

ФОРМАТ POSIT КАК ФОРМАТ ПРЕДСТАВЛЕНИЯ ЧИСЛОВЫХ ЗНАЧЕНИЙ В НЕЙРОННЫХ СЕТЯХ НА УСТРОЙСТВАХ ПРОГРАММИРУЕМОЙ ЛОГИКИ

Боровский М. А., Грабко А. Н.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: maksim.borovskij@yandex.ru, softforlutshix@gmail.com

В данной работе рассматривается возможность применения формата POSIT для представления чисел в нейронных сетях, реализуемых на устройствах программируемой логики. FPGA имеют ограниченные ресурсы обработки данных, поэтому при разработке нейронных сетей на FPGA сталкиваются с вопросом представления чисел. Формат представления должен обеспечивать точность результатов в пределах принятых разработчиками норм и предоставлять максимальную производительность при работе с ними. Формат POSIT является кандидатом на роль одного из форматов в области.

ВВЕДЕНИЕ

Многие задачи не могут быть решены с помощью строгих алгоритмов, например, задачи распознавания, классификации, прогнозирования и т.д. При их решении широко используются нейронные сети (НС), которые позволяют обойти ограничения классических вычислительных систем.

В последние десятилетия были проведены многочисленные исследования аппаратных реализаций НС. НС хорошо поддаются представлению на языках описания аппаратуры, например, VHDL и Verilog, т. к. состоят из обособленных компонентов и явных связей между ними. Аппаратные НС дают преимущество в скорости обработки поступающих данных, что ведет к более быстрым процессам обучения и непосредственного использования НС.

При проектировании аппаратных НС важным является вопрос представления числовых значений, т. к. он напрямую влияет на объем необходимого хранилища, число логических элементов для представления АЛУ, ширину полос передачи данных и скорость их обработки.

Имеется некоторое число подходов к представлению числовых значений при реализации НС на FPGA:

1. числа с плавающей точкой (сложная реализация, малая производительность, большая точность);
2. числа с фиксированной точкой (легкая реализация, большая производительность, точность регулируется числом бит дробной части).

Появившийся в 2017 году и зарекомендовавший себя в некоторых областях формат POSIT может обеспечить необходимое соотношение точности и производительности в НС на FPGA.

I. ФОРМАТ POSIT

Предложенный Дж. Густафсоном арифметический подход предлагает три формы пред-

ставления чисел: Unum Type I, II и III. Unum Type III – последняя редакция формата, POSIT. Формат POSIT, как и float, имеет секции знака, экспоненты и дробной части, также обладает дополнительной секцией режима. Формат появился после принятого упрощения по отношению к Unum Type II: последний бит не используется в качестве значащего в дробной части и не используется как ubit (бит в конце дробной части, показывающий, что число является точным либо находится в интервале соседних вещественных чисел). Размеры секций режима, экспоненты и дробной части являются варьирующимися. Значение 0 в бите знака соответствует положительному числу, 1 – отрицательному. Режимом является число повторяющихся бит (нулей или единиц) в начале строки. Например, строка 0000 имеет 4 нуля, значит, режим равен 4, 001 - 2, 1110 - 3 и т.д. Режим имеет соответствующее значение k , которое используется для вычисления масштабирующего фактора. Значение k равно $-m$, если повторяющиеся биты нули, и $m - 1$, если повторяющиеся биты единицы, где m – количество повторяющихся бит. Например, для строки 0001 k равно -3, 1110 - 2 и т.д. Масштабирующий фактор равен $useed^k$, а $useed = 2^{2^{es}}$, где es – максимальное число бит экспоненты. Экспонента является беззнаковым целым числом и предоставляет масштабирование в 2^e раз, где e – значение экспоненты. Данный формат позволяет изменять точность числа и хранить данные об этом в компактном виде. Таким образом, числа, близкие к единице по модулю, будут иметь большую точность, чем числа стремящиеся к бесконечности либо нулю. Стоит отметить, что НС обычно оперируют числами в промежутке $[-1; 1]$. Оставшиеся биты после битов режима и экспоненты представляют дробную часть, она имеет тот же формат, что и в float, но скрытый бит всегда равен 1. Значения по битовому представлению POSIT p вычисляются по следующей формуле:

$$x = sign(p) \times useed^k \times 2^e \times f$$

Однако, при $p = 0$ $x = 0$, а при $p = -2^{n-1}$ $x = \pm\infty$. Например, 5-битное POSIT число с максимальным числом бит экспоненты равным 2 имеет максимальное представимое число 4096, а минимальное по модулю – 1/4096 [1, 2].

II. СРАВНЕНИЕ ЧИСЕЛ POSIT И ЧИСЕЛ С ФИКСИРОВАННОЙ ТОЧКОЙ

Достаточно часто при проектировании НС на FPGA отдают предпочтение формату представления числовых значений с фиксированной точкой, поэтому целесообразным является сравнение данного формата с форматом POSIT.

Для сравнения будут использованы 5-битные представления чисел с фиксированной точкой в прямом коде и числа POSIT (будут представлены в трех модификациях: с максимальным числом бит экспоненты 0, 1 и 2, чтобы увидеть влияние данного параметра).

5-битное представление с фиксированной точкой может представить 31 (из-за положительного и отрицательного нулей, что можно исправить использованием дополнительного кода – 32 значения без двух нулей) значение с постоянной разницей между ними 0.125.

Для быстрого генерирования POSIT последовательностей с указанием полного числа бит и максимального числа бит экспоненты был написан скрипт на языке Python. Сгенерированные три последовательности (для простоты только в положительной области) представлены в таблицах по ссылке [3].

Итак, числа POSIT с нулевой экспонентой от 0 до 1 совпадают с числами с фиксированной точкой в этом же интервале, однако разность между числами > 1 начинает увеличиваться: 0.25, 0.5, 1 и 4 – вместо постоянной 0.125. Таким образом, видно, что точность чисел с приближением к крайним значениям падает (плавная кривая нарастания погрешности). Необходимо сказать, что покрытие числовой прямой обычными числами с фиксированной точкой равномерно и легко предсказуемо, чего нельзя утверждать о числах POSIT.

Представимый интервал также различен, для POSIT с нулевой экспонентой это [-8; 8], для чисел с фиксированной точкой это [-1.875; 1.875]. Принимая во внимание, что выходное значение функций активации НС обычно принадлежит интервалу [-1; 1] (например, сигмоида), можно сделать вывод, что увеличенный интервал представления чисел POSIT в данном случае излишен.

При сравнении чисел с фиксированной точкой и чисел POSIT с экспонентой в 1 и 2 бит видно, что придаваемый масштаб еще больше уменьшает точность чисел, в том числе и на интервале [0; 1].

Число бит экспоненты влияет на диапазон и точность чисел, поэтому для каждой задачи необходимо подбирать данный параметр, что

вносит дополнительную сложность при работе с POSIT. Однако такая особенность дает возможность использовать различное число бит экспоненты в определенных ситуациях, например, для представления результатов функций активации и представления весов сети, однако это приводит к усложнению конвертирования чисел между форматами.

Необходимо отметить, что для реализации арифметико-логического устройства для обработки чисел POSIT требуется в десятки раз больший объем вычислительных ресурсов FPGA [4].

При работе со значениями, для представления которых достаточно чисел с фиксированной точкой и 3–6 битами для дробной части, использование чисел POSIT не дает никаких преимуществ.

ЗАКЛЮЧЕНИЕ

В данной работе были проведены характеристика формата представления чисел POSIT, сравнение с числами с фиксированной точкой для представления чисел в нейронных сетях, реализуемых на FPGA. Преимущества использования формата POSIT не выявлено в случаях, где достаточно чисел с фиксированной точкой и небольшим числом бит дробной части. Также необходимо отметить, что формат POSIT является достаточно новым, представленные в источниках примеры зачастую рассматривают формат со стороны, где он лучше справляется с задачей. Однако существуют примеры, где присутствуют грубые несовпадения результатов вычислений с реальными [5]. Формат имеет достаточно большое число особенностей, которые должны быть учтены перед использованием, и формат не является универсальным, как, например, форматы с плавающей (IEEE 754) или фиксированной точкой.

СПИСОК ЛИТЕРАТУРЫ

1. Posit Arithmetic [Electronic resource] / J. L. Gustafson. – Unum and Posit, 2017. – Mode of access: <https://posithub.org/docs/Posits4.pdf>. – Date of access: 15.09.2019.
2. Beating Floating Point at its Own Game: Posit Arithmetic [Electronic resource] / J. L. Gustafson, I. Yonemoto. – Dr. John L. Gustafson, 2018. – Mode of access: <http://www.johngustafson.net/pdfs/BeatingFloatingPoint.pdf>. – Date of access: 16.09.2019.
3. its-conference-thesis-2019 [Electronic resource] / M. Borovsky. – GitHub, 2019. – Mode of access: <https://github.com/maximka777/its-conference-thesis-2019>. – Date of access: 20.09.2019.
4. Parameterized Posit Arithmetic Hardware Generator [Electronic resource] / R. Chaurasiya, J. L. Gustafson, R. Shrestha. – Unum and Posit, 2018. – Mode of access: https://posithub.org/docs/iccd_submission_v1.pdf. – Date of access: 22.09.2019.
5. Posits: the good, the bad and the ugly [Electronic resource] / F. de Dinechin, L. Forget, J. Muller. – Univ Lyon, 2019. – Mode of access: <https://hal.inria.fr/hal-01959581v3/document>. – Date of access: 22.09.2019.