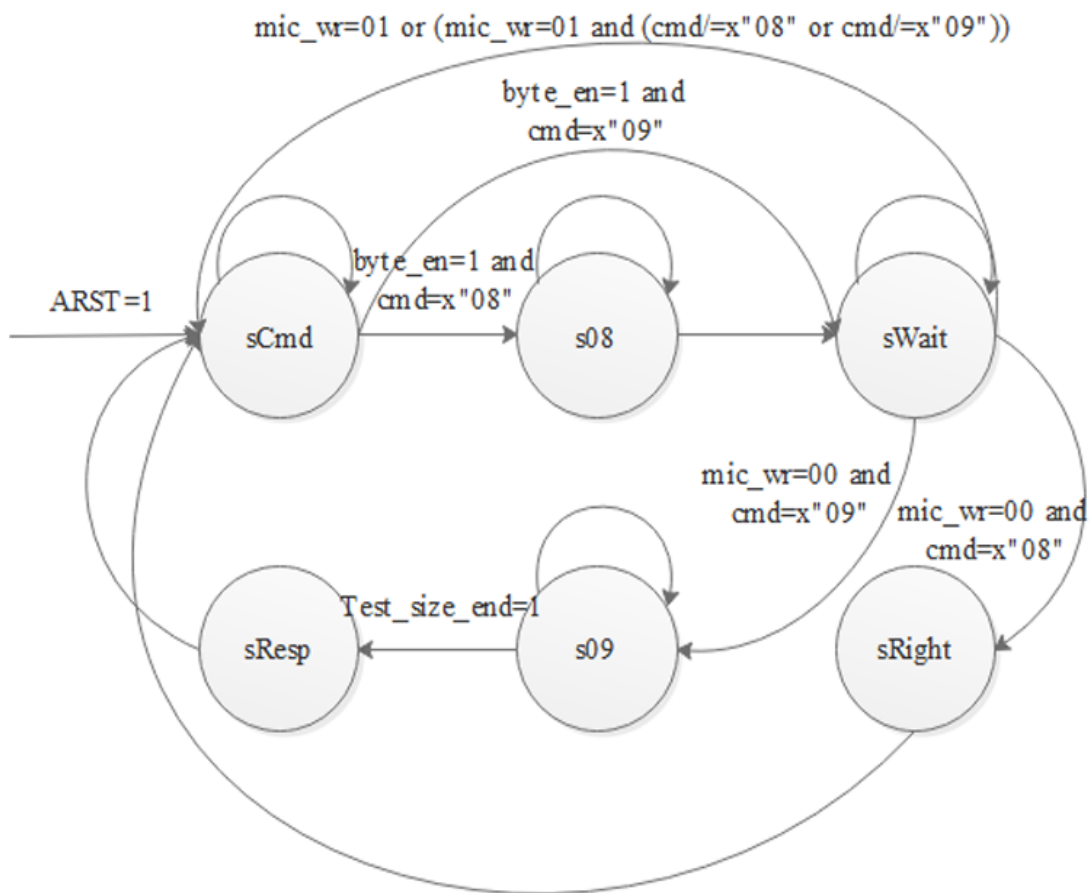
Stav	Popis stavu
sWait	Stav načtený po ARST. Stav čeká na příkaz od nadřazeného prvku, že má začít vysílat.
sSize_o	Jednotaktový stav odesílající do UARTu první bajt dat, tedy velikost.
sDat_p	V tomto stavu jsou vždy po potvrzení odeslání stop bitu nahrána další data do posuvného registru komponenty UART.
sCrc_o	Přechod do tohoto stavu je při potvrzení načtení předposledního bajtu dat. Následně stav řeší odeslání kontrolního součtu a resetování paměti kontrolního součtu při příjmu informace o odeslání posledního datového bajtu

Nadřazená komponenta mikomp\_top pouze vyhodnocuje a zpracovává povely. Obsahuje dvě osmi bitové paměti. Jedna je pro uložení povelu a druhá pro uložení jednoho přijatého bajtu dat. Pro diplomovou práci je potřeba jen dvou povelů. V případě většího počtu povelů by bylo lepší celý MIKOMP vytvořit na implementovaném mikrokontroleru na FPGA. Oba povely jsou v tab. 11.

**Tab. 11: Povely pro MIKOMP.**

Povel	Popis povelu	Odpověď na povel
X"08"	Příjem 1 bajtu jasu.	-
X"09"	Žádost o rozměry panelu.	X"09", počet sloupců, počet řádků

Celý příjem a odesílání povelů implementovaný do struktury FPGA řídí Moorův stavový automat viz obr. 38.



**Obr. 38: Stavový automat pro nadřizenou komponentu mikomp\_top.**

**Tab. 12: Popis jednotlivých stavů stavového automatu pro UART.**

Stav	Popis stavu
sCmd	Prvotní stav, v němž se čeká na příjem povelu a uložení povelu do paměti. Podle přijatého povelu je určen následující stav.
s08	V tomto stavu je povolen zápis do osmibitové paměti.
sWait	Jak název napovídá, čeká se na příjem odpovědi o správnosti přenosu. Tedy odpovídá-li kontrolní součet. Pokud neodpovídá nebo není správný povel automat se vrací do prvotního stavu. V opačném případě přechází do dalších stavů.
sRight	Jednotaktový stav řešící zápis jasu do paměti, ze které je potom čten pro výstup.
s09	Stav, který posílá povelův pro testování velikosti panelu. Po konci testování přechází do odpovědi.
sResp	Opět jednotaktový stav, který odešle do komponenty mikomp správný rámec dat a také povel k odeslání odpovědi.

## 7.3 Obrazová paměť

Obrazová paměť je velmi jednoduchý prvek. Skládá se z přepínače RAM paměti a povolení přepnutí. Povolení je potřebné při zobrazování pohyblivých obrazců. Jelikož příjem dat a odesílání dat do panelu není synchronní, není možné přepínání paměti dělat automaticky po určitém časovém intervalu. Nelze tak ani činit automaticky po příjmu dat celého obrazce. To by mělo za následek u rychlých obrazových změn špatné míchání barev snímku při přechodu. Nastávalo by chybné čtení dat v různých částích binární modulace. Jako příklad lze uvést třeba situaci, kdy se celý obrazec dat načte do RAM paměti ve chvíli, když je zobrazován čtvrtý bit BCM. Tedy první čtyři bity barvy byly zobrazeny z jedné paměti a další dva jsou zobrazeny již z druhé paměti. Je-li na nich jiný obrazec, mohlo by to v aplikacích s malou snímkovací frekvencí znamenat probliknutí nežádoucích barev.

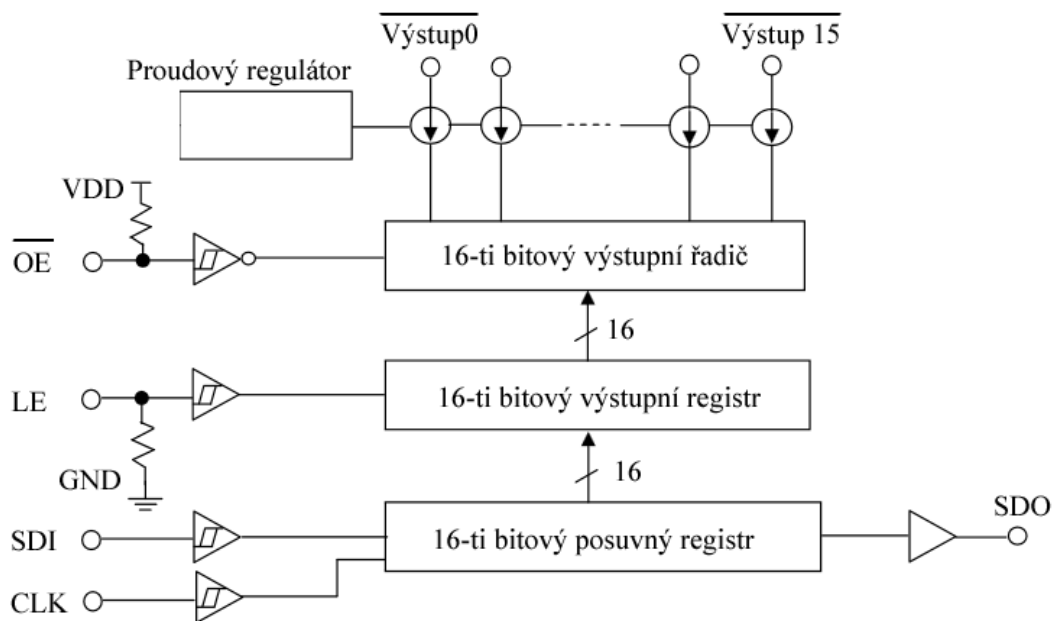
Z toho důvodu je povolení přepnutí vytvořeno jako jednobitová paměť se synchronním resetem a hradlem AND na výstupu. V případě příjmu informace o načtení dat do RAM je do paměti nahrána jednička. Ta je synchronně resetována informací o odeslání posledního řádku a bitu dat do panelu. Do hradla AND po dobu jednoho hodinového taktu přichází z výstupu paměti logická jednička a zároveň resetovací jednička, což tvoří přepínací pulz.

### 7.3.1 Přepínač paměti

Přepínač obsahuje jednobitovou paměť s negovaným výstupem přivedeným na vstup a je přepínána přepínacím pulzem. Dále obsahuje osm vstupních a dva výstupní multiplexory. Pro přiřazování správných dat na jednotlivé RAM paměti. Samozřejmě obsahuje i RAM paměti. Z důvodů uvedených v kapitole 5.2.2. Musí být obě paměti dvouportové. Pro panel o dvě stě padesáti šesti sloupcích, třiceti dvou řádcích a šestnácti bitové barvě je potřebné nastavit paměť na 8192 šestnácti bitových slov. Druhou možností je použití jednoportové paměti s 4096 třiceti dvou bitovými slovy. Takové zapojení sice ušetří tři multiplexory v paměťovém přepínači a výpočet pro druhou adresu, ale neúměrně zvětší logiku pro zápis bajtů do paměti. Z toho důvodu byla vybrána dvouportová paměť. Rozšíření bitů může být vhodné při potřebě tří a více portové paměti.

## 7.4 Ovládání panelu

O ovládání panelu se zmiňuje kapitola obr. 39, v níž je schéma monochromatického panelu, který se od plnobarevného panelu liší jen menším počtem LED řadičů. Plnobarevný panel typu Adafruit má trojnásobek řadičů MBI5024. Schéma tohoto řadiče je na obr. 39.



Obr. 39: Schéma zapojení řadiče MBI5024.[33]

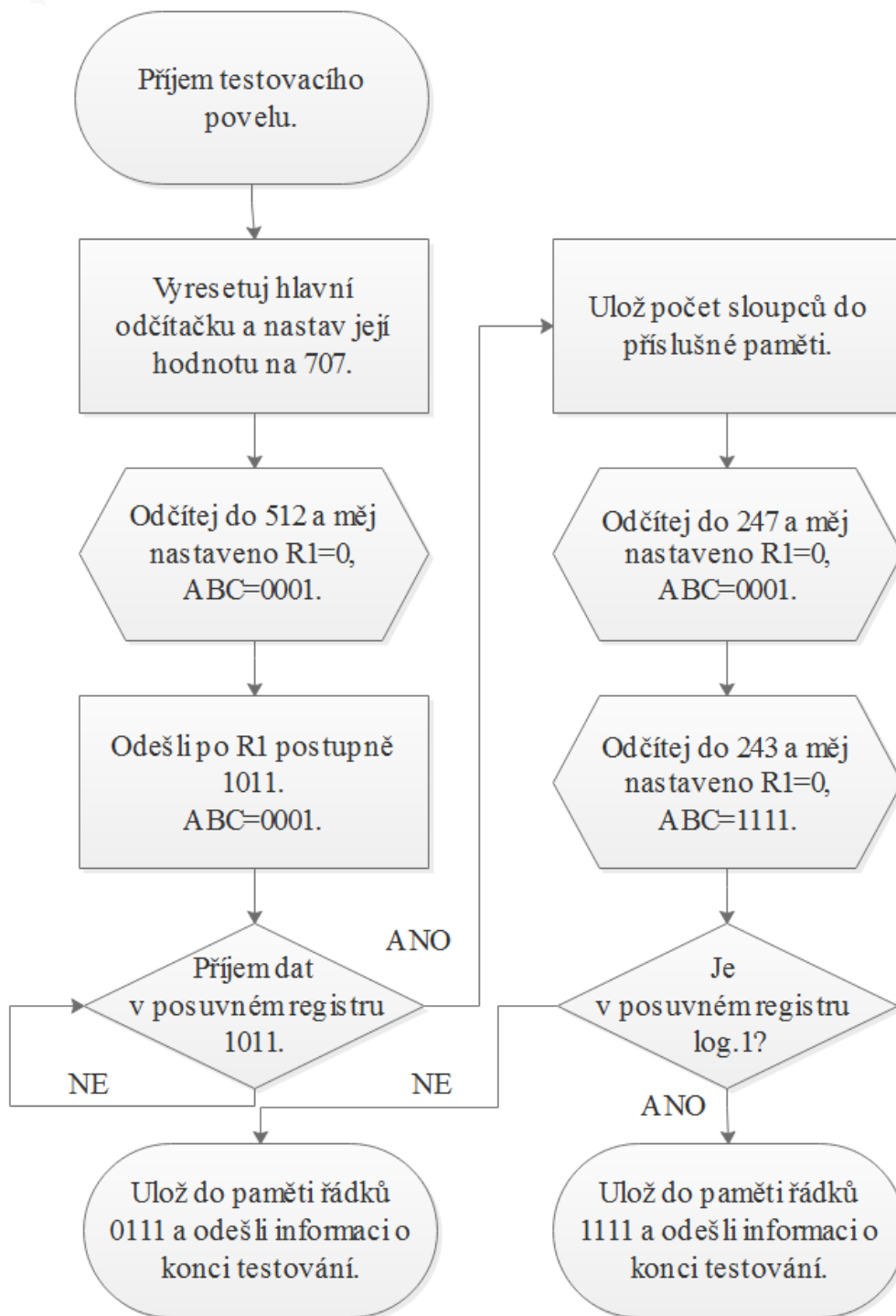
Popis pinů je následovný:

- CLK – hodinový signál,
- SDI – datový signál,
- LATCH – signál pro překlápění dat z posuvného do výstupního registru,
- OE – negovaný signál spínací výstupní řadič,
- SDO – výstupní data z posuvného registru.

Ovládání panelu je poslední ale nejsložitější prvek celého systému. Kromě samotného rozsvěcování LED na panelu obsahuje ještě obvod pro testování panelu. Hlavním prvkem celé komponenty je dvanácti bitová odčítačka. Ta má za úkol číst délky jednotlivých úseků BCM. Dále při posledních odečtení až na hodnotu dvě stě padesát šest komparátor spustí vyčítání dat z paměti a odesílání do panelu. Také je podle odčítačky řízen globální jas a testovací řetězec. Při překlápění do nového časového úseku rámce je kombinační logikou vytvořen jednotaktový pulz LATCH. Na výstupu datových vodičů jsou ještě multiplexory s klopným obvodem. Multiplexory mají za úkol odeslat správné bity dat a klopné obvody zajišťují, že se na výstupu pinů neobjeví zpoždění z kombinační části obvodu.

## 7.4.1 Testování velikosti panelu

Přijde-li signál z komponenty mikomp\_top o vyžádaném testování velikosti panelu, převezmou testovací obvody kontrolu nad odesíláním dat. Přesněji řečeno nad odesíláním dat do červené složky barvy R1 první poloviny panelu a řízení řádků ABC. Celý diagram je na následujícím obrázku.

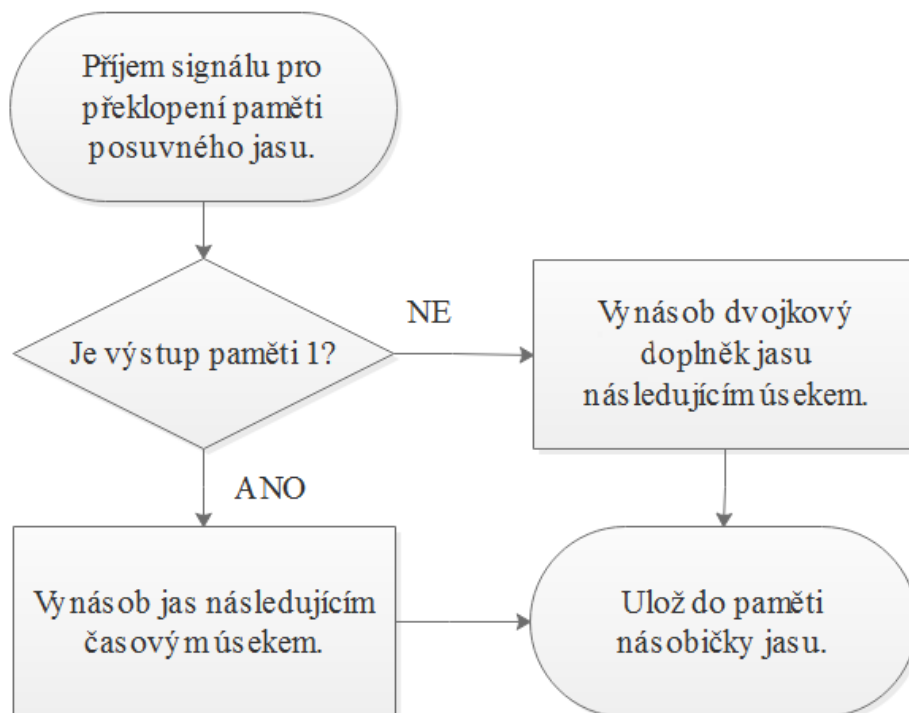


**Obr. 40: Vývojový diagram dynamického řízení BCM.**



## 7.4.4 Řízení jasu

Pro řízení posuvného jasu je potřeba mít jednobitový klopný obvod s negovaným výstupem na vstup (přepínač), paměť pro vynásobený následující jas, několik multiplexorů a násobičku jasu. Na obr. 42 lze vidět vývojový diagram pro ukládání následujícího jasu.



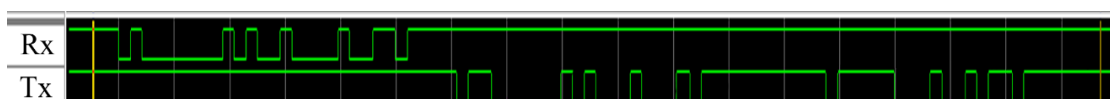
**Obr. 42: Schéma zapojení vstupní části dynamické BCM.**

Následující jas se uloží přesně ve chvíli, kdy nastává následující časový úsek. Poté je jen výstup z paměti komparován s hlavním čítačem a výsledek s přepínačem přes hradlo XOR vytváří OE.

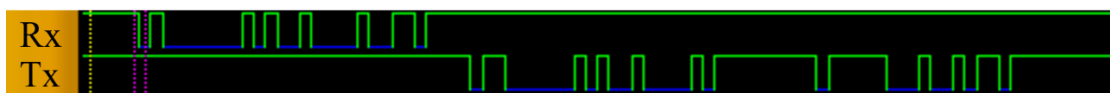
## 8 VÝSLEDKY PROJEKTU

Celý návrh programu prošel jednotlivými simulacemi a bylo tak možné odhalit a odstranit nedostatky. Dále prošel vizuálním testem, a to jak v místnosti, tak ve venkovním prostředí. Při venkovním svitu však podle očekávání nedosáhl potřebné intenzity světla. Potvrdilo se tedy, že pro venkovní návrh je nutné použití menších multiplexorů řádků.

Kromě simulací byly průběhy signálů podrobeny snímkování logickým analyzátozem. Na obr. 43 je vidět simulace odesílání testovacího povelu a odpověď na něj. Na obr. 44 je stejný průběh, ale již nasnímkovaný logickým analyzátozem. Na obrázcích jde vidět odeslaná zpráva po následná zpráva odeslaná po Rx (0h01,0h09, 0hF6) a odpověď na tuto zprávu po Tx (0h03, 0h09, 0hFF, 0h1F, 0hDA). Naměřené průběhy logickým analyzátozem byly shodné.

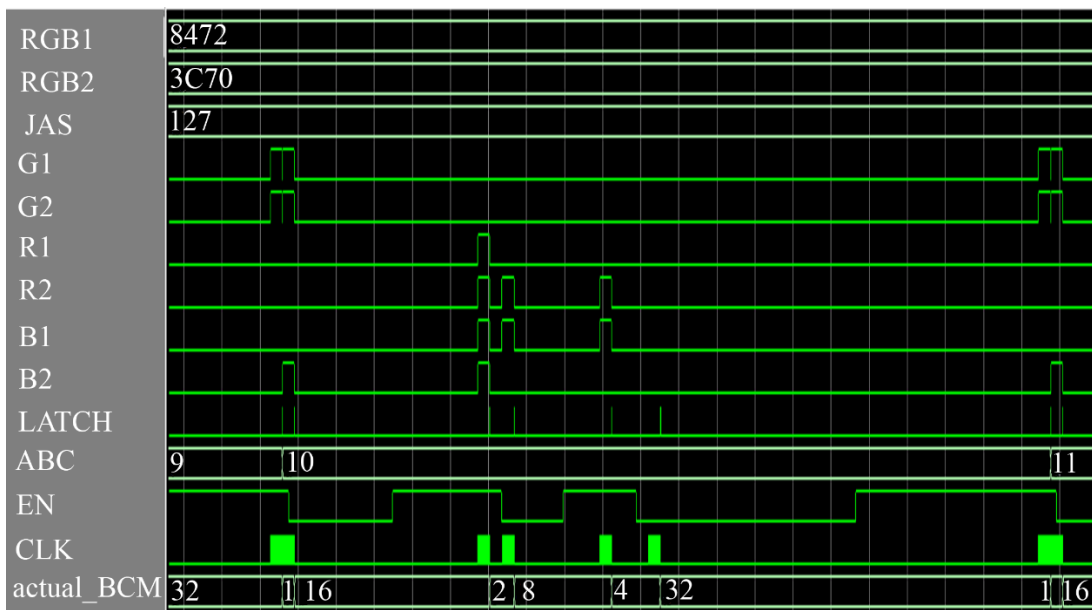


**Obr. 43: Simulace komunikace testovacího řetězce po UART.**

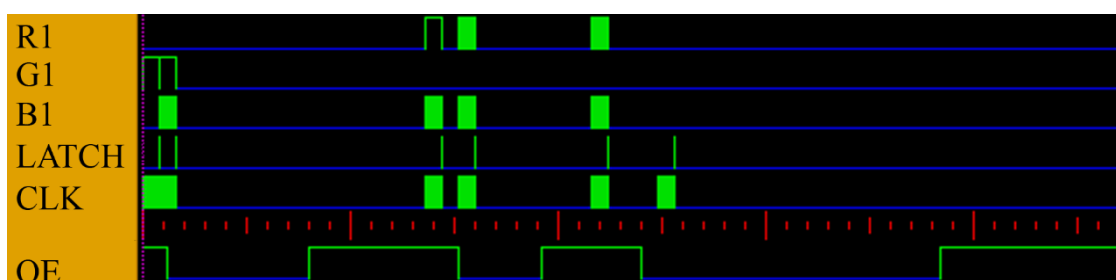


**Obr. 44: Nasnímkovaná komunikace testovacího řetězce po UART logickým analyzátozem.**

Dalším simulovanými a testovanými průběhy bylo řízení panelu. Pro lepší přehlednost. Jak pro simulaci, tak pro reálné průběhy byly odesílány jen dvě barvy (0h8472 a 0h3C70) a stejný jas o hodnotě sto dvacet sedm, tedy padesáti procentní jas. Odesílaná data barvy jsou signály R1/2, G1/2, B1/2. Na obr. 45 jde dále vidět přepínání řádků (ABC), padesáti procentní posuvný jas (OE). Podle CLK jde vidět, že data se načítají vždy těsně před překlopením do nového časového úseku. Signál LATCH tyto úseky ohraničuje. Actual\_BCM ukazuje, jaký úsek je aktuálně zobrazován. Podobné průběhy jako simulace vykazuje i snímek pořízený logickým analyzátozem viz obr. 46. Tím byl snímán jen jeden datový kanál RGB. Oproti simulaci však nebyla na každý blok použita jedna barva, ale testovací obrazec byl složen z šachovnice s barvami stejnými jako v případě simulace. Díky tomu je stejný průběh dynamické BCM, tedy dvojice 1–6, 2–8 a 4–32. Na obou dvou obrázcích je zachyceno načtení jednoho celého řádku.



Obr. 45: Simulace signálů řídicích panel.



Obr. 46: Průběh řídicích signálů panelu zachycený logickým analyzátořem.

## 8.1 Implementace na FPGA

Zápis VHDL kódu, kompilace a implementace byla vytvořena programem Quartus Prime v edici LITE. Tato edice je dostačující pro návrh řídicích systémů LED RGB panelu. Jelikož nejsou žádné nároky na vysokou pracovní frekvenci, využitou plochu a nízký odběr FPGA, je zvolen pro kompilaci vyrovnaný režim (Balanced). Pro stavové automaty je použito One-Hot kódování. Celkové využití takto kompilovaného FPGA je znázorněno v tab. 13. MAX 10 složen z logických prvků (Logic elements). Ty obsahují jak náhledové tabulky LUT, tak registry. Proto jsou v tabulce na prvním místě uvedeny použité logické prvky a následující tři řádky popisují přesnější využití buněk.

**Tab. 13: Tabulka využití FPGA.**

Název	Využito	Celkový počet	Procentuální využití
Celkově logické prvky (LE)	952	8 064	11,8
Náhledové tabulky LUT	637	8 064	7,9
Registry	43	8 064	0,5
Registry + LUT	272	8 064	3,4
RAM paměť	363 144	387 072	68
DSP devíti bitové násobičky	3	48	6
PLL bloky	1	1	100
I/O piny	27	101	27

Pro zjištění kmitočtových závislostí za programu Time Quest byl zvolen SLOW 1200mV 85C model. Ze všech tří modelů vykazoval nejmenší maximální kmitočty. Pro časovou základnu, která teď využívá 25MHz, byl maximální kmitočet 55,91 MHz a pro časovou základnu vstupního hodinového signálu to bylo 162,68 MHz. Oba dva maximální kmitočty jsou dostačující, jelikož jsou dvojnásobkem obou používaných hodinových signálů. Největší kombinační cesta, která snižuje maximální pracovní kmitočet první uváděné časové základně je od výstupu z RAM paměti až po výstupní registry RGB dat. V případě potřeby by šly vložit do cesty registry, avšak znamenalo by to minimálně o jednu latenci navíc zpoždění na výstupu. Jelikož jsou kmitočtové závislosti dostačující, není důvod další optimalizace.

## 9 ZÁVĚR

V práci byla shrnuta teorie týkající se řízení LED panelů s důrazem na plně barevné RGB panely. Cílem bylo prostudovat teorii, udělat rešerši a praktickou část diplomové práce. Dalším cílem bylo zvolit bitovou hloubku barvy, která je stanovena na šestnáct bitů (RGB 565). Tady se naskytá otázka využitelnosti panelu s šedesáti pěti tisíci pěti sty třiceti šesti barvami. Přeci jen v hromadné dopravě je taková široká paleta barev zbytečná. Využití této palety barev je například na velkoplošných obrazovkách. Dále byl navrhnut způsob řízení celého RGB panelu. Jako nejvhodnější se jeví řízení panelu pomocí dynamické binární kódové modulace spolu s posuvným jasem. Dohromady jsou velmi silnou dvojicí nezanedbatelně zvyšující šířku napěťových, resp. proudových pulzů. A tím snižují elektromagnetické vyzařování a celkové teplotní ztráty na diodách. Aby byl zjištěn celkový vliv obou těchto změn při řízení RGB panelů, bude nutné další zkoumání a testování vyzařování, jak tepelného, tak elektromagnetického, v porovnání s klasickým řízením binární kódovou modulací a pulzně šířkovou modulací. Pro řízení testovaného panelu bylo vybráno FPGA od firmy Altera MAX 10 10M08SAE144C8GES, který plně vyhovuje. Avšak pro použití do vozidel hromadné dopravy je nutné použití čipu s širším rozmezím pracovních teplot. V praktické části byl navrhnut obvod na FPGA, který přijímá obrazová data přes SPI a komunikuje pomocí UART s nadřazeným prvkem. Dále tato data ukládá a používá pro řízení LED panelu. Komunikace SPI byla navrhnutá s obvodem navíc, které zajišťují spolehlivější příjem dat. Pomocí UART komunikace je přijímán jas, který se využívá v konečném řízení panelu. Také je po UARTu posílána informace o velikosti panelu. RAM paměť byla navrhnutá jako dvouportová s přepínačem signálů. Přepínač obsahuje ochranu proti přepnutí paměti ve chvíli, kdy nejsou odeslána všechna data jednoho snímku. Celá práce je koncipována, jako návod, informační materiál a pomyslný odrazový můstek pro firmu BUSE s.r.o., která má zájem na rozšíření svého sortimentu informačních LED panelů o RGB LED panely. Zkušební panel typu Adafruit, na kterém byla diplomová práce dělána, není ideální řešení pro použití v dopravě. Ať již z důvodu, že obsahuje součástky s malým rozsahem pracovní teploty, tak i kvůli příliš velkému multiplexoru, nímž je spínáno šestnáct řádků. Při venkovním osvětlení je viditelnost panelu kriticky snížena. Proto je potřeba navrhnout panel o menším multiplexoru, který bude řídit dva, maximálně čtyři řádky. Dále je třeba se zaměřit na implementaci používanějších komunikačních protokolů, jako jsou HDMI, Ethernet a další. Jedním z cílů práce byl návrh zapojení modulu a návrh desky plošných spojů. Oba dva návrhy jsou v příloze.

## LITERATURA:

- [1] Earge learning: Light Emitting Diodes [online]. 2014 [cit. 2017-12-11]. Dostupné z: <http://eagerlearning.org/electronics/introduction/light-emitting-diodes/>
- [2] Iq scan: LED osvětlení [online]. Praha, 2017 [cit. 2017-12-11]. Dostupné z: <http://www.iqscan.cz/clanky/led-osvetleni>
- [3] LED Drive Methods and Circuit Design. Stanley Electric Co., Ltd [online]. [cit. 2017-12-11]. Dostupné z: [https://www.stanley-components.com/data/technical\\_note/TN003\\_e.pdf](https://www.stanley-components.com/data/technical_note/TN003_e.pdf)
- [4] Voltampérová charakteristika polovodičovej diódy. In: Elektrotechnické spínacie súčiastky [online]. 2014 [cit. 2017-12-11]. Dostupné z: <http://elektronicke-suciastky56.webnode.sk/dioda/>
- [5] Segment labeling for 7-, 14-, and 16-segment displays. In: Maxim integrated [online]. 2002 [cit. 2017-12-11]. Dostupné z: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1131>
- [6] LED matrix. In: Simply Smarter Circuitry Blog [online]. 2016 [cit. 2017-12-11]. Dostupné z: <https://www.circuitspecialists.com/blog/wp-content/uploads/2016/02/Getting-started-with-LED-Matrix-Tutorial-768x472.jpg><https://www.paladix.cz/clanky/barevne-videni-druhy-pohled.html>
- [7] Barevné modely. In: Stránky k výuce informatiky [online]. 2017 [cit. 2017-12-11]. Dostupné z: <http://www.ivt.mzf.cz/grafika/barevne-modely/><https://www.techfak.uni-bielefeld.de/ags/wbski/lehre/digiSA/WS0607/3DVRCG/Vorlesung/8a.RT3DCGV-R-color.pdf>
- [8] Gamma korekce. Fotoroman.cz[online]. Praha, 2011 [cit. 2017-05-23]. dostupné z:[http://fotoroman.cz/glossary/3\\_gamma.htm](http://fotoroman.cz/glossary/3_gamma.htm).
- [9] 32x16 and 32x32 RGB LED Matrix. Adafruit [online]. 2012[cit. 2017-11-17]. Dostupné z :<https://learn.adafruit.com/32x16-32x32-rgb-led-matrix/overview>.
- [10] IEEE 802.3 ETHERNET WORKING GROUP [online]. 2017 [cit. 2017-12-11]. Dostupné z: <http://www.ieee802.org/3/>
- [11] 10Base-T Ethernet RJ-45 Pinouts. In: Highteck.net [online]. 2016 [cit. 2017-12-11]. Dostupné z: <http://www.highteck.net/images/243-Ethernet-type-RJ-45.jpg>
- [12] Sítě: Fyzický přenos, MAC adresy a protokoly. In: IT network [online]. 2015 [cit. 2017-12-11]. Dostupné z: <https://www.itnetwork.cz/site/fyzicky-prenos-mac-adresy-a-protokoly>
- [13] Types of ethernet. In: Highteck.net [online]. 2016 [cit. 2017-12-11]. Dostupné z: <http://www.highteck.net/images/242-Ethernet-type.jpg>
- [14] Sítě: Ethernet a rozbočovače. In: IT network [online]. 2015 [cit. 2017-12-11]. Dostupné z: <https://www.itnetwork.cz/site/ethernet-a-rozbocovace>

- [15] LALÍK, Aleš. LVDS – snadné interní připojení (nejen) LCD. Netbook.cz [online]. 2009 [cit. 2017-12-11]. Dostupné z: <https://notebook.cz/clanky/technologie/2009/lvds>
- [16] KAPIC, Tomáš. HDMI rozhraní. Praha, 2010. Semestrální práce. ČVUT Praha.
- [17] HDMI basics. In: Areas of interest: making sense of stuff [online]. 2010 [cit. 2017-12-11]. Dostupné z: [https://friedfoo.files.wordpress.com/2010/02/hdmi\\_paper.jpg](https://friedfoo.files.wordpress.com/2010/02/hdmi_paper.jpg)
- [18] PIŠKULA, Jiří. Tisíc podob HDMI. Tv freak [online]. 2010 [cit. 2017-12-11]. Dostupné z: <https://www.tvfreak.cz/tisic-podob-hdmi/3666-2>
- [19] HDMI connector pinout. In: National Instruments [online]. 2017 [cit. 2017-12-11]. Dostupné z: [http://www.ni.com/cms/images/devzone/tut/HDMI\\_Connector\\_Pinout.png](http://www.ni.com/cms/images/devzone/tut/HDMI_Connector_Pinout.png)
- [20] BARTEK, T. Řídící 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.
- [21] Display HDMI [online]. 2014 [cit. 2017-12-11]. Dostupné z: <https://www.slideshare.net/PREMGALGAJJAR/hdmi-35120513>. Prezentace.
- [22] Understanding EDID: An Introduction to AW EDID Editor. Analog Way [online]. , 10 [cit. 2017-12-11]. Dostupné z: <http://www.analogway.com/files/uploads/white-papers/white-paper-edid--082714.pdf>
- [23] Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877. Vyoj.hw.cz [online]. Redakce HW serveru [cit. 2017-12-11].
- [24] ŠUNKA, P. Zpracování keramických střepů pro archeologii. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 36 s. Vedoucí bakalářské práce doc. Ing. Pavel Šteffan, Ph.D..
- [25] VALACH, Soběslav. Moderní metody zpracování obrazu strukturou FPGA. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Dostupné také z: <http://old.roznovskastredni.cz/dwnl/pel2009/10/valach.pdf>. Učební materiál.
- [26] KUSHPIIL, Vasilij. Programovatel ná logika. Dostupné také z: <http://ojs.ujf.cas.cz/~kushpil/02EMBS/Lecture10.pdf>. Prezentace.
- [27] [https://moodle.vutbr.cz/pluginfile.php/290290/mod\\_resource/content/4/Prednaska\\_09\\_DSP\\_IO\\_Transceivery.pdf](https://moodle.vutbr.cz/pluginfile.php/290290/mod_resource/content/4/Prednaska_09_DSP_IO_Transceivery.pdf) KUBÍČEK, Michal. Přednáška 9. Brno. [cit. 2017-12-11]. Prezentace. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií.
- [28] Programovatelné logické obvody a hradlová pole. Brno. [cit. 2017-12-11] Dostupné také z: [http://www.urel.feec.vutbr.cz/~kolouch/pld/1\\_prednasky/obr/obr06\\_02.png](http://www.urel.feec.vutbr.cz/~kolouch/pld/1_prednasky/obr/obr06_02.png). Učební text. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií.

- [29] Farnell Company [online]. [cit. 2017-12-11]. Dostupné z: <http://cz.farnell.com/>
- [30] Mouser Electronics [online]. [cit. 2017-12-11]. Dostupné z: <https://cz.mouser.com/>
- [31] MAX 10M08 Evaluation kit. In: Altera [online]. [cit. 2017-12-11]. Dostupné z: [https://www.altera.com/content/dam/altera-www/global/en\\_US/images/products/devkits/images/max\\_10\\_eval\\_board.jpg](https://www.altera.com/content/dam/altera-www/global/en_US/images/products/devkits/images/max_10_eval_board.jpg).
- [32] Imgur. [stack.imgur.com](https://i.stack.imgur.com/rOWzl.gif) [online]. [cit. 2018-05-20]. Dostupné z: <https://i.stack.imgur.com/rOWzl.gif> [33] <http://belchip.by/sitedocs/00006155.pdf>
- [33] Macroblock MBI5024 [online]. 2008 [cit. 2018-05-20]. Dostupné z: <http://belchip.by/sitedocs/00006155.pdf>