



Prirodno-matematički fakultet
Univerzitet u Kragujevcu

Obrada fotografija na PYNQ Z-2 kartici i prenosom podataka putem UART porta

PROJEKTOVANJE VLSI SISTEMA
PROJEKTNII ZADATAK

Profesor:
Aleksandar Peulić

Student:
Krunoslav Velfel (49/2018)

Uvod	3
Tehnički detalji	4
Postupak	5
KREIRANJE PROJEKTA U SOFTVERU VIVADO	5
RAD SA VITIS OKRUŽENJEM	5
POVEĆANJE STEK I HIP MEMORIJE	6
RASPORED BITOVA U BMP DATOTECI I UČITAVANJE PODATAKA	6
OBRADA FOTOGRAFIJE	7
PROGRAMIRANJE PLOČE I PRIMJER UPOTREBE	8
Nedostaci	10

Uvod

Cilj projekta je omogućiti obradu fotografija BMP ekstenzije na ZYNQ 7020 čipu. Datoteka se putem UART (serijskog, SR-232) porta preusmjerava na PYNQ Z-2 karticu, vrši se obrada, te se istim putem obrađena datoteka dostavlja računar.

BMP format datoteke kreiran je prvobitno za potrebe operativnog sistema Windows, ali se zbog jednostavnog formata i lakog tumačenja našao u širokoj primjeni. BMP format ne posjeduje ni jedan mehanizam za komprimovanje, budući da se podaci o svakom pikselu pojedinačno pamte u datoteci. Ova karakteristika omogućava da se sadržaj datoteke lako protumači, bez upotrebe složenih parsera, i samim tim olakša obrada fotografije na uređajima slabije računske moći kao što su pojedini programabilni uređaji.

PYNQ Z-2 dolazi sa ugrađenim mostom (eng. *bridge*) za UART komunikaciju. UART, kao asinhroni način prenosa podataka posjeduje brojne nedostatke, a koji će biti obrađeni u nastavku dokumenta. Takođe, biće navedeni detalji implementacije i neki od načina kako rješenje može biti efikasnije.

Tehnički detalji

Hardver:

- PYNQ Z-2 kartica sa mikro-USB kablom;
- računar sa USB portom.

Softver:

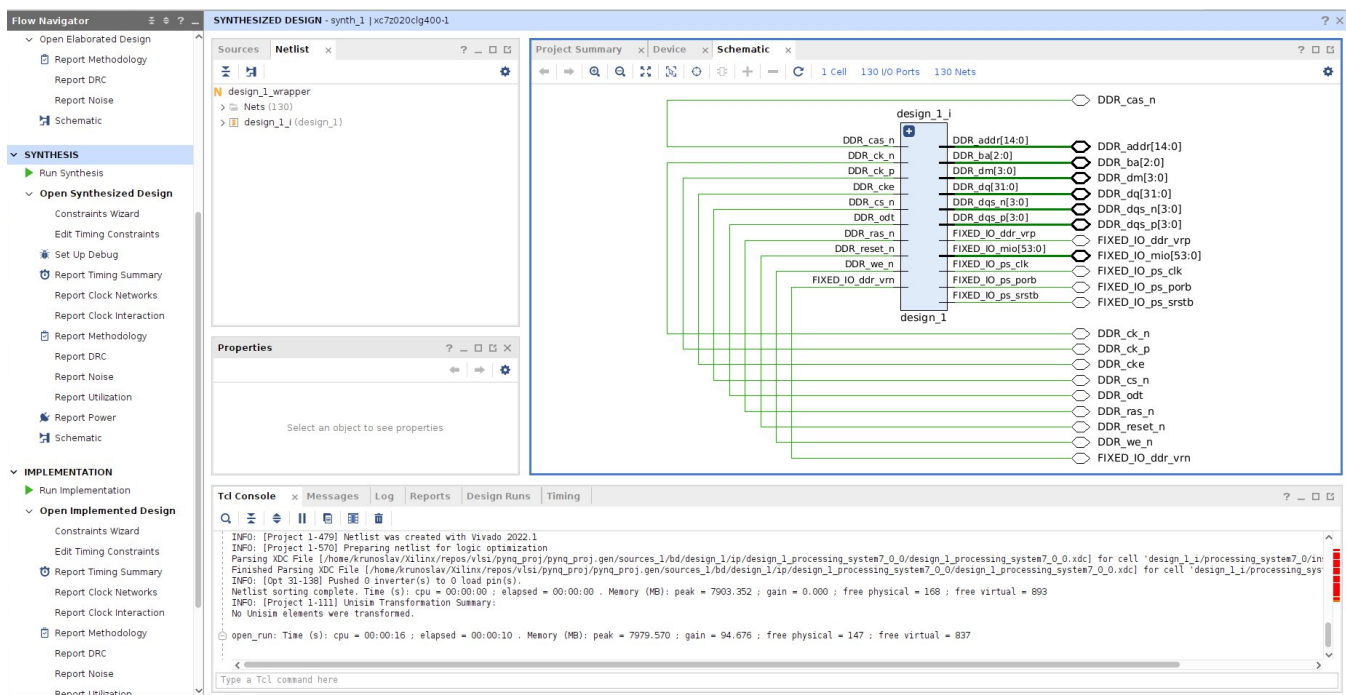
- Ubuntu 20.04LTS operativni sistem instaliran na računaru;
- Vivado 2022.1 (sa drajverima za PYNQ Z-2 karticu);
- Vitis IDE 2022.1;
- clang C kompajler, v10.0.0;
- cbmp biblioteka;
- CuteCom softver za serijsku komunikaciju.

Postupak

KREIRANJE PROJEKTA U SOFTVERU VIVADO

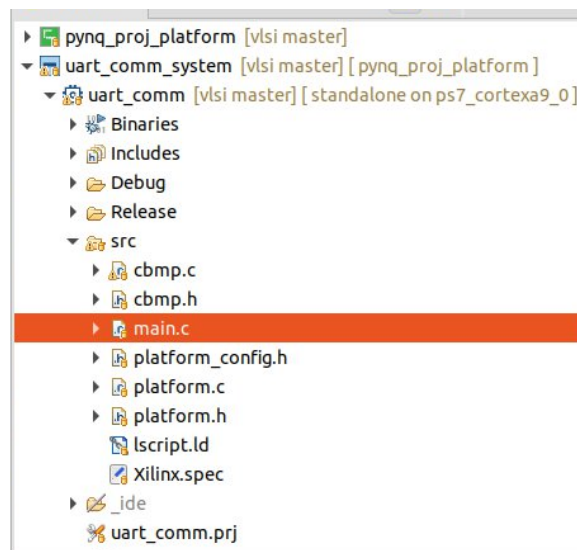
Za PYNQ Z-2 ploču, bitno je spojiti M_AXI_GP0_ACLK sa FCLK_CLK0 izlazom i kreirati novu šemu. Takođe, u podešavanjima potrebno je:

- preusmjeriti `stdio` i `stdout` na UART ulaz, nakon čega se sintetiše novi hardverski dizajn; i
- podesiti *baud rate* na učestalost 115200.



RAD SA VITIS OKRUŽENJEM

Nakon što se izveze (eng. *export*) hardverska konfiguracija u XSA datoteku, kreira se novi C projekat u Vitis-u:



POVEĆANJE STEK I HIP MEMORIJE

Kao što je već pomenuto, BMP datoteke nemaju mehanizam za kompresiju i često zauzimaju dosta memorije. Stoga, potrebno je povećati vrijednosti *heap* i *stack* memorije kako bi se svi podaci uspješno smjestili u privremenu memoriju PYNQ ploče. Shodno tome, potrebno je modifikovati parametre linkera, smještene u `lscript.ld` datoteci. Od posebne je važnosti vrijednost *heap* memorije, budući da se sva memorija u ovom projektu alocira dinamički, pozivanjem `malloc()` funkcije.

Linker Script: lscript.ld

A linker script is used to control where different sections of an executable are placed in memory. In this page, you can define new memory regions, and change the assignment of sections to memory regions.

Available Memory Regions

Name	Base Address	Size
ps7_ddr_0	0x100000	0x1FF00000
ps7_qspi_linear_0	0xFC000000	0x1000000
ps7_ram_0	0x0	0x30000
ps7_ram_1	0xFFFF0000	0xFE00

Stack and Heap Sizes

Stack Size

Heap Size

RASPORED BITOVA U BMP DATOTECI I UČITAVANJE PODATAKA

image.bmp x

42	4D	8A	00	00	00	00	00	00	00	00	7A	00	00	00	6C	00	BMè.....z...l.
00	00	02	00	00	00	02	00	00	00	01	00	18	00	00	00	00
00	00	10	00	00	00	13	0B	00	00	13	0B	00	00	00	00	00
00	00	00	00	00	00	42	47	52	73	00	00	00	00	00	00	00BGRs.....
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	02	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	FF	00	FF	00	00
00	00	FF	00	00	FF	FF	FF	00	00	+						

Svaka BMP datoteka počinje heksadekadnim vrijednostima 0x42 0x4D (“BM”). Sljedeća četiri bajta predstavljaju veličinu datoteke i broj ukupnih bajtova koje je potrebno učitati. **Bajtovi su raspoređeni *big endian* rasporedom, što znači da je veličina ove datoteke 138 bajtova.**

Učitavanje bajtova postiže se pozivom funkcije `inbyte()`, koja sa standardnog ulaza učitava i kao povratnu vrijednost vraća vrijednost veličine jednog bajta. Budući da je hardver konfigurisan na taj način, standardni ulaz i izlaz preusmjereni su na tok podataka sa UART porta.

```
for (i = 0; i < 2; i++) {
    buf[i] = inbyte();
}

unsigned int sz = 0UL;

for (; i < 6; i++) {
    sz = sz >> 8;
    buf[i] = inbyte();
    unsigned int v = 0U | buf[i];
    v = v << 24;
    sz = sz | v;
}

printf("sz = %d\n", sz);

while (sz - i > 6) {
    buf[i] = inbyte();
    i++;
}
```

U ovom kodu, učitavaju se prva dva bajta ("BM") i smiještaju u bafer. Sljedeća četiri bajta, koja predstavljaju veličinu datoteke, učitavaju se i pretvaraju u *little endian* reprezentaciju.

Svi ostali bajtovi učitavaju se u *while* petlji, i čitav bafer prosljeđuje se *cbmp* biblioteci.

OBRADA FOTOGRAFIJE

```
BMP* bmp = bopen(sz, buf);

unsigned int x, y, width, height;
unsigned char r, g, b;

// Gets image width in pixels
width = get_width(bmp);

// Gets image height in pixels
height = get_height(bmp);

for (x = 0; x < width; x++) {
    for (y = 0; y < height; y++) {
        // Gets pixel rgb values at point (x, y)
        get_pixel_rgb(bmp, x, y, &r, &g, &b);

        // Sets pixel rgb values at point (x, y)
        set_pixel_rgb(bmp, x, y, 255 - r, 255 - g, 255 - b);
    }
}

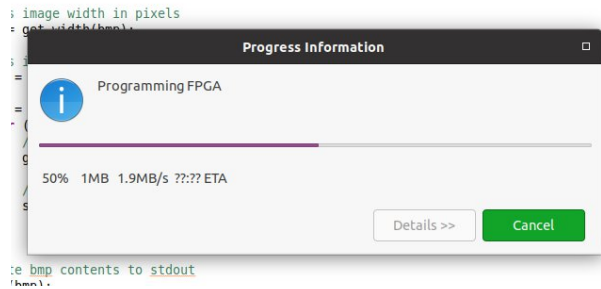
// Write bmp contents to stdout
bwrite(bmp);

// Free memory
bclose(bmp);
```

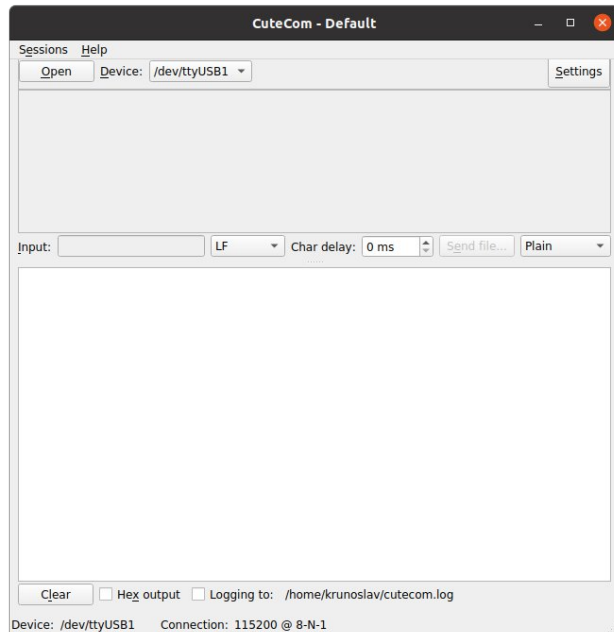
Pozivom funkcija *cbmp* biblioteke vrši se inverzija piksela. Funkcija `bopen()` je modificirana tako da, umjesto pokazivača na FILE strukturu, bude pohranjena učitanim bajtovima skladištenih u već pomenuti bafer.

Slične modifikacije urađene su i za `bwrite()` funkciju. Umjesto pisanja modificirane datoteke na fajl sistem, vrši se ispis heksadekadnih vrijednosti obrađene fotografije na standardni izlaz (tj. u ovom slučaju, na UART serijski port budući da je obavljena redirekcija).

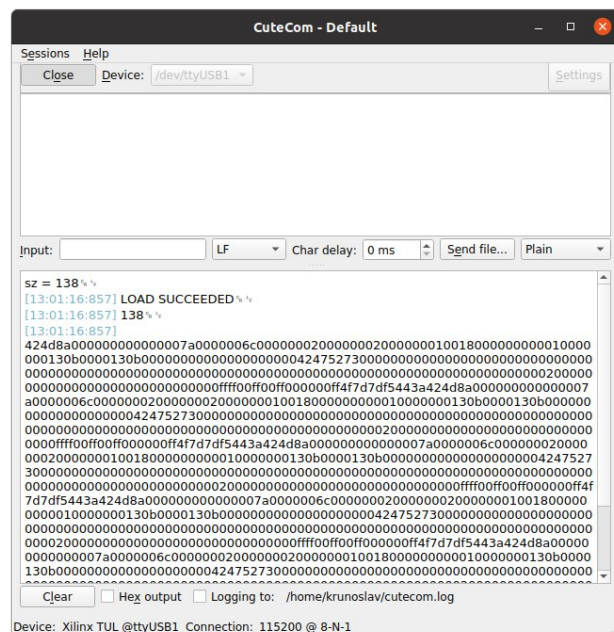
PROGRAMIRANJE PLOČE I PRIMJER UPOTREBE



Pokretanjem *debug* procesa u Vitis okruženju biće kompajliran čitav projekat i ploča će biti programirana.



Zatim se pokreće *CuteCom* softver za serijsku komunikaciju. Potrebno je izabrati odgovarajući port na koji je ploča povezana, otvoriti port, te izabrati opciju *Send file*.



PYNQ kartica će dati nekoliko izlaza:

- očekivani broj bajtova datoteke koja se šalje;
- status učitavanja (uspješno/neuspješno);
- broj uspješno učitanih bajtova;
- heksadekadne vrijednosti modifikovane

BMP fotografije.

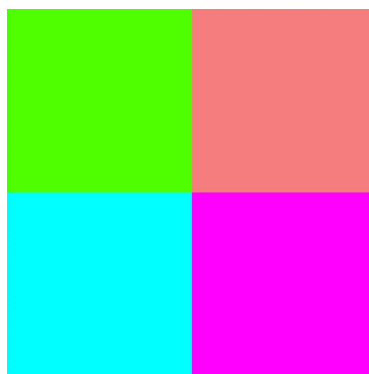
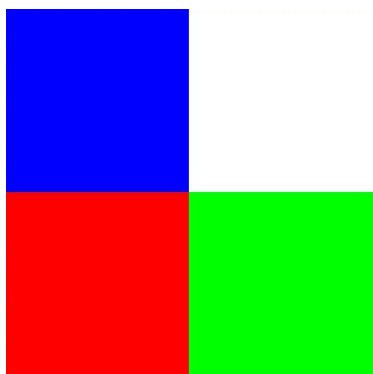

```

-Untitled - x
42 4D 8A 00 00 00 00 00 00 00 00 00 7A 00 00 00 6C 00 BME.....z...l.
00 00 02 00 00 00 02 00 00 00 01 00 18 00 00 00 00 .....
00 00 10 00 00 00 00 13 0B 00 00 13 0B 00 00 00 00 .....
00 00 00 00 00 00 00 42 47 52 73 00 00 00 00 00 00 .....BGRs.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 FF FF 00 FF 00 FF .....
00 00 00 FF 4F 7D 7D F5 44 3A 42 4D 8A 00 00 00 00 ...O}}D:BME...
00 00 00 7A 00 00 00 00 00 6C 00 00 00 02 00 00 00 ...Z...l.....
02 00 00 00 01 00 18 00 00 00 00 00 10 00 00 00 .....
13 0B 00 00 13 0B 00 00 00 00 00 00 00 00 00 00 .....
42 47 52 73 00 00 00 00 00 00 00 00 00 00 00 00 BGRs.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 FF FF 00 FF 00 FF 00 00 00 FF 4F 7D .....O}
7D F5 44 3A 42 4D 8A 00 00 00 00 00 00 00 7A 00 ...}}D:BME.....z
00 00 6C 00 00 00 02 00 00 02 00 00 00 01 00 ...l.....
18 00 00 00 00 10 00 00 00 13 0B 00 00 13 0B .....
00 00 00 00 00 00 00 00 00 00 42 47 52 73 00 00 .....BGRs...
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 02 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF .....
00 55 00 55 00 00 00 55 45 70 70 55 44 3A 42 4D ...011D:8

```

Heksadekadne vrijednosti se mogu serijalizovati u novu datoteku putem proizvoljnog *hex* softvera, i potom sačuvati.

Rezultat obrade i poređenje originalne datoteke (lijevo) sa novom (desno). Promijenjen je i profil RGB boja:



Nedostaci

UART port, zbog svoje asinhronne prirode, često uzrokuje gubitke podataka prilikom prenosa. Jedan od načina za remedijaciju ovog problema jeste sporiji *baud rate*, te korišćenje funkcija Xilinx SDK-a koje su nižeg nivoa nasuprot već korišćenih *inbyte*, *outbyte*, *printf*, i sl.

Kako je UART port izuzetno spor, a BMP datoteke često zauzimaju veću količinu memorije, bilo bi poželjno implementirati kompresiju bajtova prilikom prenosa. Jedna od biblioteka koja bi mogla poslužiti toj svrsi jeste LZMA SDK.

Moguće je izvršiti dodatne modifikacije *cbmp* biblioteke kako bi se iskoristilo ubrzanje koje pružaju asemblerske SIMD instrukcije za paralelne proračune na ARM arhitekturi.