# AES I ARM PROCESORI

*Danijela* D. Protić
Generalštab Vojske Srbije, Uprava za telekomunikacije i informatiku (J-6), Centar za primenjenu matematiku i elektroniku, Beograd

DOI: 10.5937/vojtehg61-3256

OBLAST: elektronika
VRSTA ČLANKA: stručni članak

*Sažetak:*

*Potreba za zaštitom informacija dovodi do velikih problema u izradi prenosivih uređaja kojima su limitirani snaga, memorija i energija. Ukoliko se takvim uređajima dodaju koprocesori, koji treba da obavljaju funkcije kriptozaštite, njihove se dimenzije povećavaju, pojavljuje se nefleksibilnost pa cena uređaja raste i do nekoliko puta. Na drugoj strani, algoritmi za zaštitu podataka su često memorijski zahtevni, a zbog velikog broja operacija koje je potrebno izvršavati u procesima šifrovanja i dešifrovanja, koprocesori često uspore rad osnovnog procesora. Za jedan od standarda za kriptozaštitu, AES, NIST je prihvatio Rijndaelov blokovski algoritam sa dužinom ulaznog i izlaznog bloka od 128 b, i dužinama šifarskog ključa od 128 b, 192 b i 256 b. Zbog karakteristika male potrošnje, 32-bitske arhitekture i brzog izvršavanja instrukcija, ARM procesori mogu da realizuju kriptozaštitu podataka, između ostalog i AES-om, a da ne opterete glavne procese u sistemima u kojima se koriste. Tehnologija ARM-a zaštićena je kao intelektualna svojina, pa je veliki broj proizvođača koristi za razvoj sopstvenih proizvoda, što je rezultovalo činjenicom da je u svetu proizvedeno preko 2 milijarde čipova koji su bazirani na ovoj tehnologiji. U radu su prikazane mogućnosti za poboljšanja u izvršenju algoritma AES primenom najnovijih verzija ARM procesora.*

Ključne reči*: procesori; kriptozaštita; AES.*

## Uvod

Rastuća potreba za zaštićenom komunikacijom i obradom podataka zahteva brze sisteme za izvršavanje kriptografskih algoritama. Sa sve većim korišćenjem bežičnih uređaja u svakodnevnom životu, kriptozaštita osetljivih informacija postaje krucijalna. Ovaj zadatak postaje veoma zahtevan ukoliko se koriste prenosivi uređaji kojima su limitirani

e-mail: adanijela@ptt.rs

snaga, memorija i potrošnja energije. Kada se, tradicionalnim pristupom, zahtevni kriptografski parametri ugrađuju u koprocesore, oni često postaju nefleksibilni, a njihova veličina povećava dimenzije uređaja uz koji se koriste, što utiče i na rast cene osnovnog proizvoda. Alternativa, odnosno novi pristup, je integracija jednostavnih instrukcija u glavni procesor u uređaju, s ciljem da se ubrza rad i podrže kriptografske operacije. Najnovija istraživanja ukazuju na koristi takve ekstenzije u kriptografiji s javnim ključevima (Tillich, Großschadl, 2006).

Devedesetih godina prošlog veka je Acorn kompanija[1], iz Engleske, razvila prvi RISC[2] bazirani mikroprocesor za komercijalnu namenu, visokih performansi i male potrošnje (~100mW) – ARM (Klami, et al., 2009). Za razliku od drugih proizvođača, kompanija ARM Holdings nije licencirala svoje proizvode bazirane na ARM procesorima, nego ARM tehnologiju kao intelektualnu svojinu. Iz tog razloga danas postoji nekoliko desetina kompanija koje su razvile sopstvene proizvode bazirane na ARM dizajnu (Intel, Samsung, Texas Instruments), sa ukupno oko dve milijarde proizvedenih čipova (ARM press release, 2011). ARM procesori su pogodni za ugradnju u prenosive sisteme sa baterijskim napajanjem, pa ih veliki broj proizvođača ugrađuje u smart kartice, prenosive uređaje, potrošačku elektroniku, itd. U svakom slučaju, ARM arhitektura je dizajnirana tako da izuzetno dobro podržava 32-bitske embedded[3] sisteme (Osvik, et al., 2010, pp.1-20). Embedded sistemi se mogu koristiti u raznovrsnim uređajima, od malih senzora na proizvodnim trakama, do kontrolnih sistema koje koristi npr. NASA[4] (Sloss, et al., 2004). Svaki od ovih sistema sadrži i hardverske i softverske komponente koje su izabrane tako da maksimizuju efikasnost. Elemente jednog uređaja koji je ARM baziran čine: ARM procesor, kontroleri, periferije i magistrale.

Nacionalni institut za standarde i tehnologiju[5] (NIST) Sjedinjenih američkih država (SAD) je 2001. godine, objavio rezultate konkursa kojim je stari, standardizovani algoritam za zaštitu podataka DES (Data Encryption Standard) unapređen i zamenjen novim. NIST je insistirao da na konkurs budu dostavljani simetrični, blokovski algoritmi. U simetričnim algoritmima ključevi za enkripciju i dekripciju su identični, za razliku od asimetričnih algoritama, odnosno algoritama sa javnim ključem u kojima se ključ za enkripciju razlikuje od ključa za dekripciju (Kuljanski, 2010, pp.65-77). Rijndaelovo rešenje je izabrano kao najbolje od 15 ponuđenih algoritama, zbog dobre kombinacije karakteristika, bezbednosti, efikasnosti, lake implemen-

[1] Kompanije Acorn Computers, Apple Computer i VLSI Technology danas su jedna kompanija – Advanced RISC Machines (ARM Holdings plc).
[2] RISC – Reduced Instruction Set Computing – smanjeni broj instrukcija u obradi podataka.
[3] embedded sistem – mikroračunarski sistem kod kojih su sva kola ugrađena (embedded, engl.) uz relativno mali broj pratećih integrisanih kola
[4] NASA – National Aeronautics and Space Administration
[5] NIST – National Institute of Standards and Technolgy

tacije i fleksibilnosti. U novembru 2001. godine NIST je standardizovao AES (Advanced Encryption Standard) u publikaciji FIPS[6]-197, u kategoriji standarda za računarsku bezbednost, a standard je stupio na snagu u maju 2002. godine. U objašnjenju NIST-a stoji da Rijndaelov simetrični, blokovski algoritam može biti korišćen za zaštitu elektronskih podataka „tako da se enkripcijom podaci menjaju u nerazumljiv oblik, a dekripcijom im se vraća smisao". AES se prihvata kao standardan ako se za enkripciju koriste kriptografski ključevi dužine *isključivo* 128 b, 192 b ili 256 b[7], a dužina ulaznog odn. izlaznog bloka je *fiksna* i iznosi 128 b. AES može biti implementiran u softver, firmware, hardver ili njihovu kombinaciju a, u zavisnosti od aplikacija, okruženja, primenjene tehnologije, itd., NIST preporučuje da se algoritam koristiti uz druga rešenja koja su publikovana FIPS-om (FIPS, 2001). Na osnovu rezultata komparativne analize između NIST-ovih finalista, Rijndaelov algoritam je pokazao najbolje karakteristike u poređenju sa MARS, RC6, Serpent i Twofish algoritmima. Svaki od algoritama ima izvanredne karakteristike, obzirom na fleksibilnost, mogućnost implementacije, bezbednost i efikasnost, kao što je prethodno pomenuto u tekstu. Sa stanovišta bezbednosti najsigurniji je Serpent, za njim slede MARS i Twofish, a plasman AES-a odredila je dužina ključa (za 128 b algoritam je manje efikasan), dok poslednje mesto zauzima RC6. Algoritmi su ispitivani na JAVA i C platformama, ali se JAVA pokazala oko tri puta sporija (Sternbenz, Lipp, 2002). Testovi na hardverskim komponentama nisu pokazali iste rezultate kao za softver. Serpent, koji je u softveru najsporiji, implementiran hardver postaje najbrži, Rijndael takođe garantuje veliku brzinu, dok se Twofish i RC6 mogu implementirati kompaktno na maloj površini, ali rade sporije. MARS je na poslednjem mestu po brzini. Ipak, rezultati za svaki od algoritama, implementiranih hardverski, ukazuju na mnogo veću brzinu od softverske implementacije, sa faktorom 4–20 (Dandalis, 2000). Takođe, Rijndael je pokazao najbolje karakteristike za ugradnju u smart kartice (Sano, et al., 2002).

AES je dizajniran tako da daje odlične karakteristike na mnogo različitih platformi, što posebno važi za 32-bitske platforme. Matrica stanja i šifarski ključ mogu biti predstavljeni kao celobrojne vrednosti 32-bitskih promenljivih, dodavanje ključa je pravolinijska xor operacija sa matricom stanja, implementacija S-box supstitucije je mnogo efikasnija zbog korišćenja tabele dužine 256 B, što omogućuje brzu (bajtovsku) permutaciju. ARM takođe omogućuje najbolji način za optimizaciju u funkcijama MixColumns() i InvMixColumns(), koje su detaljno opisane u tekstu FIPS-197 (Darnall, Kuhlman, 2002).

Rad je organizovan na sledeći način: nakon uvoda sledi opis AES-a po NIST-ovom standardu, sa prikazom pseudo-kodova algoritma. Treće

[6] FIPS – Federal Information Processing Standards Publication
[7] Na osnovu dužine ključa (u bitima) AES ima nazive AES-128, AES-192 i AES-256

poglavlje prikazuje karakteristike ARM procesora i daje šematski prikaz razvojne ploče za ARM926EJ-S procesor, sa oznakom PB926EJ-S. U četvrtom poglavlju opisani su procesi u izvršenju AES algoritama koje ubrzava ARM procesor, a poslednje poglavlje je zaključak rada.

# AES

AES je standardizovan Rijndaelov iterativni, blokovski algoritam, čiji su ulaz i izlaz blokovi od 128 b. Šifarski ključ *K* je varijabilne dužine od128 b, 192 b ili 256 b (Daemen, Rijmen, 1999; Daemen, Rijmen, 2000) dok, po standardu NIST, druge vrednosti nisu dozvoljene (Mladenović, et al., 2006). Šifrovanje se izvodi transformacijom matrice stanja *s* (state) koja, inicijalno, dobija vrednost otvorenog teksta sa ulaza (ulazni podaci koji nisu šifrovani), a organizovana je kao matrica od 4 x 4 bajta. Broj rundi ($N_r$) određuje dužina ključa *K* ($N_k$ = 4, 6, 8, prikazano u rečima (w) od 32b). Vrednosti dužine ključa i ulaznog/izlaznog bloka ($N_b$) koji su takođe reči od 32b, i broj rundi za AES algoritme prikazan je u Tabeli 1.

*Tabela 1*
*Table 1*

Dužina ključa i bloka, i broj rundi za AES
Key and block lenght, and the number of rounds for AES

| | $N_k$ (w) | $N_b$ (w) | $N_r$ |
|---|---|---|---|
| AES-128 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

Pre šifrovanja potrebno je izvesti ekspanziju šifarskog ključa, pri čemu se dobijaju ključevi rundi. Tom prilikom se generiše $N_b$*($N_r$+1) 32-bitskih reči koje možemo označiti sa *w*[*i*], pri čemu je $0 \le i < N_b$*($N_r$+1). Ekspanziju šifarskog ključa određuje sledeće:

(1) SubWord() funkcija uzima 4 B ulazne reči i primenjuje S-box transformaciju na svaki bajt čime se generiše 32-bitska reč,

(2) RotWord() rotira reč za jedan bajt u levo,

(3) Rcon[*i*] su konstante.

Prvih $N_k$ reči ekspandovanog ključa čini niz bitova šifarskog ključa, koji su predstavljeni kao 32-bitske reči. Svaka reč *w*[*i*] dobija se xor-ovanjem prethodne reči *w*[*i*-1] sa reči koja se nalazi na $N_k$ pozicija unazad, odnosno *w*[*i*-$N_k$]. Za reči na pozicijama koje su proizvodi $N_k$, *w*[*i*] se dobija tako što se *w*[*i*-1] prvo rotira za jedan bajt u levo (funkcija RotWord()), zatim xor-uje sa Rcon[*i*/$N_k$], na tako dobijenu reč se primeni funkcija SubWord(), i na kraju se ovaj rezultat xor-uje sa *w*[*i*-$N_k$].

Pseudo-kod algoritma za ekspanziju ključa: KeyExpansion():

Key Expansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)

```
begin
word temp
i = 0
while (i ≤ Nk)
w[i] = word(key[4*i], key[4*i + 1], key[4*i + 2], key[4*i + 3] )
i = i+1
end while
i = Nk
while (i < Nb*(Nr + 1))
temp = w[i-1]
if (i mod Nk = 0)
temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
else if (Nk > 6 and i mod Nk = 4)
temp = SubWord(temp)
end if
w[i] = w[i - Nk] xor temp
i = i + 1
end while
end
```

Za šifrovanje, *s* se prvo inicijalizuje ulaznim blokom, pri čemu se bajtovi upisuju po kolonama, zatim se prvi ključ runde (*w*[0] ... *w*[3]) xor-uje sa *s*, nakon čega se ona modifikuje $N_r$ puta, s tim što se finalna runda neznatno razlikuje od prethodnih. Za šifrovanje, AES algoritam koristi četiri (bajtovski orijentisane) transformacije: supstitucija korišćenjem supstitucione tabele - SubBytes(), pomeranje vrsta matrice stanja za različite ofsete – ShiftRows(), transformacija u okviru jedne kolone matrice stanja – MixColumns() i dodavanje ključa runde matrici stanja - AddRoundKey(). U finalnoj rundi nema transformacije MixColumns(). Algoritam se, pored finalne, izvodi $N_r$ -1 rundi, na sledeći način:

(1) SubBytes(): menja svaki ulazni bajt matrice stanja sa vrednošću matrice S-box, čije su vrednosti prikazane u standardu,

(2) ShiftRows(): *i* –ta vrsta matrice stanja ciklično se pomera u levo za *i* bajtova, pri čemu je $0 < i < 3$,

(3) MixColumns(): množi svaku kolonu, koja se posmatra kao polinom stepena manjeg od četiri, sa koeficijentima iz GF($2^8$), fiksnim polinomom po modulu drugog fiksnog polinoma,

(4) AddRoundKey(): xor-uje ključ runde *r* sa matricom stanja *s*.[8]

Pseudo-kod Cypher() prikazuje korake šifrovanja za AES:

```
Cypher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
```

[8] Detaljna objašnjenja svih transformacija i vrednosti matrica S-box i InvS-box data su u NIST-ovom dokumentu FIPS-197, pa ti podaci nisu navedeni u tekstu.

```
state = in
AddRoundKey(state, w[0, Nb])
for round = 1 step 1 to Nr-1
SubBytes(state)
ShiftRows(state)
MixColumns(state)
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
end for
SubBytes(state)
ShiftRows(state)
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
out = state
end
```

Svaka transformacija kod šifrovanja ima svoju inverznu transformaciju: AddRoundKey() (je, istovremeno, i sopstvena inverzija), InvMixColumns(), InvShiftRows() i InvSubBytes(). Koraci u dešifrovanju se izvršavaju u smeru suprotnom od izvršavanja koraka u šifrovanju.

Pseudo-kod algoritma za dešifrovanje InvCypher() ima sledeći oblik:

```
InvCypher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
byte state[4, Nb]
state = in
AddRoundKey(state, w[Nr*Nb, (Nr+1)* Nb])
for round = N-1 step -1 downto 1
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
InvMixColumns(state)
end for
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[0, Nb-1])
out = state
end
```

## ARM procesori

Izbor pogodnog procesora koji se integriše u embedded sistem zavisi od domena aplikacije tog sistema. Aplikacije mogu biti aritmetički ili upravljački intenzivne, što može da bude problem u uređajima za masovnu upotrebu (visoka cena, potrošnja energije). Imajući to u vidu, da bi bio ostvaren efikasan dizajn, neophodno je koristiti različite klase procesora,

čije se arhitekture međusobno razlikuju, kao što su npr. mikrokontroleri, digitalni signal procesori (DSP), aplikaciono specifični procesori, multimedija procesori ili RISC procesori (iz kojih je razvijena ARM arhitektura).

RISC dizajn namenjen je za primenu jednostavnih instrukcija koje se izvršavaju u jednom ciklusu, pri najvećoj brzini koju daje taktni sat (clock). Osnova dizajna je redukovanje kompleksnosti hardverskih instrukcija, jer su fleksibilnost i mogućnosti softvera veće od hardvera. RISC procesor zato ima veće zahteve za kompajler, (kod CISC[9] dizajna veće je oslanjanje na hardver). Za RISC dizajn važe četiri karakteristike: (1) procesori imaju ograničeni skup instrukcija, koje se izvršavaju velikom brzinom, a sve instrukcije su istog obima i za izvršenje zahtevaju isti broj taktova, (2) izvršenje instrukcija se može učiniti protočnim (pipeline), čime se postiže veća propusnost u odnosu na mehanizam sekvencijalnog izvršavanja, (3) RISC je load-store[10] arhitektura, a procesor radi sa podacima koji se nalaze u registrima, (4) postoji veliki broj registara opšte namene, čijim se korišćenjem smanjuje broj obraćanja memoriji (u svakom registru može se nalaziti podatak ili adresa). Format podataka je fiksan (32 b) a format instrukcija tro-adresni. Problem RISC-a je uslovno grananje (Sloss, et al., 2004).

ARM procesori su razvijeni na osnovu RISC procesora a ono što ih razlikuje je namena – da budu deo većeg, embedded sistema. Ključno u ovakvom sistemu nije snaga samog procesora, već efikasnost. Osnovno je postići maksimalne performanse sistema i minimalnu potrošnju energije. ARM instrukcije su 32-bitske i, u strukturi ARM7, izvode se u tri koraka: (1) prijem, (2) dekodovanje i (3) izvršenje, dok kod procesora ARM9 slede još dva koraka: (4) pristup memoriji i (5) upis (Sloss, et al., 2004). ARM instrukcija služi za obradu, skladištenje ili prenos podataka, što zahteva (1) load-store arhitekturu, (2) tro-adresne naredbe za obradu podataka, (3) uslovno izvršenje svake od naredbi, (4) uključenje (jakih) višestrukih registara za load-store operacije, (5) mogućnost izvođenja opštih operacija pomeranja i opštih aritmetičko-logičkih operacija u jednoj instrukciji, u jednom taktu, (6) proširenje na otvoreni skup naredbi preko skupa naredbi koprocesora, što uključuje i dodavanje novih registara i tipova podataka programerskom modelu i (7) mogućnost 16-bitskih Thumb instrukcija (ARM Limited, 2001). Instrukcije ARM procesora se, od instrukcija za RISC procesor, razlikuju u pet karakteristika:

– varijabilni broj ciklusa za izvršenje određenih naredbi (npr. load-store-multiple kod transfera višestrukih registara),

– in-line barrel shifter, za preprocesiranje jednog od ulaznih registara pre nego on bude korišćen za instrukciju (proširuje mogućnosti instrukcija poboljšavajući im osnovne karakteristike, utiče na gustinu koda),

[9] CISC – Complex instruction set computer
[10] load-store (engl.) – učitaj-sačuvaj (podatke)

– Thumb (16-bitske) instrukcije koje obezeđuju da ARM izvršava ili 16- ili 32-bitske instrukcije, koje mogu da ubrzaju rad do 30% u odnosu na standardne 32-bitske instrukcije fiksne dužine,

– uslovnim grananjem redukuju se funkcje grananja i povećava gustina koda,

– poboljšane (enhanced) instrukcije – instrukcije DSP-a dodaju se standardnim ARM instrukcijama za 16x16 bitsko brzo množenje i saturaciju. Ovim instrukcijama poboljšavaju se karakteristike koje bi dali ARM procesor i DSP ukoliko bi se koristili zajedno (Sloss, et al., 2004).

ARM Holding je dizajnirao velik broj procesora, počevši od prve verzije ARMv1. Oznake verzija i brojevi serija opisuju ARM dizajn, snagu i različite funkcije koje su rezultat unapređivanja prethodnih verzija. U ovom radu akcenat je stavljen na procesore verzije 4 i 5, a po Slossu i saradnicma (2004) presek njihovih karakteristika dat je Tabelom 2.

*Tabela 2*
*Table 2*

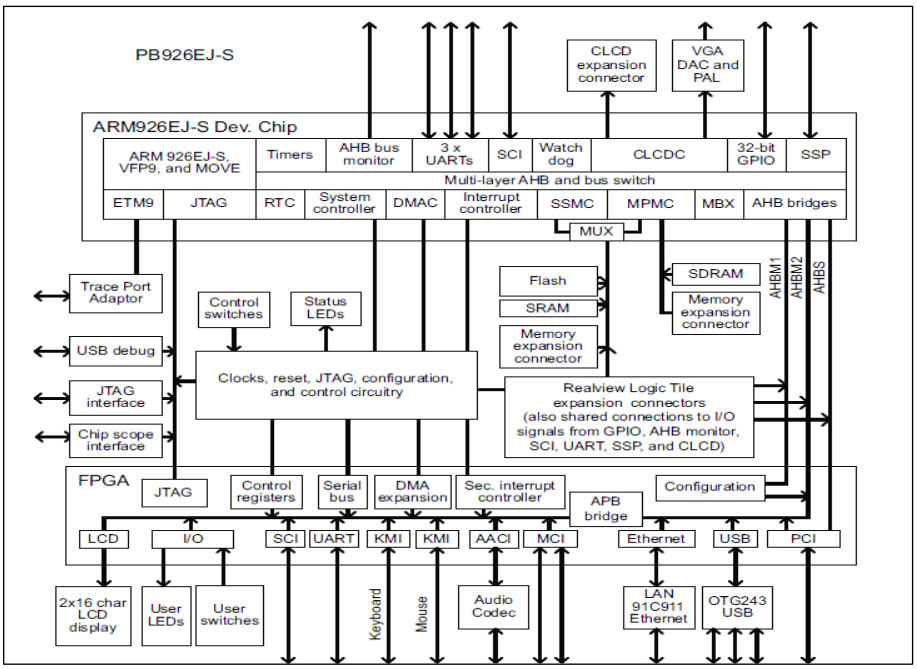
Karakteristike ARM čipova
ARM chips characteristics

| Revizija | Naziv osnovnog čipa | Poboljšanja |
|---|---|---|
| ARMv4 | StrongARM | Load-store naredbe za signed/unsigned polu-reči (16b). |
| ARMv4T | ARM7TDMI,[11] ARM9T | Thumb instrukcije. |
| ARMv5TEJ | ARE7EJ, ARM926EJ | Java ubrzanje. Bolje multiprocesorske i nove multimedia instrukcije. |

ARM9 familija predstavljena je 1997. godine. Zbog pet faza u izvršavanju naredbi ARM9 procesor može da radi na višim frekvencijama, od prethodnih familija. Dodatne faze u izvršavanju mogu da utiču na poboljšanje svih karakteristika procesora. Memorijska struktura je organizovana tako da sledi Harvardov model arhitekture u kojem su odvojene magistrale za prenos podataka od magistrala za instrukcije. Do verzije procesora ARM7 arhitektura je bila von Neumanova, tj. koristila se jedna magistrala i za podatke i za instrukcije (Daemen, Rijmen, 2001). Poslednji ARM9 procesor predstavljen je 2000. godine i nosi oznaku ARM926EJ-S. Dizajniran je za upotrebu u malim, prenosivim, na Javi-baziranim uređajima kao npr. u 3G telefoniji, ili za PDA[12] uređaje. Procesor ARM926EJ-S je prvi i po tome što koristi Jazelle tehnologiju, koja ubrzava izvršenje Java koda. Ovaj procesor podržava operativne sisteme kao što su Windows CE, Symbian OS, PalmOS i Linux. Slikom 1. prikazana je arhitektu-

[11] T – Thumb (palac), D – debug hardware specified in the architecture (debag hardver specifičan za arhitekturu), M – enhanced multiplier (poboljšani množač), I – embedded ICE for JTAG debugger connections (ugrađeni ICE za JTAG debager konekcije)
[12] PDA – Personal Digital Assistant (lični digitalni pomoćnici).

ra ploče PB926EJ-D[13] (u daljem tekstu: ploča) za razvoj proizvoda koji su bazirani na ARM926EJ-S čipu (Andrews, 2005).




*Slika 1* – Razvojna ploča PB926EJ-S
*Figure 1* – Platform Baseboard PB926EJ-S

Ploču je moguće koristiti kao samostalni razvojni sistem sa napajanjem i JTAG interfejsom (za debagovanje i testiranje), ili uz ekspanziju dodavanjem statičke ili dinamičke memorije, VGA monitora, CLCD[14] displeja, MMC[15], SD[16] ili SIM[17] kartice, periferija preko USB portova, sinhronog serijskog porta ili UART-a[18], audio CODEC-a, 32-bitskog GPIO[19], Ethernet-a, PCI kartice, itd. Strukturu i funkcije ploče čine:

– arhitektura sistema,

– ARM926EJ-S PXP Development Chip (u daljem tekstu: čip), sa procesorom koji je opisan u Development Chip Reference Manual-u (Daemen, Rijmen, 2001),

[13] Platform Baseboard for ARM926EJ-S
[14] CLCD – Color LCD Controler
[15] MMC – MultiMedia Card
[16] SD – Secure Digital card
[17] SIM – Security Identity Module
[18] Uloga UART-a je serijsko-paralelna konverzija podataka primljenih sa perifernog uređaja, i paralelno-serijska konverzija podataka koji se prenose ka periferijama
[19] GPIO – General Purpose Input Output

– PB926EJ-S FPGA20, za kontrolu sistema i konfiguracione funkcije pa ploča radi kao stand-alone razvojni sistem, ili je proširena RealView Logic Tiles ili PCI karticom,

– displeji – LCD (konverzija signala sa čipa u konfigurabilan VGA signal), i displej 2x16 karaktera (za debagovanja ili izlaz iz aplikacija),

– RealView Logic Tile ekspanzija, za razvoj AMBA AHB[21] i ABB[22] periferija,

– memorije – SSRAM i SDRAM[23] (mogu biti proširene statičkim ili dinamičkim memorijama) i flash memorija (može biti proširena statičkom memorijom),

– clock generatori (kristalni oscilatori, programabilni ili externi clock),

– debag i test interfejsi – JTAG konektor obezbeđuje opremu za debagovanje za hardver (paralelni kabl, Ethernet kabl); JTAG signali mogu biti kontrolisani preko USB debug port kontrolera, itd.

## Poboljšanja u izvršenju AES algoritma primenom ARM procesora

Softverska implementacija kriptografskih parametara u opštem slučaju nudi najviši stepen fleksibilnosti, ali može da dovede do slabih performansi embedded sistema u smislu procesne snage, memorije ili potrošnje energije. Direktan način za premošćavanje ovog problema je integracija koprocesora koji oslobađa glavni procesor od kriptografske obrade. U zavisnosti od aplikacije, koprocesor može da redukuje i memoriju potrebnu za izvršenje kriptografskog algoritma, a dodatne instrukcije mogu da redukuju ukupno procesno vreme, i uštede energiju. Kriptografski hardver je tipično mnogo brži, i energetski efikasniji od softvera koji bi bio ugrađen u glavni procesor. Međutim, kako odabrati pravu kombinaciju hardversko-softverskog dizajna, koji sistem i arhitekturu koristiti, koje komponente i platforme odabrati, problem je za koji nije dovoljno samo izabrati najbolju kombinaciju svih ovih elemenata, već je potrebna verifikacija hardvera, softvera i svih drugih parametara koji treba da dovedu do traženih rešenja i zadovolje zahteve tržišta. Na početku i u toku ovakvog procesa potrebno je izvršavati verifikacije koje su, ne retko, scenarija bez zajedničkih, ili sa malo zajedničkih obeležja. U dizajnu hardvera treba biti svestan da nove metode koje će se izvršavati u hardveru povlače posebne tehnike debagovanja i za hardver i za softver. Ovo važi za ceo dijapazon od najsporijih metoda izvršenja sa fleksibilnim debagovanjem do najbržih, pri kojima je debagovanje teško. Zato se koriste četiri metoda za har-

[20] FPGA – Field Programable Gate Array
[21] AHB – Advanced High-performance Bus
[22] ABB – Advanced Peripheral Bus
[23] RAM – Random Access Memory

dverski dizajn: (1) logička simulacija, (2) ubrzavanje simulacije, (3) emulacija hardvera i (4) izrada hardverskih prototipova, a sve u skladu sa postavljenim pitanjima o strukturi, efikasnosti i ispunjenju zadatih uslova. Očito je da ovakva kompleksnost u rešavanju zadataka zahteva kooperaciju inženjera različitih profila (koverifikacione metode) (Andrews, 2005). Hardver-inženjeri orijentisani su ka protoku po magistralama i centralnim procesorskim jedinicama, što se odnosi na interfejs za transakcije i na verifikacione platforme za logičke simulacije i kasnije za akceleraciju i emulaciju. Softver-inženjeri zahtevaju dobar model centralnog procesora i alate za debagovanje. Za svaki softver njima mora da bude dostupan *model* hardvera. Koverifikaciona matrica prikazana je u izvornom obliku (na engleskom jeziku),[24] i data Tabelom 3.

*Tabela 3*
*Table 3*

Koverifikacona matrica
Co-verification matrix

| | CPU Bus Model | Software Model of ARM CPU | Hardware Model of ARM CPU |
|---|---|---|---|
| Software Logic Simulation | 1. Block Level Tests | 2. Initialization Software | 3. I-Circuit Interface Testing |
| Simulation Acceleration | 4. Directed Tests | 5. Diagnostic Software | 6. RTOS Porting |
| Emulation | 7. Constrained Random Tests | 8. Device Drivers | 9. Application Software |

ARM procesori mogu da poboljšaju karakteristike AES algoritma kako na nivou hardvera tako i na nivou softvera, na više načina, npr.:

– primenjenom arhitekturom (npr. jezik C i assembler za 64 bitne procesore imaju bolje karakteristike od Java koda za 32 bitne procesore),

– ekstenzijom arhitekture (manji broj obraćanja memoriji, manji broj instrukcija),

– optimizacijom algoritama za ekspanziju ključa, šifrovanje i dešifrovanje (npr. kod transpozicije matrice stanja),

– poboljšavanjem/dodavanjem instrukcija za optimizaciju operacija AES-a (npr. smanjenje koda, smanjenje broja xor operacija),

– povećanjem brzine izvršenja algoritma (npr. nove instrukcije Intela AES-NI ubrzavaju izvršenje algoritma 3–10 puta (Intel Advanced Encrypton Standard), određene operacije mogu da se paralelno, istovremeno izvršavaju u jednom registru procesora (Bertoni, et al., 2006) što dovodi do smanjenja broja ciklusa po bajtu (Osvik, et al., 2010),

– smanjenjem utroška energije (npr. koršćenje 32-bitnog low-power ARM procesora), itd.

[24] Izvorni, engleski jezik, korišćen je da bi bile otklonjene eventualne nedoumice oko naziva elemenata matrice. U tekstu koji sledi će, u zavisnosti od potrebe, biti prevedeni i opisani oni izrazi u matrici koji će direktno ukazivati na implementaciju AES-256 u ARM926EJ-S procesor.

Autori Irwin i Page (2003) predložili su ekstenziju za PLX – RISC arhitekturu opšte namene, i razmatrali korišćenje ekstenzija za multimediju za 128-bitski PLX procesor, kako bi implementirali AES sa što manjim brojem obraćanja memoriji. Međutim, prezentovani koncept je teško prilagodljiv 32-bitskoj arhitekturi (Tillich, Großschadl, 2006; Bertoni, et al., 2006).U tradicionalnim sistemima za prenos slike i video sadržaja celokupni sadržaj najpre se kompresuje nekim od algoritama kompresije. Zatim se tako dobijeni niz podataka u celosti zaštiti primenom nekog od standardnih kriptografskih algoritama (DES, IDEA[25], AES). Specifične karakteristike podataka ovog tipa, velika bitska brzina prenosa podataka i ograničena dozvoljena širina propusnog opsega, čine standardne kriptografske algoritme neadekvatnim za primenu u ove svrhe (Jovanović, 2010). Bertoni i njegovi saradnici ponudili su različite instrukcije, o kojima su detalji i procenjene karakteristike za Intel StrongARM procesore prikazani u (Irvin, Page, 2003). Osnova njihovog rada je optimizacija algoritma u funkcijama MixColumns() i InvMixColumns() u kojima je, između ostalog, bitno redukovan broj xor operacija. Ovo je dodatno ubrzala grupa autora okupljenih oko Atasu-a (ALaRI-USI), iz Lugana (Atasu, et al., 2004), koji su koristili StrongARM SA-1110 Mikroprocesor (SA-1110), koji je integrisani 32-bitski RISC mikroprocesor za Intel dizajn i procesnu tehnologiju, dok je efikasnost, u smislu utroška energije, deo ARM arhitekture (Intel Corporation, 1998). Darnall i Kuhlman (2006) testirali su AES implementiran u jedan od najpopularnijih ARM procesora – ARM7TDMI, koji se koristi kod mobilnih telefona, pejdžera i mp3 plejera, jer je procesor mali potrošač. Pokazali su da postoji niz mogućnosti da programer, na uštrb jedne karakteristike, poboljša čitav niz drugih, kao što je npr. ušteda ROM-a koja malo usporava vreme za kriptozaštitu ili, obrnuto, bitno će biti smanjeno vreme enkripcije ukoliko se iskoristi više ROM-a (Intel Corporation, 2000).

## Zaključak

Kod uređaja kojima je limitirana snaga, memorija i potrošnja energije, što je karakteristika svih prenosivih modula koji zahtevaju veliku brzinu obrade podataka i koriste baterijsko napajanje, potrebno je koristiti čipove koji su pogodni da zadovolje ove uslove i istovremeno im obezbede kvalitetan rad. ARM procesor je gotovo idealan za realizaciju ovih uređaja, a obzirom da je ARM zaštićen kao intelektualna svojina a ne kao dizajn, veliki broj proizvođača koristi ovu tehnologiju za ugradnju u smart kartice, potrošačku elektroniku, senzore u proizvodnim sistemima i slično. Međutim, ARM nije isključivo namenjen za smanjenje potrošnje energije ili dimenzija uređaja, već su nje-

[25] IDEA – International Data Encryption Algorithm

gove karakteristike da doprinosi efikasnosti, pogotovo zbog činjenice da može da bude deo embedded sistema. ARM instrukcije su 32-bitske i izvode se u 3–5 koraka, a služe za obradu, skladištenje i prenos podataka. Zbog svojih karakteristika ARM se često koristi da ubrza rad algoritama za kriptozaštitu. Jedan od takvih algoritama je NIST-ov AES, koji je standardizovan kao Rijndealovo rešenje dužine ulaznih i izlaznih podataka od 128b i dužina ključa od 128b, 192b i 256b, u kategoriji standarda za računarsku bezbednost. AES može biti implementiran softversko-hardverski, fleksibilan je i ima dobru kombinaciju karakteristika bezbednosti i efikasnosti.

ARM na različite načine obezbeđuje izvršavanje AES algoritma, hardversko-softverskom optimizacijom na nivou primenjenih softverskih paketa, ekstenzijom arhitekture, optimizacijom algoritama za ekspanziju ključa, šifrovanje i dešifrovanje, dodavanjem novih i poboljšanjem postojećih instrukcija za optimizaciju operacija AES-a, povećanjem brzine izvršavanja algoritma i smanjenjem utroška energije.

## *Literatura*


Andrews, J., R., 2005, *Co-Verification of Hardware and Software for ARM SoC Design*, Elsevier.

Atasu, K., Belveglieri, L., Macchetti, M., 2004, Efficient AES Implementations for ARM Based Platforms, *SAC'04*, Nicosia, Cyprus, March 14–17

Bertoni, G., Breveglieri, L., Farina, R., Regazzoni, F., 2006, Speeding Up AES By Extending a 32-Bit Processor Instruction Set, pp.275–282, In *Proceedings of the 17th IEEE International Conference on Application-Specific Systems, Architectures and Processors ASAP 2006*, Sep 11–13.

Daemen, J., Rijmen, V., 1999, Efficient Block Ciphers for Smartcards, pp.29–36, In *USENIX Workshop on Smartcard Technology Smartcard '99*, May 10–11.

Daemen, J., Rijmen, V., 2000, The Block Cipher Rijndael, pp.288–296, In Quisquater, J. and Schneier, B., editors, Smart Card Research and Applications, *Volume 1820 of Lecture Notes in Computer Science*, Springer, Berlin.

Daemen, J., Rijmen, V., 2001, Rijndael, the Advanced Encryption Standard, *Dr. Dobb's Journal*, 26(3), pp.137–139.

Dandalis, A., Prasanna, V. K., Rolim J. D. P., 2000, A Comparative Study of Performance of AES Final Candidates Using FPGAs, pp.125–140, In C¸. K. Ko¸c and C. Paar, editors, *Proc. Cryptographic Hardware and Embedded Systems Workshop (CHES'00)*, Volume 1965 of LNCS, Springer-Verlag.

Darnall, M., Kuhlman, D., 2006, AES Software Implementation on ARM7TDMI, pp.424–435, In Barua, R., and Lange, T. (Eds.) *Progress in Cryptology – INDOCRYPT 2006, 7th International Conference on Cryptology in India*, Kolkata, India, December 11–13.

Irwin, J., Page. D., 2003, Using Media Processors for Low-Memory AES Implementation, pp.144–154, *In Proceedings of the 14th IEEE International Conference on Application-specific Systems, Architectures and Processors ASAP 2003*, June 24–26.

Jovanović, B., 2010, Algoritmi selektivnog šifrovanja – pregled sa ocenom performansi, *Vojnotehnički glasnik/Military Technical Courier*, 10(4), pp.134–154.

Klami, K., Hammond, B., Spencer, M., 2009, *ARM Announces 10 Billionth Mobile Processor*, Dostupno na: http://www.arm. com/news/24403.html, Preuzeto 10.01.2013.

Kuljanski, S., 2010, RSA algoritam i njegova praktična primena, *Vojnotehnički glasnik/Military Technical Courier*, 10(3), pp.65–77.

Osvik, D. A., Bos, J. W., Stefan, D., Canright, D., 2010, *Fast Software AES Encryption*, pp.1–20.

Sano, F., Koike, M., Kawamura, S., Shiba, M., 2002, Performance Evaluation of AES Finalists on the High-End SMART Card, pp.82–93, *Third AES Candidate Conference*, New York, USA, April 13–14.

Sloss, A. N., Symes, D., Wright, C., 2004, *ARM System Developer's Guide, Designing and Optimizing Software*, Morgan Kaufmann Publishers (Imprint of Elsevier).

Sternbenz, A., Lipp, P., 2002, Performance of the AES Candidate Algorithm in JAVA, pp.161–165, *Third AES Candidate Conference*, New York, USA, April 13–14.

Tillich, S., Großschadl, J., 2006, Instruction Set Extensions for Efficient AES, pp.270–284, In Goubin, L. and Matsui, M. (Eds.): CHES 2006, LNCS 4249, *Implementation on 32-bit Processors*, International Association for Cryptologic Research.

Federal Information Processing Standards Publication 197, 2001, *Announcing the Advanced Encryption Standard (AES)*, Dostupno na: http://csrc. nist. gov/publications/, Preuzeto 12.10.2012.

Intel, Intel Advanced Encryption Standard (AES) Instruction Set – Rev 3.01, [internet], Dostupno na: < http://software. intel. com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set i>, Preuzeto 08.01.2013.

Intel, 1998, *Intel StrongARM SA-110 Microprocessor Instruction Timing, Application Note 278194–001*, Intel Corporation.

Intel, 2000, *Intel StrongARM SA-1110 Microprocessor, Developer's Manual 278240–003*, Intel Corporation.

*Q4 revenue came from the sale of 1.8 billion ARM-processor based chips*, 2001, ARM press release.

AES AND ARM PROCESSORS

FIELD: Electronics
ARTICLE TYPE: Professional Paper

*Summary*

*The need for information security leads to big problems in the development of portable devices which have limited available memory space and power consumption. Also, if coprocessors for encryption are*

*added to core processors, dimensions of such devices grow, they become inflexible and their price increases several times. It is, also, very well known that algorithms for data encryption are memory demanding and, because of a large number of operations that has to be executed during encryption and decryption, coprocessors often slow core processors. For one of cryptography standards, AES, the NIST has accepted Rijndael's block algorithm; the length of the input and output data stream is 128 bits, and the lengths of the cipher key can be 128, 192 and 256 bits. Due to the characteristics of low power consumption, a 32-bit architecture format, as well as fast execution of instructions, ARM processors implement the AES algorithm and do not burden the main processes in the system in which it is implemented. The ARM technology is protected as intellectual property, not as a processor design. As a result, many manufacturers have developed their own ARM-based products, so today over 2 billion chips are produced. This paper presents the possibilities for improving the performance of the AES algorithm using the latest version of the ARM processor.*

## Introduction

*The growing need for information security requires fast execution of cryptographic algorithms. With the increasing usage of wireless and portable devices in everyday life, cryptographic protection becomes crucial. This task becomes extremely challenging when using portable devices that have limited power, memory and power consumption. In the traditional approach, if the demanding cryptographic coprocessors are installed, the core processor becomes inflexible. The alternative is the integration of new simple instructions, to accelerate the processor and support the cryptographic operations.*

*In the nineties, the Acorn company developed the first high-performance and low-power microprocessor for commercial purpose – ARM. Unlike other manufacturers, the ARM Holdings did not register the ARM processor as a product but as intellectual property, which was probably the reason why many other companies, such as Intel, Motorola and Texas Instruments, developed their own products, with a total of about 2 billion chips produced. ARM processors are suitable for use in portable battery-powered systems such as smart cards, consumer electronics, etc. The ARM architecture is designed to support 32-bit embedded systems that can be used in a variety of devices, from small sensors on the assembly line to the NASA control systems. The elements of ARM-based devices are the ARM processor, controllers, peripherals and busses.*

*In 2001, the US National Institute of Standards and Technology (NIST) released the results of the competition for a new algorithm for data protection, which had to replace the Data Encryption Standard (DES) with the new one. The NIST insisted that symmetrical block algorithms had to be submitted to the tender. Rijndeal's solution was chosen as the best algorithm out of 15, due to a good combination of safety, efficiency, easiness of implementation and flexibility. In 2001, the*

*NIST standardized the Advanced Encryption Standard (AES) in the category of standards for computer security. The standard came into effect in May 2002.*

*ARM processors speed up the AES algorithm execution and release main processors of the cryptographic processing. The advantages of the ARM processors implementation are presented in this paper.*

## AES

*AES is Rijndael's standardized block iterative algorithm for data encryption whose input and output sequences lengths are 128 bits. The cipher key has a variable length of 128, 192 or 256 bits. Encryption is performed in four steps, meaning four byte-oriented transformations: substitution using substitution tables, shifting state matrix (s) for different offsets, transformation of one column of the state matrix and round key adding. Each transformation has its inverse transformation for decryption. Deciphering steps run in the opposite direction of the steps in encoding. Encryption begins with a state matrix transformation. At first, the plain text (input data that is not encrypted) is assigned to the state matrix, which is organized as a matrix of 4x4 bytes. The number of rounds ($N_r$) determines the key length K (32-bit words). Decryption goes in the opposite way – from the ciphered data to the plain text.*

## ARM processors

*The choice of a suitable processor that can be integrated in the embedded system depends on the application domain. Applications can be arithmetic or control intensive, which can be a problem for devices for mass and large scale usage. With that in mind, it is necessary to use different classes of processors such as microcontrollers, digital signal processors (DSPs), application specific processors, multimedia processors and RISC processors. ARM processors are developed based on RISC processors; however, what distinguishes them is their purpose – ARMs are to be a part of a larger, embedded system. In this system, the focus is not only on processor's power but also on its efficiency. It is essential to achieve the maximum system performance and take care of low power consumption.*

*The ARM instructions are 32-bit and run in three to five steps: receiving, decoding, execution, memory access and data storage. They require a load-store architecture, three-address command for data processing, conditional execution of commands, powerful multiple registers for load-store operations in one clock, performing shift and arithmetic and logic operation in one clock cycle. The expansion to the open set of commands via the coprocessor instruction set is possible with 16-bit Thumb instructions.*

*The ARM Holdings have designed a number of processors, but this paper discusses ARM7 and ARM9 processors from the ARMv5EJ series, whose improvements were JavaScript acceleration, better multiprocessor instructions and new multimedia instructions.*

Improvements in the performances of the AES algorithm by the ARM processors usage

*Software implementation of cryptographic parameters generally offers the highest degree of flexibility, but it can lead to poor performance of the embedded system in terms of processing power, memory and energy. A direct way of overcoming this problem is the integration of coprocessors that „liberates" the main processor of the cryptographic processing. Depending on the applications, the coprocessor can reduce the memory required for the execution of cryptographic algorithms and additional instructions can reduce the total processing time and energy. Cryptographic hardware is typically much faster and more energy efficient than the software installed in the main processor. However, how to choose the right combination of hardware and software design is a problem that requires cooperation of engineering of different profiles in order to reach the required solution and fulfill the market demands.*

*ARM processors can improve the characteristics of the AES algorithm at both hardware and software levels in various ways, such as:*

*– architecture (e. g. C and the assembler for 64-bit processors have better performances than the Java code for 32-bit processors),*

*– architecture expansion (e. g. less memory addressing, a smaller number of instructions),*

*– optimization algorithm for key expansion, encoding and decoding (e. g. transposition of the state matrix),*

*– improvement of old and adding new instructions for optimization (e. g. reducing code density),*

*– increasing the speed of algorithm execution (e. g. Intel added a new instruction, Intel AES-NI),*

*– reduction of energy consumption (e. g. using 32-bit low-power ARM processors), etc.*

Conclusion

*For devices that have limited processing power, memory and energy consumption, which are the characteristics of all portable modules that require high-speed data processing and have a battery, there is a need for chips suitable to meet these conditions and, at the same time, to provide them with high operational quality. ARM processors are almost ideal for the realization of these devices and considering that the ARM is protected as intellectual property, not as a design, a large number of companies uses this technology for smart cards, consumer electronics, sensors in production systems, and the like. However, the ARM is not only designed to reduce energy consumption or dimension of devices, but its characteristics contribute to the efficiency, particularly because of the fact that it can be a part of the embedded system. The ARM 32-bit instructions perform in the 3–5 steps; they are used for the processing, storage and transmission of data.*

*Due to its characteristics, the ARM is often used to speed up the cryptographic algorithms. One of them is NIST's AES, a standardized Rijndael's solution in the category of standards for information security; lengths of input and output data are 128-bits and the key lengths are 128, 192, and 256 bits. The AES can be implemented in software, it is flexible, and has a good combination of security and efficiency. The ARM provides better execution of the AES by software and hardware optimization at the level of applied software, by architecture expansion, by the optimization of algorithms for key expansion, encryption and decryption, by adding new instructions for the AES optimization and enhancing the existing ones,, by increasing the computational power and by reducing energy consumption.*

Key words: *processors; encryption; AES.*

Datum prijema članka/Paper received on: 19. 01. 2013.
Datum dostavljanja ispravki rukopisa/Manuscript corrections submitted on:
07. 02. 2013.
Datum konačnog prihvatanja članka za objavljivanje/ Paper accepted for publishing on:
09. 02. 2013.