УДК 004.383.3

Параллелизм инференса глубоких свёрточных нейронных сетей на СБИС К1879ВМ8Я

Сергей Владимирович Ландышев

АО НТЦ «Модуль», Россия, 125190, г. Москва, 4-я ул. 8 марта, д.3. landysh@module.ru

Аннотация — В статье рассматриваются вопросы реализации глубоких свёрточных нейронных сетей на отечественной СБИС К1879ВМ8Я. Описывается архитектура СБИС с точки зрения реализации параллельных вычислений, приводятся методики оптимизации инференса нейронных сетей с применением параллелизма по данным и операциям. Приводятся возможные режимы обработки нейронных сетей, масштабируемые решения и многопроцессорные конфигурации.

Ключевые слова — параллельные вычисления, нейронные сети, СБИС К1879ВМ8Я, тензорный процессор, parallel computing, CNN, ONNX, NeuroMatrix

Inference parallelism of deep convolutional neural networks on the NM6408 ASIC

Sergey Landyshev

RC Module, Russia, 125190, Moscow, 3 Eighth March 4th Street. landysh@module.ru

Abstract — This article is devoted to implementation of deep convolutional neural networks at the NM6408 chip. The chip architecture is described considering the implementation of parallel computing for optimal neural networks inference by data and operation parallelism. Various modes of processing neural networks, scalable solutions and multiprocessor configurations are considered.

Keywords: Parallel computing, CNN, NeuroMatrix, NM6408 chip, tensor processor

Воля, которая стремится к познанию, никогда не удовлетворяется оконченным делом.

Бруно, Джордано

Введение

Насущная необходимость в современных средствах автоматизации в народном хозяйстве и обороне выводит вопросы создания высокопроизводительных систем на новый актуальный уровень. Стремление разработчиков автоматических систем получать решение с применением глубоких свёрточных нейронных сетей связано с устоявшейся репутацией «чёрного ящика», дающего ответы на вопросы, которые обычно рассматривались в контексте решения человеком. Искусственный интеллект, который сегодня, главным образом, ассоциируется с глубокими нейронными сетями, применяется в самых широких смыслах – для детектирования объектов, их классификации и сегментации, для задач машинного перевода, теории управления, минимизации ошибок и т.д. Традиционные, дескрипторные методы обработки информации, повсеместно вытесняются (часто спорно) нейросетевыми методами.

Описанная ситуация была бы невозможна без появления специализированных процессоров с оптимизированными вычислениями нейросетевых операций, прежде всего свёрток и субдискретизаций (пулингов) — это устройства GPGPU[1] с расширенной функциональностью, тензорные и нейроморфные процессоры[2]. Такие устройства производятся ведущими мировыми брендами и являются результатом многолетнего поступательного развития.

Выбор предлагаемых на рынке устройств от российских производителей, даже учитывая потенциальный выход на рынок в будущем, не богат. Все предлагаемые решения на практике пригодны только для «проигрывания» уже обученных сетей, для систем обучения сетей эти устройства не могут предоставить требуемые вычислительные мощности. Эти разработки не имеют предыстории: нет линеек и модификаций существующих вычислителей. Ситуация усугубляется внешними обстоятельствами, которые требуют безусловного импортозамещения в некоторых отраслях. В этой связи очевидны трудности интеграторов при переходе на отечественные платформы, теперь цена инференса существенно выше. Для субъективной оценки положения на отечественном рынке играют роль завышенные ожидания и требования потребителей, традиционно ведущих разработку на импортных фреймворках и аппаратуре – в новых условиях приходится пересматривать критерии допустимого качества полученного решения.

Для существующих отечественных процессоров и систем нейросетевых вычислений всегда наиболее актуальны качественные характеристики, позволяющие нивелировать недостатки количественных характеристик, которые обусловлены техпроцессом изготовления микроэлектроники. Здесь подспорьем может являться наличие школы и традиций параллельных вычислений и оптимизации численных методов в РФ и в СССР (работы Э.А. Трахтенгерца ИПУ РАН[3]). В настоящей статье рассматриваются вопросы реализации глубоких свёрточных нейронных сетей на отечественной СБИС К1879ВМ8Я[4]. Описывается архитектура СБИС с точки зрения реализации параллельных вычислений, приводятся методики оптимизации инференса нейронных сетей с применением параллелизма по данным и операциям. Приводятся возможные режимы вычисления нейронных сетей, масштабируемые решения и многопроцессорные конфигурации.

СБИС К1879ВМ8Я

СБИС содержит встроенные блоки векторно-матричных вычислений над числами с плавающей точкой одинарной (32 бита) и двойной (64 бита) точности. Микросхема является гетерогенной многопроцессорной системой на кристалле (СнК), в состав которой входят 16 процессорных ядер NeuroMatrix Core 4 (NMC4) и 5 ядер ARM Cortex-A5. Упрощённая схема (основные блоки) СБИС приведена на рисунке 1. К1879ВМ8Я содержит четыре юнита, каждый из которых содержит одно ядро ARM Cortex-A5 (CPUx) и четыре процессорных ядра NMC4 (NMPUx). Кроме этого, выделяется одно процессорное ядро ARM для общего управления системой (CCPU). СБИС имеет пять интерфейсов с внешней памятью типа DDR3 – по одному интерфейсу в каждом юните и один для выделенного процессорного ядра. Основной скоростной внешний интерфейс с хост-процессором выполнен по стандарту PCIe2.0, также имеются четыре высокоскоростных интерфейса для связи с внешними процессорными системами. В каждом юните СБИС имеется один контроллер ПДП. Контроллеры обеспечивают обмен данными между разными блоками внутренней памяти и банками внешней DDR памяти. Поддерживается одно- и двухмерный обмен данными.

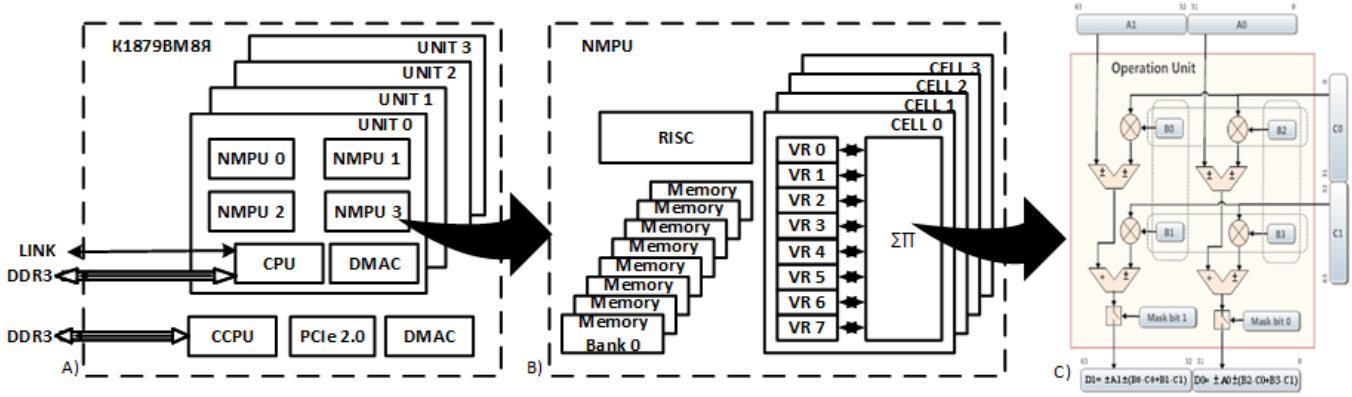

Рисунок 1 — Вычислительные блоки К1879ВМ8Я

Ниже будут рассматриваться параллельные вычисления и их особенности, присущие различным аппаратным срезам микросхемы. На самом низком доступном программисту уровне располагаются аппаратные ресурсы процессорных ядер NMC4, в частности RISC ядро, четыре независимых векторно-матричных АЛУ (cell0..cell3) и их регистровое окружение (см. рисунок 1B). Каждый из юнитов СБИС образует более высокий абстрактный уровень реализации параллелизма (см. рисунок 1A). Самый высокий уровень образует СБИС К1879ВМ8Я, включающая четыре юнита и интерфейс обмена данными с хост-компьютером по шине PCIe2.0.

Здесь следует отметить, что в руководстве по эксплуатации СБИС К1879ВЯ8Я[4] применяется термин «кластер», вместо «юнита». С точки зрения описания аппаратуры – это синонимы. Термин «юнит» был введён в контексте нейросетевой обработки, для обособления минимального вычислительного узла, на котором можно выполнить инференс нейронной сети.

NMDL – библиотека для инференса нейронных сетей

Рассматриваемые в статье принципы параллельной обработки во многом воплощены в программной библиотеке NMDL (NeuroMatrix Deep Learning) [5]. ПО поддержки инференса и портирования нейронных сетей пользователя состоит из динамически линкуемых библиотек и набора утилит, запускаемых под управлением ОС Windows и Linux. Входными данными для портирования или компиляции являются модели, описанные в форматах ONNX[6] и DarkNet[7]. Результатом компиляции является модель нейронной сети, оптимизированная для запуска на вычислительных модулях на базе СБИС К1879ВМ8Я или на симуляторе модуля МС127.05[8].

Библиотека функций NMDL предоставляет пользователю API (интерфейс C) для загрузки модели нейронной сети на модуль и входных тензоров.

Вычисления на NMPU

Основными вычислительными блоками NMPU являются RISC ядро и векторный сопроцессор. Сопроцессор является блоком, состоящим из четырёх одинаковых АЛУ (Cell0 … Cell3), каждое из которых выполняет следующие операции над векторами в формате чисел с плавающей точкой:

- Копирование с переупаковкой;
- Унарные операции (neg, abs);
- Поэлементные бинарные операции (*, +, -);
- Умножение вектора на матрицу со смещением ($Z_m = \sum_{n=0}^{1}(X_n * W_{nm}) + Y_m$, m=0, 1).

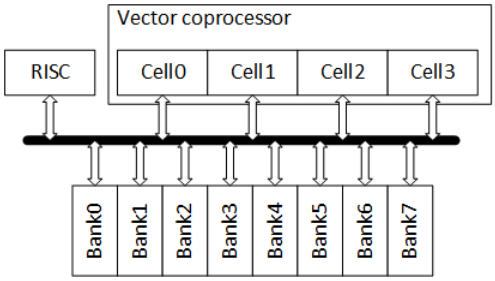

Рисунок 2 — Обобщённая схема NMPU

Основная задача RISC ядра – выборка и исполнение команд для настройки и управления векторным сопроцессором.

На векторном сопроцессоре реализуются функциональные примитивы. С поддерживаемым набором операций можно реализовать унарные, бинарные операторы над векторами данных, копирование с перестановками элементов, а также произведение вектора на матрицу и произведение матриц. С той или иной степенью эффективности реализуются все операторы ONNX, при этом входные данные разделяются между четырьмя АЛУ максимально равномерно. Наиболее эффективной является реализация свёртки – для данных большого размера достигается производительность до 90% от пиковой[9].

Однако, существуют ограничения, связанные с особенностями аппаратной реализации, в частности, при обращении векторного сопроцессора к памяти.

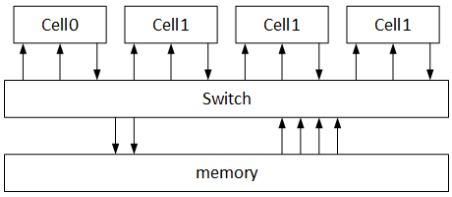

Рисунок 3 — Взаимодействие АЛУ с памятью

На рисунке 3 стрелками показаны шины данных, соединяющие АЛУ СБИС с памятью через коммутатор. Они отражают ограничения на возможность одновременных транзакций, например, каждое АЛУ может отдавать наружу только один вектор данных одновременно, а весь векторный сопроцессор может писать в память одновременно только два вектора. Кроме этого, эффективность обработки существенно зависит от типа используемой памяти. Доступ к скалярам из внешней памяти DDR3 в восемь раз медленнее, по сравнению с обращением к внутренним банкам памяти.

Исходя из вышеперечисленных особенностей можно сформулировать основные правила эффективной обработки на NMPU:

- Минимизация обращений к внешней памяти.
- Использование данных с регулярной структурой.
- Максимизация длины обрабатываемых данных («разгон» конвейера векторного сопроцессора).
- Максимизация количества операций без выгрузки данных за пределы сопроцессора.

– Использовать перемещения данных между разными АЛУ одного сопроцессора.

Наименее эффективными оказываются вычисления, для которых требуется только один «проход» данных через векторный сопроцессор (копирование, унарные и бинарные операторы), когда источник и результат хранится во внешней памяти. Такое размещение данных характерно для инференса, так как размер обрабатываемых данных существенно превосходит объём внутрикристальной памяти.

В соответствии с правилами эффективной обработки для максимизации количества операций внутри сопроцессора целесообразно применение каскада преобразований, когда вместо вызова примитива применяются процедуры, объединяющие смежные операции без выгрузки промежуточных результатов. Для инференса существуют характерные паттерны обработки, присущие различным топологиям нейронных сетей, например можно выделить следующие каскады:

– BatchNormalization + Conv
– Conv + Bias + Activation (Relu, PRelu)
– Conv + MaxPool (AvgPool)
– Mul (Div) + Bias + Activation (Relu, PRelu)
– UnaryOperation + Bias + Activation (Relu, PRelu) – имеется в виду любая унарная операция.

Вычисления на юните

На более высоком уровне иерархии аппаратных средств СБИС К1879ВМ8Я относительно ядра NMPU является юнит. Юнит содержит четыре ядра NMPU, одно ядро ARM Cortex A5, интерфейс с памятью DDR3 до 1 ГБ и контроллер ПДП (см. рисунок 1А).

Простым и эффективным решением задачи распараллеливания вычислений на юните является параллелизм по данным или геометрический параллелизм. В этом случае, входной тензор делится по одной из осей на четыре равные (по возможности) части. Каждая часть данных далее обрабатывается независимо на «своём» NMPU. На рисунке 4 входной тензор делится по ширине.

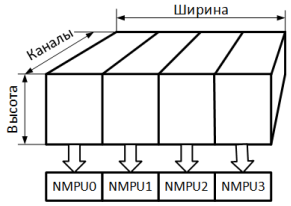

Рисунок 4 — Распределение данных в юните

Разделение данных происходит виртуально – структура тензора в памяти не изменяется. При хранении тензора с высотой *h*, шириной *w* и количеством каналов *c* в пиксельном формате содержимое памяти в порядке возрастания адресов выглядит так: $X_{0,0,0}$, $X_{0,0,1}$, …, $X_{0,0,c-1}$, $X_{0,1,0}$, …, $X_{0,w-1,c-1}$, $X_{1,0,0}$, …, $X_{h-1,w-1,c-1}$, где первый индекс – высота от 0 до h-1, второй индекс – ширина от 0 до w-1, третий индекс – каналы от 0 до c-1. Имея такую структура данных, достаточно только сформировать указатели на начала порций данных для каждого из NMPU.

Для инференса нейронных сетей характерна обработка больших объёмов данных. Объём внутренних банков памяти существенно меньше размеров входных и выходных тензоров, поэтому при выполнении ONNX операторов или каскада операторов приходится выполнять загрузку и выгрузку данных во внешнюю память. Так как тензор обрабатывается непосредственно из DDR3, при виртуальном разделении данных автоматически решаются проблемы, связанные с частичным отсутствием локальности, когда данные одной части могут зависеть от другой части, например, при перекрытии смежных пикселей в оконной обработке (свёртки и пулинги).

Полное отсутствие локальности характерно для операций, связанных с изменением структуры тензора, например, его транспонирование. Попытка выполнить перестановку элементов параллельно, задействовав все NMPU, приведёт к постоянным блокировкам со стороны контроллера DDR3 при одновременном обращении процессоров к внешней памяти. Подобные операции целесообразно выполнять через каналы ПДП.

Вычисления на СБИС

Архитектурное решение разместить в одной СБИС четыре юнита предоставляет разработчику широкие возможности для реализации различных концепций параллелизма. Каждый юнит имеет один интерфейс с внешней памятью DDR3 объёмом до 1ГБ, причём первые 512 МБ доступны для других юнитов. Таким образом, взаимодействие между юнитами можно реализовать через общую внешнюю память. Кроме этого, для межпроцессорного обмена можно использовать специальные коммуникационные каналы (см. рисунок 1). Синхронизация между юнитами может выполнятся как аппаратно, через сигналы, так и программно из общей памяти.

Рассмотрим реализацию параллелизма по данным. Такой тип обработки эффективен для большого числа однородных действий над однородными данными, которые можно так разбить на группы, что при обработке каждой группы не потребуется обращений к данным из других групп – это свойство полной локальности алгоритма. Для инференса нейронных сетей можно выделить группы операторов, обладающие разной локальностью. Некоторые из операторов приведены в таблице 1.

Таблица 1.

1. Локальный алгоритм	2. Почти локальный алгоритм	3. Не локальный алгоритм
• Все унарные и бинарные операторы, например: Mul, Add, Neg, … • Поточечная обработка: Cos, Relu, Sigmoid, … • Concat (ось конкатенации не равна оси разделения данных по юнитам)	• Conv • MaxPool • AvgPool • MatMul • Dense • GlobalAveragePool	• Transpose • Flatten • Shape • Concat (ось конкатенации равна оси разделения данных по юнитам)

Операторы из первой группы реализуются тривиально – входные данные равномерно разбиваются на четыре части и обрабатываются в юнитах одновременно и независимо.

Для второй группы характерны пред- или пост-вычисления. Например, реализации свёртки на нескольких юнитах предполагает предварительное копирование смежных строк из соседних по высоте групп данных входного тензора для обеспечения бесшовной свёртки. Для операции Dense необходимо выполнить постобработку, сложив поэлементно выходные группы данных. Пре- и пост-обработка использует данные из соседних групп и поэтому здесь требуется барьерная синхронизация юнитов.

Алгоритмы третьей группы наиболее «неудобны» для распараллеливания, так как по сути являются поэлементными перестановками. В случае, когда операции из этой группы не входят в каскад обработки (например, transpose + поточечная обработка) эта работа выполняется с применением многомерного ПДП.

Описанная выше реализация с геометрическим параллелизмом используется в NMDL как работа в режиме «multi unit», то есть подразумевается, что один инференс выполняется на всех четырёх юнитах. Входные тензоры делятся по высоте, образуя полоски данных, каждая из которых обрабатывается в соответствующем юните. На рисунке 5 схематично иллюстрируется такая обработка. Пунктирными линиями показаны разделённые по юнитам данные.

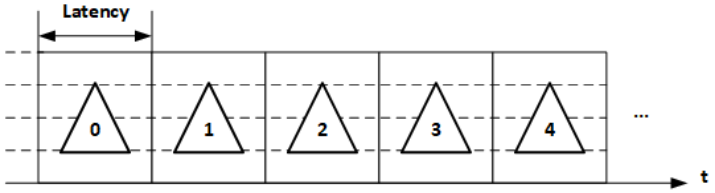

Рисунок 5 — Обработка входных тензоров (кадров) в режиме «multi unit»

Так как в режиме «multi unit» входные и выходные данные распределяются по юнитам, то возможна обработка больших моделей.

Рассмотрим режим пакетной обработки. В классическом понимании обработка в этом режиме сводится к последовательным вызовам одного и того-же оператора для разных входных данных, в инференсе это означает послойную обработку, то есть кадры обрабатываются не последовательно, один за другим в одном направлении, а как бы перпендикулярно. На рисунке 6 иллюстрируются последовательная обработка (A) и пакетная (B) для двух-кадрового пакета и одной операции (один слой). Стрелками показаны обращения к данным во времени.

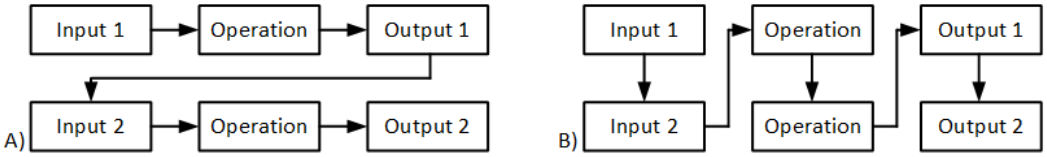

Рисунок 6 — Последовательная (A) и пакетная (B) обработка входных тензоров

При достаточно больших размерах пакета можно достичь существенного прироста производительности за счёт «разгона» аппаратного конвейера, так как послойная обработка предполагает многократные вызовы одних и тех-же

процедур и предоставляет возможности для оптимизации циклов. Размер пакета определяется особенностями вычислителя. Так, для СБИС К1879ВМ8Я, оптимальными размерами будут значения, кратные 32 (32, 64, 96 и т. д.).

Реализация пакетного режима требует больших объёмов памяти: нужно держать в памяти промежуточные буферы для обработки всего пакета, то есть в N раз больше, чем при последовательной обработке, где N – размер пакета. Далеко не все современные модели можно разместить в доступной внешней памяти микросхемы при пакетной обработке.

Самый существенный недостаток этого режима – большая латентность, то есть время от начала обработки входных данных до получения результата обработки. Выдача результата задержится до полной обработки всего пакета. В этой связи, режим пакетной обработки имеет очень ограниченное применение – системы обработки в режиме оффлайн, когда имеется значительный и уже сформированный набор входных данных.

В текущей реализации NMDL пакетный режим в описанном выше варианте не реализован. Вместо этого есть режим с реализацией инференса на одном юните «single unit». Здесь предоставляются возможности для гибкой диспетчеризации обработки, с выполнением инференса разных нейронных сетей на одном процессоре. На рисунке 7 показана обработка входных тензоров во времени, когда на все юниты загружена одна нейронная сеть.

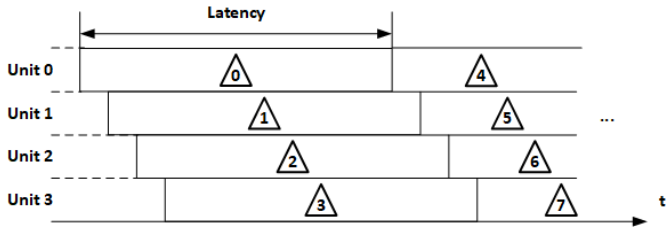

Рисунок 7 — Обработка входных тензоров (кадров) в режиме «single unit»

В этом режиме достигается максимальная производительность за счёт минимизации накладных расходов, которые возникают в не полностью локальных алгоритмах в режиме «multi unit». Однако, по сравнению с «multi unit» существенно возрастает латентность. Прирост производительности сильно зависит от использования не локальных алгоритмов в той или иной нейронной сети – чем их больше, тем существеннее прирост. В таблице 2 представлены результаты инференса некоторых нейронных сетей для двух режимов.

Таблица 2.

NN	Multi unit		Single unit	
	FPS	Latency (ms)	FPS	Latency (ms)
alexnet (227x227)	12,6	79	13	308
resnet 50 (224x224)	12,2	82	20,6	194
squeezenet (224x224)	74,4	13	100	40
yolo v5s (640x640)	4,7	212	5,3	754

Применяя режим «single unit», возможно построение комплексных систем нейросетевой обработки на базе СБИС К1879ВМ8Я. На четыре юнита микросхемы можно загрузить четыре произвольных модели нейронных сетей и решать разные задачи на одном вычислителе. На рисунке 8 схематично иллюстрируется такая обработка.

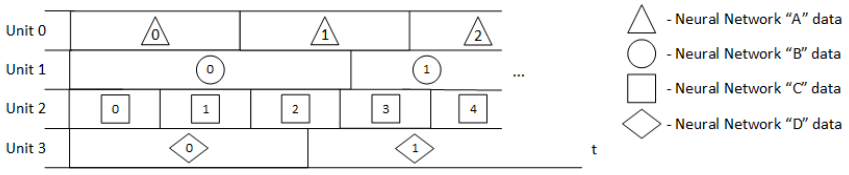

Рисунок 8 — Независимая обработка моделей в режиме «single unit»

Многопроцессорная обработка

В контексте организации многопроцессорной системы взаимодействие между СБИС К1879ВМ8Я может осуществляться по двум интерфейсам: соединение по шине PCIe и коммуникационным линкам. Соединение с хост-компьютером (управляющая ПЭВМ или процессор приложений) осуществляется только по PCIe, если не используются специально разработанные адаптеры. Наиболее простая схема организации многопроцессорной системы – это подключение процессоров к одному хост-компьютеру по интерфейсу PCIe с топологией «звезда» (рисунок 8).

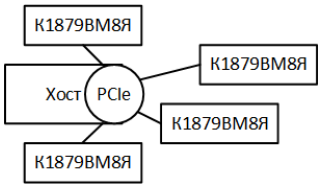

Рисунок 9 — Звездообразное подключение ускорителей к хосту.

На рисунке 9 условно показаны подключения СБИС непосредственно к шине PCIe. Для упрощения не показаны подключения на уровне вычислительных модулей с несколькими процессорами, PCIe концентраторы и т.д., которые в реальности могут образовывать сложную многокаскадную систему. Эта схема применима для потоковой обработки видео кадров или аналогичных задач с непрерывной обработкой. Хост-компьютер реализует конвейер обработки поочерёдно загружая все ускорители. Каждый из ускорителей может работать в разных режимах. В режиме «multi-unit» на один процессор загружается один кадр. В режиме «single-unit» один процессор обрабатывает последовательность из четырёх кадров.

Для звездообразного подключения просто оценить прирост производительности – он будет расти пропорционально количеству СБИС. Однако, существует ограничение на масштабируемость системы, связанная с пропускной способностью шины PCIe и с конкретной реализацией драйвера. Можно рассчитать верхнюю оценку для максимального количества СБИС в системе с одним хостом и ускорителями на шине PCIe, которые можно задействовать без простоев и потерь производительности.

Условием загруженности процессоров, когда обработка осуществляется без простоев, например, в режиме «single unit» является:

$T_p \geq 4 * C * (T_w + T_r)$, где

- T_p – время обработки одного кадра на одном юните в секундах. $T_p = 1/V_p$, где V_p – производительность обработки на одном юните (кадры в секунду);
- T_w – время записи одного кадра. $T_w = S_w / V_w$, где S_w – размер кадра в байтах, V_w – скорость записи (байт/с).
- T_r – время чтения результата. $T_r = S_r / V_r$, где S_r – размер результата в байтах, V_r – скорость чтения (байт/с).
- C – количество СБИС.

$C \leq 0{,}25 / (V_p * (S_w / V_w + S_r / V_r))$.

Например, расчёт максимального количества СБИС для Yolo v5s 640x640 для хоста ПЭВМ Эльбрус e2k-8c2:
Vp = 1,325 кадров/с
Sw = 6553600 байт
Vw = 960090187 байт/с
Sr = 2142000 байт
Vr = 987272545 байт/с
Cmax = 0,25 / (1,325 * (6553600 / 960090187 + 2142000 / 987272545)) ≈ 20.

То есть, для заданного примера верхний предел эффективного ускорения может быть достигнут на 20 СБИС, или 20 * 4 * 1,325 = 106 кадров/с. В реальной системе возникнут расходы на пре- и пост-обработку, скажутся особенности работы операционной системы с вытесняющей многозадачностью на хост-компьютере и т. д., в итоге фактические показатели окажутся в 1,5 – 2 раза ниже расчётных.

Заключение

СБИС К1879ВМ8Я имеет в своём составе специализированные аппаратные блоки, обеспечивающие эффективность операций над тензорами. По сути – это набор АЛУ, параллельно выполняющих векторно-матричные операции с данными, находящимися на разных уровнях иерархии доступа к памяти. Векторно-матричные сопроцессоры, содержащие АЛУ имеют свою систему команд, которые выбираются и выполняются в глубоком конвейере. Нейросетевые операции в микросхеме являются программно определяемыми, предоставляя большую гибкость при их реализации и расширяя область применения СБИС. Состав аппаратуры, её кластерная структура и многослойная организация памяти позволяет использовать различные методы распараллеливания алгоритмов, в частности алгоритмов инференса нейронных сетей.

В то же время, программно определяемая сущность не позволяет выйти на пределы эффективной обработки. Нейросетевые операции требуют сверхвысокого быстродействия, которое достигается введением специализированных вычислительных блоков, с аппаратной поддержкой каскадных вычислений над тензорами данных, разменивая гибкость и многообразие программных решений на производительность. Наработки, связанные с реализацией параллелизма инференса, могут быть использованы при проектировании новых систем, в которых возможен баланс высокой производительности при аппаратной реализации типичных алгоритмов и гибкости программно определяемых блоков обработки при решении нетипичных или новых задач.

В статье рассмотрена архитектура СБИС как иерархия вычислительных устройств и памяти. На нижнем уровне определяются процессорные ядра NMC4 со «своей» внутренней памятью.

На этом уровне реализуются вычислительные примитивы и их каскады. Каждый из юнитов СБИС образует второй уровень, на котором из вычислительных примитивов составляются нейросетевые операторы – свёртки и субдискретизации. Юнит взаимодействует со «своим» контроллером DDR3, реализуя независимую обработку входных данных в соответствии с заданной моделью нейронной сети. Важную роль здесь играет контроллер ПДП, в котором параллельно с вычислениями в АЛУ выполняются многомерные выборки данных для операций копирования и переформирования данных.

Третий уровень иерархии устройств образует сама СБИС К1879ВМ8Я, включающая четыре юнита и интерфейс обмена данными с хост-компьютером PCIe2.0. В статье рассмотрены два режима обработки – «multi unit», когда единый инференс выполняется сразу на всех четырёх юнитах одной СБИС и «single unit», когда на каждом юните СБИС выполняется свой инференс. В первом случае достигается минимальное время от момента получения входных данных до выдачи результата инференса. Во втором случае можно достичь максимальной производительности.

В статье также рассмотрена параллельная обработка на нескольких СБИС. Коммуникационные возможности К1879ВМ89Я позволяют строить многопроцессорные вычислительные комплексы различных топологий, которые масштабируются для решения самых ресурсоёмких задач.

Список литературы

1. Hyesoon Kim, Richard Vuduc, Sara Baghsorkhi. Performance Analysis and Tuning for General Purpose Graphics Processing Units (GPGPU). — Morgan & Claypool Publishers, 2012.
2. M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. Fonseca Guerra, P. Joshi, P. Plank, S. R. Risbud Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook PROCEEDINGS OF THE IEEE | Vol. 109, No. 5, May 2021.
3. Трахтенгерц Э.А. Програмное обеспечение параллельных процессов. Академия наук СССР 1987.
4. Микросхема интегральная 1879ВМ8Я. Руководство по эксплуатации. Электронный ресурс https://www.module.ru/uploads/products/18798-b192775dfa.pdf.
5. NMDL. Руководство пользователя. Электронный ресурс https://www.module.ru/uploads/pages/nmdl_ru.pdf.
6. ONNX. Электронный ресурс. https://onnx.ai.
7. YOLO: Real-Time Object Detection. Электронный ресурс. https://pjreddie.com/darknet/yolo/.
8. Модуль МС127.05. Электронный ресурс. https://www.module.ru/products/2-moduli/39-12705-nm6408-devkit.
9. Реализация глубоких свёрточных нейронных сетей на СБИС К1879ВМ8Я: Материалы международного форума «Микроэлектроника 2019», Алушта, 2019 г. С.В. Ландышев, М.Ю. Клименко