

Povijest i disekcija igraće konzole Atari 2600 kroz softversku emulaciju sklopovlja

Tomislav Kućar

Algebra/Primjenjeno računarstvo, Zagreb, Hrvatska
kucar.tomislav@gmail.com

Sažetak - Atari 2600 igraća je konzola razvijena i predstavljena tržištu 1977. godine od strane tvrtke Atari. Konzolu pokreće 8-bitni mikroprocesor MOS 6507, inačica procesora MOS 6502 koji je svojom pojavom na tržištu 1975. izazvao pravu revoluciju.

Ključne riječi – Atari 2600, emulacija, 650x procesori

I. UVOD

Atari 2600, prvotno nazvan Atari VCS predstavlja bitnu točku u povijesti sklopovlja pošto je popularizirao sklopovlja bazirana na mikroprocesorima sa rom karticama koje su sadržavale kod.

U ovom radu analiziram Atari 2600 kroz emulaciju njegovog sklopovlja. Objašnjavam pojmove emulacije te alate potrebne da bi se razvijao kod za platformu (assembler i disassembler). Veliki dio rada je fokusiran na sami procesor mos6507.

Radu je pridružen izvorni kod emulatora, te su na kraju samog rada priložene upute za kompajliranje i korištenje emulatora.

A. Emulacija

U suštini emulacija je rekreiranje ponašanja procesora i individualnih komponenti emulirane platforme[1]. Svaki dio originalnog sistema se rekreira softverski, kao što bi se spajale same žice u sklopovlju.

Postoji nekoliko načina izvršavanja softvera namjenjenog za jedno računalo, na drugom, kao što su virtualizacija, prekompajliranje izvornog koda, cross kompajliranje, simulacija i naravno emulacija. Emulacija je po pitanju performansi često najgori izbor ali najvjernije oponaša izvornu platformu pošto se nesvjesno implementiraju čak i nepoznati nedostaci i bugovi platforme.

Emulacija je direktno povezana sa dizajnom virtualnih mašina poput Java primjerice [2]. Java ima svoj virtualni procesor koji konzumira bytecode. Java VM možemo u stvari promatrati kao emulator za sklopovlje koje nikad nije bilo implementirano. Dakle, emulator možemo promatrati kao virtualne mašine koje nam omogućuju da pokrenemo kod originalno napisan za emulirano sklopovlje. Upravo kao i kod virtualnih mašina, čest način implementacije emulatora je interpreter koji konzumira kod i odmah ga izvršava, bez kompajlacije.



Sl. 1 Atari VCS "wood veneer" verzija iz 1980

II. ATARI (VCS) 2600

A. Povijest

Atari 2600 (poznat kao Atari VCS prije 1983) je igraća konzola razvijena i predstavljena tržištu 1977. godine od strane tvrtke Atari. Dostavljala se zajedno sa dva joystick kontrolera, dva spojena paddle kontrolera te igricom, u početku su to bili „Combat” i kulturni „Pac-Man”.

U trenutku izlaska na tržište konzola je koštala 199 američkih dolara (804\$ uz inflaciju danas). Unutar Atari tvrtke konzola je imala kodno ime Stella no dobila je tržišni naziv VCS kao konkurencijski odgovor na Fairchild-ovu konzolu VES. Čim su doznali da je navik koji je Atari odabrao Fairchild mijenja ime svoje konzole u Channel F. Time počinje natjecanje dviju tvrtki u snižavanju cijena što je za rezultat imalo da su mnoge kompanije koje su radile klonove prestale poslovati. Samo u 1977 Atari je prodao više od 250000 primjeraka konzole.

U 1978 prodano je samo 550000 jedinica konzole od proizvedenih 800000 no već 1979 prodano je čak milijun primjeraka. Nove vrste igrica i programerske tehnike maksimalnog iskorištavanja sklopovlja konzole dovele su do tog naglog rasta.

Zbog nezadovoljstva onosom kompanije prema razvijateljima grupa programera je napustila Atari[3] i osnovala tvrtku Activision 1979. Activision je prva samostalna razvijateljska tvrtka igrica na svijetu.

Od 1977 do ukidanja 1992 Atari 2600 prodan je u više od 30 milijuna komada.



Sl. 2 Pac-Man

U 1977 kada je konzola izbačena na tržište 9 igrica je bilo razvijeno za prodaju uz novo predstavljenu konzolu. Popis izdanih igrica broji nešto više od 560 igrica, podjeljenje po kategorijama na one koje su razvili Atari i Sears, te one koje su razvile treće osobe (primjerice Activision).

Najprodavanija igrica za VCS je bila PacMan koji je ukupno prodan u 7 milijuna primjeraka.

B. Slopovlje

U srcu VCS-a nalazi se 8bitni MOS Technology 6507 processor, verzija poznatog procesora 6502, koji radi na 1.19Mhz. 6507 za razliku od 6502 ima samo 13 adresnih pinova, nadalje odabравši jeftini format kartiča sa samo 12 adresnih linija maksimalna adresibilna memorija iznosi samo 4kb[4]. Programeri kasnije koriste bankswitching metodu kako bi adresirali čak do 32kb.

Kartice Read Only Memorije (ROM cartiges) su služile za prijenost software-a no mogle su služiti i kao proširenje sklopovlja u nekim slučajevima.

Uz procesor najbitniji dio konzole je tkz. Television Interface Adaptor (TIA). TIA je izrazito složen čip koji generira sliku, zvučne efekte i čita ulaze kontrolera. Na sam dizajn je utjecao pokušaj minimizacije količine ram-a potrebnog da se upravlja prikazom slike pošto su cijene RAMa bile eksremne, u rangu tisuća dolara po megabajtu[7]. 6507 komunicira sa TIAom pišući a ponekad i čitajući TIA registre.

Address range	Function
\$0000-\$007f	TIA registers
\$0080-\$00ff	RAM
\$0200-\$02ff	RIOT registers
\$1000-\$1fff	ROM

Rane video igre općenito su koristile dvije različite vrste grafike, "igračice" koje su kontrolirali igrač ili računalo (danas poznati kao sprites), i "playfield", ili pozadinska grafika obično iscrtana pod igračima. Igrači se neprestano kreću, dok su igrališta općenito statična i mijenjaju se samo na točno definiranim vremenima (kako bi se izbjegli artefakti nastali crtanjem u pogrešno vrijeme,

nastali ograničenjem sklopovlja). Konvencionalni način crtanja igrališta je korištenje bitmap-a koji se drži u framebufferu. Svako mjesto memorije u framebufferu ima vrijednost koja opisuje piksele na zaslonu. Sklopovlje zaslona očitava vrijednosti iz buffera i koristi ih za generiranje analognog signala za prikaz na CRT monitoru. Mapiranje memorije na mjesta zaslona ili piksela često je ograničeno sklopovljem zaslona.

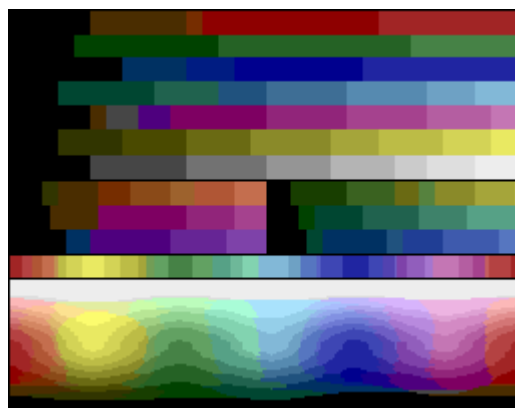
Tipični prikaz od 320 do 200 piksela s čak samo jednim bitom po pikselu zahtijevao je 8000 bajta memorije za pohranjivanje framebuffera. [6] To ne bi bilo prikladno za platformu koja je imala za cilj koštati samo nekoliko stotina dolara. Čak i dramatična smanjenja rezolucije ne bi smanjila troškove memorije na razumne razine. Umjesto toga, dizajnerski tim odlučio je potpuno ukloniti memorijski framebuffer.

TIA se razlikuje od uobičajenog pristupa framebufferu po tome što se slika na zaslonu sastoji od manipulacije pet pokretnih grafičkih objekata (2 igrača, 2 projektila i 1 lopta) i statičkog playfield objekta. To se sve generira na svakoj liniji skeniranja iz registara, za razliku od tehnike koja se koristi u modelu mapiranom preko framebuffera, no program je potrebno ažurirati na svakoj liniji skeniranja.

Vodoravna rezolucija nije jednolična, jer njezina veličina ovisi o određenom grafičkom objektu. Najmanjoj jedinici piksela odgovara 1 takt boja ciklusa čipa, od čega na liniji ima 160 vidljivih. Potrebno je 76 ciklusa kako bi se iscrtala linija, sto znaci da bi se iscrtao ekran treba proći 160 x 70 ciklusa.

Objekt Playfield sastoji se od registra od dva i pol bajta, koji se može reflektirati simetrično ili kopirati kao na desnoj polovici zaslona za ukupno 40 bita (svaki bit je širok 4 ciklusa boja). Boja je iscrtana ako je bit 1 ili ako je 0 odabrana je iz unaprijed definirane palete do 128 boja i drži se u drugim registrima.

Na konvencionalnoj televiziji u boji NTSC, maksimalne rezolucije općenito su između 256 i 320 piksela po liniji, tj 192 do 240 pixela na zaslonu[5]. Emulator priložen radu radi na sličan način, tako da dekodira vrijednosti iz video morije koja se nalazi na lokacijama 81ff do f9ff. VCS je koristio NTSC ili PAL paletu, ovisno o vrsti televiyijskog formata. NTSC paleta ima 128 boja dok PAL drži samo 104 boje[7].



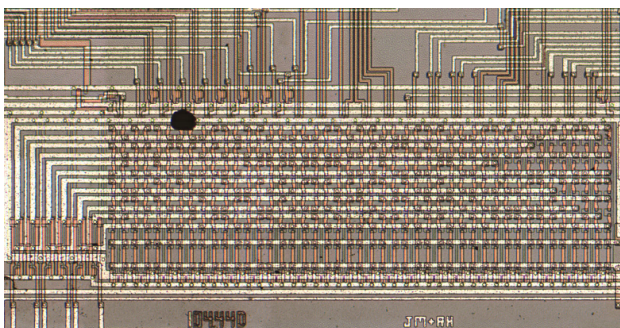
Sl. 3 NTSC paleta boja

Budući da su registri držali podatke samo za jednu liniju zaslona, iscrtavanje cijelog zaslona zahtijevalo je ažuriranje registara "u letu", proces poznat kao "Racing the beam".

Za početak procesa, program koji se pokreće na CPU-u učitava u TIA registre podatke potrebne za crtanje prve linije zaslona. TIA tada čeka da televizor bude spreman za crtanje linije i pročita registre kako bi stvorila signal za tu liniju. Tijekom razdoblja horizontalnog pražnjenja između linija, CPU bi brzo promijenio TIA-in registar na podatke potrebne za sljedeći redak. Taj je proces nastavio niz zaslon.

Taj proces je otežala činjenica da je MOS 6507 korišten u VCSu je oklaštrena verzija 6502 i ne podržavala hardverske prekide. Općenito, zaslon bi generirao prekid kada je završio crtanje linije i pripremao se za sljedeću. Prekid bi pokrenuo kod koji je potreban za ažuriranje zaslona, a zatim se vratilo na "glavni" program. Zbog toga su programeri trebali pažljivo pratiti svoje programe kako bi se izvodili u točnom broju ciklusa potrebnih za različite događaje povezane sa zaslonom. Program koji je potreban za brojanje ciklusa programa za iscrtavanje nazivao se "kernel" i bio je izuzetno složen u usporedbi s drugim sustavima.

TIA može proizvesti različite tipove impulsa i zvukova iz svojih dvaju oscilatora (ili kanala) AUD0 i AUD1. Svaki oscilator ima 5-bitni razdjelnik frekvencije i 4-bitni audio kontrolni registar koji manipulira valnim oblikom. Tu je i 4-bitni registar kontrole glasnoće po kanalu. Frekvencije se generiraju uzimanjem 30 kHz i dijeljenjem s 5-bitnom vrijednošću. Rezultat je jeftini frekvencijski razdjelnik sposoban za netonirane note i neparne ugođene frekvencije. TIA nije glazbeni čip osim ako skladatelj radi unutar granica učestalosti ili modulira između dvije netonirane frekvencije kako bi stvorio vibrato ugođenu notu. Registar audio kontrole generira i upravlja impulsnim valom radi stvaranja složenih impulsa ili zvuka.



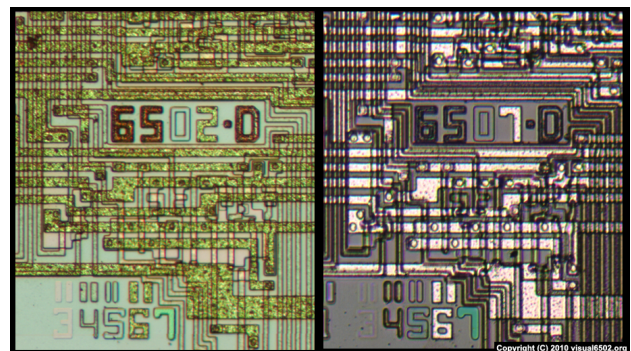
SI.5 Makro slika silicija dekodera TIA čipa

III. MOS6507

6507 je 8-bitni mikroprocesor tvrtke MOS Technology, Inc. 6507 je u biti 6502 čip u manjem, jeftinijem 28-pinskom paketu. 6502 je nastao kao rezultat napora četvorice inženjera koji su napustili motorolu kako bi izradili mikroprocesor pristupačne cijene. Uspjeli su u svojoj namjeri i predstavili 8 bitni procesor za samo 25\$ kada je najjeftiniji procesor koštao 179\$. No Atari je

trebao procesor od 12\$ dolara. MOS je trebao hitno klijenta pa je skinuo adresne linije i prekide te smjetio čip u 28-pinski paket. Takav osakaćeni procesor mogao je adresirati samo 8K memorije, koja je za aplikacije u to vrijeme (1975) bila prihvatljiva i nije bila pretjerano restriktivna. Cijela obitelj CPU-a 65xx izvorno je bila zamišljena kao linija vrlo jeftinih mikroprocesora za male ugrađene sustave, a ne za računala opće namjene i svakako ne za interaktivna osobnim računalima (koja još uvijek nisu niti postojala).

Čipovi 6507 i 6502 koriste iste temeljne silikonske slojeve i razlikuju se samo u završnom sloju metalizacije. To povezuje linije prekida s njihovom neaktivnom razinom, tako da nisu ranjive na stvaranje lažnih prekida iz elektromagnetičke buke. Prve tri znamenke identifikatora čipa dio su silicijskih slojeva, a posljednja znamenka nalazi se u metalizacijskom sloju. Mikrofotografija 6502 i 6507 pokazuje tu razliku.



Od 28 pinova u 6507, 13 ih je za adresiranje te 8 za podatke. Sedam preostalih pinova koristi se za napajanje, CPU timer clock, resetiranje CPU-a, za traženje stanja čekanja sabirnice (RDY pin) i za čitanje/zapisivanje naredbi memoriji (ili MMIO uređaja) iz CPU-a. Ne postoji IRQ (prekidni) ili NMI pin na procesoru.

RDY pin nije uključen u sve ostale 28-pinske srezane verzije 6502[9]. Unutar Atari 2600, RDY se koristi za sinkronizaciju CPU tajmera na televizijski video okvir. Ova je funkcija neophodna za tkz. "Racing the beam" koju koristi 6502 i Atari adapter televizijskog sučelja televizora za generiranje televizijskog video signala. Kao odgovor na određeni pristup registru, TIA će naložiti RDY da produži taj ciklus sabirnice do kraja trenutne linije skeniranja videozapisa.

A. Emulacija i arhitektura 650x procesora

Emulacija se može podijeliti na dijelove. Emulacija CPUa vrlo je različita od emulacije zvuka ili grafičkog sustava. CPU prenosi bajtove iz memorije i izvršava funkcije ovisno o tome što to oni znače. Te bajtove nazivamo operacijski kodovi. Uz operacijske kodove favorimo i o operandima na kojima operacijski kodovi izvršavaju radnje. Grafički uređaj koristi video memoriju i naredbe koje šalje CPU kako bi generirao sliku na televizoru ili na zaslonu monitora. Zvučni uređaj koristi podatke i naredbe za generaciju električnog signala koji se šalje u zvučnike kako bi stvorio zvuk. Algoritmi

uključeni u grafičku generacija ili generaciju zvuka nemaju nikakve veze sa algoritmima koji su uključeni u CPU emulaciju što izrazito otežava proces emulacije. Upravo iz tog razloga emulator priložen radu vjerno emulira samo CPU dok se način rada RAMa i video memorije razlikuje od originalnog Atarija 2600.

Glavni dio emulatora je petlja jer način rada računala može se promatrati kao ponavljajući zadatak. CPU radi u bilo kojem trenutku dohvaćanje, dekodiranje i izvršavanje petlje s uputama. Iako u stvarnosti svi djelovi emuliranog uređaja rade paralelno zbog težine paralelne emulacije i platforme na kojoj emuliramo sklopovlje (x64) neću ulaziti u problem implementacije paralelnih ili dretvenih emulata.

```
U suštini glavna petlja emulatora izgleda kao slijedeće:
while(true) {
    // dohvati operacijski kod iz memorije
    opcode = Read(pc++);

    // dekodiraj ga
    instr = InstrTable[opcode];

    // izvrši instrukciju
    Exec(instr);
}
```

Samo dekodiranje instrukcija se može napraviti na više načina od kojih je jako čest veliki switch case sa svim operacijama, većina modernih kompajlera će pretvoriti switch case u jump table kako bi se postigle bolje performanse no jump table se također može implementirati direktno u kodu emulatora.

Jedan od problema kod implementacije i kasnije programiranja procesora je endian, 6507(2) je little endian procesor što znači da će kod spremanja više bitnog broja u memoriju spremati manje važni bit prvi te onda više važni bit. Prikaz u tablici ispod za zadani asemblerski kod:

```
LDA $1234
STA $1000
```

Big endian		Little endian	
\$1000	\$12	\$1000	\$34
\$1000	\$34	\$1000	\$12

Pošto su CPU jezgre teške za isprogramirati iz razloga da je potrebno jako puno vremena za implementaciju svih instrukcija i njihovo testiranje, za svoj emulator sam iskoristio i prilagodio 6502 jezgru otvorenog koda.

Za početak, da bi emulirali jezgru procesora potrebne su nam određene varijable za ugrađene registre. 6502 ima 6 ugrađenih registara: programsko brojilo, stack pointer, statusni registar, X,Y registre opće namjene te akumulator koji je srce sustava posto odrađuje svu aritmetiku i logiku. Statusni registar se sastoji od jednobitnih zastavica (zero, sign, overflow, break, decimal, interrupt i carry zastavica).

Nadalje moramo simulirati resetiranje procesora. Obično prilikom paljenja sustav drži procesor u stanju resetiranja (Power on reset) kako bi se natjeralo interne registre da poprime poznato stanje.

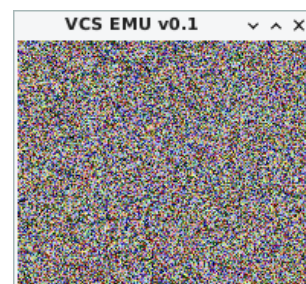
```
void mos6502::Reset() {
    A = 0x00;
    Y = 0x00;
    X = 0x00;
    sp = 0xFD;
    status |= CONSTANT;
    cycles = 6; // reset traje 6 ciklusa
    return;
}
```

Nadalje implementiramo instrukcije samog procesora. 6402 ima 13 načina adresiranja. Ovisno o načinu adresiranja assembler generira drukčiji opcode za instrukcijsku riječ. Primjerice operacija LDA \$01 biti će asemblirana u heksadecimalni zapis a9 01 dok će operacija LDA \$0101 biti asemblirana u ad 01 01.

```
Jump table:
instr.addr = &mos6502::Addr_IMM;
instr.code = &mos6502::Op_LDA;
InstrTable[0xA9] = instr;
instr.addr = &mos6502::Addr_ABS;
instr.code = &mos6502::Op_LDA;
InstrTable[0xAD] = instr;
```

```
implementacija:
void mos6502::Op_LDA(uint16_t src) {
    uint8_t m = Read(src);
    SET_NEGATIVE(m & 0x80);
    SET_ZERO(!m);
    A = m;
}
```

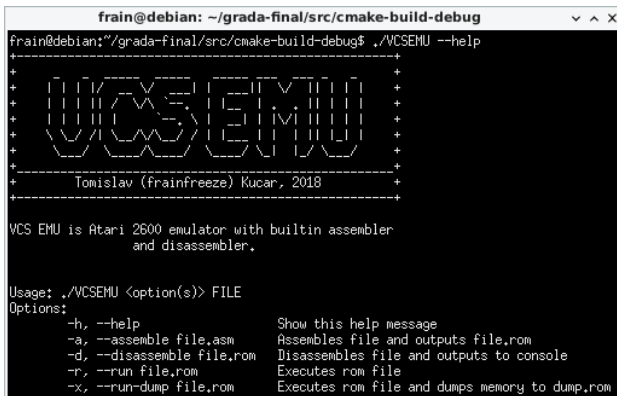
Nakon što je jezgra implementirana dodjelimo memoriju emuliranom procesoru pomoću arraya. Različite memorijske lokacije imaju različita svojstva, mogu biti read only, write only, te R/W. Nadalje dio memorije služi kao video/IO/audio memorija itd. Originalni layout memorije atarija je naveden predhodno u tablici. Za potrebe testiranja kompleksnijih programa priloženi emulator ima nešto više od 8kb memorije. Memorijska lokacija \$752d (29997) sadrži novi random byte nakon svake instrukcije. Memorijske lokacije od \$752F (29999) do \$81fe (33278) rezervirane su za zvuk iako taj dio emulatora još nije implementiran. I na poslijetku memorijske lokacije \$81ff (33279) do \$f9ff (63999) mapiraju piksele na ekranu (192x160 piksela). Kako bi se isctala boja na ekranu potrebno je učitati broj boje (0-127) na memorijsku lokaciju piksela. NTSC color dekođer potom broj pretvara u rgb vrijednost i isctava ju.



Sl. 6 nasumične vrijednosti upisane u video memoriju

IV. DODATAK: UPOTREBA EMULATORA

Emulator je kompajliran i testiran na Debianu 9 sa cmake 3.9, make 4.1, gcc 6.3, python 3 te sfml 2.0. Za kompajliranje je potrebno u root direktoriju koda pokrenuti „cmake.” te nakon toga „make”. Pokretanjem dobivene izvršne datoteke dobivaju se instrukcije o korištenju.



```
frain@debian: ~/grada-final/src/cmake-build-debug
frain@debian:~/grada-final/src/cmake-build-debug$ ./VCSEMU --help
+-----+
+ VCSEMU +
+-----+
+ Tomislav (frainfreeze) Kucar, 2018 +
+-----+
VCSEMU is Atari 2600 emulator with builtin assembler
and disassembler.

Usage: ./VCSEMU <option(s)> FILE
Options:
-h, --help Show this help message
-a, --assemble file.asm Assembles file and outputs file.rom
-d, --disassemble file.rom Disassembles file and outputs to console
-r, --run file.rom Executes rom file
-x, --run-dump file.rom Executes rom file and dumps memory to dump.rom
```

U emulator je ugrađen disassembler generiran python skriptom iz cvs datoteke instrukcija procesora stoga je potrebno imati obje datoteke prisutne te ne modificirati direktno disassembler.h datoteku.

V. ZAKLJUČAK

Atari 2600 i MOS 6502 bili su važne točke u počesti računalstva, Atari kao preteča današnje industrije igračih konzola i igrice a MOS 6502 jeftinijih i pristupačnih uređaja.

ZAHVALE

Ovim putem zahvaljujem Karlu Kegljeviću koji je napisao NTSC dekodier i manje isječke pzhon koda pomoću kojih sam generirao asembler za testiranje jezgre procesora.

LITERATURA I IZVORI

- [1] <https://en.wikipedia.org/wiki/Emulator>
- [2] https://en.wikipedia.org/wiki/Virtual_machine
- [3] http://www.gamasutra.com/view/feature/132160/atari_the_golden_years__a_.php?page=8
- [4] Racing the beam - Atari VCS, MIT Press, str. 26
- [5] <http://atarihq.com/danb/files/stella.pdf>
- [6] McCallum, J.C., „Memory prices (1957-2012)”
- [7] https://en.wikipedia.org/wiki/List_of_video_game_console_palettes
- [8] Chris Crawford, "ANTIC and the display list"
- [9] <https://marc.info/?l=classiccmp&m=110668039327589&w=2>