Платформы обработки сетевого трафика на ПЛИС с аппаратными СФ-блоками

А.С. Коваль, email: koval@cs.vsu.ru

Воронежский государственный университет

Аннотация. *Решение проблем обработки трафика сетей в реальном времени в задачах обеспечения информационной безопасности или повышения QoE требует применения специализированных аппаратных средств. Рассматриваются варианты платформ, построенных на основе ПЛИС с аппаратными СФ-блоками (процессоров и периферии) и средства разработки.*

Ключевые слова: *классификация трафика, ПЛИС, СнК, FPGA, SoC, QoS, QoE.*

Введение

Требование обработки трафика сетей в реальном масштабе-времени (РВ) обычно возникает в задачах обеспечения информационной безопасности и QoS/QoE. Аппаратные решения уменьшают время обработки, таким образом, упрощая достижение необходимого режима РВ (жесткого или мягкого)[1]. Кроме того, аппаратные решения дают дополнительные гарантии невозможности злонамеренной модификации алгоритмов обработки. Поэтому актуальна задача выбора платформ разработки аппаратных решений, как для оценки идей, исследований, так и для апробирования прототипов: оценки работы устройства или сети из нескольких устройств (сеть с OpenFlow-свичами [2]) на реальном трафике.

1. Варианты платформ

Помимо высокой скорости обработки потока пакетов, аппаратная платформа должна позволять реализовывать как можно более широкий класс алгоритмов обработки и их параметров. Так называемые SmartNIC – это вариант с наименьшими подобными возможностями, т.к. реализованы на ASIC и отличаются от обычных сетевых контроллеров повышенной вычислительной мощностью и дополнительными приложениями обработки. SmartNIC, созданные для дата-центров, содержат приложения для Network Function Virtualization (NFV), Deep Packet Inspection (DPI). Для обозначение последних устройств часто используется термин Data Processing Unit (DPU). Наиболее известны

SmartNIC компаний-производителей Mellanox, Broadcom, Nvidia. Google, Amazon - так же имеют подобные разработки. Помимо ограниченного набора функций п приложений, практически все SmartNIC используют проприетарые закрытые технологии и мало подходят в качестве исследовательской платформы [3].

Платформы основанные на ПЛИС не имеют вышеуказанного недостатка SmartNIC: с помощью средств разработки реализуются произвольные алгоритмы обработки пакетов. К сожалению средства и длинная последовательность этапов разработки - сложны. Для сокращения времени создания прототипа на данный момент существует несколько подходов: высокоуровневый синтез (HLS); предметно-ориентированные репрограммируемые платформы с кодовой базой (NetFPGA, Alveo); экосистемы типа PYNQ. Помимо этого, в рамках подхода Система-на-Кристалле (СнК, англ. SoC) существует отдельный класс ПЛИС с аппаратными сложнофункциональными (СФ) блоками, которые работают под управлением ОС [4].

2. Платформа на основе СнК-ПЛИС

В данной работе использовались ПЛИС Zynq-7xxx компании Xilinx, содержащие два аппаратных ядра ARM и необходимую для них периферию. Запуск ОС на аппаратных процессорах СнК-ПЛИС решает еще одну важную задачу в контексте обсуждаемых платформ: изоляцию системы управления (control plane) и системы передачи данных (data plane), повышая надежность и уменьшая задержки в тракте передачи данных. Доступный вариант плат разработки с ПЛИС Zynq-7010 – плата управления от биткоин-майнера Ebit E9, помимо ПЛИС, содержит NAND-память 128MiB, DDR3-память SDRAM 256MiB, разъем micro SD, контроллер Ethernet WIZnet IP101GA:

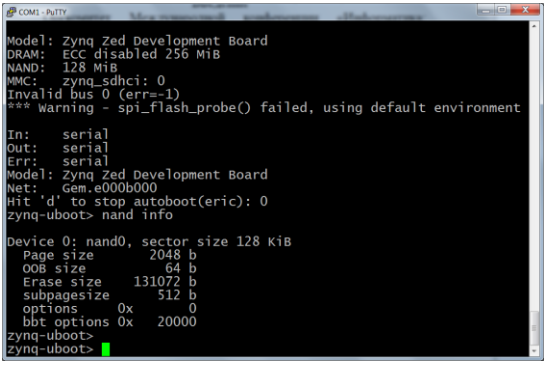

Рис. 1. Загрузчик управления биткоин-майнера Ebit E9

При подаче питания загружается ОС Linux из NAND и далее запускается процесс управления майнинг-платформой.

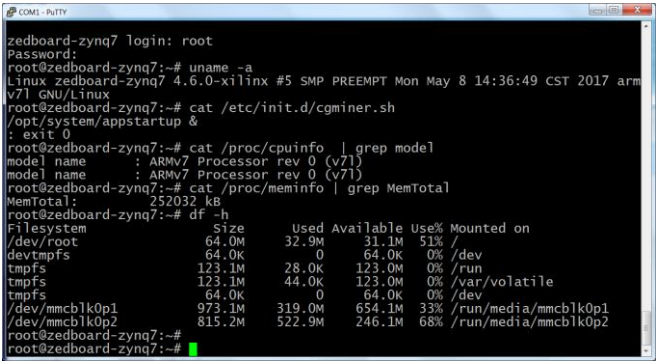

Рис. 2. ОС биткоин-майнера

В ходе работы был создан проект с загрузкой ОС Linux с micro SD карты. Проект содержит аппаратную конфигурацию ПЛИС, и компоненты ОС: ядро, корневую файловую систему и загрузчик u-boot.

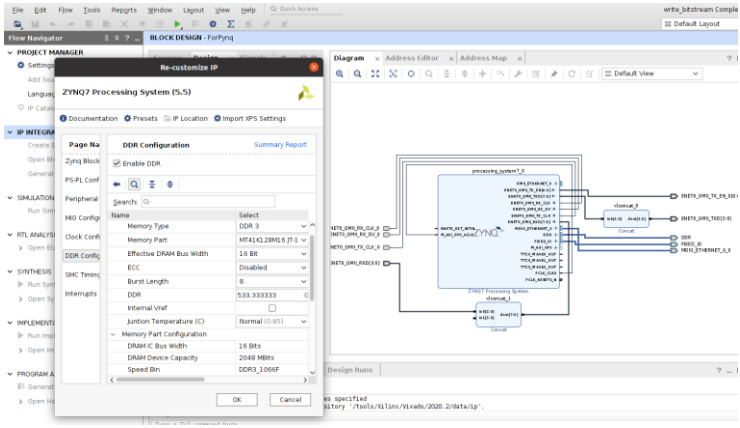

Рис. 3. Конфигурация аппаратной части базового проекта

Стандартные средства проектирования для СнК Zynq включают компонент PetaLinux Tools для создания корневой системы, сборки ядра и адаптации загрузчика к конкретной платформе.

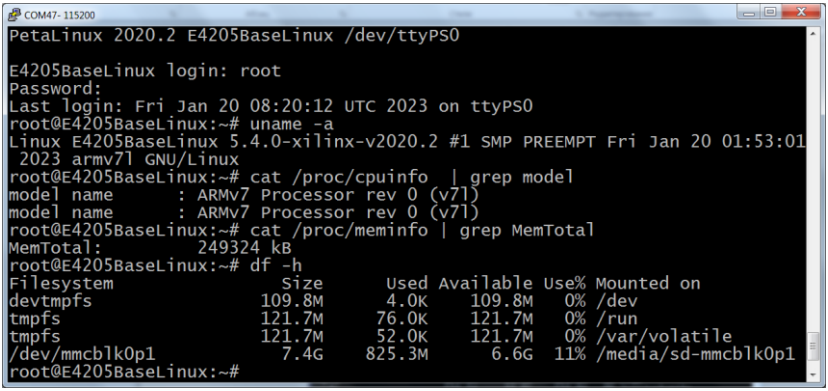

Рис. 4. Загрузка сборки ОС Linux базового проекта

ОС Linux и приложения выполняются на аппаратных СФ-блоках, процессорах ARM, это часть называется Processing System (PS). В реконфигурируемой части микросхемы – Programmable Logic (PL) можно размещать аппаратную часть обработки [4]. Разработанный первоначально базовый проект может использоваться для формирования в PS ускорителей обработки. Как отмечалось выше, для ускорения разработки существует несколько подходов. В данной работе применяется подход экосистемы PYNQ.

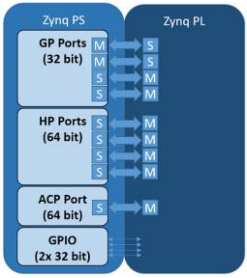

Рис. 5. PS и PL части ПЛИС Zynq

3. PYNQ: передача данных между PS и PL

PYNQ - проект с открытым кодом компании Xilinx для ПЛИС Zynq, Zynq UltraScale+, Zynq RFSoC и ускорителей Alveo. Основная идея: запуск в PS-части ПЛИС ОС Linux и Python, которые будут управлять загрузкой и работой в PL-части ПЛИС так называемых оверлеев.

На рис. 5 показаны: 4 интерфейса AXI Master HP (High Performance), 2 - AXI GP (General Purpose), 2 - AXI Slave GP порта, порт AXI Master ACP и 64 GPIO линий. Оверлей состоит из ПЛИС-конфигурации и Python API, позволяющих задействовать СФ-блоки и передавать данные между PS и PL Разработаны 4 класса для переноса данных между PS и PL:

1. pynq.gpio.GPIO - General Purpose Input/Output
2. pynq.mmio.MMIO - Memory Mapped IO
3. pynq.buffer.allocate() - Memory allocation
4. pynq.lib.dma.DMA - Direct Memory Access

Заключение

В ходе работы был создан базовый Linux-проект для СнК Zynq. Сформирован загрузочный образ для неподдерживаемой PYNQ аппаратной платформы. В результате стало возможным выполнять «блокноты» Jupyter Notebook PYNQ [5] с ускорителями на PL ПЛИС. Это упрощает применение аппаратной обработки во многих сферах, в том числе и в задачах «глубокого» анализа пакетов. В дальнейшем планируется использовать PYNQ для аппаратной классификации трафика методами машинного обучения.

Список литературы

1. Эннс, В. СнК, БМК или ПЛИС: выбор варианта исполнения цифровой интегральной схемы / В. Эннс // Компоненты и технологии. – 2018. – №4 (201). – С. 100-102.

2. Kang, J. A comparative study of zynq-based openflow switches in a software/hardware co-design / J. Kang, X. Hei, J. Song // Lecture Notes in Computer Science. – 2017. – Vol. 10658 LNCS. – P. 369-378.

3. "Облачная матрица" Pluribus заработала на "процессорах данных" Nvidia // Открытые системы. СУБД. – 2022. – № 2. – С. 7.

4. SoCs with Hardware and Software Programmability [Электронный ресурс]. – Режим доступа : https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html

5. PYNQ embedded community projects [Электронный ресурс]. – Режим доступа : http://www.pynq.io/community.html