Mrežni procesori

IBM PowerNP 4GS3

Dalibor Hrg

Sadržaj:

1. Uvod

Mrežni procesori su specijalizirani procesori za potrebe implementacije mrežnih protokola pri velikim brzinama prijenosa podataka. Njima se ne implementiraju čitavi protokoli već samo dijelovi protokola koji troše više vremena na obradu paketa nego uobičajeno. Mjesto njihove ugradnje je između sučelja mreže (*network interface*) i sučelja premosnika ili usmjerivača (*switch fabric in a switch/router*). Ta specijalizacija se odrazila ne samo na arhitekturu već i na samo programsko sučelje potrebno da procesori prorade. Pod mrežnim procesorom podrazumijeva se niz više procesora koji imaju programsku podršku i osiromašeniju arhitekturu a sve zbog njihovog spajanja u višeprocesni sustav radi brže obrade nad paketima podataka (PDU-*protocol data units)*. Mrežni procesori su vrlo korisna stvar budući da omogućuju niz dodatnih usluga kao kvaliteta usluge (QoS-*quality of service*), kompresija podataka, sigurnost (IPsec), virusna zaštita i dr. a to sve pri velikim brzinama poput mreže SONET OC-48 (2.488 Gbps).

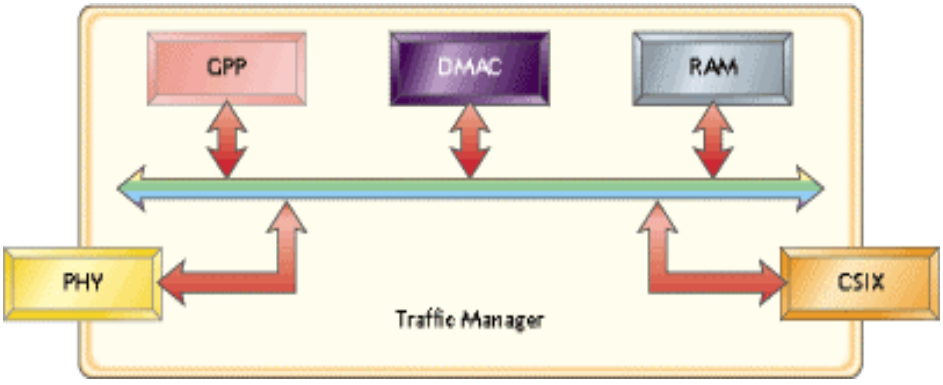
U ovom seminaru se opisuju i daju karakteristike mrežnih procesora, kako se upotrebljavaju i što su zamijenili. Također su navedeni problemi oko brzine i kompleksnosti. Dan je opis općenite arhitekture mrežnog procesora, konfiguracija procesnih elemenata mrežnih procesora i načina njihovog programiranja. U nastavku je opis IBM porodice mrežnih procesora i njihova usporedba s Motorolinim procesorima. Na kraju se detaljno opisuje IBM PowerNP 4GS3 koji je od *Micro Design Resources* proglašen najboljim mrežnim procesorom za 2001. i 2002.

2. Mrežni procesori i NTM (*network traffic manager*)

Mrežni procesor se upotrebljava u NTM-u (NTM-*network traffic manager*, upravljač mrežnog prometa) koji se nalazi između samog sučelja mreže i prklopnik/usmjernik uređaja. NTM određuje na temelju prispjelog podatka, kome, kako i kada treba poslati podatak. NTM posjeduje dva sučelja: PHY (*physical interface*) sučelje prema mreži i CSIX (*common switch interface*) sučelje prema preklopniku. Slika 1. prikazuje način ugradnje NTM-a u jednu mrežu.

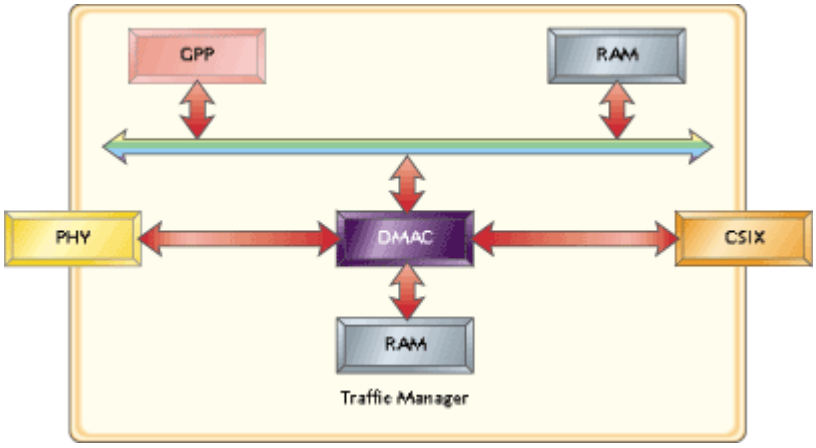
Slika 1: Način spajanja NTM-a.

Često se umjesto NTM-a koristi samo izraz TM (*traffic manager*). Prve izvedbe TM-a su se sastojale od GPP-a (GPP-*general purpose processor,* procesor opće namjene) koji je preko DMAC-a (*direct memory access controller*) komunicirao sa RAM-om (*random access memory*). Tu je bila i jednostavna I/O jedinica. Mrežni promet unutar TM-a se sastojao od obrade PDU-a (*protocol data unit*) koju je prvo dohvaćao GPP preko DMAC-a iz RAM-a te je potom prosljeđivao prema sučelju PHY ili CSIX. PDU je inače generički naziv za bilo koji podatak ili jedinku podatka nekog protokola. Tako npr. u IP (*Internet protocol*) PDU je paket. Prvi TM-ovi su se sastojali od navedenih jedinica koje su prikazane na slici 2.



Slika 2: Arhitektura TM-a.

Arhitektura TM-a promjenjena je kada je brzina prijenosa podataka mrežom prestigla brzinu klasičnih procesora i brzinu prijenosa putem sabirnica. To je zahtijevalo integraciju sučelja PHY i CSIX unutar jedinstvenog ASIC-a (ASIC-*application-specific integrated circuit,* aplikacijski specifični integrirani sklop) čime je skračeno vrijeme prijenosa PDU-a do RAM-a jer nije bilo potrebno koristiti sabirnicu u tom smjeru (Slika3).

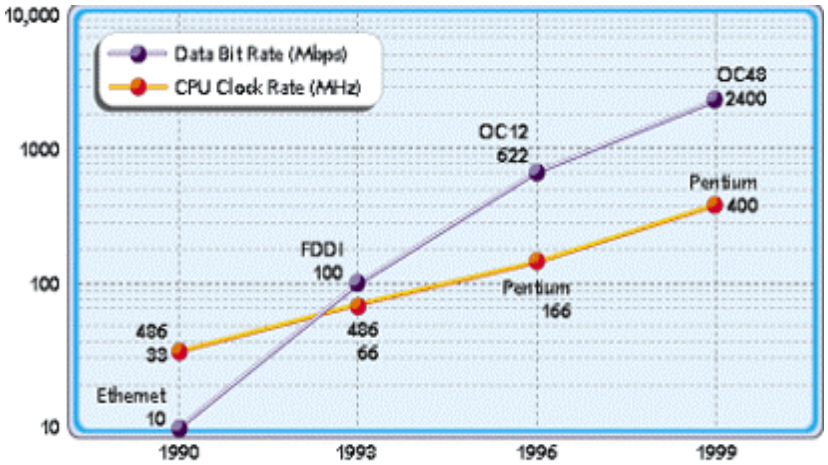


Slika 3: ASIC bazirana arhitektura TM-a.

Protokoli su unutar ove nove arhitekture bili izvedeni u ASIC-u koji se jednom programiraju za određenu funkcionalnost. To se postiže određenim stupnjem ožičavanja (*hard-wired)* u samoj arhitekturi ASIC-a. ASIC je propuštao GPP-u samo one PDU-ove koji su zahtijevali obradu ili su to neki kontrolni i signalni PDU-ovi. U svrhu povećanja funkcionalnosti uvedeni su umjesto ASIC-a mrežni procesori koji su programirljivi i time se mogu ostvariti nove usluge na razini mrežnog prometa kao kvaliteta usluge (QoS-*quality of service*), kompresija podataka, sigurnost (IPsec), virusna zaštita i dr.

3. Problem brzine i kompleksnosti

Dva su općenita problema vezana uz mrežne procesore: brzina i kompleksnost. Budući da reguliraju mrežni promet pri brzinama reda Gbps, ostaje malo vremena za procesiranje svakog PDU-a. Uočeno je da propusnost mreža baziranih na optičkim vlaknima s vremenom puno brže raste nego performanse i brzine procesora koji se razvijaju na siliciju. U posljednjih desetak godina propusnost mreža je porasla za faktor 240 dok je brzina procesora za 12. To se jasnije vidi na slici 4.

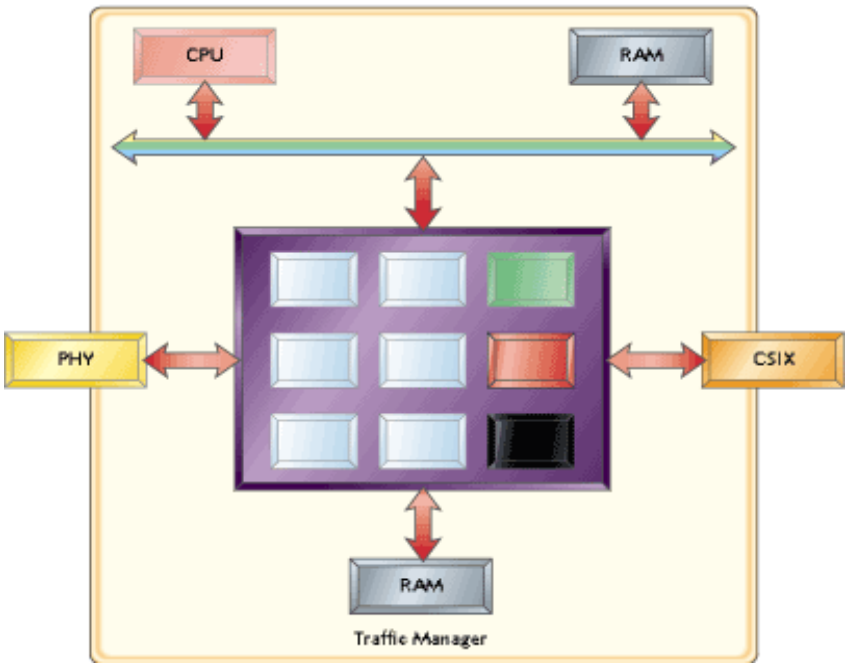


Slika 4: Usporedba propusnosti mreže i takta procesora

Kompleksnost je vezana uz pojavu novih usluga koje su potpuno programski izvedene. Npr. ako posjedujemo Sonet OC-48 (2.488 Gbps) mrežu,133 MHz mrežni procesor te uzmemo da IP paket zajedno s zaglavljem zauzima 49 okteta, može se jednostavno pokazati da ostaje 21 procesorski takt po paketu za obradu što je malo. Klasična obrada paketa se sastoji od smanjivanja 8 bitnog brojila svaki puta kada ga neki usmjerenik prihvati, izračunavanja kontrolne sume i provjere ispravnosti paketa te njegovog usmjeravanja na odredišnu adresu. Međutim, uvođenjem novih kompleksnijih usluga koje bi bilo potrebno vršiti nad tim paketima zahtijevalo bi previše sklopovskih elemenata ili kod ASIC-a ožičavanja pa se to mora izvesti programski. Tipično se uvode usluge detekcije i zaštite od virusa te kompresija podataka uz brzine reda Gpbs što povećava performanse same mreže.

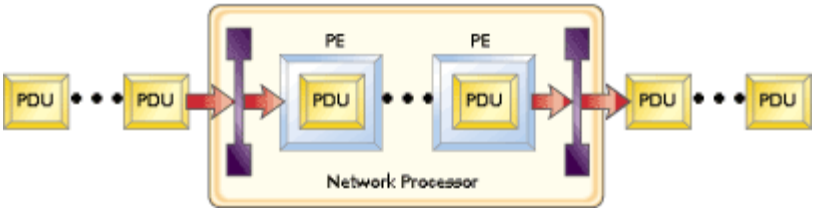
4. Arhitektura mrežnih procesora

Kod mrežnih procesora bitno je uočiti razliku prema ostalim procesnim elementima poput GPP-a koji su najčešće Intel Pentium ili PowerPC. Jedna od glavnih razlika je da su dizajneri kod GPP-ova gledali da je omogućena dobra instrukcijska iskoristivost što je očitovano velikim spremničkim memorijama (*cache memory*), nizom registara, protočnom (*pipeline)* strukturom i dr. Mrežni procesori ne nasljeđuju sve osobine klasičnih GPP-ova ali su bazirani i na višeprocesorskoj arhitekturi. Sastoje se od više procesnih elemenata (PE-*processing element*) koji rade paralelno čime se dobivaju performanse bolje od samo jednog GPP-a. Paralelnim načinom rada više PE-ova postiže se obrada N PDU-ova u vremenskom trajanju obrade samo jednog PDU-a umjesto da se jedan PDU obrađuje N puta brže. Mrežni procesori su najčešće bazirani na RISC arhitekturom kao Motorola C5 mrežni procesor koji sadrži 16 RISC PE-ova. Slično je i sa IBM PowerNP-ovima koji imaju 8 PE-ova i dvije tzv. *picocode engine* procesne elemente koji mogu izvršavati dvije programske niti paralelno i zbog performansi programiraju se u asembleru. Slika 5. prikazuje klasičnu arhitekturu TM-a gdje se vidi zamjena ASIC-a sa NP-om koji se sastoji od više PE-ova.

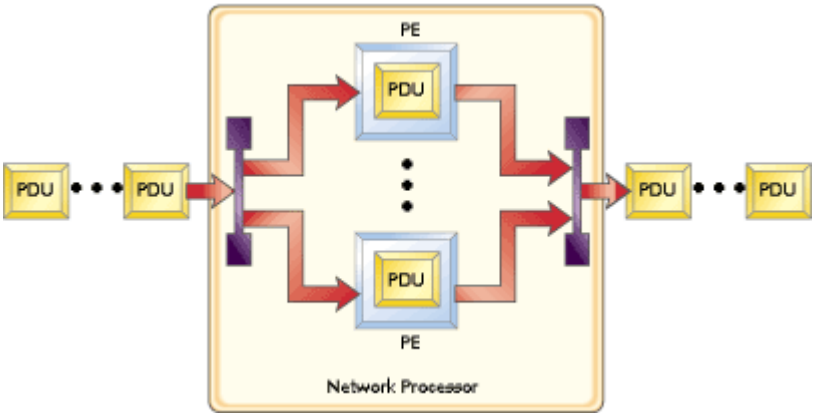


Slika 5: Arhitektura TM-a sa mrežnim procesorom

Dva su načina izgradnje višeprocesnog sustava: protočna (*pipeline)* arhitektura kod koje svaki PE ima točno određenu funkciju za obraditi nad PDU-om ili paralelno raspoređeni PE-ovi tako da se istovremeno procesira više PDU-ova na način da jedan PE obrađuje u potpunosti njemu dodijeljeni PDU. Slika 6. prikazuje protočnu (*pipeline)* arhitekturu dok slika 7. paralelno raspoređivanje PE-ova.



Slika 6: Protočna (*pipeline)* arhitektura mrežnog procesora



Slika 7: Paralelno raspoređivanje procesnih elemenata

Logično je pitanje koliko je idealno potrebno PE-ova u jednom mrežnom procesoru? To se promatra ovisno o propusnosti mreže za koju ćemo to primijeniti. Vrijeme propagacije (*latency*) paketa u LAN-u je reda 1μs, dok recimo na udaljenostima od 6000 km (na razini Interneta) kašnjenje iznosi oko 10 ms. Obično se dozvoljava kašnjenje od 10 μs po mrežnom procesoru, pa se za određivanje idealnog broja PE-ova koristi podatak propusnosti mreže kako bi se izračunalo koliko PDU-ova dolazi u intervalu od 10 μs. Upravo toliko iznosi idealni broj PE-ova u mrežnom procesoru. Radi primjera pogledajmo idealni broj PE-ova za SONET OC-48 (2.488 Gbps). Veličina paketa iznosi 40 okteta na kojih moramo dodati 9 okteta zaglavlja PPP-a (*point to point protocol*) i još 3.5% od tih 49 okteta koje unosi zaglavlje SONET-a. Ukupno je za jedan paket potrebno 406 bita.

$$\frac{2.488 \cdot 10^9 \frac{\text{b}}{\text{s}} \cdot 10\mu\text{s}}{406\text{b}} = 61$$

Prema tome ukupno je potrebno 61 procesnih elemenata. Vidi se da njihov broj ovisi o propusnosti mreže ali i o samoj veličini paketa. U tablici 1. dani su primjeri za ostale mreže:

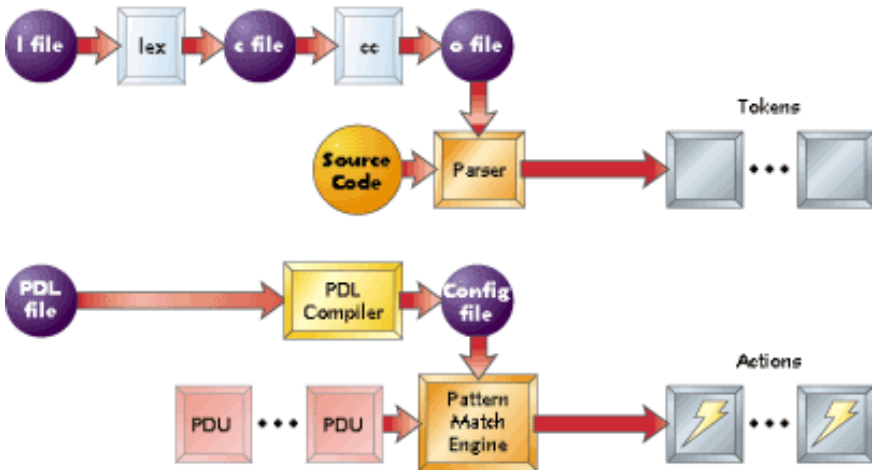
Standard	Propusnost mreže(Gbps)	Idealni broj PE-ova
100Mb Ethernet	0.1	2
OC-3	0.155	3
OC-12	0.622	15
OC-48	2.4	61
OC-192	9.6	244

Tablica 1: Idealni broj procesnih elemenata ovisno o mreži

Kod navedene dvije izvedbe arhitekture treba uočiti i neke probleme. Brzina obrade nekog PDU-a kod protočne arhitekture ovisi o najsporijem PE-u tako da je to i ukupna brzina mrežnog procesora. Dobra strana ove arhitekture je da nije potrbno voditi brige oko redoslijeda obrade PDU-ova budući da ih se redom procesira. Kod paralelne izvedbe problem nastaje nakon što se obrade PDU-ovi, kada ih je potrebno otpustiti. Budući da neka obrada može prije završiti potrebno ih je stavljati u red kojeg treba reorganizirati po prispijeću PDU-ova u mrežni procesor što može predstavljati dodatno gubljenje vremena.

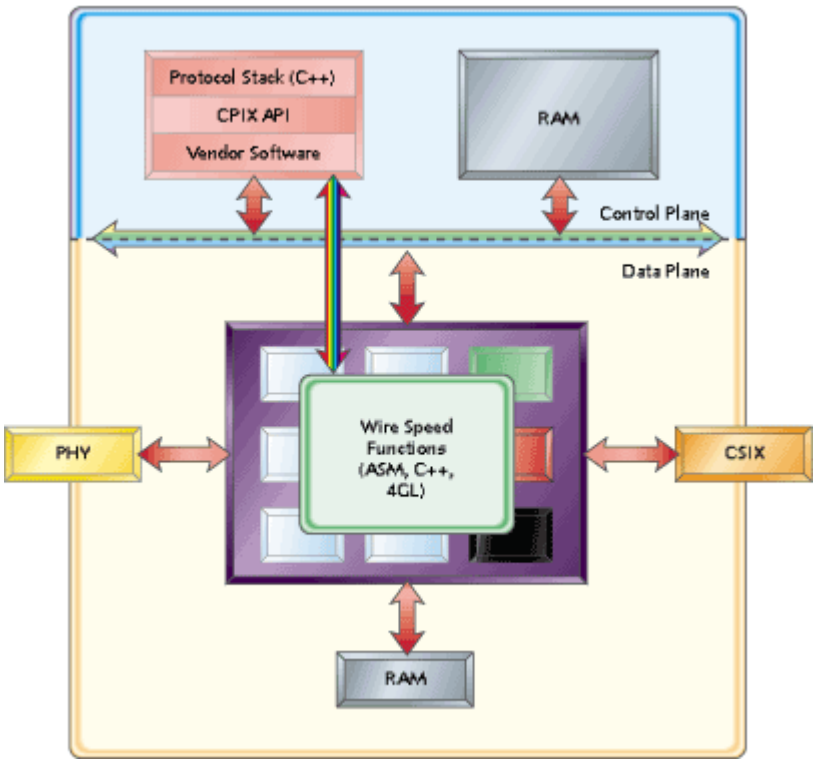
5. Programska podrška za mrežne procesore

Općenito, kod programiranja najviše su zastupljeni imperativni i funkcionalni jezici. Imperativni jezici postižu funkcionalnost postavljanjem vrijednosti varijabli, pozivima procedura i ispitivanjem vrijednosti varijabli. Osim toga naredbe se izvršavaju slijedno ili postoje skokovi na neke naredbe. Tu spadaju C, C++, Pascal, Ada i dr. Funkcionalni jezici su građeni od poziva i definicija procedura. Program specificira što se događa kad nastupi uvjet a ne kako se izvodi. To uvodi olakšanje pri programiranju. Budući da je zadatak mrežnih procesora obrada PDU-a točnije provoditi neku klasifikaciju nad PDU-ovima te eventualno pokrenuti neku od usluga za razliku od GPP-a koji se bavi kontrolnim i ispitnim PDU-ovima, dolazi u pitanje upotreba funkcionalnih jezika. Primjer funkcionalnog jezika je jezik koji prihvaća leksički analizator lex. Lex prihvaća jezik koji specificira uzorke koji se mogu pojaviti na ulazu i fragmente C koda koji govore što se radi za neki uzorak. Nadalje, analizator proizvodi C kod programa koji stvori logičku cjelinu koja služi kao parser za neki prevoditelj ili interpreter. Slična je stvar i sa PDL (*pattern description language*). Uzorci s ulaza su PDU-ovi a ne nizovi znakova kao u obradi jezika. PDL prevoditelj ima sličan zadatak kao leksički analizator. On generira kod koji će izvršavati PE i time raditi klasifikaciju PDU-a i dodatne potrebne usluge. Slika 8. prikazuje cijeli postupak upotrebe PDL-a i lex-a.



Slika 8: Usporedba lex-a i PDL-a

Mnogi proizvođači mrežnih procesora kao Motorola, IBM, Lucent (Agere) i Solidum nude jezike visoke razine (HLL-*high level languages*) kao programsku opremu. Slika 8. prikazuje lex-a i PDL-a za potrebe zadataka poput klasifikacije PDU-a. S druge strane imperativni jezici isto stoje na usluzi da se primjene pa se najčešće koriste asembler i C++ kao što je navedeno. Slika 9. prikazuje upotrebu pojedinih jezika kod GPP-a i PE-a.



Slika 9: Programska podrška mrežnog procesora

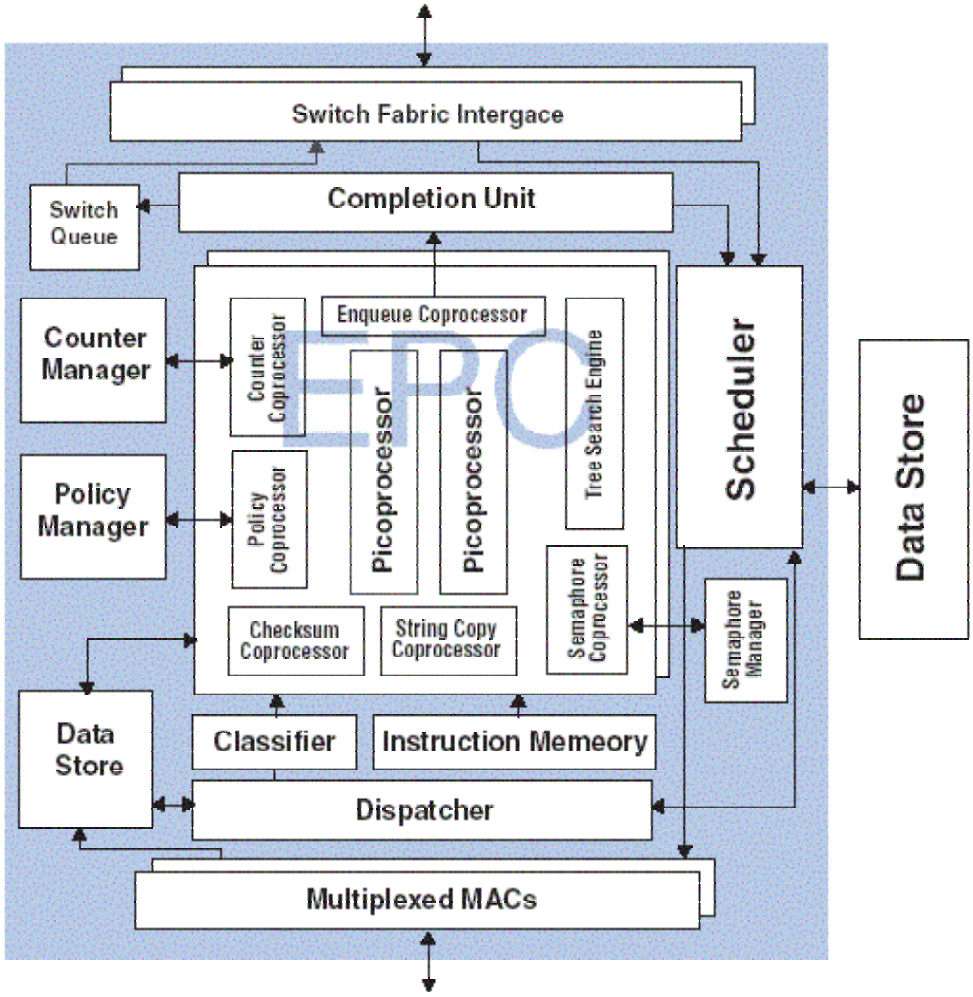
Kod odabira mrežnog procesora mora se voditi računa da se odabere onaj kod kojeg postoji dobra programska podrška. To podrazumijeva mogućnost programiranja mrežnog procesora u asembleru, C++ i funkcionalnim jezikom. Mora postojati dobro definiran API kojim GPP komunicira s mrežnim procesorom kao i jasno iskorištavanje paralelnosti i višeprocesne moći.

6. IBM mrežni procesori (PowerNP 4GS3)

Ovisno o proizvođaču mrežnog procesora na tržištu se nalaze ove tri vrste arhitektura mrežnog procesora:

- Potpuno sklopovska izvedba ili konfigurabilni ASIC
- Potpuno programibilni mrežni procesori
- Hibrid ili programibilni ASIC

Svaka od ovih arhitektura ima svoje mane i prednosti. Sklopovska izvedba je namijenjena brzim operacijama i dobro se uklapa u mreže velike propusnosti dok programibilni mrežni procesori omogućuje veću fleksibilnost. Očito hibridi čine najbolji odabir pa je i tržište orijentirano prema njima. IBM se je odlučio za hibridnu izvedbu koja je bolje poznatija kao IBM PowerNP. U PowerNP arhitekturi, sklopovski su izvedene one funkcije za obradu paketa koje zahtijevaju velike brzine rada. Procesni elementi ili pikoprocesori za programsku podršku koriste asembler. Slika 10. prikazuje prvi mrežni procesor 4GS3 u izvedbi PowerNP arhitekture.



Slika 10: IBM PowerNP 4GS3

Prema slici 10. NP4GS3 se sastoji od:

- **EPC** (*Embedded Processor Complex*) – objedinjuje višeprocesorski sustav i sklopovske akceleratore (*coprocessors*)
- **Instruction memory** - memorije za instrukcije
- **Ethernet MAC-ova** – omogučuju spajanje na lokalnu mrežu
- **sučelje prema preklopniku (*switch*)**
- **ostali fiksno-logički elementi**

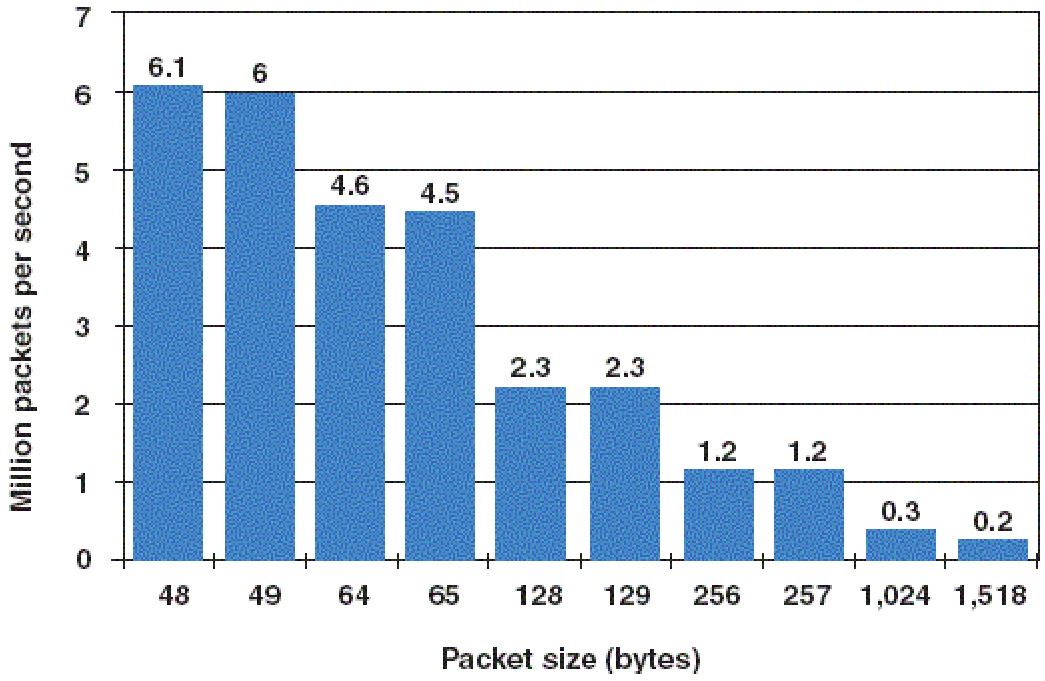
Srce NP4GS3 je **EPC** koji se sastoji od 16 programibilnih pikoprocesora, višestrukih sklopovskih akceleratora i PowerPC 405 kontrolnog procesora. Svaki 32-bitni pikoprocesor radi na 133 MHz i izvodi dvije programske niti istovremeno. Brzina pikoprocesora iznos 2128 MIPS-a. PowerNP arhitektura podržava paralelnu izvedbu procesnih elemenata tako da svaki pikoprocesor procesira neki paket u potpunosti. Pikoprocesorima je zajednički dostupno 128 KB djeljene memorije za spremanje instrukcija. **EPC** se ujedno sastoji od oko 80 sklopovskih akceleratora. Dolje su navedeni neki koprocesori ili sklopovski akceleratori te njihova uloga:

- **Enqueue coprocessor** – vrši određene funkcije oko redova paketa i njegovog reorganiziranja.
- **Classifier** – opcionalno se koristi za potrebe klasificiranja bitnih dijelova okvira.
- **Checksum coprocessor –** provjerava i izračunava zaštitu nad okvirima.
- **Flow Control, Counter manager, Scheduler, Policy manager –** brinu se oko standardnog mrežnog prometa kao dekrementiranja brojila nekog paketa, stavljene u red, kontrola toka podataka i sl.

NP4GS3 sadrži 384KB memorije na čipu za spremanje tablica podataka ili sličnih podatkovnih pravila. Čip ima mogućnost i adresirati vanjski RAM ukupne veličine 4M x 64 bajta. Totalna propusnost podataka iznosi 120 Gbps što omogućuje više od 500,000 tabličnih promjena u sekundi. Što se tiče multipleksiranih MAC-ova, omogućeno je 40 10/100Mb Ethernet portova ili 4 Gigabit Ethernet porta. Alternativno, arhitektura podržava 1 x OC-48, 4 x OC-12 ili 16 x OC-3 POS (*Packet over SONET*) porta. Ovime se smanjuje potreba dodatnih vanjskih sklopova a time i cijena sustava.

Performanse samog NP4GS3 su testirane u funkciji usmjernika na OC-48 mreži. U tablicu ruta je učitano 30.000 upisa i provjeravala se propusnost usmjernika s obzirom na razne veličine paketa uz poznatu propusnost mreže. Uočeno je da se ne opada propusnost

koja je određena mrežom i veličinom paketa uz prisustvo samog mrežnog procesora, a to je odličan rezultat. Slika 11. prikazuje rezultate izvedenog testa.
Još je jedno svojstvo važno kada se govori o performansama a to je mogućnost spajanja više NP4GS3 procesora u zajednički sustav. Ova arhitektura omogućuje spajanje 64 mrežnih procesora preko DASL-a (*on-chip data aligned synchronous link*) na preklopnik (*switch)* sučelje. Time se mogu dobiti veći sustavi različitih i brojnih funkcionalnosti.



Slika 11: Performanse NP4GS3 u funkciji routera

7. Programska podrška za NP4GS3

IBM programska podrška za mrežne procesore omogućuje da se pomoću simulatora procesora potpuno isproba rad bez korištenja ikakvog mrežnog procesora. Uz standardnu skupinu alata ili SDT (*Software Developer's Toolkit*) kojim se programira i simulira, nudi se u svrhu lakše izrade samih programa dva paketa: BSO (*Base Software Offering*) ili osnovna programska podrška i ASO (*Advanced Software Offering*) za razvoj programske podrške. BSO paket sadrži sljedeće:

- Primjere pikokoda za ostvarenje funkcija kao dijagnostika, obrada okvira, održavanje memorije, *power-up,* rad nad tablicama i sl.
- Pikokod za funkcije obrade i prosljeđivanje paketa

ASO paket uključuje mogućnost rada na različitim platformama i laku portabilnost koda, kod je izvorno pisan za Red Hat Linux ali se lagano prenosi na druge platforme što je i jedan od zadataka ASO-a. ASO uključuje podršku za različite WAN i LAN protokole. Ostale važne stvari ASO-a su:

- Layer 2 functions
- IP forwarding
- Multiprotocol Label Switching (MPLS)

Software Developer Toolkit (SDT) se sastoji od sljedećih alata:

- **Core Simulation Model -** osnovni simulator
- **NPASM** - asembler
- **NPDump** - interpreter pikokoda
- **NPScope** - debager
- **NPSim** - mrežni simulator
- **NPProfile** - koristi se za određivanje performansi sustava
- **NPTest** – generator testnih uzoraka

Zbog kompatibilnosti SDT je dostupan za Sun Solaris, Red Hat Linux i Microsoft Windows platforme. U radu sa SDT-om koristi se *Tool Command Language* (TCL) i *Toolkit* (Tk). Korisnik može pisati vlastite TCL skripte uz postojeće demonstracijske

skripte kao pomoć u korištenju samog *Toolkit*-a. U nastavku se opisuju pojedini alati SDT-a.

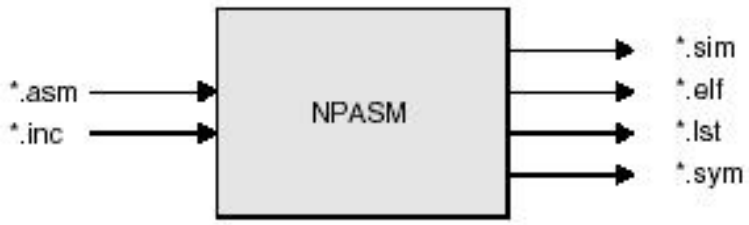
Core Simulation Model

Ovo je simulator funkcija za NP4GS3 mrežni procesor. Može izvoditi oko 1000 pikoinstrukcija u sekundi čime omogućava brzo testiranje.

NPASM

Asembler, proizvodi izvršni kod za NP4GS3. Skup asemblerskih instrukcija je posebno pripremljen za obradu paketa. Rad sa instrukcijama nalikuje na programiranje jezikom više razine budući da je omogućena manipulacija stringovima i cijelim brojevima. Koriste se uvjeti, petlje, makroi i sl. NPASM može ovisno o navedenim opcijama generirati različite izlazne datoteke kao:

- **Picocode load file for the core simulation model (*.sim)** – kod koji se koristi za simuliranje.
- **Symbol table that cross-references label names with addresses (*.elf**) – izvršni kod.
- **Listing of the assembled code (*.lst)** – listing asemblerskog koda.
- **Symbol table listing (*.sym) –** tablica prikaza korištenih simbola, konstanti i sl.



Slika 13: NPASM ulaz i izlaz

NPDump

Ovo je alat koji uzima izvršni kod (*.elf) proizveden NPASM-om, i prikaže ga ASCII tekstom. Omogućuje se programerima da usporede i provjere podatkovne strukture s podacima koji ih predstavljaju.

NPScope

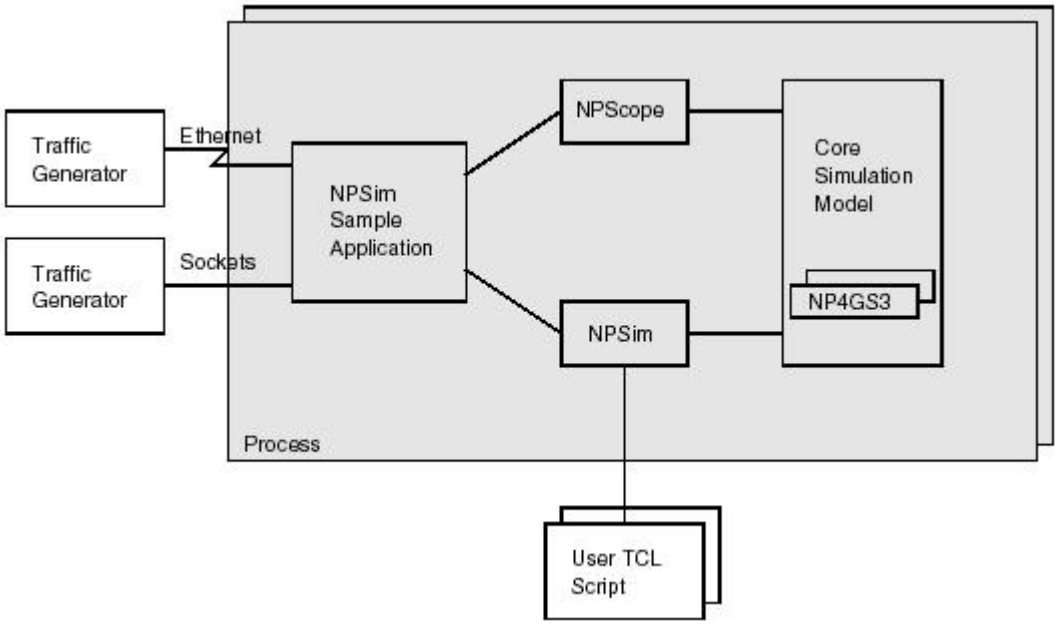
Debugger s TCL/Tk grafičkim sučeljem (GUI). Omogućuje korisniku niz funkcija poput:

- Simulacija i monitorski prikaz registara, podatkovne memorije, instrukcija, redova.
- Prikaz pikokoda i mogućnost izvođenja korak po korak uz upotrebu prekidnih značka.
- Automatsko pronalaženje pogrešaka u pikokodu koje se odnose na krivo referenciranje adresa memorije.
- Konfiguracija Core Simulation Model-a.
- Testiranje NP4GS3 sklopovskih komponenata.

NPSim

NPSim omogućuje upotrebu TCL interpretera na Core Simulation Model-u uz dodane podatkovne primitive. NPSim-om se ostvaruje sljedeće:

- Unos pikokoda u memoriju
- Stvaranje *socket* sučelja sa potrebnim kontrolnim funkcijama koje primaju podatke od simulacijskog procesa
- Spremanje i snimanje prometa paketa
- TCL procedure za prihvaćanje Ethernet okvira



Slika 14: NPSim i usko vezane komponente

NPSim *Sample Application* se koristi za stvaranje testnih mrežnih konfiguracija različitih veličina i kompleksnosti.

NPProfile

Analizator performansi pikokoda. Na temelju podataka dobivenih od NPScope-a generira podatke performansi pikokoda i grafički ih prikazuje. Iz tog grafa se mogu dobiti odgovori na pitanja kao:

- Koliko je potrebno ciklusa za obradu jednog okvira?
- Koje se instrukcije izvode?
- Koliko se često izvode instrukcije i koliko čekaju na izvođenje?

NPTest

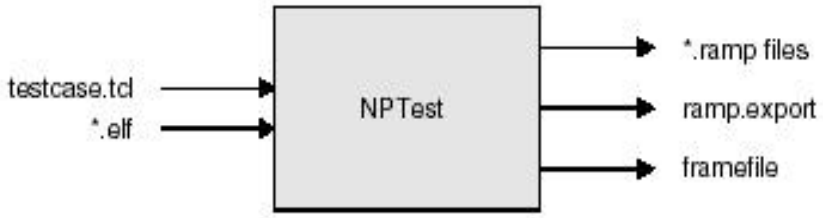
Generator test-primjera. Koristi TCL/Tk skripte i sučelje s funkcijama potrebno za stvaranje test-primjera kod simuliranja. NPScope izvoditi generirani primjer. NPTest nas oslobađa potrebe bilo kakvih vanjskih događaja ili sklopovlja. Neke od važnijih NPTest funkcija su:

- Konfiguracija portova
- Alociranje i inicijalizacija brojila

- Kreiranje kontrolnih i podatkovnih paketa
- Konfiguracija NP4GS3 koprocesora

Kao ulaz NPTest prihvaća:

- Testni primjer s TCL komandama za izvršavanje NPTest funkcija (testcase.tcl)
- Izvršni kod koji je statički alociran od NPTest-a (*.elf)



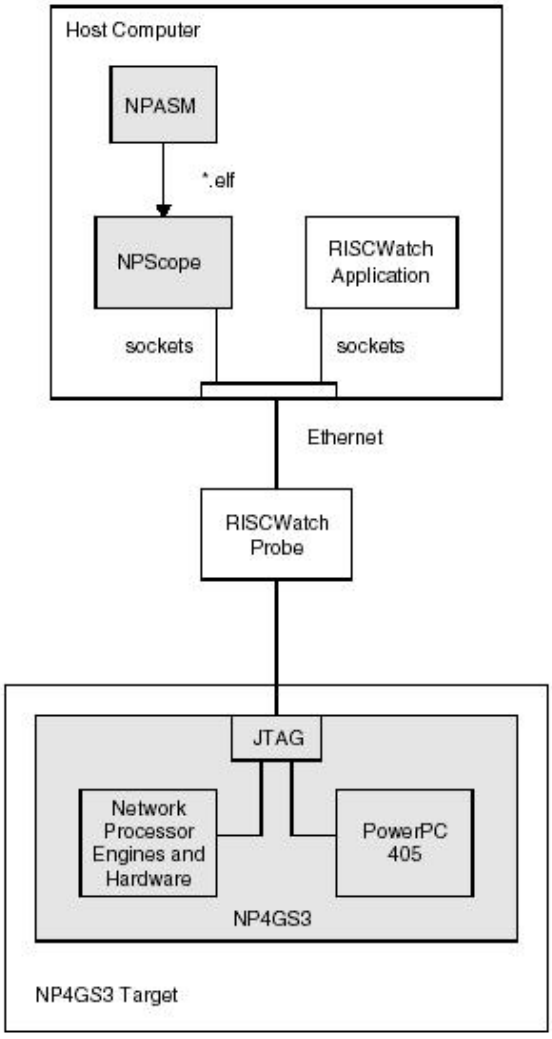
Slika 15: NPTest

NPTest generira na izlaz sljedeće fileove:

- Testni primjer kojeg izvodi NPScope (ramp.export)
- Referenca od ramp.export-a, isto koristi NPScope (*.ramp)
- Generira se kad se stvaraju paketi, koristi NPScoope (framefile)

RISCWatch

RISCWatch aplikacija i RISCWatch sklop nisu u SDT-u i isporučuju se posebno. Slika 16. prikazuje ulogu RISCWatch sklopa (*RISCWatch Probe*).



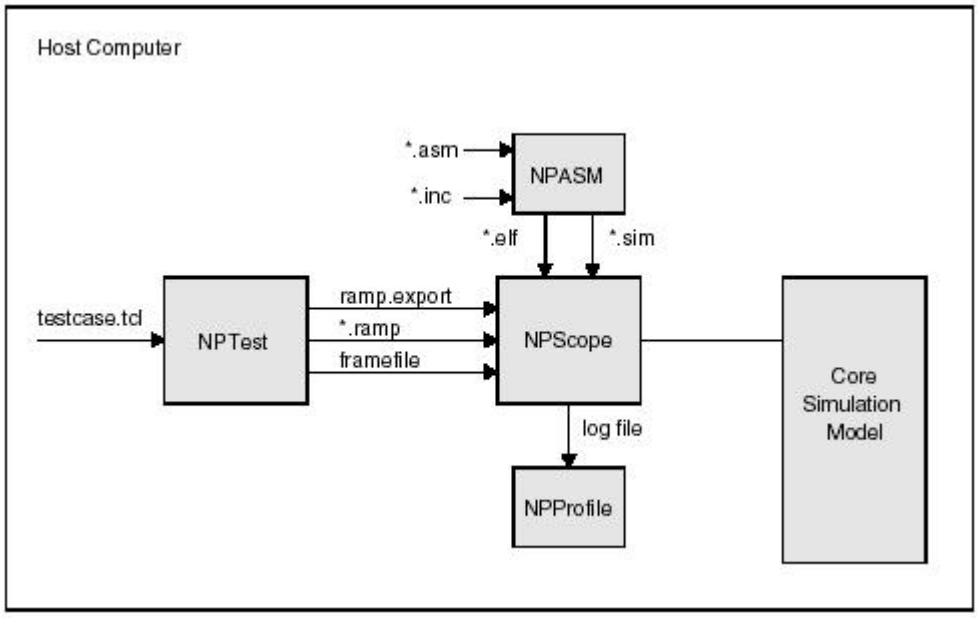
Slika 16: Povezivanje NP4GS3 i host računala preko RISCWatch sklopa

RISCWatch povezuje host računalo preko Etherneta sa NP4GS3 mrežnim procesorom na JTAG port. Preko RISCWatch sklopa je omogućeno simultano debagirati NP4GS3 mrežni procesor s NPScope-om i ugrađeni PowerPC 405 procesor s RISCWatch aplikacijom. RISCWatch nudi korisniku sljedeće mogućnosti:

- Ethernet na JTAG sučelje
- Podrška za spajanje više udaljenih računala
- Korištenje grafičkog (GUI) sučelja
- Mrežna potpora tokom gradnje sustava
- Primjeri C/C++ izvršnih programa
- Detaljne informacije o stanju mreže i sustava

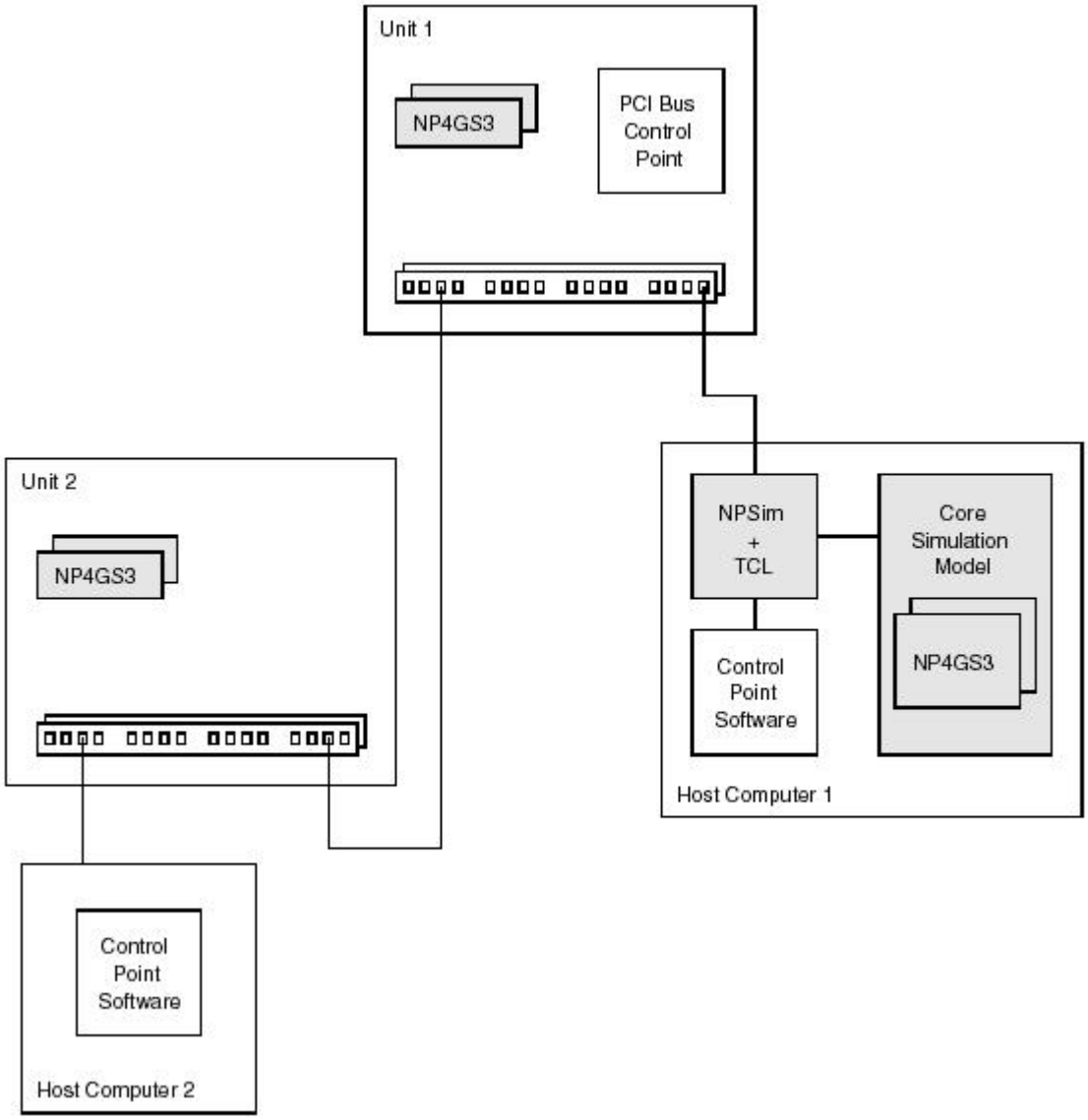
8. Sklopovska i programska integracija

Da bi smo prenijeli izvršni kod NPScope-om u simulator mrežnog procesora, potrebno je prvo koristiti asembler NPASM. NPASM se koristi za dobivanje izvršnog koda. NPScope-om ujedno debagiramo NP4GS3 dok RISCWatch aplikacijom debagiramo kontrolni procesor PowerPC 405. Da bi bilo omogućeno debagiranje uopće ali i sve ostale funkcije koje želimo ugraditi, mora postojati RISCWatch sklop koji povezuje host računalo na kojem se izvodi SDT i ciljni mrežni procesora. Slika 17. prikazuje način korištenja već opisanih SDT alata.



Slika 17: Korištenje SDT alata

Sklopovska i programska integracija vidljive su već pri upotrebi RISCWatch-a na slici 16. Međutim često puta se zahtjeva složenija testna integracija u smislu provjere ispravnosti rada sustava. Budući da se mrežni procesori najviše pojavljuju u funkciji usmjernika ili preklopnika, prikazana je sklopovska i programska integracija sustava u funkciji usmjernika. Slika 18. prikazuje takav testni sustav. Unit 1 na slici 18. predstavlja mrežni uređaj čiji je kontrolni dio procesor spojen na PCI sabirnicu. Unit 2 je isto mrežni uređaj kojemu je kontrolni dio neko vanjsko korisničko računalo 2. Korisničko računalo 1 šalje i prima pakete preko Unit 1 i Unit 2 mrežnog uređaja, odnosno Unit 1 i Unit2 predstavljaju usmjernike. Unit 1 ima dodanu PCI sabirnicu i procesor čime ima i dodatne procesne uloge nad paketima.



Slika 18: Testna konfiguracija u funkciji usmjernika

9.Zaključak

IBM NP4GS3 arhitektura mrežnog prosecora predstavlja hibrid ili programibilni ASIC u svrhu postizanja najboljeg omjera performansi i fleksibilnosti za danu cijenu. Performanse su vidljive iz same sklopovske arhitekture, broju pikoprocesora, raspoložive *on-chip* memorije i ostalih dodatnih sklopova. IBM se uistinu potrudio napraviti bogatu programsku podršku (SDT) kojom je omogućeno simuliranje čitavog rada mrežnog procesora bez potrebe ijednog sklopa. Kada smo zadovoljni sa simulacijom i dobivenim performansama koda (NPProfile alat SDT-a), možemo pristupiti sklopovskom projektiranju. SDT (*Software Developer's Toolkit*) pruža niz alata kojima se postiže jednostavnost programiranja. Prvenstveno NPASM asembler koji omogućuje rad nad stringovima, cijelim brojevima, makroima, petljama, uvjetima i pomalo sliči na jezik više razine. NPSim pruža potporu TCL skripti kojima se pozivaju funkcije za rad nad Ethernet okvirima, otvaranju pristupnih točka (*sockets*) i prebacivanje izvršnog koda u memoriju.
Sam SDT nije dostupan kao *freeware* već ga je potrebno kao i sve dodatne primjere programiranja posebno naručiti. Visoke ocjene na testovima i performansama u mrežnom prometu, dovele su NP4GS3 na mjesto najboljeg mrežnog procesora u 2001. i 2002. godini prema *MicroDesign Resources* izvještaju.

10. Literatura

Korišteni linkovi na Internetu:

- http://www.intel.com/technology/itj/2002/volume06issue03/index.htm
  Intel Technology Journal

- http://www.embedded.com/story/OEG20010730S0053
  Embedded systems programming

- http://www-3.ibm.com/chips/products/wired/products/network_processors.html
  IBM mrežni procesori

- http://www-3.ibm.com/chips/products/wired/products/devcorner.html
  Sve za programere IBM mrežnih procesora, Developers Toolkit

- http://www-3.ibm.com/chips/techlib/techlib.nsf/productfamilies/PowerNP_Network_Processors
  PoewerNP porodica mrežnih procesora

- http://www-3.ibm.com/chips/techlib/techlib.nsf/products/PowerNP_NP4GS3
  Sva potrebna dokumentacija o NP4GS3 mrežnom procesoru