

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zavod za elektroničke sustave i obradu informacija

Sustavi za praćenje i vođenje procesa

Seminarski rad

Mrežni procesori

30. svibnja 2005.

Hrvoje Kalinić  
0036387298

# Sadržaj

Sadržaj .....	1
Apstraktno .....	2
Problem .....	3
Mrežni procesor .....	5
➤ Osnovne (opće) funkcije mrežnog procesora .....	5
➤ Tipovi aplikacija ugrađenih u mrežnim procesorima .....	6
➤ Značajke arhitekture mrežnih procesora .....	7
Summa summarum .....	10
Zaključak .....	11
Dodatak .....	12
Literatura .....	13

## **Apstraktno**

U tekstu je prikazan problem zbog kojeg je došlo do razvoja mrežnih procesora, u grubo su objašnjeni nedostaci prijašnjih tehnologija ta varijacije koje su dovele do razvoja mrežnih procesora kakvi se koriste danas.

Dalje se raspravlja o općim funkcijama mrežnog procesora, tipičnim aplikacijama, te na kraju i arhitekturi koja se koristi u mrežnim procesorima.

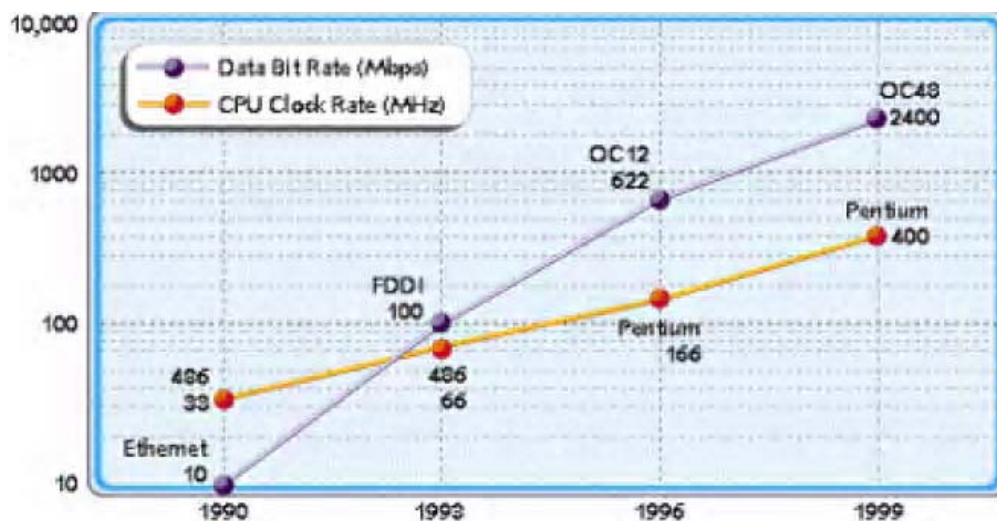
Ukratko, ovo je samo kratak uvod u to područje, za koji sam nastojao da bude što općenitiji, postupno šireći definiciju pojma, pokušavajući slijediti i povijesni razvoj.

## Problem

Gotovo svi uređaji na mreži koriste neke oblike obrade paketa. Ethernet preklopnici („switch-ovi“), AMT preklopnici, IP usmjerivači („rout-eri“), vatrozidovi (*firewalls*), uređaji za presretanje, web serveri, web preklopnici (...) su samo neki od njih. U prošlosti, kada je protok podataka bio relativno sporiji u odnosu na mogućnosti procesora, a mrežni protokoli jednostavniji, mrežni uređaji su bili izvedeni poput programa koji su nastanjivali procesore opće namjene (i njima bili pokretani). No, kao se povećavao protok podataka, a time i zahtjevi za složenijim mrežnim protokolima brzina sabirnice i takta procesora opće namjene je bila ubrzo premašena.

Naravno, možemo postaviti pitanje kao je moguće da imamo problem s brzinom, s obzirom da je opće poznato da je brzina razvoja novih tehnologija u siliciju iznimno brza.

Odgovor možemo vidjeti na Slici 1 gdje je prikazano kako frekvencijski pojas optičkih vlakana raste i većom brzinom no razvoj silicija.



Slika 1. Usporedan rast brzine procesora i komunikacijskih kanala

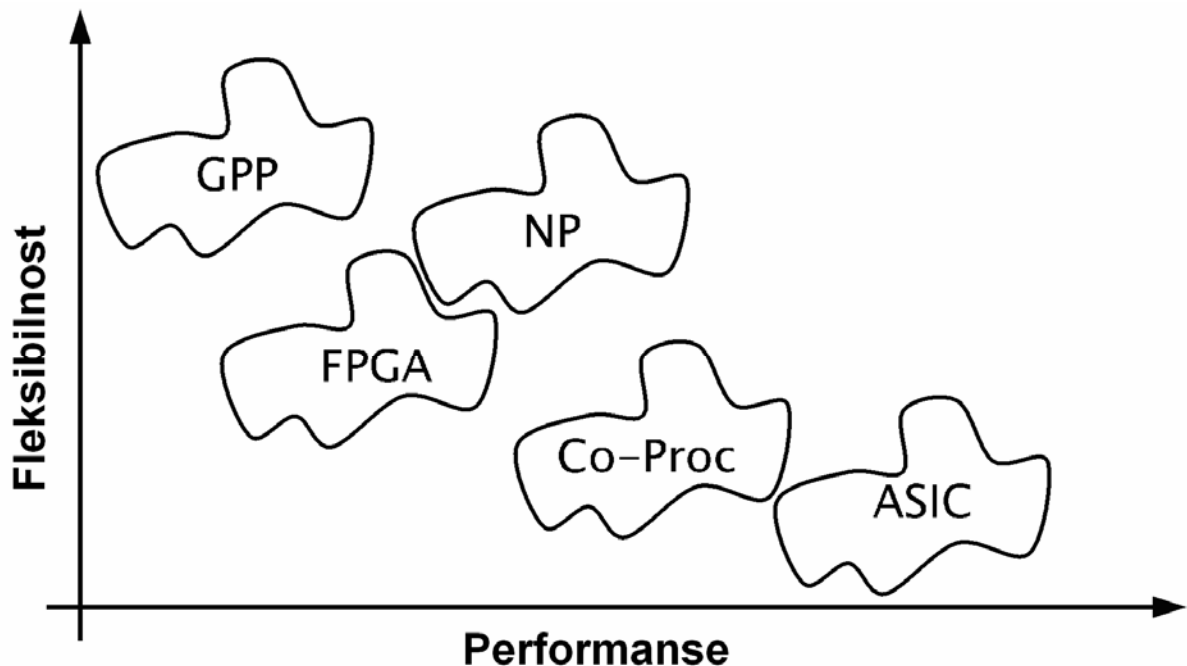
Rast poluvodičkih komponenti je već legendaran pa je možda zapanjujuće da neka druga tehnologija može nadmašiti taj rast. Slika 1 pokazuje rast brzine takta procesora u usporedbi s brzinom protoka podataka kroz mrežu u posljednjem desetljeću 20og stoljeća. Zanimljivo je primijetiti da se brzina takta procesora povećala za 12 puta dok se brzina mreže povećala za 240 puta. Daljnji eksponencijalni rast se očekuje prvenstveno iz razloga što su temeljne barijere dovoljno udaljene.

Drugi razlog možemo pronaći u sve složenijim protokolima, većim zahtjevima na mrežu, raznim aspektima sigurnosti (...) dakle, rast složenosti obrade paketa, jednako kao i njihovoj količini.

Danas, u jezgri mreže postoje sustavi brine i do 10Gbps, a pojavljuju se i brzine do 40Gbps. Pristupne i rubne mreže rade na brzinama do 1Gbps, pa čak i do 2.5Gbps. Smjer u razvoju današnjih mreža okrenut je prema sve složenijim protokolima i dodatnim uslugama. Neke od navedenih usluga uključuje poboljšanje kvalitete usluge (QoS), sigurnost, i poznavanje s IPv6 protokola. Navedena unaprjeđenja zahtijevaju značajnu procesorsku moć, koja u većini slučajeva nadilazi sposobnosti procesora opće namjene. Donedavno je jedino rješenje bilo u

ASIC<sup>1</sup> procesora. No korištenje ASIC-a značilo je odricanje programibilnosti nauštrb brzini i procesorskoj snazi. To je kompromis u kojem svakako gubimo. To je postalo jasno daljnjim razvojem mreža koje su posebno na svojim rubnim i pristupnim dijelovima mora biti prilagodljiva, da bi i noviji protokoli mogli biti podržavani. To zahtjeva određeno dozu programibilnosti, da bi mrežni uređaji mogli biti unaprijeđeni tokom svog rada na mreži. To je nešto što ASIC ne omogućava. Uz to, razvojno vrijeme ASIC sklopa je dugotrajno – nadmašuje 18 mjeseci u većini slučajeva. S obzirom na brzinu razvoja mreža, zahtjevi na razvoj takvog sklopa su da mora biti rađen u najnaprednijim sub-mikronskim tehnologijama što dodatno komplicira razvoj (s obzirom na pouzdanost novih tehnologija) i, naravno, dodatno povećava vrijeme razvoja. Također bitno je napomenuti da ti isti razlozi uzrokuju i implementaciju sklopa na većoj površini silicija, a sve to zajedno, naravno, pridonosi i povećanju cijene. Tako dugi razvojni ciklusi zahtijevaju preciznije predviđanje mrežnih protokola koje će trebati podržati u trenutku kada izađu na tržište. To naravno, obično rezultira mijenjanjem proizvoda tokom samog razvojnog procesa. Ako tome pribrojimo mogućnost da protokol nije doraden ili u potpunosti definiran, problemi kao da niču na svakom koraku. Također, da bi proizvodnja bila isplativa, potrebno je imati velike serije. Što je za namjenske sklopove kao što je ASIC dosta teže ispuniti.

Kao drugo rješenje samo nam se nameće u ovom trenutku FPGA sklopovlje. FPGA sklopovlje ima kraći razvojni ciklus no ASIC, no ipak duži, s obzirom na programsko rješenje ugrađeno u procesorima opće namjene. Kao glavni nedostatak i FPGA sklopovlja možemo navesti visoku potrošnju i cijenu. Fiksna arhitektura FPGA sklopa je također nedostatak za složenije i brže aplikacije, također FPGA nema toliku propusnost kao ASIC, a također i manju unutarnju memoriju.



Slika 2. Usporedne karakteristike različitih familija procesora

<sup>1</sup> Eng. „Application Specific Integrated Circuit“ – Integrirani krug dizajniran s posebnom namjerom, s jedinstvenim funkcijama ugrađenim u sebe. Koristi se s namjerom da se poveća brzina obrade podataka, smanji potrošnja energije ili cijena.

Zbog svih navedenih razloga, postaje jasno da uređaj koji nam je potreban za obradu paketa na mreži treba biti programibilan, uz zadovoljavanje specifičnih zahtjeva mrežnih aplikacija. Slika 2<sup>2</sup> prikazuje usporene karakteristike različitih familija procesora u ovisnost o njihovoj fleksibilnosti i performansama. Fleksibilnost možemo dobiti većom programibilnošću sklopa, bolje performanse arhitekturo.

Takav uređaj je poznat pod nazivom mrežni procesor.

## Mrežni procesor

Jednostavna definicija mrežnog procesora može izgledati nešto poput:

Mrežni procesor je programibilni uređaj posebno dizajniran s namjerom da obrađuje podatke mrežnom brzinom.

Naravno, to nam baš i ne govori mnogo o tome što to mrežni procesor točno obrađuje i koje su njegove funkcije. Stoga ćemo se prvo tome posvetiti malo pažnje.

### Osnovne (opće) funkcije mrežnog procesora

U svojoj općoj namjeni kao procesor mrežnih paketa mrežni procesor ima neke tipične značajke koje uključuju:

Izdvajanje obrazaca (uzoraka) (eng. „Pattern Matching“)

- Sposobnost da pronalazi određene uzorke bitova (ili bajtova) unutar paketa, što mu omogućava da klasificira tipove paketa. To je bitno npr. ukoliko treba odlučiti je li paket IPv4 ili IPv6 tipa.

Pogledavanje po ključu (eng. „Key Lookup e.g. Address Lookup“)

- Sposobnost brzog pregledavanja baze podataka koristeći ključ (najčešće adresu paketa), a u svrhu pronalaženja bitnih informacija, najčešće o usmjeravanju.
- Podatke pregledava obično u vlastitoj tablici koja može biti i dio unutarnje kao i vanjske memorije. Za mala polja to može biti i tablica 1n1 koja zahtjeva samo jednu provjeru, no za velika polja mogu se zahtijevati višestruke provjere zbog pretraživanja po stablu. Tipična provjera se odnosi na provjeru određene adrese s namjerom da se odredi adresa sljedećeg skoka.

Obrada podataka (eng. „Data Manipulation“)

- Promjena podataka sadržanih u paketu, tokom njegove obrade.
- Neki od primjera zašto nam je potrebna ova sposobnost mrežnog procesora mogu biti primjena CRC<sup>3</sup> algoritma (provjera ispravnosti), rastavljanje i ponovno sastavljanje podataka, zaštitno kodiranje ili dekodiranje podataka.

---

<sup>2</sup> Za nešto detaljnije karakteristike pogledati tablicu u dodatku

<sup>3</sup> Eng. Cyclic redundancy check

Menadžment nizova (eng. „Queue Managment“)

- Kako se podaci primaju i obrađuju tako se za daljnje slanje sortiraju u nizove (queue<sup>4</sup>).
- Raspoređivanje u nizove ima svoje prioritete i svaki paket dobiva svoj redoslijed definirajući na taj način promet na mreži i prioritete QoS-a.

Kontrola obrade paketa (eng. „Control Processing“)

- Manji procesi koji obrađuju pakete kontrolirani su sa više razine koja uključuje komunikaciju i usklađivanje s ostalim čvorovima sustava.

Računanje

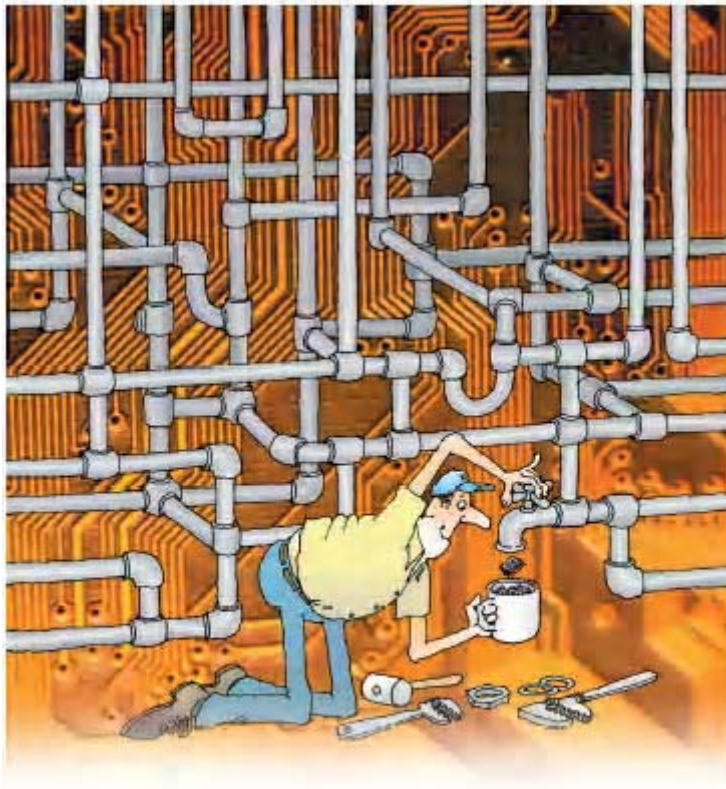
Koristeći opće funkcije mrežnog procesora, program implementiran kao aplikacija koju mrežni procesor izvršava rezultira komadićem fizičke opreme koja obavlja zadanu funkciju. Stoga na jednom mjestu nabrojimo još i aplikacije koje se obično ugrađuju u mrežne procesore

### **Tipovi aplikacija ugrađenih u mrežnim procesorima**

- Filtriranje ili prosljeđivanje paketa ili okvira; dakle, osnovne operacije usmjerivača („router-a“) ili preklopnika („switch-eva“).
- Unapređivanje kvalitete usluge (eng. „QoS“) – identificiranje različitih tipova ili klasa paketa i posebno obrađivanje tih tipova i klasa nauštrb nekih drugih tipova i klasa. Dakle, kontroliranje vjerojatnosti da se određeni paket obradi.
- Kontrola pristupa – odlučivanje hoće li se određenom paketu (ili nizu paketa) dopustiti prijelaz preko tog dijela mrežne opreme.
- Zaštitno kodiranje toka podataka (eng. „Encryption of data streams“) – ugrađeni mehanizmi koji na fizičkoj osnovi omogućavaju kodiranje individualnim (zasebnim) tokovima podataka.

---

<sup>4</sup> Queue dolazi od latinske riječi cauda, koja znači rep.



Promotrimo također i neke značajke arhitekture koje su proizvođači mrežnih procesora usvojili, s namjerom da postignu željene performanse.

### Značajke arhitekture mrežnih procesora

U definiciji mrežnog procesora smo već spomenuli da mrežni procesor programabilan uređaj. No, programibilne uređaje smo također već spomenuli kao uređaje izvedene na procesorima opće namjene i primijetili da takvi uređaji nisu pogodni za primjene na mreži, osim, možda, za najsporije aplikacije, iz razloga što nisu dovoljno brzi. Postavlja se, dakle, pitanje: „Po čemu su to mrežni procesori toliko različiti od procesora opće namijene?“

Odgovor je da je njihova glavna značajka mehanizmi višestruke obrade podataka.

Mehanizmi višestruke obrade podataka (eng. „Multiple Programmable Processing Engines“)

- Svojstvo mrežnog procesora je da posjeduje više mehanizama višestruke obrade podataka (eng. PPEs<sup>5</sup>) ili procesorskih jezgara, za razliku od procesora opće namijene koji imaju samo jednu jezgru procesora.

Posljednjih godina mnoga kompanije započele su rad na vlastitim mrežnim procesorima, jednako poznate tvrtke, kao i mnoge male tvrtke. To je dovelo do raznih arhitektura mrežnih procesora, no svima im je zajedničko to što imaju višestruke jezgre unutar procesora. Mehanizmi višestruke obrade podataka se iz istog razloga mogu pronaći pod raznim nazivima

---

<sup>5</sup> Eng. „Programmable Processing Engines“



i s različitim funkcijama, ovisno o proizvođaču<sup>6</sup>. Neke od izvedbi procesora su bazirane na RISC<sup>7</sup> jezgrama. No, opće RISC jezgre nisu sasvim pogodne za manipulaciju bitovima, dok je manipulacija bitova glavna funkcija mrežnih procesora. Stoga je manipulacija bitovima dodavana jezgri RISC-a. Općenito mrežni procesori bazirani na RISC jezgrama imaju svoje mehanizme obrade podataka organizirane paralelno, tako da svaki paket koji dođe u mrežni procesor je dodijeljen svojoj jezgri procesora, na taj način da ona vrši čitavu obradu nad njim. Također mehanizmi višestruke obrade podataka mogu biti organizirani u protočnoj strukturi, ili u kombinaciji protočne i paralelne strukture, pa čak i da budu podesive da primjene najpogodniju kombinaciju paralelno-protočne strukture za danu aplikaciju. Kao alternativa RISC arhitekturi može poslužiti VLIW arhitektura. Procesori s VLIW arhitekturom imaju veći broj naredbi i stoga riječi njihovih naredbi iznimno dugačke odakle im i naziv (eng. „Very Long Instruction Word“). No VLIW procesori također imaju i prednost da za gotovo svaku operaciju koju trebaju izvršiti imaju naredbu, tako da čak i one složenije mogu izvesti s relativno malim skupom naredbi. To se koristi i u dizajnu mrežnih procesora. S VLIW jezgrama mehanizme obrade podataka možemo podijeliti na način da svakoj jezgri dodijelimo određenu zadaću. Kao posljedicu toga imamo slučaj u kojemu većinu naredbi u pojedinoj VLIW jezgri ne koristimo, sažimajući na taj način broj naredbi potrebnih danoj jezgri za izvršavanje zadaće. To može dovesti do značajnog poboljšanja u performansama ukoliko iskoristimo te posebnosti za optimiziranje mehanizama obrade podataka. Dakle, u ovoj izvedbi jezgre su organizirane na način da svaka obavlja točno određenu funkciju u protočnoj strukturi, za daljnje poboljšanje performansi ništa nas ne sprječava da koristimo višestruku paralelnu protočnu strukturu u obradi nadolazećih paketa. No, takva namjenska struktura pojedinih dijelova VLIW arhitekture ima za posljedicu da su mrežni procesori s takvim jezgrama teži za programiranje i manje fleksibilni. Mnogi mrežni procesori, a posebno oni s arhitekturu baziranom na RISC jezgrama koriste oblike višeprotočnosti<sup>8</sup> u svojim mehanizmima obrade podataka. To je jasno s stajališta povećanja performansi jer se može dogoditi tijekom izvršavanja naredbe da neka od jezgara treba čekati da se završi neka druga operacija. Za primjer nam može poslužiti čitanje podataka iz zajedničke memorije. Ukoliko neka druga jezgra upravo ima pristup memoriji, sabirnica je zauzeta, te naša jezgra mora čekati (svoj red). Neaktivno stanje neke od jezgara je uzaludno trošenje procesorskog vremena koje se izravno očituje na performansama uređaja. Iz tog razloga se koristi višeprotočnost koja omogućava da se više od jedne zadaće izvršava na jednoj jezgri, prebacujući se s jedne zadaće na drugu ukoliko jedna od njih treba čekati na svršetak nekog drugog procesa. Da bi se osiguralo dovoljno brzo prebacivanje između zadaća jezgre mrežni procesor ima sklopovlje koje to prebacivanje čini praktički trenutnim, što je još jedna od razlika prema procesorima opće namjene kojima mora proći određeni broj instrukcija da bi se prebacili s zadaće na zadaću. Kao, možda jedini nedostatak, u takvom pristupu napomenut ćemo da višeprotočnost kao i sami višestruki mehanizmi obrade podataka se izravno očituju u složenosti koda programa koji se izvršavaju na takvom procesoru.

Namjensko sklopovlje (eng. „Dedicated Hardware“)

- Mrežni procesori, posebno oni temeljeni na RISC jezgrama, posjeduju posebno namijenjeno sklopovlje, ili integrirane koprocesore koji služe za obavljanje

---

<sup>6</sup> Eng. Neki od naziva su: „micro engines“, „channel processors“, „task optimised processor“... S obzirom na šturost literature o ovom pojmu na Hrvatskom jeziku, smatrao sam da te nazive nema smisla prevoditi.

<sup>7</sup> Procesor s reduciranim setom naredbi (eng. „Reduced Instruction Set Computer“). Arhitektura RISC-a ima vrlo jednostavan skup naredbi, stoga, da bi izvršio složenije naredbe, zahtjeva veći broj operacija.

<sup>8</sup> Eng. multithreading, thread- pobijati se, progurati, specifično označava i dio programa koji se može izvršiti sam za sebe. Razlikovati od višezadačnosti (multitaskinaga) jer se zadaće u tom slučaju mogu bitno razlikovati jedna od druge. Prevodim taj pojam kao višeprotočnost po uzoru na protočnu strukturu.

uobičajenih funkcija poput pregledavanja po ključu, menadžment nizova, računanje CRC algoritma ili zaštitnog kodiranja. To dodano sklopovlje također može biti napravljeno ciljano za jednu (i upravo samo za tu) funkciju.

#### Sučelje prema mreži (eng. „Network Interface“)

- Ključno obilježje mrežnih procesora je njihovo mrežno sučelje, kao mjesto gdje paketi odlaze na mrežu i s mreže dolaze na mrežni procesor.
- Neki od standardnih sučelja koji se danas koriste su UTOPIA 2 i 3, SPI-3, a za 10Gbps promet SPI4.2.
- Također SPI4.2 se koristi i kako sučelje za izradu preklopnika (eng. „Switch Fabric Interface“), s obzirom da se mnogi od mrežnih procesora koriste i kao usmjerivači.

#### Kontrola obrade paketa (eng. „Control Processing“)

- Glavna zadaća mrežnog procesora je da obrađuje pakete podataka mrežnom brzinom. No, neki od paketa, takozvani kontrolni<sup>9</sup> paketi ne trebaju biti obrađeni tolikom brzinom. To nam pokazuje da neki paketi imaju prioritet pred drugima i zahtjeva nam obraćanje pažnje na iznimne pakete i preslagivanje redosljeda obrade. Kontrolni paketi obično zahtijevaju složeniju obradu no ostali paketi, te ukoliko budu obrađivani standardnom procedurom u nekoj od jezgri mogu značajno usporiti protok podataka kroz nju. Ti podaci su stoga izdvojeni iz uobičajenog toka podataka i proslijeđeni odvojenom procesoru koji nazivamo kontrolni procesor. Taj procesor je obično odgovoran i za neke druge funkcije poput statističke obrade podataka, a funkciju mu može izvršavati i procesor opće namijene poput ARM-a. Iako mnogi mrežni procesori imaju integriranu jezgru koja obavlja funkciju kontrolnog procesora, može postojati i vanjski procesor opće namijene spojen sabirnicom s ostatkom sklopa. Je li kontrolni procesor unutar ili van mrežnog procesora obično je definirano položajem mrežnog procesora u samoj mreži. Tako se zbog cijene obično ugrađuje u mrežni procesor za mreže bliže korisniku, dok unutrašnjost mreže (veće brzine) zahtjeva postojanje zasebnog procesora.

#### Memorija

- Mrežni procesor koristi memoriju za pohranjivanje programskog koda, tablica za pregledavanje<sup>10</sup>, podataka o paketima itd. Iako je izvedivo da memorija bude unutar mrežnog procesora, količina memorije potrebne u nekim slučajevima čini neizbježnim da se koristi i vanjska memorija. Namjenska memorija se često koristi za pohranu programskog koda, nalazi se lokalno do svake jezgre. Kod potreban za obradu paketa obično je sažet i zbijen tako da memorija potrebna za to može biti prilično malena, npr. 4Kbyta po jezgri za mrežne procesore s manjim zahtjevima. Nasuprot tome, kod potreban kontrolnom procesoru može biti drastično velik, što je izravna posljedica iznimki koje mora obraditi. Stoga je memorija potrebna kontrolnom procesoru izvedena kao vanjska, iako će naravno imati nešto priručne<sup>11</sup> unutarnje memorije. Dodatno, osim tog komadića memorije dodijeljenog lokalno svakoj jezgri procesora postoji i zajednička unutarnja memorija, koju jezgre međusobno dijele.
- Većina mrežnih procesora ima barem dva sučelja prema vanjskoj memoriji, jedno prema SRAM-u<sup>12</sup>, a jedno prema DRAM-u<sup>13</sup>, dok mrežni procesori s većim

---

<sup>9</sup> Eng. „control and management packets“

<sup>10</sup> Eng. „lookup tables“

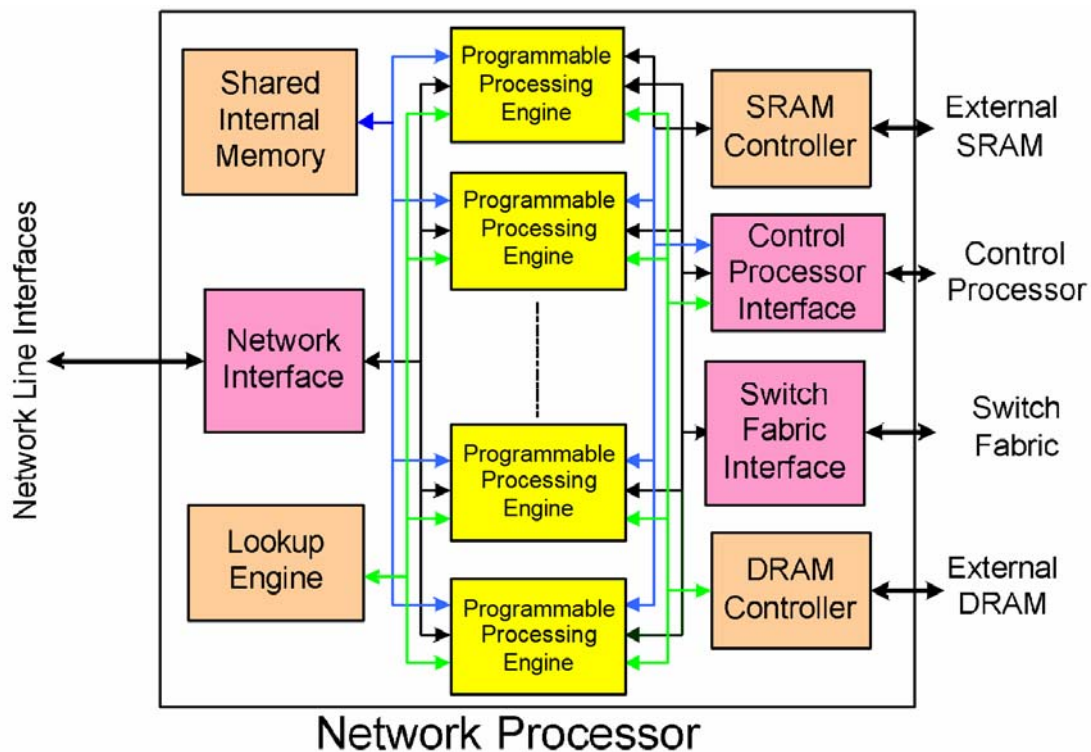
<sup>11</sup> Eng. „cache“

<sup>12</sup> Eng. „Static Random Access Memory“

<sup>13</sup> Eng. „Dynamic Random Access Memory“

zahtjevima imaju i više od jednog sučelja za svaki tip memorije. Pristup vanjskoj memoriji je usko grlo u svakoj arhitekturi, stoga proizvođači koriste posljednje tehnologije da bi omogućili brže pristupe memoriji, neki od primjera su DDR (Eng. „Double Data Rate“) ili QDR (Eng. „Quadruple Data Rate“) SRAM, DDR SDRAM (Eng. „Synchronous DRAM“), FCRAM (Eng. „Fast Cycle RAM“) ili RDRAM (Eng. „Rambus DRAM“).

Dosadašnji pregled trebao bi nam barem donekle dati uvid u arhitekturu mrežnog procesora. Sve navedene karakteristike arhitekture prikazane su na Slici 3 koja prikazuje opću arhitekturu mrežnog procesora.



Slika 3. Opća arhitektura mrežnog procesora

## Summa summarum

Kako su rasli mrežni protoci podataka pojavio se zahtjev za sve naprednijim mrežnim protokolima, te su se zahtijevali sve brži procesori na mreži. Uskoro je dosegnuto da su željene brzine mogli postići samo ASIC sklopovi koji su imali predug razvoj i prekratko zadržavanje na tržištu, s obzirom na nedostatak fleksibilnosti. Iz tog razloga su se pojavili zahtjevi za programabilnim sklopovima što je dovelo do razvoja mrežnih procesora.

Mrežni procesori ostvaruju svoju prednost zahvaljujući mehanizmima višestruke obrade podataka koji su ugrađeni u mrežni procesor kao višestruke jezgre. Također sadrže i dodatno sklopovlje koje im pomaže u sučeljavanju s mrežnim prometom.

## Zaključak

Moram priznati da je moj odabir teme bio potpuno slučajan, da nisam uopće bio upoznat s ovim pojmom kada sam ga prvi put pročitao. Tako da je bio priličan izazov proučiti sve to u tako kratkom vremenu i odglumiti stručnjaka. Nadam se da sam u tome barem donekle uspio. Također, nastojao sam i sve termine prevesti na hrvatski jezik, jer smatram da sam to dužan prema svojoj kulturi, no svjestan sam da bi to svima otežalo daljnje istraživanje bilo po ključnoj riječi bilo po literaturi. Stoga se nadam da sam ostavi dovoljno napomena u tekstu za sve koji će se dalje baviti ovim pojmom. Također, ispričavam se ukoliko sam unio zabunu s kojim od svojih prijevoda, jer iako sam se trudio da istražim kako su pojedini pojmovi već prevedeni na hrvatski jezik, nisam za sve pojmove uspio, dijelom stoga što su oni rijetki autori koji pišu na hrvatskom jeziku išli često jednostavnijim putem koristeći se *slang*-ovnim pojmovnikom, ili jednostavno zato što mi nisu dostupni.

Nadam se ujedno da će ovaj tekst svima koji ga budu čitali poslužiti kao kratak uvod u ovo područje. Kao što i rekoh, teško je biti stručnjak iz područja za koje imaš manje od mjesec dana da mu se posvetiš, tako da su neke stvari sigurno šture, a neke možda prejednostavno opisane. Toga sam svjestan, no dijelom sam to učinio i namjerno, s namjerom da tekst već u prvom čitanju bude razumljiv. Nastojao sam u njemu ocrtati način na koji sam i sam učio o mrežnim procesorima.

## Dodatak:

	Complete Programmability	Simple Programming Model	Maximum System Flexibility	Massive Processing Power	High Functional Integration	Stable Programming Interfaces	Third-Party Support
Network Processors	++	++	++	++	++	++	++
Custom ASICs	--	--	--	++	+	--	--
Configurable Processors							
Configurable SOC	+	--	+	-	+	--	-
Configurable Processor Cores	+	-	+	-	+	--	--
Application-Specific Standard Products (ASSPs)							
Smart MACs	-	-	-	--	-	--	--
Single Function Components	-	--	-	+	--	-	--
Programmable Communications Components							
DSPs	+	-	+	+	-	+	+
State Machine Engines	+	--	-	-	-	--	--
Special Purpose Devices	+	-	+	-	--	-	-
Pattern Matching Processors	+	-	-	+	--	--	--
Switching Chipsets							
L2 chipsets	--	-	--	--	++	--	--
Network Processor chip sets	+	--	-	-	+	--	-

++ is excellent; + is good; - is fair; and -- is poor

## Literatura:

Herity, Dominic: „Network Processor Programming“

<http://www.embedded.com/story/OEG20010730S0053>

<http://www.s3group.com/pdf/ESP2001-NP-Programming.pdf>

Heppel, Andrew: „An Introduction to Network Processors“

[http://www.roke.co.uk/download/white\\_papers/network\\_processors\\_introduction.pdf](http://www.roke.co.uk/download/white_papers/network_processors_introduction.pdf)

Husak, David: „Network Processors: A Definition and Comparison“

<http://www.mi.chu.edu.tw/~ykchang/cial/IP-Routers/network-processor-definition-comparison.pdf>

[http://www.freescale.com/files/netcomm/doc/white\\_paper/COMMPROCWP.pdf](http://www.freescale.com/files/netcomm/doc/white_paper/COMMPROCWP.pdf)

Intel: „Next Generation Network Processor Technologies“

<http://www.intel.com/design/network/papers/27905001.pdf>

De Angelis, G.: „A Model Based Approach to Design Applications for network Processors“

<http://pacinotti.isti.cnr.it/ERI/VIworkshop/DeAngelis.pdf>

Ostali internet izvori:

<http://computing-dictionary.thefreedictionary.com/network%20processor>

<http://www.intel.com/education/highered/Networking/Lectures.htm>

[http://en.wikipedia.org/wiki/Network\\_processor](http://en.wikipedia.org/wiki/Network_processor)

[http://newsroom.cisco.com/dlls/2004/hd\\_051904c.html](http://newsroom.cisco.com/dlls/2004/hd_051904c.html)

