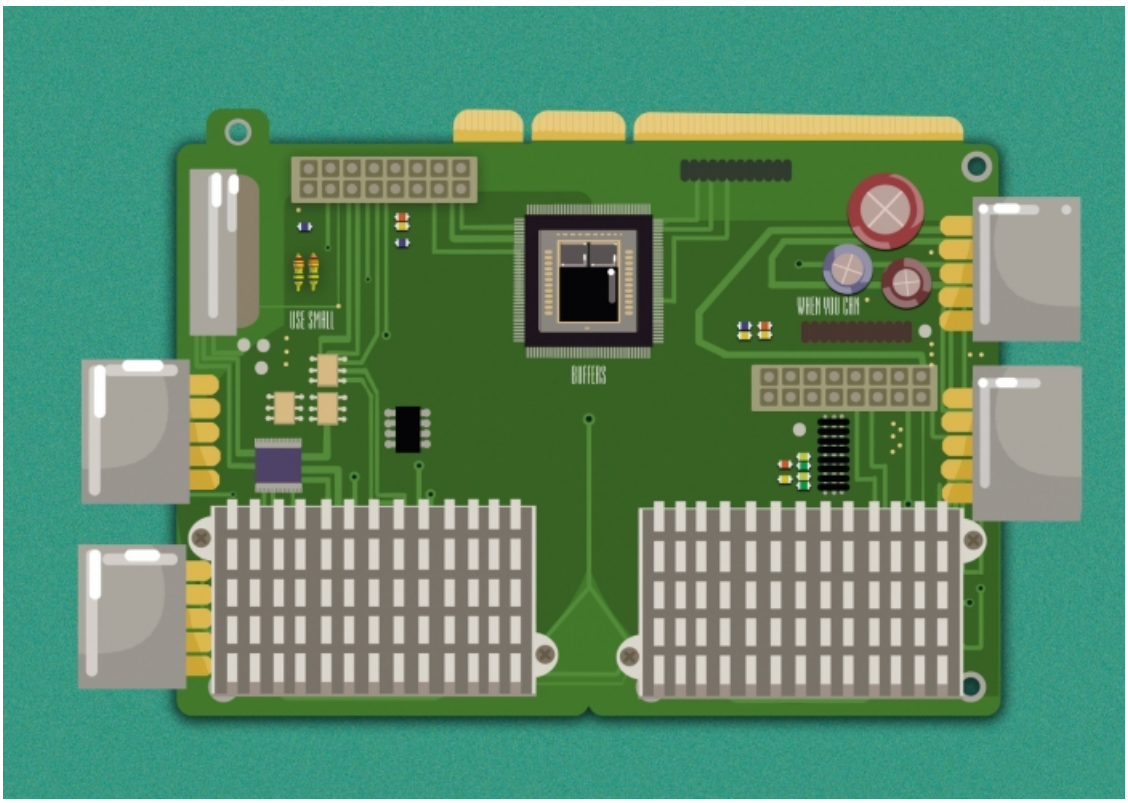
Интеллект сети: как передать пакет

Самый мощный сетевой чип сегодня умеет обрабатывать 25 600 000 000 000 бит за одну секунду. 30 миллиардов транзисторов должны провернуть уйму работы, доставляя биты со входного интерфейса на выходной. Новые типы интерфейсов, увеличивающиеся скорости, утончающиеся техпроцессы, новые функции - бросают ежедневные вызовы инженерам, разработчикам, производителям. У меня давно уже был праздный интерес к аппаратной начинке сетевых коробок, а в этот раз я решил поковыряться отвёрткой ещё глубже и, сняв крышечку с сетевых микросхем, взглянуть на Pipeline"ы да SerDes"ы. Как вы уже, вероятно, поняли, в этой статье будет запредельно много сленга, англицизмов и слов на английском, потому что переводить их на русский неблагодарно, а порой и кощунственно.



Но знаете, какая мысль действительно не даёт мне покоя всю дорогу? Как вся эта сложнейшая конструкция: миллиарды транзисторов, размером около десятка нанометров, напечатанных на пластине размером с большую печеньку, которая полдюжиной тысяч ног припаяна к многослойной плате, напичканной мириадами других микрочипов, сопротивлений и конденсаторов и являющейся частью многоюнитовой модульной коробки, управляемой операционной системой, состоящей из сотен прошивок для разных чипов, инструкций по работе с ними, реализаций стандартизированных и проприетарных протоколов и механизмов поддержания жизнедеятельности многочисленных вентиляторов, блоков питания, линейных карт, фабрик коммутации, интерфейсов, процессоров, памяти - вообще хоть как-то работает, а тем более работает настолько стабильно в течение многих лет. Настоятельно рекомендую ознакомиться с 14-м выпуском СДСМ перед тем, как приступить к чтению.

Содержание

- Терминология.
- Архитектура сетевого устройства.
- Типы чипов.
  - Компромиссы: скорость, функциональность, гибкость, цена.
  - CPU.
  - ASIC.
  - FPGA.
  - NP.
  - Programmable ASIC.
  - Чипы для датацентровых коммутаторов.
- Архитектура ASIC.
  - Физическое устройство.
    - SerDes.
    - PHY, Silicon Photonics.

    - Логическое устройство.
        - Pre-Ingress processing.
        - Parser.
        - Ingress Match-Action.
        - Traffic Manager + MMU.
        - Egress Match-Action.
        - Deparser.
    - Pipeline.
    - Programmable Pipeline.
- Существующие чипы.
    - Commodity/Merchant.
    - Custom.
    - Merchant vs Custom.
- Полезные ссылки.

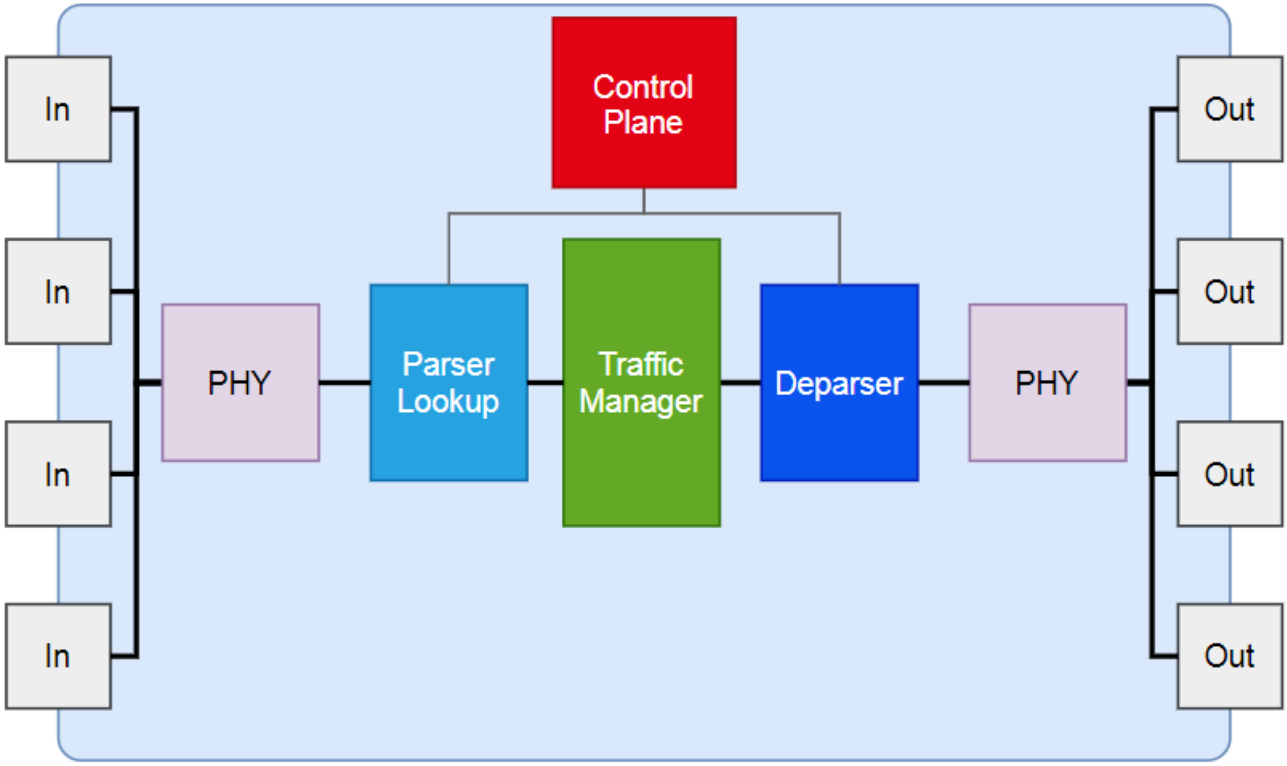
Терминология

Задача этого параграфа не объять все непонятные слова, употребляемые в статье, а лишь внести некую ясность в неразбериху русско-английских терминов.

- **Чип коммутации**, **сетевой процессор**, **(Packet) Forwarding Engine**, **PFE** - микросхема, способная коммутировать пакет из входа в нужный выход с нужным набором заголовков.
- **Pins**, **пины**, **ножки** - металлические контакты на микросхеме для соединения с основанием.
- **Data Path** - путь внутри устройства (или чипа), по которому передвигается пользовательский трафик.
- **Lookup** или **лукап** или **поиск** - поиск адресата в таблицах (FIB, LFIB, ARP Adjacencies, IPv6 ND Table итд.)
- **Pipeline** или **конвейер** - набор действий, которые происходят с пакетом по мере его продвижения от входа в чип до выхода из него.
- **Single-chip**, **одночиповый** - устройство, внутри которого только один чип.
- **Fixed**, **фиксированный**, **pizza-box** - немодульный коммутатор. Обычно внутри него нет фабрик коммутации. Часто эти термины используются как синоним Single-chip, хотя это не совсем верно - внутри может стоять два (back-to-back) чипа или даже больше.
- **Сериализация/Десериализация** - процесс перевода данных из параллельного низкоскоростного интерфейса (МГц) в последовательный высокоскоростной (ГГц) и наоборот. Например, из чипа в интерфейс или из чипа в фабрику.
- **Память** - физическая микросхема для хранения.
- **Буфер** - некий участок памяти, выделенный для хранения пакетов. Здесь и далее в производных словах, таких как "буферов", ударение на "У".
- **Очередь** - абстракция над буфером, позволяющая **виртуально** выстраивать пакеты в упорядоченную очередь. Фактически в памяти они, конечно, хранятся "как попало". **OCB** - On-Chip Buffer - память, встроенная в чип.
- **Programmable ASIC** - ASIC, логику работы которого можно изменить путём перепрошивки.
- **Programmable Pipeline** - чип с возможностью для сторонних компаний программировать конкретные части ASIC в ограниченных пределах с помощью предоставляемого вендором компилятора.

Архитектура сетевых устройств

Итак, что такое сетевое устройство? Будь то коммутатор, маршрутизатор, файрвол, балансировщик, программный или аппаратный, его задача - доставить пакет со входа на правильный выход, и состоит оно из следующих частей:



Электрический или оптический сигнал, попадая на устройство через входной физический порт, восстанавливается до потока битов, из него вычленяются отдельные Ethernet-кадры, далее на основе заголовков (Ethernet, IP, MPLS и т.д. или, иногда, содержимого) принимается решение о том, в какой выходной порт этот пакет должен быть далее отправлен и с каким набором заголовков. На своём пути от входного к выходному порту пакет ещё проходит через модуль Traffic Manager, где с ним могут происходить следующие вещи: буферизация, полисинг, шейпинг, обработка по приоритетам. Это путь самого пакета.

И отдельно от пути пакета - Control Plane Module, который отвечает за то, чтобы путь вообще появился. Это всяческие протоколы маршрутизации, обмена меток и прочее.

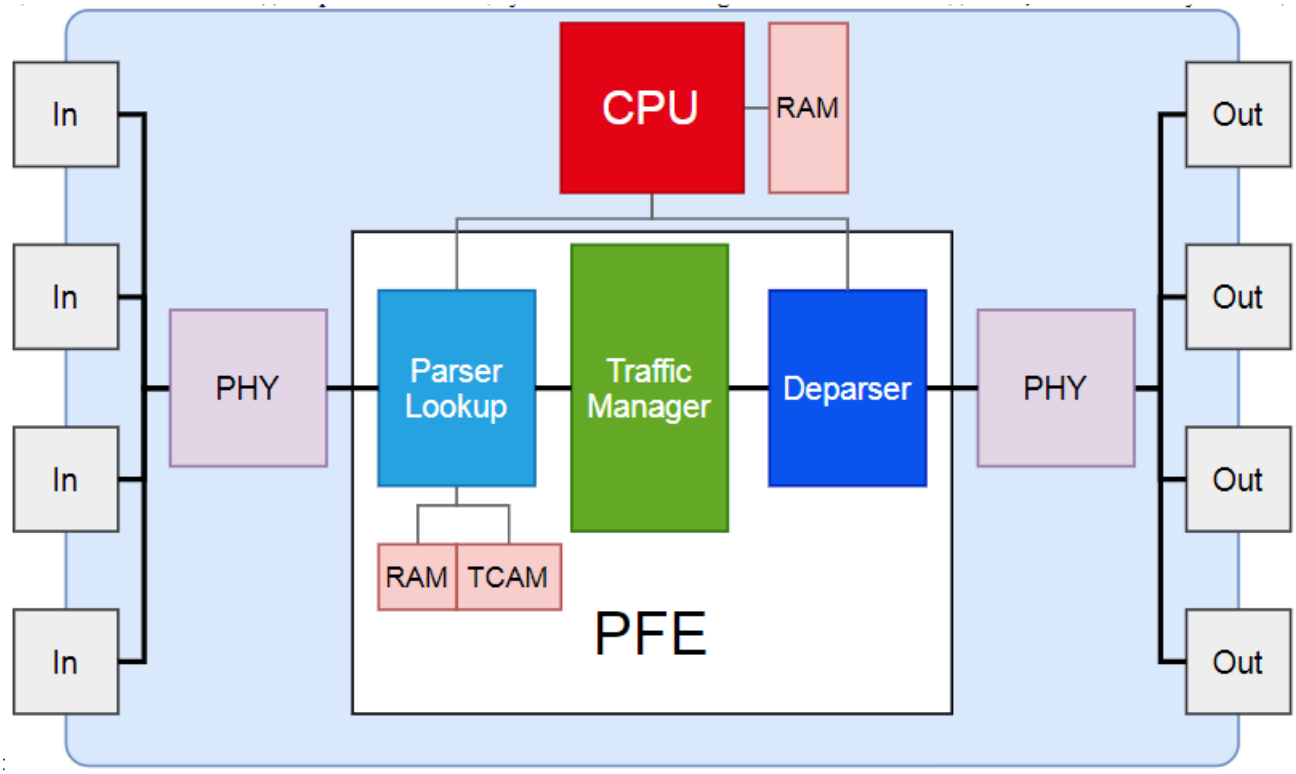
Это компоненты, которые присутствуют всегда и во всех сетевых устройствах.

Реализация же этих функций уже зависит от того, о чём именно мы говорим. Например, на обычном x86 всю работу, кроме PHY могут взять на себя CPU и оперативка. Более типично, что функции канального уровня заберёт на себя NIC - Ethernet, проверка контрольных сумм. А можно в компьютер доставить SmartNIC"и, которые аппаратно могут делать, например, туннелирование. Но мы не будем сегодня про программные реализации сетевых функций. Поговорим о старых добрых материальных коробках, которым всё равно никуда никогда не деться.

Вообще об этом я уже писал, поэтому повторяться не буду. Точнее буду, но не сильно. Точнее сильно, но я добавлю здесь ещё смысла.

Обычно на каждый блок задач выделяется специализированный чип. Так, всем Control Plane"ом занимается всё тот же CPU+память. Организация взаимодействия со средой передачи и преобразование битов в сигнал и наоборот - специальные чипы PHY. Почти всегда они реализуются на ASIC. Разбор заголовков и поиск пути - Packet Forwarding Engine. Это может быть ASIC, Network Processor, реже FPGA и даже CPU. К ним в помощь идёт или обычная память RAM или специальная CAM/TCAM для хранения таблиц лукапа. Traffic Manager - если вынесен отдельно, то опять же - узкоспециализированные ASIC"и и плюс к ним память. Но может быть встроенным в чип коммутации.

Один из вариантов реализации (single-chip устройство):



В этой статье сосредоточимся на PFE и TM, которые и могут вносить вариативные задержки в доставку, потому что могут хранить пакеты.

Типов-Чипов

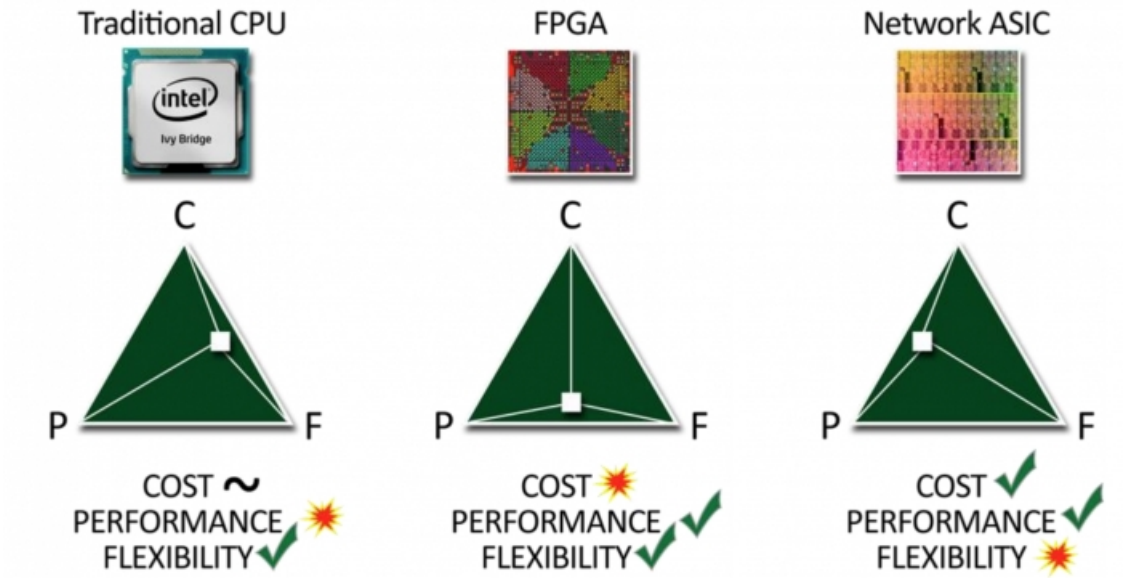
Было очень страшно начинать эту статью, потому что чипов чудовищное многообразие и простого короткого ответа на вопрос, какой выбрать - нет. Речь и про типы, и про изготовителей, и про серии, и про характеристики. Ниже я попытаюсь разложить все эти штуки на составляющие детали.

А начнём с типов чипов.

Компромиссы: скорость, функциональность, гибкость, цена

В IT всё есть компромисс. Всегда приходится чем-то жертвовать во благо другого.

Вот классический цискин треугольник про компромиссы:



Здесь не хватает ещё NP и Programmable ASIC. Источник.

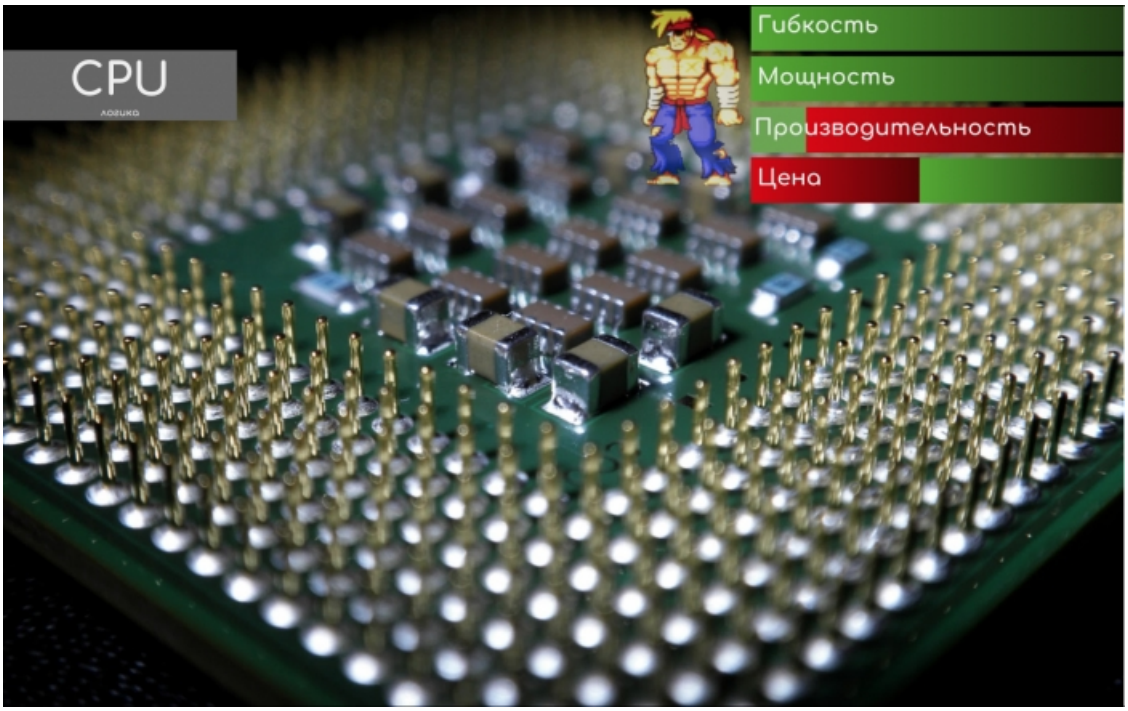
Вендоры всегда балансируют в периметре этого треугольника.

Нельзя сделать CPU, проворачивающий через себя 25 Тб/с. Нельзя сделать универсальный ASIC - зачастую они могут быть аппаратно ограничены по функциям (например, **или** VXLAN Lookup **или** IP). Нельзя сделать дешёвый FPGA.

Кроме того, при таком количестве портов, сегодня вступают в игру энергопотребление, место в стойке, тепловыделение и, конечно, цена.

Итак, на сегодняшний день существуют следующие типы чипов, которые могут быть использованы в качестве пакетных процессоров:

CPU



- Неограниченная гибкость.
- Неограниченная функциональность.
- Приемлемая цена.
- Полный провал по производительности.

Идея в том, что всё управляется кодом. Инструкции записываются в оперативную память. Для обработки каждого пакета потребуется сходить много раз из CPU в RAM. С другой стороны для изменения логики работы достаточно переписать программу. Обновления CPU не требуется.

Область применения: домашние и SOHO-маршрутизаторы, устройства уровня доступа, файрволы, DPI и т.д. Например, абсолютно все рутеры Mikrotik используют CPU для маршрутизации пакетов. Иными словами, CPU годится там, где не гонимся за ультраскоростями, а важен широкий набор функций и невысокая цена.

Впрочем, не без исключений: бывает, что и большие штуки коммутируют в CPU.

Важное замечание: CPU является необходимой частью любого сетевого устройства, потому что берёт на себя задачи Control Plane. А это автоматически означает, что ему придётся работать с протокольным трафиком: OSPF, BGP, LDP, LLDP итд. Кроме того, есть exception-трафик - когда у пакета TTL истёк или когда у него стоит бит Router Alert. Ещё CPU нужно самому генерировать трафик тех же протоколов - Self-generated. Можно ли считать это участием в коммутации? Скорее да, чем нет.

Программная маршрутизация на CPU

Для последнего десятилетия характерна тенденция к маршрутизации в софте. Для всех сетевых функций предлагаются программные альтернативы. Отсюда и DMA, DPDK, VPP, SR-IOV, которые и правда позволяют творить невиданные прежде вещи.

Более того, современные CPU обладают дополнительными блоками инструкций. У Intel это **SSE** - Streaming SIMD Extensions, позволяющие значительно ускорить обработку трафика. Тут обычные CPU

уже заходят в зону NP (Network Processor) - процессоров, которые можно программировать на языках высокого уровня вроде C, и обладающих большим набором спец. инструкций для работы с сетевым трафиком.

Одним из узких мест современных процессоров ещё является шина доступа - PCIe. В один процессор сейчас более-менее можно загнать 100 с небольшим Гбит/с. Но как бы производители программных решений ни продвигали идею, что "все можно сделать в софте", однако скорости выше 500 Гбит/с пока что можно достигать только с помощью специализированных асиков. И давайте ещё прикинем. Выдержка с сайта про VPP:

> Recent testing of FD.io release 17.04 shows impressive gains in performance on Intel’s newest platform when switching and routing layer 2⁄3 traffic. With the prior generation Intel® Xeon® Processor E7-8890v3, FD.io testing showed aggregate forwarding rate of 480 Gbps (200 Mpps) for 4-Socket machine (using 4 of E7-8890v3 CPU configuration); however, the same FD.io tests run on two 2-Socket blades (e.g. a modern 2RU server) with the new Intel® Xeon® Platinum 8168 CPUs (using four of 8168 CPUs in two by two-socket configuration), within the same power budget, show increase of forwarding rate to 948 Gbps (400 Mpps) benefiting from the PCIe bandwidth increase of the new CPUs, and the overall decrease in cycles-per-packet due to CPU micro-architecture improvements.

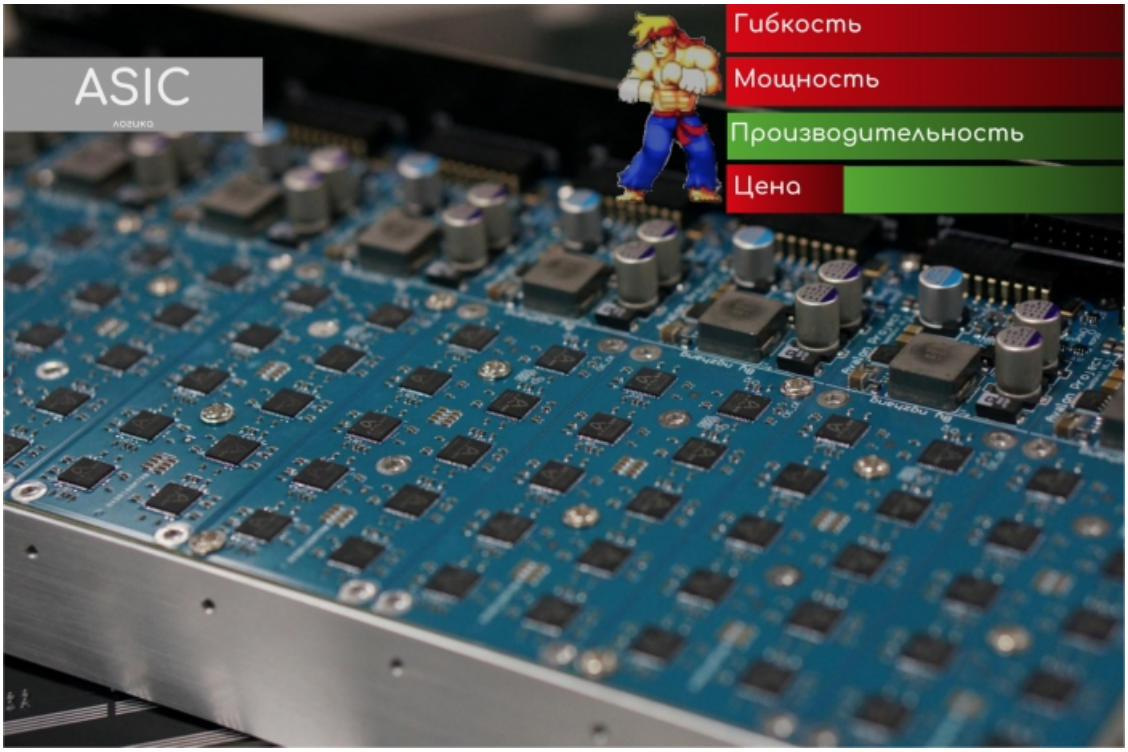
А теперь посчитаем.

Xeon E7-8890v3: Рекомендуемая цена $7174.
4 проца по 18 ядер = 72 ядра = 480 Gbps (200 Mpps)
$28696 только за процы.

Xeon Platinum 8168. Рекомендуемая цена% $5890.
2 проца по 24 ядра = 48 ядер = 948 Gbps (400 Mpps).
$11780 только за процы.

Без обвязки. А ещё кушать электричества он будет как голодный шакал. Не самый дешёвый получится рутер. Зато гибкий.

ASIC



- Околонулевая гибкость.
- Ограниченная функциональность.
- Низкая цена.
- Ультравысокая производительность.

Идея в том, что инструкции закодированы аппаратно в виде транзисторов. Сначала очень долго пишется код, реализующий логику на специальном языке программирования, вроде Verilog, далее он

преобразуется в интегральную схему, отлаживается, проверяется и отправляется в тираж. После этого поменять что-то в логике чипа можно, только произведя новый чип. Каждый пакет обрабатывается, просто прогоняясь по конвейеру из транзисторов, совершающих заранее определённые действия. Это называется Pipeline.

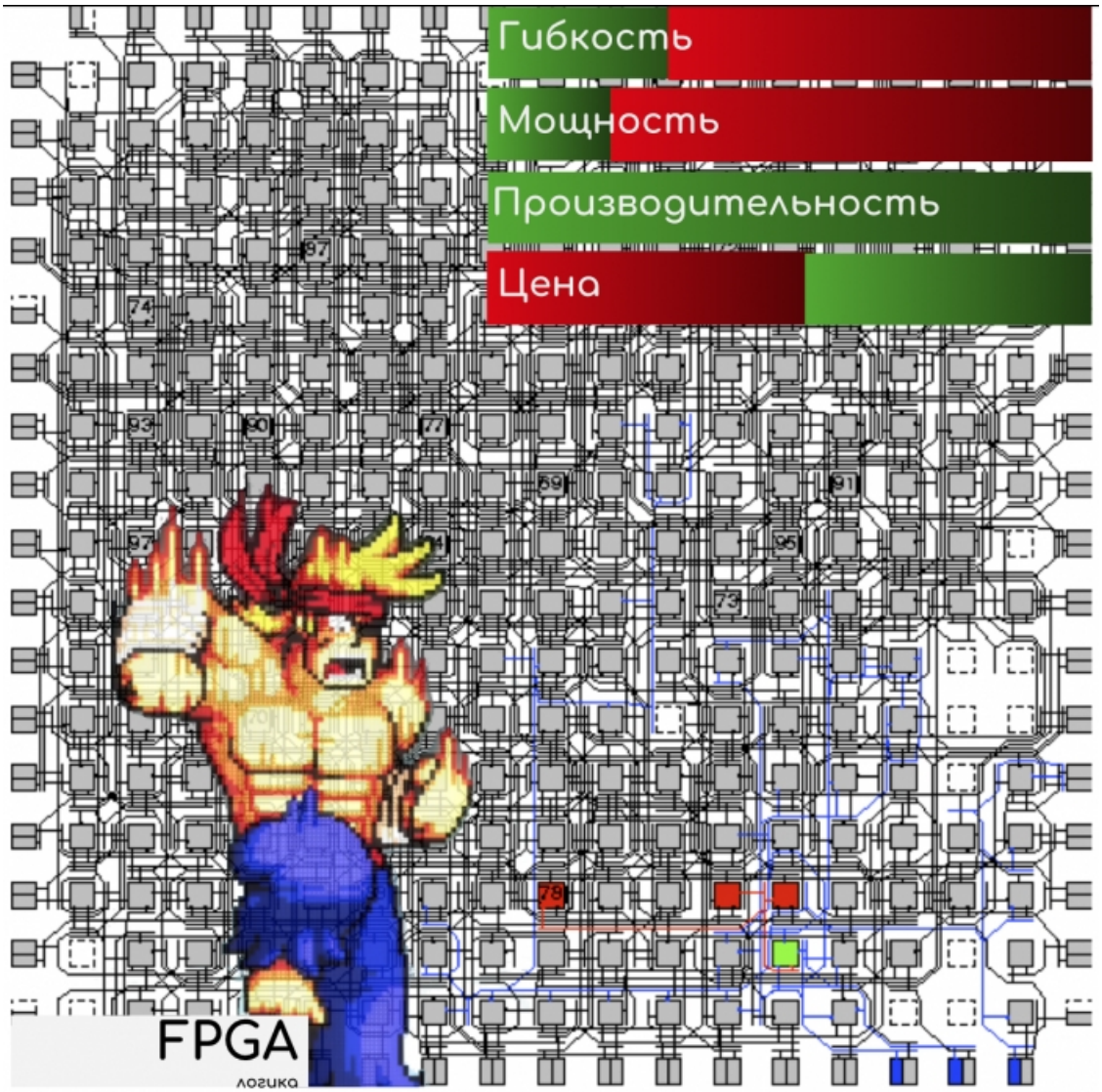
Область применения: почти любые коммутаторы и многие маршрутизаторы.

Впрочем, не без исключений: Juniper в своей линейке маршрутизаторов MX многие годы использует ASIC Trio. Вообще по книге об MX: PFE это блок ASIC"ов: PFEs are made of several ASICs, which may be grouped into four categories:

- Routing ASICs: LU or XL Chips. LU stands for Lookup Unit and XL is a more powerful (X) version.
- Forwarding ASICs: MQ or XM Chips. MQ stands for Memory and Queuing, and XM is a more powerful (X) version.
- Enhanced Class of Service (CoS) ASICs: QX or XQ Chips. Again, XQ is a more powerful version.
- Interface Adaptation ASICs: IX (only on certain low-speed GE MICs) and XF (only on MPC3E)

Речь здесь о классических ASIC - **Applicaton Specific** Integrated Circuit - статических кусках кремния с аппаратной логикой. Последние лет 10 в области сетевых микросхем произошёл сдвиг в направлении программируемых ASIC"ов, о которых чуть ниже.

FPGA



Русский термин - ПЛИС - Программируемая Логическая Интегральная Схема.

- Вполне удовлетворительная гибкость.
- Вполне удовлетворительная функциональность.
- Цена успешного полёта Апполона до Луны и обратно.
- Отличная производительность.

В отличие от ASIC"ов, где на транзисторах реализованы сами функциональные блоки, в FPGA транзисторами реализуются базовые строительные блоки - регистры, память, LUTы. Из которых потом **можно** создавать нужные функциональные блоки. Что это даёт? А то, что FPGA полностью программируемый - логику работы блоков, из которых он состоит, можно поменять. Для этого потребуется обновить прошивку чипа, что менее удобно, чем с CPU, но гораздо удобнее, чем ASIC. Так, если поддержка какой-то функции (условный Geneve) не была заложена изначально, её всегда можно добавить потом новой прошивкой.

Однако за такую программируемость приходится дорого платить.

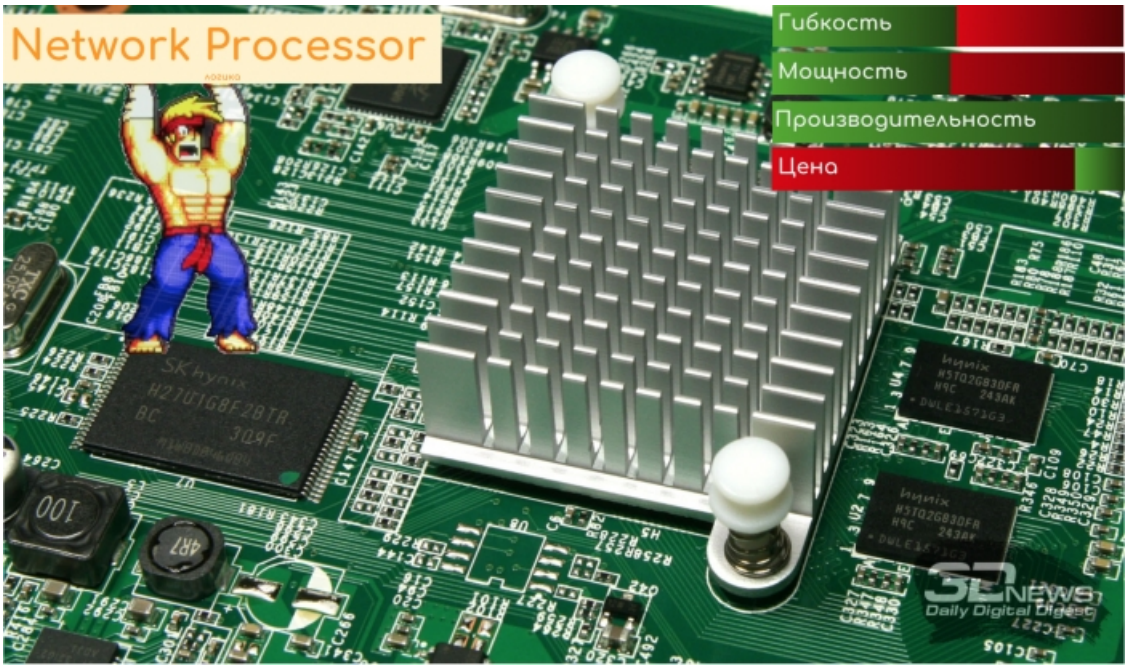
Область применения: POC или низкоскоростные решения для энтерпрайз-сегмента.

Впрочем, не без исключений: собеседовался я как-то раз в контору, в которой модульную коробку для операторов собирали полностью на FPGA, включая фабрику.

У этого даже есть основания: задолго до появления Programmable ASIC"ов на FPGA можно было делать любую обработку пакетов. И даже через несколько лет после производства плисину легко перепрошить и получить поддержку новой функции. Автору неизвестны вендоры, которые бы на ПЛИС сделали PFE на скорости более 100 Гбит/с, по всей видимости, потому что частная компания не обладает для этого достаточным капиталом. Но для рынка энтерпрайз такие решения могут вполне подойти.

Однако, я слышал, что в процессе разработки ASIC возможен такой подход, когда сначала разрабатывается FPGA, программируется нужным образом, тестируется, а потом с неё делают слепок для производства ASIC. Но пруфов нет.

NP - Network Processor



- Отличная гибкость.
- Отличная функциональность.
- Цена весьма высокая.
- Производительность весьма высокая.

NP или **NPU** - Network Processor Unit. Идея в том, что это почти CPU, который однако заточен под сетевые задачи и изготавливается специально под них. Он, как и CPU, обычно состоит из нескольких ядер, каждое из которых отвечает за свой сегмент. Для изменения логики так же достаточно переписать код приложения. NP позволяет делать более сложные штуки - например, выполнять циклы (чего лишены ASIC и FPGA), делать NAT, почти любые инкапсуляции, пушить и попать условно произвольное число меток и т.д. Долгое время NP позиционировался, как серебряная пуля для всех сетевых приложений. Но производительность уступает ASIC"ам и FPGA.

Большим преимуществом является то, что писать программы для NP можно на C. Это значительно ускоряет процесс, кроме того, где-то можно переиспользовать код.

Область применения: маршрутизаторы агрегации и ядра.

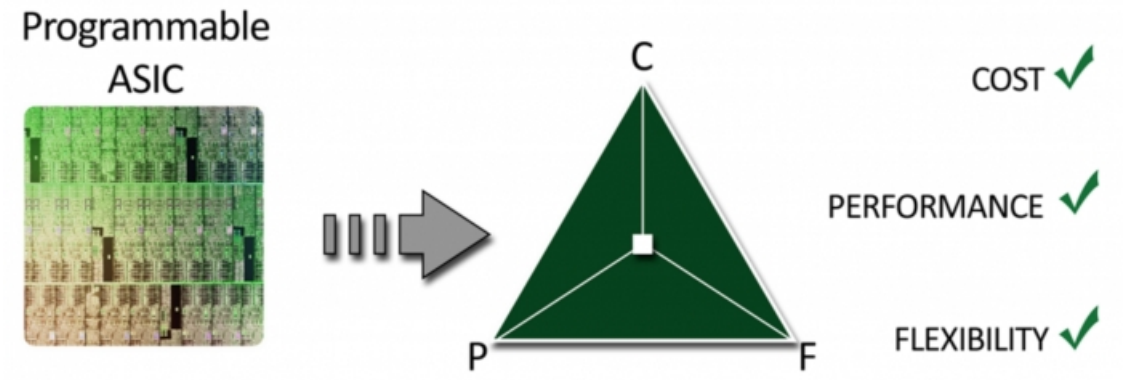
Впрочем, не без исключений: например Smart-NIC Netronome в начале своего пути использовал Intel IXP.

Programmable ASIC



- Приемлемая гибкость.
- Ограниченная функциональность.
- Низкая цена.
- Ультравысокая производительность.

А вот это уже настоящая серебряная пуля последнего десятилетия. "Почему бы нам не взять ASIC и сделать его немножечко программируемым?" - таким вопросом, видимо, задались разработчики и выдали замечательную вещь, которую циска в своём треугольнике поместила в самую середину, хотя это и не совсем так, потому что производительность программируемого ASIC"а такая же, как и у обычного. Им удалось вырваться из 2D.



Область применения: коммутаторы, маршрутизаторы. Большинство датацентровых коммутаторов и некоторые маршрутизаторы уровня границы ДЦ работают на программируемых асиках.

Чипы для датацентровых коммутаторов

Чтобы упростить себе жизнь, я продолжу далее разговор только об ASIC"ах под датацентровые коммутаторы, не пытаясь обнять Джабба Хатта.

До недавних пор на этой ниве пахал только Broadcom со своей оружейной палатой: Tomahawk и Trident - и израильскими городами: Qumran, Jericho и т.д. Выбор - особо не разбежишься - ну или разрабатывать своё (как делают Huawei, Juniper и Cisco).

Сегодня конкуренцию ему пытаются составить Mellanox со своими собственными чипами Spectrum (ныне уже Nvidia), Innovium Teralynx, Barefoot Tophino (ныне Intel). Своим появлением эти компании раскачивают рынок и провоцируют среди вендоров тренд на переход от внутренних разработок к готовым чипам их производства.

Мы в конце статьи взглянем на их модельные ряды, но пока давайте обсудим, чем же чипы

характеризуются и могут отличаться друг от друга.

А для этого надо понять, какие они задачи решают.

В датацентровых сетях есть три основных типа устройств:

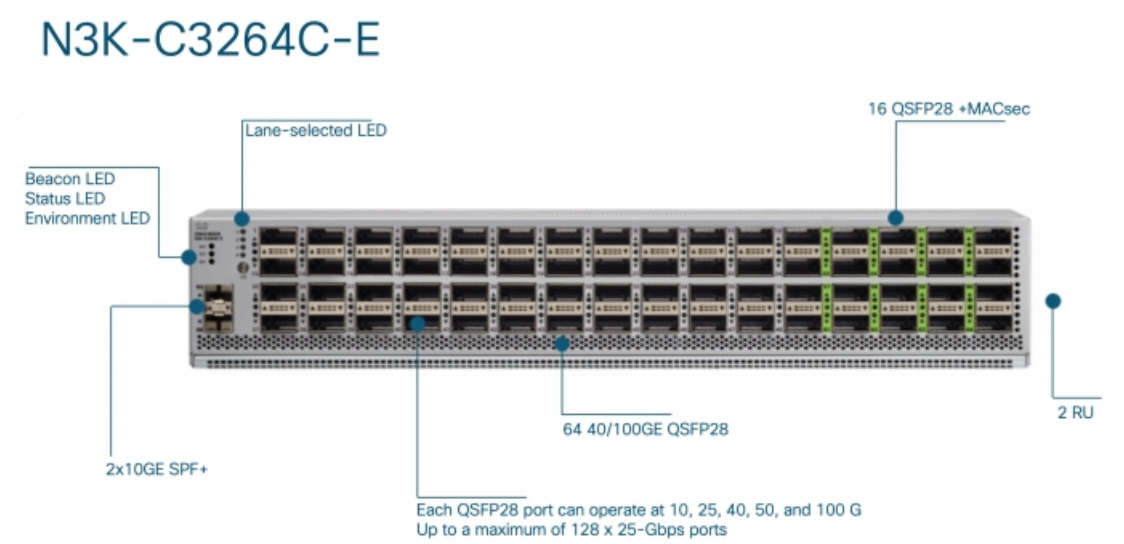
Spine - сравнительно простая железка, требующая самый минимум функций - её задача просто молотить трафик. Очень много и очень быстро. Зачастую, это просто IP-маршрутизация. Но бывают и топологии, в которых Spine играет чуть более важную роль (VXLAN anycast gateway). Но обычная практика - держать конфигурацию спайнов максимально простой.

Leaf - чуть более требователен к функциям. Может терминировать на себе VXLAN или другие оверлеи. Здесь могут реализовываться политики QoS и ACL. Зато не нужна такая большая пропускная способность, как для спайнов. Кроме того, в некоторых сценариях (VXLAN) leaf знает о сервисах за подключенными машинами (клиентских сетях, контейнерах), соответственно, ему нужно больше ресурсов FIB для хранения этой информации.

Edge-leaf - это устройства границы сети ДЦ и здесь уже фантазия ограничивается только свободой мысли сетевых архитекторов - MPLS, RSVP-TE, Segment Routing, всевозможные VPN"ы. При этом, наименее требовательны к производительности.

На каждом устройстве, соответственно, разные требования к возможностям чипов - как по пропускной способности, так и по набору функций и по количеству ресурсов для хранения чего-либо.

И надо сказать, вендоры чипов и железа добились тут поразительных успехов. Типичный спайн сегодня - это 64-128 100GE портов на 2-4 юнита с энергопотреблением около 400 Вт. И ценой порядка пары десятков тысяч долларов.



Производителям чипов приходится нелегко не только из-за попыток найти золотую середину, но и из-за возрастающих скоростей передачи данных и конкуренции. Средняя скорость аплинков с торов сегодня 200-800 Гб/с. Чтобы собрать минимально рабочую сеть ДЦ, нужны спайны с пропускной способностью 3,2 Тб/с.

Всё более и более производительные чипы нужно выпускать уже примерно каждые полтора-два года.



Актуализированная мной картинка из видео PP.

Конкурирующие производители чипов идут ноздря в ноздрю - почти одновременно у всех (Broadcom, Mellanox, Innovium, Barefoot) выходят микросхемы с почти идентичными характеристиками, а вслед за ними и коммутаторы.

Ещё одним компромиссным вопросом является размер буфера, но об этом мы поговорим в следующей статье.

Помимо скорости и обязательных функций по маршрутизации и оверлеям, есть ещё много менее

заметных вещей, которые ожидают потребители. Мы про них говорить сегодня не будем, но не упомянуть было бы ошибкой.

Это, например, **телеметрия** в реальном времени: наблюдать за утилизацией буферов, видеть бёрсты, дампы отброшенных пакетов, профиль трафика по размерам и типам пакетов. Кроме того, сегодня набирает популярность **INT** - Inband Network Telemetry.

Для многих незаметно, но уже почти жизненно важно, начинает работать **динамическая балансировка трафика**: чип отслеживает потоки (flows) и дробит их на флоулеты (flowlets) - короткие куски трафика одного потока, разделённые между собой паузой в несколько миллисекунд. Эти флоулеты он может динамически распределять по разным путям (ECMP или членам LAG), чтобы обеспечить более равномерную балансировку. Особенно важно это для Elephant Flows, оккупирующих один интерфейс.

Пользователям всё чаще хочется иметь возможность **управлять распределением буфера**, ну а **перераспределение** ресурсов FIB - это уже функциональность, отсутствие которой будет вызывать вопросы.

В условиях датацентров, ECMP и балансировка силами сети - это воздух, вендорам нужно обеспечить нужное количество как **ECMP-групп**, так и общее **количество Next-hop"ов**.

Поэтому нет одного чипа или тем более SoC, решающего сразу все задачи. Под каждую роль разработаны свои чипы. Одни из них ориентированы на пропускную способность, другие на широкую функциональность, третьи на низкие задержки.

Посочувствуем же бедным вендорам и будем выбирать долларом.

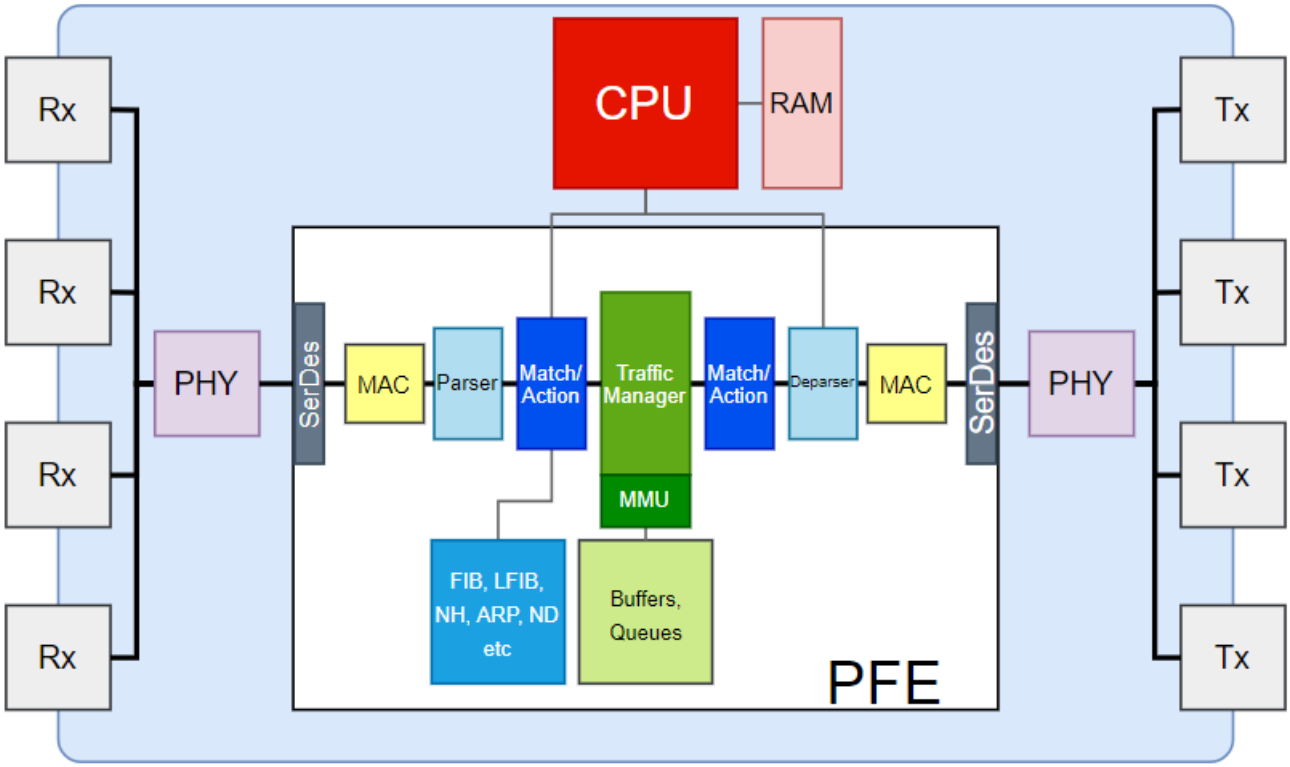
Архитектура сетевых ASIC

Сначала мы взглянем на физиологию чипа - из каких компонентов он состоит. А далее разберёмся что с пакетом в этих компонентах происходит.

Физическое устройство

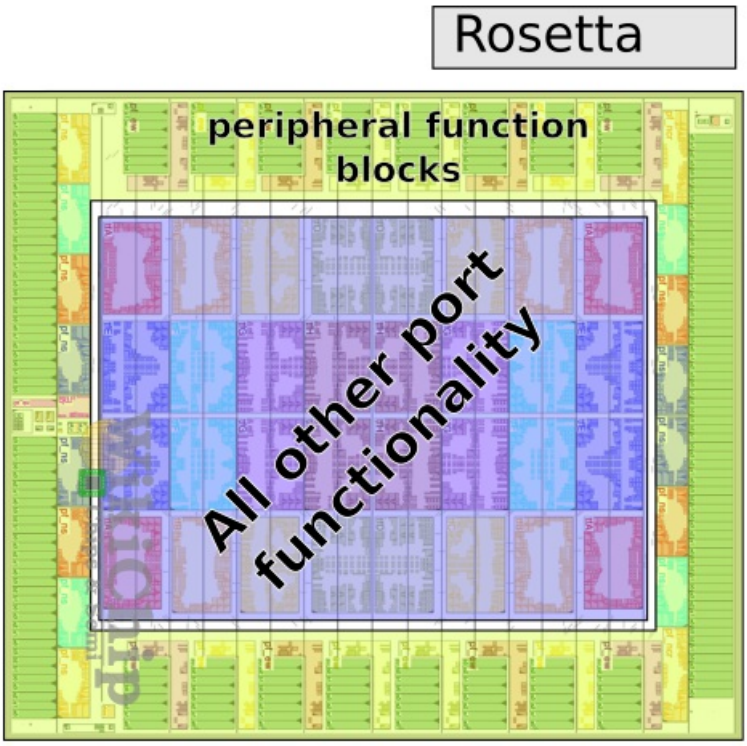
Итак, для успешной коммутации пакета нужны следующие блоки:

- Парсер заголовков (Parser).
- Лукап (Match): FIB/LFIB, Nexthop-группы, ARP Adjacencies, IPv6 ND Tables, ACL и т.д.
- Блоки преобразований (Action).
- Блок управления памятью (TM/MMU).
- Сборщик пакета (Deparser).
- SerDes.
- Память для буферизации пакетов.
- Блок, реализующий MAC.
- Чип PHY.
- Физические порты/трансиверы.



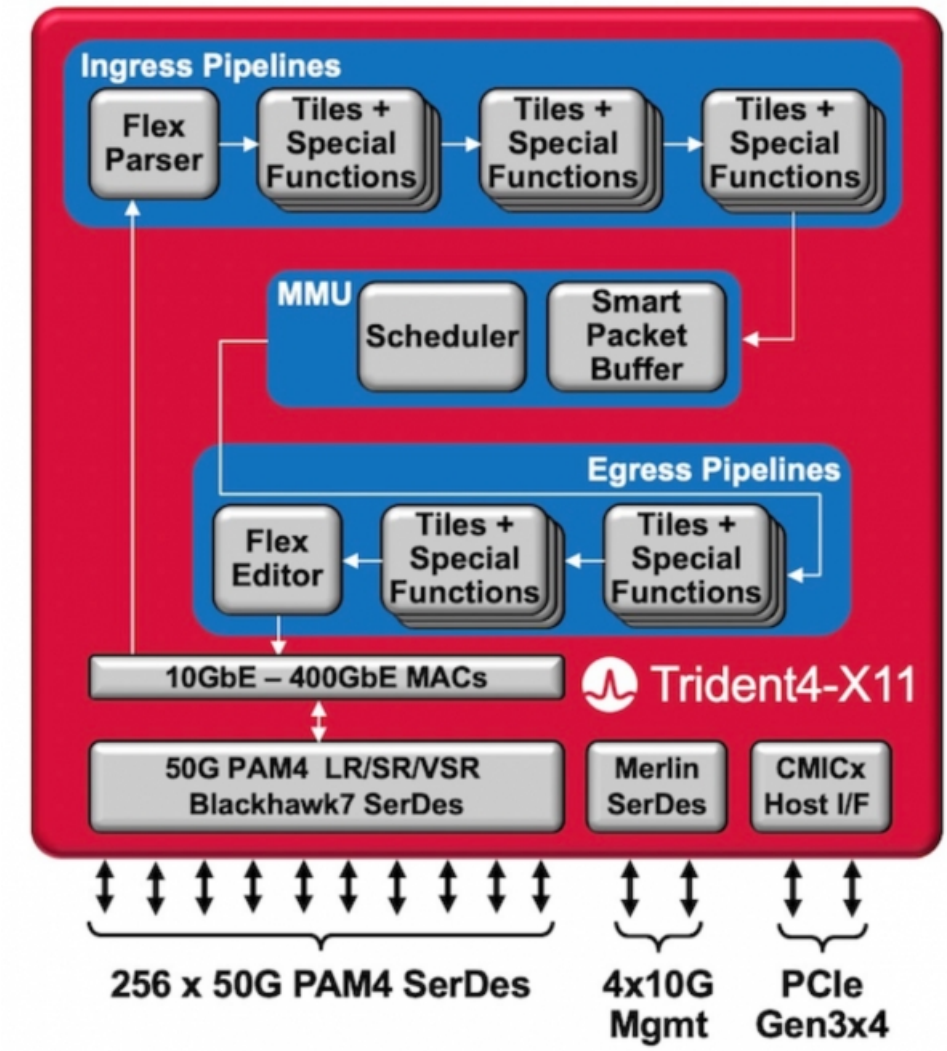
Крупными мазками: оптический или электрический сигнал попадает на порт (**Rx**), тот его передаёт на **PHY**. Модуль PHY реализует функции физического уровня и передаёт биты на входные пины PFE, где сигнал блоками **SerDes** преобразуется в удобоваримый для чипа вид. Блок **MAC** из потока битов восстанавливает Ethernet-кадр и передаёт его **парсеру**. Парсер отделяет необходимые ему заголовки и передаёт их на анализ следующему блоку **Match/Action**. Тот их исследует и применяет нужные действия - отправить на правильный порт, на CPU, энкапсулировать, дропнуть итд. Тело пакета всё это время хранилось в **буферах**, управляемых **MMU**, и теперь пришло время **Traffic Manager"y** проводить все обряды QoS. И потом процесс раскручивается в обратную сторону. Снова **Match/Action**. Потом собрать пакет с новыми заголовками (**Deparser**), преобразовать кадр в поток битов (**MAC**), сериализовать (**SerDes**), осуществить действия физического уровня (**PHY**) и передать через выходной порт (**Tx**) в среду.

В простейшем случае, вообще почти все блоки являются частью одного монолитного кристалла кремния. То есть они - продукт одного процесса печати на "вафле". Отдельные, составляющие чип компоненты, реализующие законченный набор функций, называются IP-core (не тот, что ты мог подумать, сетевой инженер!). То есть, SerDes, MAC, TM - это всё отдельные IP-core. Зачастую, они производятся сторонними компаниями, специализирующимися конкретно на данных компонентах, а потом встраиваются в микросхему. Особенно это касается SerDes - сложнейшей детали, в которую вендоры сетевых чипов не готовы вкладывать силы R&D. Один из крупных производителей SerDes - Inphi.

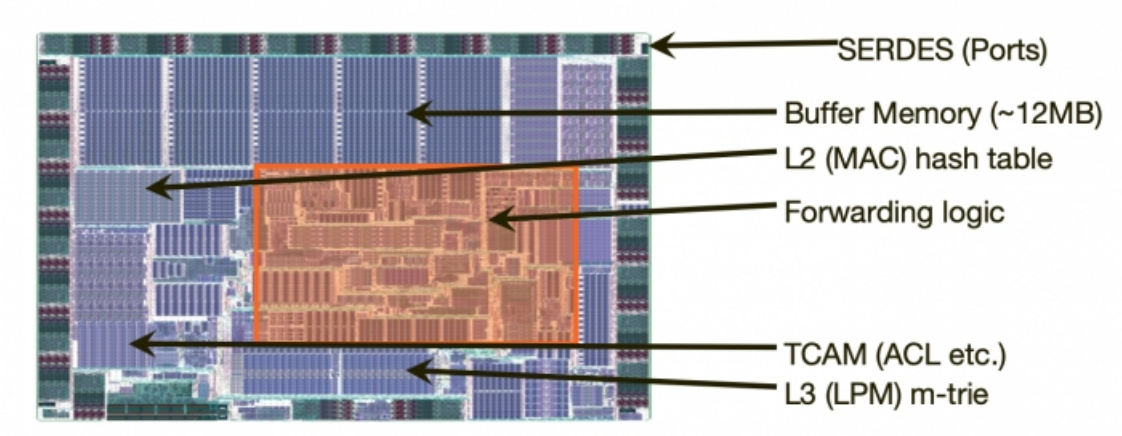


Монолитный чип Rosetta. Источник.

Все свои чипы Broadcom позиционирует как монолитные:

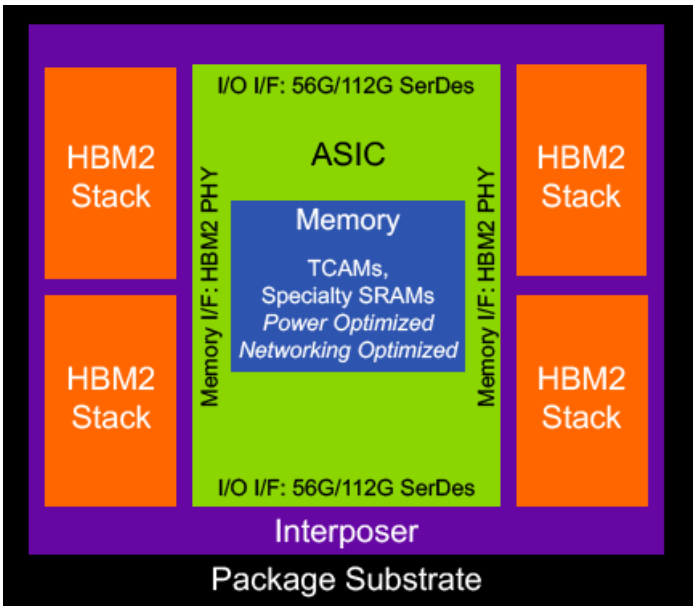


Источник.

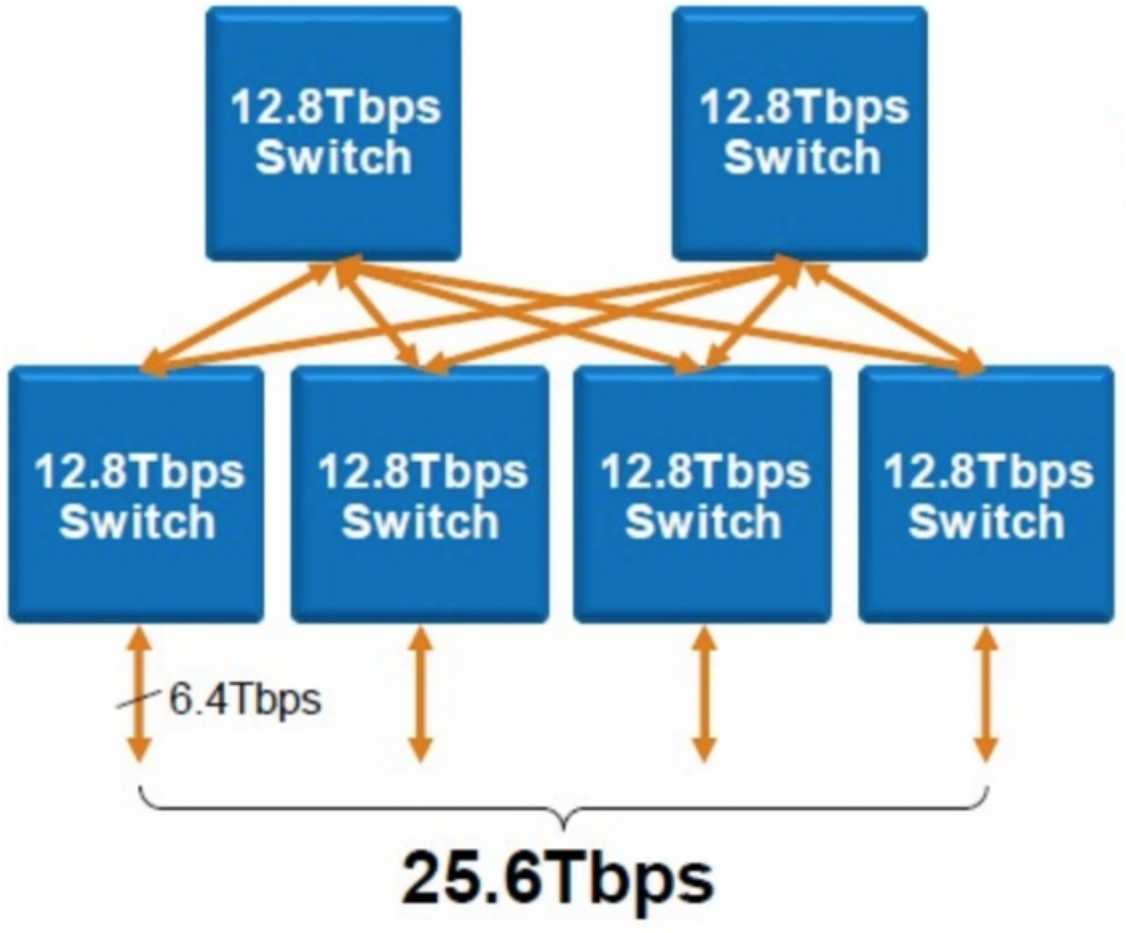


Источник.

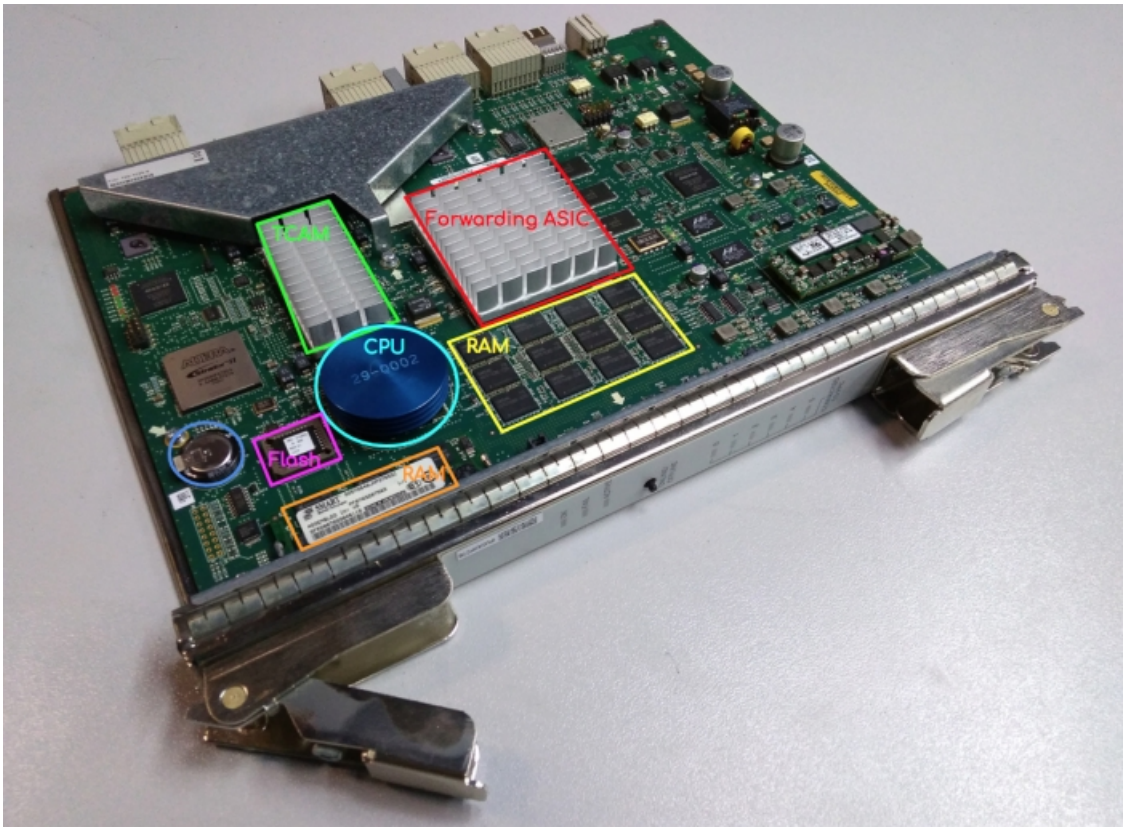
Другой распространённый вариант: в одном чипе сочетать несколько разных кусочков с интерконнектом между ними. Так, например, off-chip память HBM коммерческого производства выносится за пределы кристалла сетевого ASIC:



Под крышкой одного производительного чипа могут быть собраны несколько, так называемых, менее производительных чиплетов (chiplet), которые, объединённые в фабрику, дают бóльшую пропускную способность:

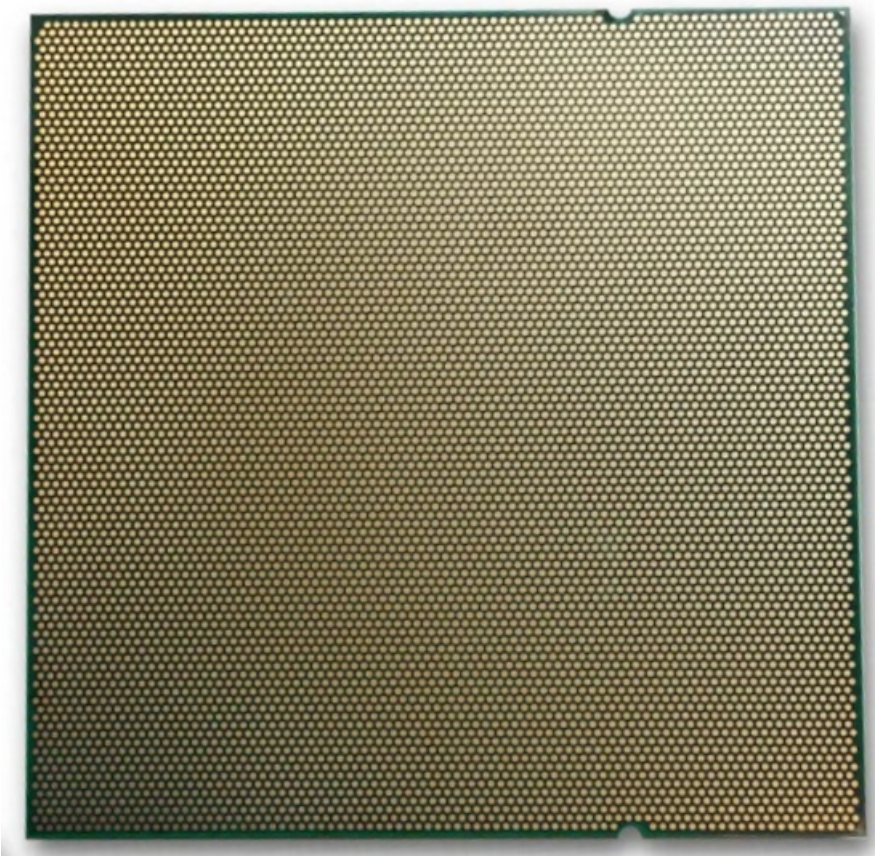


Для кастомных решений рядовая практика - вообще все ресурсы выносить за пределы сетевого ASIC"а:



В случае Juniper, кстати, их Trio - это не один ASIC - это их набор, каждый из которого реализует свои функции.

Но, как бы ни был устроен сам чип, ему нужно общаться с миром. И поэтому на животике у него есть несколько тысяч пинов:



Одни пины нужны для того, чтобы подключить к чипу интерфейсы.
Другие - чтобы подключиться к внешней памяти (CAM/TCAM/RAM), если она есть.
Третьи - к фабрике коммутации, если коробка модульная. А почти половина всех пинов - это земля и питание.

Два пина образуют дифференциальную пару для передачи данных в одном направлении. То есть, две пары пинов нужны для полнодуплексной передачи.

Вот так оно потом выглядит в программах для проектирования (для случая на порядок более простой микросхемы):

[Интеллект сети: как передать пакет]
Источник.

Теперь пришло время разобраться с тем, что же такое загадочный SerDes. Нет, это не ножки на микросхеме.

SerDes

Если коротко - то это блоки (IP-core) сетевого ASIC"а, которые позволяют получить сигнал с пинов и, наоборот - передать его туда.

Теперь с этимологией. Аналогично "модему" и "кодеку", ставшим уже такими родными в кириллическом написании, SerDes составлен из двух слов: **Serializer-Deserializer**. Так чего же он сериализует и десериализует?

Всё дело в скорости работы сетевых чипов. Независимо от типа (ASIC, NP, CPU) - их частота находится в пределах сотен МГц - единиц ГГц. А частота передачи данных с порта 10Гб/с - 10 ГГц (100 Гбит/с = 4х25ГГц). Соответственно, каким-то образом нужно понизить частоту внутри чипа. И как раз это достигается тем, что сигнал из пары вводных пинов распараллеливается на множество внутренних линий - десериализуется. В обратную сторону - сигнал с нескольких линий нужно сериализовать в пару пинов.

Блоки SerDes всегда являются составными частями кристалла сетевого чипа.

[Интеллект сети: как передать пакет]
Источник.

[Интеллект сети: как передать пакет]
Источник.

Один SerDes - это 4 пина на пузике асика - два для Tx, два для Rx.
Скорость одного SerDes"а - величина скачкообразно растущая с годами . Наиболее распространённые сегодня - это 10 Гб/с и 28 Гб/с. Но в скором будущем датацентровый масс-маркет начнут заполонять устройства с 56 Гб/с SerDes и даже со 112 Гб/с.

Выглядит довольно сложным. Для чего же вообще устраивать эту сериальную вакханалию, а не сделать просто пинов по числу реальных линий в чипе?

Зачем?

Ну, давайте прикинем, спайн-коммутатор с 64x100Гб/с портами несёт под кожухом ASIC ёмкостью 6,4 Тб/с. Если каждые 4 пина чипа могут обеспечить 28 Гб/с в полном дуплексе, значит, на нём должно быть 64*4*4=1024 пинов. Это уже как минимум 32x32 пинов на 40 см^2. И они там сидят довольно плотненько. Легко ли будет кратно нарастить их количество?

Однако проблема не только и не столько с количеством пинов.
Тенденция к серийным интерфейсам намечается уже давно.
На смену переферийным параллельным портам пришли серийные (USB на смену параллельному).
На смену IDE - SATA (Serial ATA). На смену SCSI - SAS (Serial SCSI).
На смену PCI - PCI Express.

Это жжж неспроста. Ведь казалось бы, чем больше проводов, тем больше данных можно передать за единицу времени? На самом деле нет - при параллельной передаче с повышением скорости всё острее и острее встают вопросы синхронизации между этими самым проводами. Схемотехники даже связанные дорожки на платах проектируют так, чтобы они были максимально одинаковой длины. В какой-то момент задача усложнилась настолько, что дальнейшее развитие пошло по увеличению полосы пропускания обычной дифф-пары вместо параллелизма. В частности для 100 Гб/с другого варианта просто не существует.

Модуляция

То, каким образом множество параллельных сигналов укладывается в один, зависит от метода модуляции. Для SerDes с пропускной сопособностью 10 Гб/с и 28 Гб/с используется **NRZ** - Non-Return-to-Zero. С ним же добились и 56Гб/с. Но всё же стандартом de facto для 56Гб/с и de iure для 112Гб/с является **PAM4** - 4 -level Pulse Amplitude Modulation. Есть и ленивые пессимистичные взгляды в сторону PAM8.

Итого, учитывая современные реалии (NRZ), для того, чтобы запитать данными интерфейс 100Гб/с нужно подвести к нему 4 SerDes"а по 28 Гб/с (или 16 дорожек). Отсюда и берётся "лейновость" 100Ж-портов: 4 лейна - это 4 канала по 28 Гб/с.

И это то, что позволяет 1x100Ж порт разбить на 4x25Ж с помощью гидры.

[Интеллект сети: как передать пакет]

В случае PAM4 для 100Ж нужно только 2 SerDes"а по 56Гб/с, то есть, два лейна.

GearBox"ы

Сложности с переходом на новые методы модуляции заключаются в том, что устройства на разных сторонах должны использовать одинаковые, либо нужно ставить дополнительные конвертеры. То есть, просто подключить сотками коммутатор с PAM4 к NRZ не получится.

Но когда индустрия бросала своих участников, не предлагая им решений? Для того, чтобы устройства с разными модуляциями могли взаимодействовать друг с другом, изобрели коробки передач, которые из малого количество высокоскоростных линий делают много помедленнее и наоборот. Так, в новейшие коммутаторы, выпускаемые сегодня, ставят дополнительные чипы, чтобы их можно было использовать в сети с более старым оборудованием. Использование гирбоксов также упростит вендорам и переход на 400G - не придётся менять ASIC - достаточно заменить/убрать гирбокс.

PHY

Этот зверь тоже по-своему интересен. Его задачи незамысловаты:

- Конвертация сигнала между средами (оптика-медь), если это нужно
- Восстановление битов из сигналов и наоборот
- Коррекция ошибок
- Синхронизация
- И другие задачи физического уровня.

Если хочется знать больше, и не пугают забористые тексты со страшными картинками: PHY Interface for PCI Express, SATA, USB 3.1, DisplayPort, and Converged IO Architectures.

Что действительно любопытно и достойно обсуждения - так это его расположение.
сли порт медный - RJ45, то чип PHY - это ASIC, установленный на плате.

[Интеллект сети: как передать пакет]

Если порт оптический, то в подавляющем большинстве случаев эти функции возьмёт на себя DSP PHY, встроенный в трансивер (та самая штука, называемая нами модулем и вставляемая в дырку в коммутаторе).

[Интеллект сети: как передать пакет]
Источник.

Однако, тенденции последних лет - это **Silicon Photonics**. Самый производительный коммерческий чип сегодня выдаёт 25,6 Тб/с. Это серьёзнейший инженерный вызов. И нет никаких оснований полагать, что гиперскейлеры и экзаскейлеры умерят свои аппетиты и решат остановиться на этом. Скорость будет расти.

Чип PHY находится на трансиверах, а SerDes - на кристалле сетевой микросехмы. Сигнал между ними идёт по электропроводящей среде - по металлической дорожке. С увеличением скоростей растёт и конструктивная сложность и потребляемое электричество. Рано или поздно (скорее, рано) мы во что-нибудь упрёмся. В случае silicon photonics микросхема PHY переносится внутрь самого чипа коммутации. В кристалл "встраиваются" фотонные порты, позволяющие осуществлять коммуникации между чипами на скорости света через оптическую среду.

[Интеллект сети: как передать пакет]
Источник.

Идея не новая, и только ожидавшая своего времени, а именно, когда технологии достигнут нужного уровня зрелости. Проблема была в том, что материалы и процессы, используемые для производства фотонных чипов, были фундаментально несовместимы с процессом производства кремниевых чипов - CMOS.

Из возможных альтернативных решений: установка на плату отдельного чипа, преобразующего электрический сигнал в оптический, или его установка внутрь сетевого чипа, но не на сам кристалл (всё ещё требует конвертации среды).

Но эта технологическая плотина смыта упорными разработками в этом направлении, и в скором будущем микроэлектронику ждут большие изменения. Весьма занимательная статья с историей вопроса и сегодняшними реалиями.

[Интеллект сети: как передать пакет]

И прямо во время написания этой статьи 5-го марта 2x20 Intel опубликовал в своём блоге новость о том, что они продемонстрировали первый свитч, в котором им удалось интегрировать свой интеловский silicon photonics в барефутовский Tofino2.

[Интеллект сети: как передать пакет]
Источник.

Теперь от вещей мирских к тому, сколько кругов пакет проходит в чипе.

Логическое устройство

Типичный сетевой ASIC представляет из себя конвейер, по которому пакет передаётся от входного интерфейса к выходному, а по пути с ним случаются приключения. Английский термин для этого - Pipeline.

Хотя с виду и не скажешь:

[Интеллект сети: как передать пакет]

В самом общем виде Pipeline выглядит так:

[Интеллект сети: как передать пакет]
Источник.

Пакет проходит через все стадии как минимум один раз, но для реализации сложных действий,

вроде дополнительной инкапсуляции (VXLAN), может отправиться повторно.

Parser

Сначала на вход попадает пакет с неизвестным набором заголовков.
Парсер разбирает все заголовки, отделяя их от, собственно, данных.
Если это L2-коммутатор, то его заинтересует только заголовки Ethernet и VLAN.
Если это MPLS-коммутатор, он заглянет в MPLS заголовки.
Для L3, соответственно, IPv4 и IPv6.
Если это VXLAN-терминатор, ему понадобится UDP и, собственно, заголовок VXLAN.
Для целей ECMP и ACL, парсер заглянет в UDP/TCP.

Строго говоря, на какие заголовки и какие поля в них надо смотреть, определяет фантазия разработчика. Сколько заголовков забрать? Как правило парсер вынимает фиксированное для чипа значение байтов от пакета и разделывает уже их. И это значение является одним из ограничений чипа. Почему парсинг заголовков задача нетривиальная рассказывается в совместном исследовании Стэнфорда и Майкрософта: Design Principles for Packet Parsers.

Pre-Ingress processing

Иногда логически выделяют этот блок, который совершает действия, не являющиеся ни парсингом, ни как таковым лукапом - он кладет пакет в нужный VRF, ставит внутренний Traffic Class итд.

Ingress Match-Action

Когда парсер, разобрался с чем он имеет дело, он передаёт заголовки дальше - в модули Match-Action.
Здесь происходит lookup. Для L2 поищем MAC"и, для L3 - прошерстим FIB, для MPLS - просмотрим в LFIB.
И здесь же принимается решение, что с пакетом делать дальше: пропустить/дропнуть, побалансировать, доложить CPU, в какой порт отправить, какие заголовки навесить, с каким приоритетом внутри чипа обработать, полисить ли/шейпить ли его и т.д.

Собственно действие записано в той же таблице, в которой происходит lookup. Это если коротко. А если чуть подлиннее, то:

L2 short pipeline

Приходящий L2-пакет всегда ассоциируется с VLAN. Тег VLAN проверяется через **VLAN Lookup Table**. Если VLAN lookup успешен, просматриваются таблицы: **VLAN STP**, **VLAN Port Bitmap**, **Port Filtering Mode** (PFM). Если же тег неверный, то пакет сбрасывается (или нет). После пакет проходит стандартную обработку: запомнить SRC MAC, посмотреть в таблице DST MAC, но, при этом, могут быть применены дополнительные флаги, например, отправить на CPU unknown sender.

L3 short pipeline

Если DST MAC является MAC"ом самого устройства, то процессинг передается в L3-модуль. Следующий шаг - Destination Lookup. Сначала используется **L3 Table Lookup**, в этой таблице как правило directly attached хосты. Если адрес найден, то выдается index в **L3 Interface Table**, в котором выходной порт, MAC, VLAN. Если же в L3-table не найден адрес, то делается LPM поиск (Longest Prefix Match). Результат такого поиска - index в L3 Table Lookup таблице, который должен использоваться для форвардинга. После удачного поиска, чип поменяет SA/DA/VID пакета (L2), посчитает FCS, поменяет TTL и IP checksum.

Traffic Manager + MMU

В этом блоке происходят следующие операции:

- Постановка пакетов в очередь.
- Их хранение (буферизация).
- Контроль перегрузок.
- Диспетчеризация.
- Репликация.

Он состоит из двух частей - MMU и TM. Первый отвечает за управление памятью и буферами, второй - за QoS и мультикаст. **MMU** - Memory Management Unit - компонент чипа, который управляет физической памятью. Одна из его функций аналогична MMU (блоку управления памятью) обычного

компьютера - доступ приложений к физической памяти и её защита. Но список его обязанностей гораздо шире, поскольку заточен он на работу именно с пакетами. Он отслеживает как память распределяется между интерфейсами и насколько она занята в каждый момент времени, можно ли поместить пакет в буфер, если да, то в какой, как разбить его на более мелкие ячейки, ну и, конечно, как его оттуда забрать. **TM** - Traffic Manager - решает более высокоуровневые задачи - выделение очередей, помещение в них трафика, диспетчеризация, шейпинг, полисинг, управление перегрузками. В общем, всё, что относится к QoS, а также и к мультикасту.

С мультикастом история, право, интересная (как всегда). Репликацией мультикастовых пакетов занимается блок TM. В модульных устройствах это происходит в два этапа: сначала на входном чипе создаётся столько копий, сколько выходных чипов должны получить этот пакет, а затем на выходных чипах ещё столько копий, сколько портов на этой плате должны его получить. Делается это для того, чтобы лишними копиями не загружать фабрику. Любопытный момент с буферизацией и контролем перегрузок: входной чип должен учитывать занятость выходного порта, прежде, чем отправлять пакет, потому что именно входная плата управляет VOQ. Поскольку Traffic Manager оперирует не самими пакетами, а по сути информацией о них, то ему необязательно делать сразу копий по числу выходных портов, а достаточно записать об этом информацию в VOQ.

MMU - это не совсем часть TM - это, скорее, два взаимодействующих друг с другом блока. Память и буферизация на сетевых устройствах - это настолько масштабная тема, что ей я посвятил отдельную статью, которая выйдет на nag.ru прямо следом за этой. В ней мы разберёмся с устройством и архитектурой памяти, видами и расположением буферов, арбитражем и поднимем самый горячий вопрос современности - что лучше: большие и маленькие буферы.

Egress Match-Action

Далее над заголовками пакетов могут быть совершены дополнительные акты - например, выходной ACL, туннельные инкапсуляции, сбор статистики итд.

Deparser

К этому моменту на основе результатов обработки в блоках Match-Action сформирован список новых заголовков, и он может воссоединиться с телом пакета.
Сам пакет теперь готов в последний путь внутри этого чипа, чтобы выйти через выходной интерфейс.
Кроме того, здесь может собираться дополнительная статистика о длине пакетов и сообщаться блоку TM и зеркалироваться исходящий трафик.
Вышеуказанные стадии могут быть выполнены в пределах одного чипа, а могут быть и разнесены на разные.
Так, в случае single-chip-коробки - все они скомпонованы в один кусочек силикона, площадью с фотку на паспорт.
На модульных коробках Parser и Ingress Match-Action - это входной чип коммутации, Egress Match-Action и Deparser - выходной, TM стоит отдельно между чипом коммутации и фабрикой, и может быть разделён на Ingress и Egress.

Кроме того в модульных устройствах могут существовать ещё и отдельные чипы Fabric Interface, которые разбивают пакеты на ячейки одного размера и отправляют в фабрику.

Pipeline

Пайплайном называется весь процесс доставки пакета от парсера до депарсера.
Что хорошо - можно делать несколько Match-Action подряд, например, отлукапив сначала Ethernet, потом IP, потом ещё и TCP для ECMP.

Что плохо - число этих действий строго ограничено - ASIC вещь достаточно детерминированная.
Это ведёт к тому, что некоторые вещи становятся аппаратно невозможны. К примеру, на старых Trident"ах нельзя было сделать и VXLAN и IP lookup последовательно для одного пакета. Или в другой ситуации коробка у меня не могла снять метку, сделать рекурсивный IP-lookup и навесить две новые метки.

Однако у таких трудностей есть как минимум три решения:

1. Второй чип. Тогда можно и разнести невозможные прежде операции на два этапа. И история знает такие решения.
2. Рециркуляция. Многие чипы позволяют закольцевать выход чипа на вход и прогнать пакет дважды. Тогда на второй итерации ему можно задать уже другой набор Match-Action. Но за это придётся заплатить - удвоенной задержкой и уменьшенной полосой пропускания чипа. А ещё можно упереться и в пропускную способность самого рециркулятора.
3. Купить другой чип... Другой коммутатор... Поменять работу...

Programmable Pipeline

В большинстве современных коммутаторов конвейер обработки пакетов запечён производителем в софт (если не в кремний) - он фиксирован и может быть изменён только вендором чипсета.
Не путать с Programmable ASIC. Программируемые микросхемы - уже давно реальность. Многие сетевые чипы - это ASIC с возможность программирования. Но эта возможность есть только у производителя микросхемы. Порграммируемый конвейер же - это возможность изменять логику работы чипа в определённых пределах, которую предоставляет производитель микросхемы покупателям.
Не так давно появился Barefoot Tofino, у которого полностью программируемый Pipeline - с ним можно задавать совершенно любые условия для парсера, поля для Match и действия для Action - хоть калькулятор пишите, или распределённое хранилище на кластере коммутаторов.

На сегодняшний день выпускать сетевую микросхему на рынок без возможности программирования Pipeline"а, становиться плохим тоном. Так, последние чипы Broadcom тоже уже программируемы.

Не то, чтобы теперь каждый домовой оператор кинется переписывать себе пайплайны, нанимая студентов для разработки под P4 или NPL, но это возможность, которая позволяет вендорам железа и крупным потребителям вроде гугла быть гораздо более гибкими.
Так, например, если в вашей сети все линки p2p, то зачем вам Ethernet? Тратить на него такты ASIC"а ещё - просто выкидываем его. Правда, "**можно** всё запрограммировать" превращается в наших реалиях в "**придётся** всё запрограммировать". На сегодняшний день готовых конструктивных блоков, вроде парсинга Ethernet, IP, подсчёта статистики и т.д. - не существует - всё с нуля. Если не использовать вендорские бинари, то весь Pipeline **придётся** написать самому. А если использовать, то ничего за пределами SDK не запрограммируешь.

Но на большинстве современных коммутаторов всё ещё фиксированный конвейер, который выглядит примерно так:

[Интеллект сети: как передать пакет]

Дальнейшее чтиво: Cisco всё ещё делает классную документацию, а презентации с Cisco Live - кладезь технических сокровищ. Например, они рассказывают о бродкомовских чипах больше, чем сам Бродком (если, конечно, вы не подписали NDA кровью): Cisco Nexus 3000 Switch Architecture.

Вот у P4 есть неплохое описание конвейера и его программируемости: P4 Tutorial, Hot Chips 2017.
Более фундаментальное и низкоуровневое описание **RMT** - Reconfigurable Match Tables, необходимых для возможности программирования: Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN.

Существующие чипы

До начала этого разговора следует сказать о разделении:

- **Custom/In house Silicon**. Это то, с чего всё начиналось - чипы, разработанные внутри R&D вендоров сетевых устройств: Cisco, Juniper и т.д. Используются они только внутри своего же оборудования и не продаются наружу.
- **Commodity/Merchant Silicon**. Чипы широкого производства. Можно их назвать рыночными или коммерческими. Это чипы, производимые сторонними компаниями: Broadcom, Innovium, Barefoot, Marvell и т.д.

Commodity/Merchant Silicon

Многие утверждают, что за рыночными чипами будущее (если уже не настоящее). Поэтому давайте с ними и познакомимся сначала.

Broadcom

Broadcom делает уйму разных чипов уже тыщу лет.
Исторически, он был одним из первых вендоров, кто начал производство высокоскоростных ASIC для сетевых устройств.
В 2010-м году они выпустили свой чип Trident 64x10G, а в 2020 они начнут поставлять 64x400G.

[Интеллект сети: как передать пакет]

[Интеллект сети: как передать пакет]

Все сетевые ASIC Broadcom принадлежат двум семействам: StrataXGS и Strata DNX.

StrataXGS - shallow-buffer чипы преимущественно для датацентровых коммутаторов. Названы в честь ракет. Это семейство делится на: Многофункциональные: Trident, Trident2, Trident2+, Trident3, Trident4, Maverick. Высокоскоростные: Tomahawk, Tomahawk+, Tomahawk2, Tomahawk3, Tomahawk4.

Strata DNX - чипы с глубокими буферами, рассчитанные на маршрутизаторы, модульные коммутаторы. Скорости при этом из всех семейств наиболее низкие. Названы в честь израильских городов: Arad, Qumran, Jericho, Jericho+, Jericho2. По назначению, грубо говоря, делятся они примерно так: Trident"ы на роль ToR"ов, где нужно немного интеллекта: EVPN, VXLAN, пожирнее FIB, побольше ACL. Tomahawk"и на роль Spine"ов - быстро перекладывать много пакетиков из одного интерфейса в другой. Jericho - граница датацентра, где выход во внешний мир и DCI. Обычно тут не требуются сверхвысокие скорости, потому что основной трафик - это East-West в пределах ДЦ. Зато что здесь требуется, так это весь стек сетевых технологий, большие таблицы и глубокие буферы. VXLAN, MPLS, SR, L3VPN, различные Option"ы и всё прочее, что уже основательно забылось после СДСМ.

[Интеллект сети: как передать пакет]

Но если уйти за пределы ДЦ в любую сторону - в энтерпрайз, в провайдинг, в операторов, то Broadcom свои Трезубцы позиционирует уже как универсальные чипы, которые везде и на любом уровне сети сгодятся. Почти два часа видео весьма технического склада:

- Introduction to Broadcom"s Switch Portfolio.
- Broadcom Trident4: Disrupting the Enterprise Data Center & Campus.
- Broadcom Tomahawk4: Industry"s Highest Bandwidth Chip.
- Jericho2: Driving Merchant Silicon Revolution.

Хотел бы я, пожалуй, быть таким седовласым дедком, бегущим по лезвию современных технологий.

Mellanox

Долгое время Broadcom был единоличным властелином всех сердец датацентровых коммутаторов, что позволяло ему диктовать правила игры.
Пока в 2013-м году известный производитель Inifinband-коммутаторов Mellanox не выпустил свой чип **Spectrum** и Ethernet-коммутаторы на его основе. Чип обладал производительностью 3.2 Тб/с и мог обслуживать 32 100Гб порта или 64 порта меньшей скорости.
Это было внезапно.

На сегодняшний день у них продаются свитчи на чипе **Spectrum 2** с мощностью 6,4 Тб/с.

Оба чипа shallow-buffer, рассчитаны на коммутаторы уровней Leaf и Spine - высокая скорость, низкая задержка, не самая богатая функциональность.

Ходят слухи о разработке **Spectrum 3**, от которого ожидается 12,8 Тб/с, что позволит Mellanox"у почти догнать Broadcom.

Относить ли Mellanox к числу производителей рыночных ASIC"ов - вопрос в целом дискуссионный, но все делают именно так. Увы (а может и нет), они делает чипы только для своих коммутаторов, и не продают их наружу.

В их пользу говорит то, что Mellanox - это whitebox-коммутаторы, на которое можно устанавливать сторонние операционные системы. И тем самым он участвует в конкурентной борьбе и способствует снижению цен, гонке скоростей и открытости технологий. Мимими.
И кроме того, это так называемая Fabricless - компания, которая не имеет заводов, а заказывает изготовление чипов на стороне.

Barefoot

Совсем молодой игрок на рынке коммерческих ASIC"ов. В 2013-м появился, в 2016-м вышел из тени с чипом Tofino с пропускной способностью 6,5 Тб/с.
Сегодня они готовы продавать уже Tofino 2 - 12,8Тб/с.

Чип трудится на коммутаторах Cisco Nexus 3464C, Nexus 34180YC и Arista 7170.
Чипы так же shallow-buffer и рассчитаны на спайны и лифы.

Однако сегодня нельзя просто выйти на рынок и сказать "я лучше Бродкома, купите меня". Нужно что-то предложить. Barefoot предлагает программируемый Pipeline. Это позволяет, используя специальный язык программирования P4, полностью определять, что будет происходить с пакетом в коммутаторе. Можно написать свою логику без оглядки на существующие стандарты - выбросить Ethernet вообще, заглядывать на любую глубину заголовков, выискивать какие угодно флаги итд. Грубо говоря: если 4 бита со сдвигом от начала пакета в 16 бит равны 0110, то нужно поменять 8 бит

со сдвигом 32 бита на 01001001 и отправить в интерфейс 100GE1/0.

Эта гибкость позволяет как вендорам, так и (теоретически) конечным клиентам встраивать в ASIC свою логику, а не довольствоваться встроенными правилами.
Впрочем programmable pipeline - это уже совсем другая история. Но сегодня программируемым конвейером хвастаются со сцен своих маркетинговых конференций уже почти все.

Marvell

Если загуглите в Яндексе "Marvell switch ASICs", то не так уж много ссылок вас проведут туда, где вам будут рады. Marvell определённо делает интегральные микросхемы, и даже вполне определённо делает сетевые интегральные микросхемы, но назвать его фаворитом этой гонки язык не поворачивается.

У них есть три сетевых ASIC"а, вполне конкурентоспособных по заявленным функциям и мощностям:

- Prestera CX - 12.8 Тб/с, обещают программируемый Pipeline.
- Prestera PX - по всей видимости, что-то около 1 Тб/с на роль тора.
- Falcon - 12.8 Тб/с - видел несколько упоминаний о нём, но даже на самом сайте Marvell"а информации о нём нет.

Пожалуй, из последних предложений вы можете сделать закономерный вывод, что больше про Marvell я ничего не знаю.

Innovium

У Innovium, основанного выходцами из Intel и Broadcom, есть два сетевых чипа: Teralynx 5 и Teralynx 7, обещающих знакомые скорости: 6.4 и 12.8 Тб/с. Они установлены в паре цискиных коробок: Nexus 3408 и 3432D.

Другие

Есть и другие игроки, не снискавшие успеха среди гиперскейлеров.

Один из примеров - это **Cavium**. Приходилось слышать? Это вендор, купленный не так давно Marvell"ом и производящий NP для энтерпрайз-маршрутизаторов и (!!) базовых станций.

Буквально в феврале, кстати, появилась крайне любопытная новость: Ubiquiti UniFi USW-Leaf Overview 48x 25GbE and 6x 100GbE Switch. Современный Leaf-коммутатор с 30ГБ SSD за $2000.
Немного пораскрутив публикацию, я обнаружил, что внутри сокрыт некий Taurus, разработанный **Nephos** - дочерней компанией MediaTek. И у них даже вполне любопытное портфолио.
С такой ценой, возможно, появляется новый игрок на рынке.

Для полноты картины приведу так же парочку малоизвестных компаний, которые производят низкоскоростные чипы коммутации на FPGA:

- Ethernity Networks.
- Arrive Technologies.

Нашлось, кстати, тут место и для отечественных разработок. Например, вот такого малыша вместе с отладочным комплектом можно приобрести себе для доморощенного L2-коммутатора:

- Миландр 1923КХ028.

Custom silicon

Настало время поговорить про вендоров сетевого оборудования, которые всё ещё в состоянии содержать свой гигантский штат R&D. Но именно с них когда-то всё и начиналось, рыночных чипов не было, а каждый производитель разрабатывал и изготавливал свои сетевые процессоры и ASIC"и. И это сложно сегодня, потому что когда другие могут сосредоточиться только на аппаратной обвязке вокруг чипа и софте, другим приходится выделять ресурсы на фундаментальные разработки. Стоит отдать дань уважения вендорам за это.

Занимательный факт. Во время санкционной войны лета 2019 у Huawei был невоображаемый сценарий вылететь с рынка - американская компания Broadcom заморозила поставку ASIC"ов для их линейки CloudEngine. Всё, конечно, завершилось хорошо. Но почти одновременно с этим вышел

модульный коммутатор CE16800 на чипах собственного производства, и обещали пицца-боксы. Ясное дело, что занимались они этой разработкой уже довольно давно, наверно, лет 5. Однако сей факт намекает на то, что, возможно, не так уж и плоха идея вкладываться в разработку своих чипов.

Очевидно, что и у них не по одному типу ASIC"ов, развивающихся планомерно и интегрирующихся во все новые устройства - они делятся по сериям железок, по их ролям. Не забываем и о том, насколько большие компании любят поглощения.

Я не собираюсь здесь погружаться в детали, перечисляя все их виды и возможности. Пройдёмся только поверхностно по наиболее известным из них.

Juniper

В первую очередь это, конечно, легендарный **ASIC Trio**, который бьётся внутри всех маршрутизаторов MX и коммутаторов EX. Марат Бабаян в своё время написал прекрасную статью о его работе. Внутри их магистральных коробок - **ASIC Express** (ZX, TRITON). А датацентровые коммутаторы, вроде QFX 10000, заряжены чипом **Q5**.

Huawei

У этих ребят тоже давняя история разработки своих чипов.
Когда-то начиналось с интеловских асиков, потом были эксперименты с чипом Marvell, а потом масть пошла. И новые поколения сетевых процессоров стали появляться один за другим.
Они отличаются от поколения к поколению, от класса к классу, от серии к серии.
Но для внешней людей их объединили в две линейки:

- **Ascend** для датацентров.
- **Solar** для провайдеров и операторов.

Cisco

Можно сказать, что новейшие линейки программируемых чипов это:

Cisco Silicon One, установленный в линейку Cisco 8000.

- **UADP** - Unified Access Data Plane - программируемый ASIC для каталист и некоторых нексусов.
- **QFP** - Quantum Flow Processor - для ASR и ESP.

А в глубине веков начинается такой, зоопарк, что я просто кину несколько, даже не пытаясь докопаться до глубин:

- Sasquatch, Strider - каталисты 29xx и 3xxx.
- K1, K2, K5, K10 - каталисты 4000 и 4500.
- EARL1>EARL8 - ещё разнообразные каталисты и Nexus 7000.
- Monticello - Nexus 3548.
- Big Sur - Nexus 6000.
- F3 - Nexus 7000/7700.
- nPower X1 - сетевой процессор для NCS.

Замечательные слайды, кстати, о производстве цискиных чипов и их устройстве.

И слайды про использование рыночных чипов в нексусах и их (чипов) архитектуру.

Отечественная микроэлектроника

Похвастаться терабитами, увы не можем, но вот есть 88Гб/с L3-коммутатор от "Цифровых решений" с поддержкой 1Гб/с, 10 Гб/с портов на собственном FPGA: Феникс-1/10G.
Обещали на базе этого FPGA потом выпустить ASIC, но чем история закончилась неизвестно.

Merchant vs Custom

Несмотря на всё вышеперечисленное, вендоры, конечно же, используют рыночные чипы. Те же Nexus (3xxx) или NCS5500 используют бродкомовские, инновиумские и барефутовские чипы. Juniper QFX5xxx и Huawei Cloud Engine - бродкомовские.

За

Что заставляет их использовать рыночные вместо собственных?
Ну, тупо проще взять готовенькое, впаять в плату и продать. Пресловутый Time To Market - хотя и с оговорками.
Экономия ресурсов на разработке такой сложной штуки, как микросхема по технологии 7нм с производительностью 25,6 Тб/с.
Компания, которая сосредоточена на разработке только чипов, развивает их гораздо быстрее как в плане производительности, так и функциональности.
Кто-то уже постарался и собрал информацию о том, какие функции нужны клиентам и все их реализовал

Против

В таком случае, что вендоров заставляет делать свои чипы, раз всё так хорошо?

А вот и оговорки с TTM - внести изменения в свой ASIC и выпустить следующую исправленную версию - проще и быстрее, чем запускать бюрократический маховик большой сторонней компании.

Кроме того, до недавних пор не существовало программируемых рыночных чипов - вендору нужно было полагаться на фиксированный пайплайн или весьма урезанный SDK, предоставляемый поставщиком. И производители сетевого оборудования сами начали учиться это делать, чтобы новые функции внедрять не перепаиванием асиков, а релизом новой версии софта.

Кроме того, едва ли бродком возьмётся за реализацию проприетарных протоколов или других функций, которые нужны одному покупателю - это же массовый рынок.

Никуда не спрятать вопрос безопасности. Тут своё родное - знаешь каждую закладочку.

Не всегда может устроить реализация QoS на рыночных чипах. На своих можно замутить любые мутки. Свой SDK - свои разработчики, которые его поправят. Чужой SDK - чужие баги, давайте сдавать, решать, тестировать - времени требуется значительно больше.

В общем, хорошо, что мы находимся сейчас в ситуации высококонкурентного рынка, и мы каждые полтора-два года можем получать ещё более мощные коммутаторы по ещё более низким ценам.

Ну а если вдруг вам надоели все эти наши Клозы и масс-маркет-силиконы, и душа просит чего-то совсем экзотического, то вот почитайте, как строятся суперкомпьютеры для High Performance Computing"a.

Полезные ссылки

В этот раз хоть под кат убирай. Но, поверьте, я оставил тут только самые хорошие источники, прочитанные лично моими глазами и отобранные лично моими руками.

Архитектура сетевых устройств

- Anatomy of Internet Routers.
- Cisco Nexus 3000 Switch Architecture.
- Juniper Hardware Architecture.
- Hardware Defined Networking.
- Как сделать коммутатор?
- Сети для самых маленьких. Часть четырнадцатая. Путь пакета.

Архитектура ASIC

- Packet Pushers. Understanding ASICs For Network Engineers (Pete Lumbis).
- Cisco Enterprise ASICs.
- Broadcom Ships Jericho2.
- Broadcom Launches Another Tomahawk Into The Datacenter.
- Programmable Pipeline.
- The Most Important Development Of This Century

Программируемость

- Design Principles for Packet Parsers.
- Fast Programmable Match-Action Processing in Hardware for SDN.
- P4 Tutorial, Hot Chips 2017.

Ну и немного мотивационных видео от Broadcom

- Introduction to Broadcom"s Switch Portfolio.
- Broadcom Trident4: Disrupting the Enterprise Data Center & Campus.
- Broadcom Tomahawk4: Industry"s Highest Bandwidth Chip.
- Jericho2: Driving Merchant Silicon Revolution.

Заключение

Что ещё сказать после семидесяти пяти тысяч символов? Только то, что многое из того, что вы прочитали в этот раз - частные случаи, которые могут быть (и будут) несправедливых в других ситуациях. Как только сетевой инженер смещает свой фокус со стандартизированных протоколов в область обработки пакетов, он падает в пропасть бесконечных компромиссов, где нет универсальных ответов, нет RFC, нет исчерпывающих мануалов. И чем глубже он падает, тем страшнее становится разнообразие деталей и нюансов.

Следом за этой статьёй об интеллекте сети выйдет следующая, посвящённая памяти сети и буферам, где я разберу, как пакет в сети сохранить, но не похоронить. Собственно с короткой заметки о буферах и начались эти две циклопические публикации. Сначала разбираешься как хранятся пакеты в чипе, потом смотришь, где располагается память, потом как устроен кристалл микросхемы, следом узнаёшь про SerDes"ы и Silicon Photonics. Здесь бы и тормознуть, но верёвка уже оборвалась.

Спасибы

- Андрею Глазкову (glazgoo) за рецензию и дельные замечания о коммерческих чипах.
- Михаилу Соколову (insektazz) за ликбез по устройству чипов, SerDes и Silicon Photonics.
- Александру Клименко (v0lk) за обнаружение точек роста в вопросах Pipeline"ов.
- Дмитрию Афанасьеву (fl0w) за дополнения ко всем частям статьи.
- Артёму Чернобаю за КДПВ.