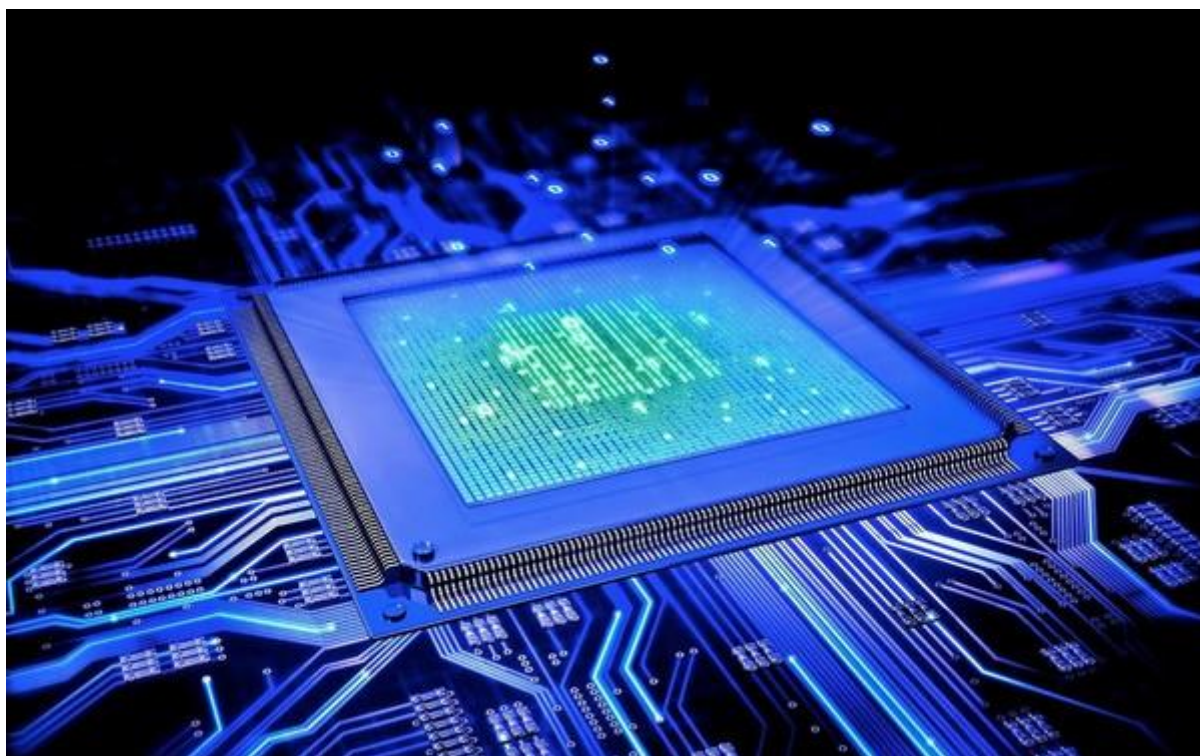


## ***FPGA-ПРИСКОРЮВАЧІ ЗЛІТАЮТЬ У «ХМАРИ»***



Поява на ринку *FPGA*-прискорювачів, які можна перепрограмувати як угодне число раз, причому на мові високого рівня типу *C*, стало справжнім проривом в області високопродуктивних обчислень. Але не меншим проривом стала можливість використовувати технологію *FPGA*, не купуючи ці дорогі адаптери, а просто орендуючи виділений сервер з прискорювачем в «хмарі» провайдера.

### **Вступ або про мікросхеми *FPGA* в 3-х абзацах**

Мікросхема *FPGA* (англ. *Field-programmable gate array*, програмувальна користувачем логічна інтегральна схема, ПЛІС) - це інтегральна мікросхема (ІМС), яку можна реконфігурувати під будь-які складні обчислювальні задачі. В індустрії існує потреба в спеціалізованих мікросхемах (*ASIC, application-specific integrated circuit*, інтегральна схема спеціального призначення) – від керування космічними апаратами і до розрахунків за фінансовими моделями.

Однак до появи *FPGA* сильним і одночасно слабким місцем спеціалізованих ІМС була жорстка функціональність, закладена в мікросхему, а також висока

складність проектування і вартість запуску у виробництво. Якщо функціонал потім потрібно змінити, або на етапі проектування відбулися помилки, то потрібно було створювати по суті нову ІМС.



*FPGA-прискорювач з ІМС FPGA Intel Altera Arria 10 і портами 2x10GE*

Поява на ринку *FPGA*-прискорювачів, які можна перепрограмувати необмежену кількість раз, причому на мові високого рівня типу *C*, стало справжнім проривом в ніші високопродуктивних обчислень. Це дозволило прискорити час розробки і виходу продуктів на ринок. З'явилися абсолютно нові можливості для розробників апаратних засобів, в т.ч. працюючих над програмуванням спеціалізованих інтегральних схем типу *ASIC*.

*FPGA*-процесори пройшли вже два етапи з точки зору доступності цієї технології і сьогодні активно входять в третій етап. Перші *FPGA* з'явилися в 1985 році, але їх програмування як і раніше вимагало знання мови низького рівня типу асемблера. На другому етапі, який почався приблизно в 2013 році, і завдяки зусиллям компанії *Altera*, з'явилася можливість програмування на *C*-подібній мові високого рівня. Це кардинально розширило застосування *FPGA*, але висока вартість ІМС і раніше стримувала розширення кола клієнтів, які могли б собі дозволити цю технологію.

Традиційно маршрут проектування і верифікації ПЛІС вкрай трудомісткий і вимагає високої спеціалізації, по своїй складності маршрут наближається до проектування *ASIC*. Це обмежує використання ПЛІС розробниками. Особливо, це

стосується обчислювальних додатків, де учасники процесу, – програміст, математик, алгоритміст, – бажають сфокусуватися на своїй задачі, а не на її апаратній реалізації.

Вирішуючи цю проблему компанія *Altera* в 2013 році вивела на ринок для своїх ПЛІС підтримку відкритого стандарту програмування гетерогенних обчислювальних платформ *OpenCL*, що розширило можливість застосування апаратури розробниками обчислювальних додатків, малознайомих з апаратурою ПЛІС, мовами *HDL*, маршрутом проектування і верифікації. Але залишилася проблема – дорога апаратура і засоби проектування.

І, нарешті, приблизно з 2016 року можна говорити про третій етап, який ознаменувався доступністю для широкого кола клієнтів повністю готових серверів (фізичних і віртуальних) з *FPGA*-процесорами в «хмарах» найбільших дата-центрів – *MS Azure, Amazon Web Services (AWS), Cloud Alibaba* і *Huawei Cloud*.

Навіщо можуть знадобитися *FPGA*-прискорювачі? Потіки даних все ростуть з одного боку, а з іншого відзначені складності в нарощуванні обчислювальної потужності без збільшення розмірів і споживання обчислювальної системи. Як правило, в додатку є задачі керування і ресурсномісткої обробки даних. Доцільно залишити керування на ЦП, а обробку відправити на спеціалізований ресурс. Динамічна реконфігурація «на льоту» під особливості задачі є однією з головних властивостей ПЛІС-технологій. Вона дозволяє досягти майже максимальної продуктивності при рішенні конкретної задачі.

Синтез обчислювального ресурсу на ПЛІС під задачу також повинен дати вигоду і в зниженні енергоспоживання. Також, на ПЛІС присутні конфігуровані внутрішня швидка пам'ять і комунікаційна частина, що дозволяє організувати практично всі відомі протоколи обміну інформацією. Наприклад, для організації кеш-пам'яті, апаратних *DSP*-блоків, контролерів пам'яті і т.д. Іншими словами, це розвинена система на кристалі (*SOC, System-On-Chip*), що дозволяє синтезувати конкретне обчислювальне ядро під конкретну задачу.

## **Базові відмінності *FPGA* від *CPU* та *GPU***

Які типи прискорювачів доступні на сьогоднішній день? Багатоядерні процесори *CPU*, *GPU* і *FPGA*. Розглянемо їх нижче. Також виникає питання - чи можна їх використовувати за єдиним стандартом?

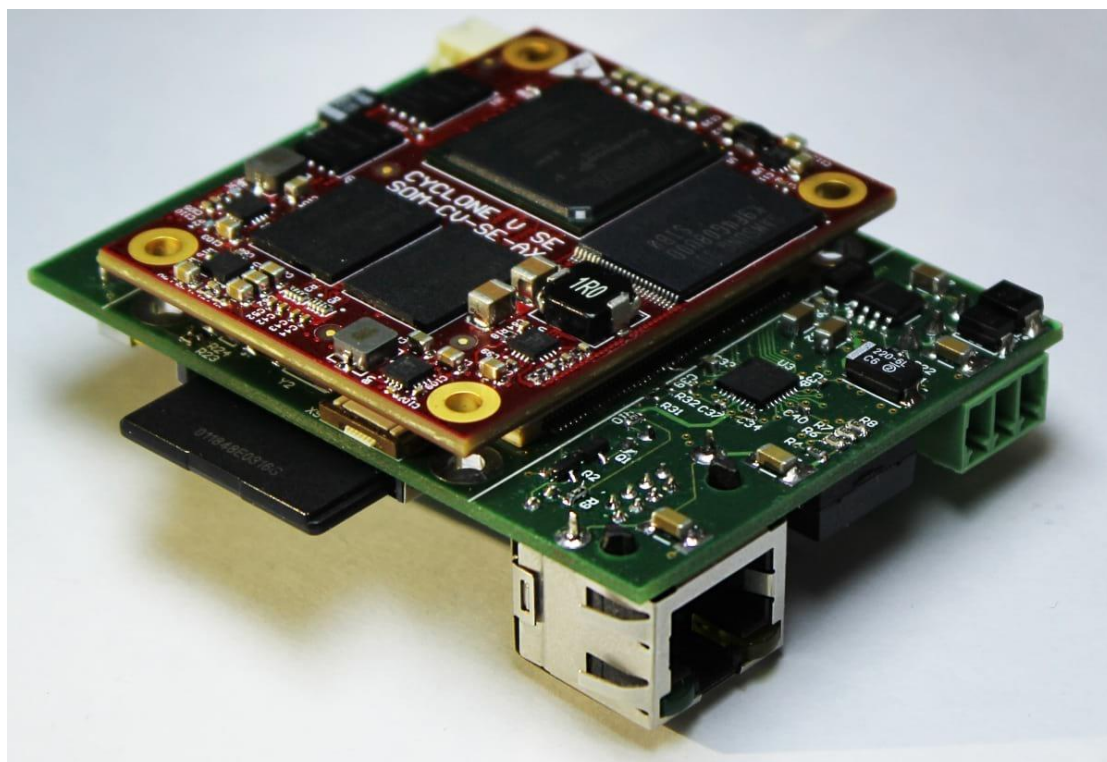
У кожного з них є свої переваги й недоліки. *CPU* забезпечують прийнятне співвідношення продуктивності й універсальності. Майже 99% всіх програм написані для виконання на *CPU*. У графічних процесорів *GPU* більше число ядер і векторна архітектура, висока швидкість обміну з пам'яттю й обміну інформацією. *FPGA* мають найбільшу продуктивність на одиницю енергоспоживання за рахунок апаратних властивостей, але вимагають трудомісткого програмування.

Про ці відмінності докладніше. Універсальні процесори *CPU* – це по суті, «робочі конячки» ІТ-індустрії. Їх можна використовувати для найрізноманітніших задач, але через свою архітектуру *CPU* не настільки ефективні для масивно-паралельних обчислень. В останні роки ця проблема частково вирішується за рахунок збільшення в ІМС процесора кількості обчислювальних ядер, але не більше сотні. Графічні процесори (*GPU*) довгі роки працювали тільки в ніші відображення інформації на екрані. І лише відносно недавно *GPU* стали застосовуватися для задач високопродуктивних обчислень (наприклад, «видобуток» криптовалюти, матричні обчислення, прискорення наукових обчислень, тощо). Робота з графікою призвела до майже ідеального пристосування архітектури *GPU* для масивних паралельних матричних або векторних обчислень. Сучасні *GPU* містять в ІМС вже тисячі обчислювальних ядер. Як результат, у порівнянні з *CPU* їх продуктивність на подібних задачах виросла на порядки.

*FPGA*, на відміну від універсального і графічного процесорів, можна перепрограмувати відповідно до особливостей розв'язуваної на них обчислювальної задачі. Маємо адаптацію синтезуємого спеціалізованого обчислювача під структуру й особливості будь-якої задачі з майже максимальною реалізацією паралелізму, властивому їй. Іншими важливими відмінностями *FPGA* є знижене енергоспоживання в розрахунку на одиницю обчислювальної потужності, а також в загальному випадку масивно-паралельна дрібнозерниста гетерогенна архітектура.

При цьому число ядер в ІМС *FPGA* може досягати значно більшої кількості в порівнянні з *GPU*.

Адаптер *FPGA*-обчислювача може виконуватись в різному форм-факторі (*VPX*, *PC-104*, *Com-express*, *PCIe* і т.д.), яка крім самих ІМС *FPGA* може містити на платі пам'ять типу *SRAM* і *DRAM*, включаючи й ультра-нову типа *HBM*, а також високошвидкісні інтерфейси обміну інформацією (10/40/100 *GE*, *PCI Express*, надшвидкісні *LVDS*-порти, тощо). *FPGA*-прискорювачі також випускаються в форм-факторі одноплатного комп'ютера для реалізації вбудованих систем, що популярно в промислових системах і відеоаналітики.



*FPGA*-прискорювач в форм-факторі *SOM* (одномодульна система)

Кожен ІМС *FPGA* містить масив з мільйонів логічних елементів, який і може бути перепрограмований під різні функціональні задачі. Крім того, є додаткові апаратні ресурси – сигнальні процесори, блоки пам'яті, цифрової обробки й інтерфейсів.

Чому *FPGA* переважає *ASIC* в продуктивності? Завдяки більш досконалим техпроцесам створення кристалів. Відповідно, на тій же площі кристала у *FPGA*

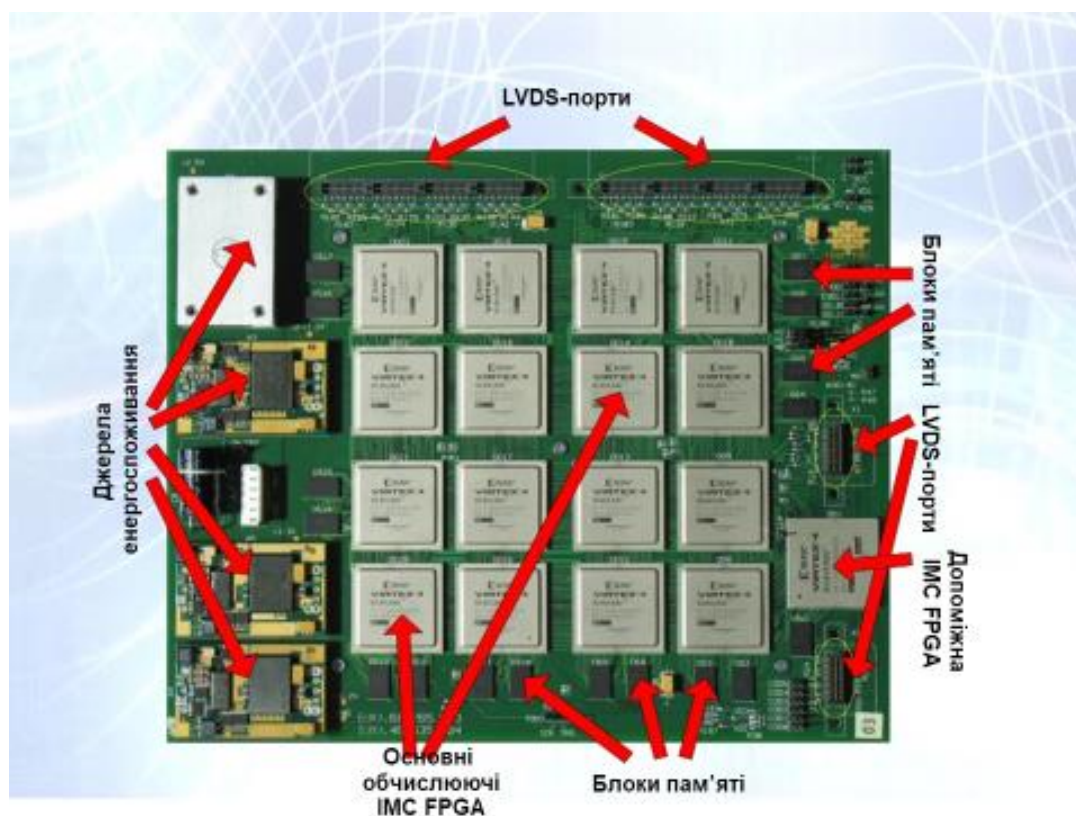
можна розташувати в рази більше число апаратурних ресурсів, ніж у *ASIC*, що і забезпечує вигрaш в продуктивності.

### Області застосування *FPGA*

Одним з базових напрямків застосування *FPGA* є дрібно- і середнє серійне виробництво різних спеціалізованих обчислювальних й керуючих пристроїв (в т.ч. прискорювачів). Виготовлення їх на базі мікросхем *ASIC* економічно недоцільно.

На початок 2018 року за приблизними даними провідних розробників подібних пристроїв, сфери застосування *FPGA*-прискорювачів виглядали приблизно так:

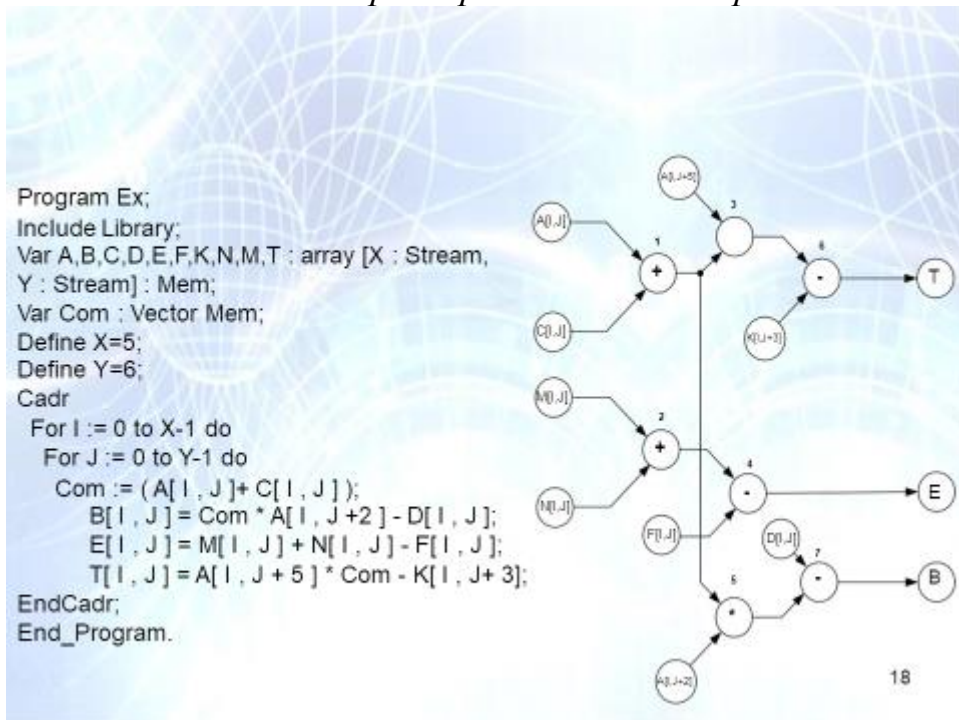
- 50% - військово-промисловий комплекс;
- 10% - промислові застосування;
- 20% - телекомунікації (обладнання базових станцій *GSM* і ін.);
- 10% - обробка відеопотоків (відеостудії, відеоаналітика);
- 10% - вузько спеціалізоване застосування (включно наукові обчислення).



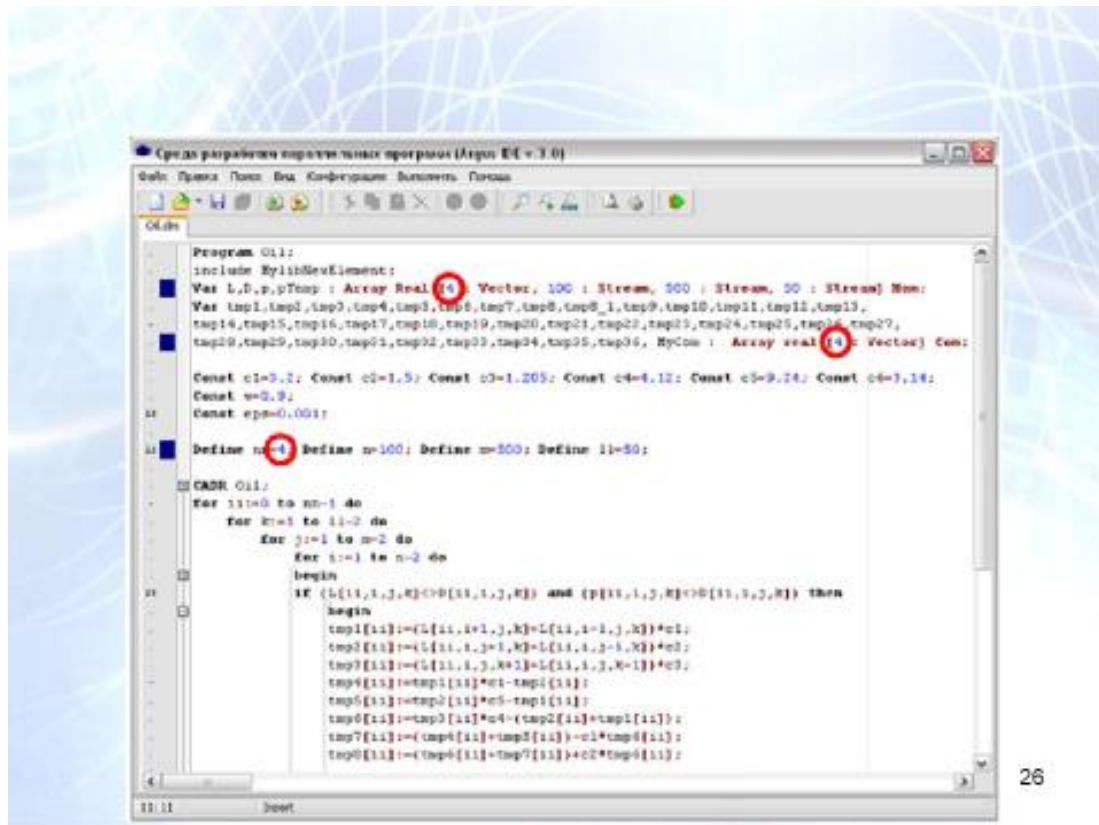
*Модуль *FPGA*-прискорювача на базі *FPGA XILINX VIRTEX-4* для побудови високопродуктивних реконфігурованих обчислювальних систем*



*Високопродуктивні реконфігуровані обчислювальні системи на базі FPGA-прискорювачів для «хмарних» обчислень*



*Паралельна програма та відповідний інформаційний граф потоку обчислень*



26

Приклад масштабування паралельної програми в спеціалізованому середовищі розробки

Однак, незважаючи на переважно військове застосування в минулому, зараз сфера цивільного користування *FPGA*-прискорювачів зростає набагато швидше. У 2015 році *Intel* придбала одного з найбільших виробників *FPGA* – компанію *Altera*. Розробки *Altera* тепер втілюються в кремній вже під брендом *Intel*, в тому числі при проектуванні *CPU* нових поколінь. Наприклад, лінійка *FPGA Intel Cyclone 10* демонструє надвисоку продуктивність (до 134 *GFLOP*) і має розширені можливості обміну інформацією.

Підключення до інших пристроїв виконується через мережевий порт *10GE* або по шині *PCI Express x4*. Ці *FPGA* призначені для систем машинного зору, спостереження, відео трансляцій, а також робототехніки. Молодша модель *Cyclone 10 LP* реалізована як обчислювальний ядро для інженерних систем - управління комплексами датчиків, контролерами двигунів і так далі.

Крім лінійки *Cyclone*, у виробничій програмі *Intel* присутні і інші серії ІМС *FPGA*, успадковані від *Altera*: *MAX*, *Arria* і *Startix*. Останні дві серії – найпотужніші



ІМС *FPGA* з існуючих на ринку. В 2018 році вони оновилися до рівня *Arria 10* і *Startix 10*. Побудовані на гіперфлекс-архітектурі, досягаючи продуктивності у 10 терафлопс (майже на 3 порядки потужніше *Cyclone 10*).

Серії *Cyclone*, *MAX*, *Arria* і *Startix* частково перекривають одна одну по продуктивності, але *Intel* позиціонує кожен серію окремо. Для *Arria* це сигнальні процесори для приладобудування, для *Startix* – високопродуктивні обчислення в дата-центрах, телекомунікації. Область застосування сімейства *Cyclone* варто згадати особливо – «Інтернет речей» (*IoT*).

### **Поняття образу в технології FPGA**

Але щоб використовувати плати *FPGA*-прискорювачів в серверах, потрібно придбати і здійснити програмування логіки їх ІМС під конкретну прикладну задачу. Якщо треба абстрагуватися від маршруту проектування і сфокусуватися на обчислювальній задачі, можна скористатися *OpenCL* і *Intel FPGA SDK for OpenCL*. Для цього буде потрібно пакет підтримки плати *BSP*. Такий пакет надається, якщо для плати заявлена підтримка *OpenCL* (*OpenCL BSP*). Маючи подібний пакет підтримки, можна отримати «середу розробника ПЗ», де є модель платформи, функція для прискорення, бібліотека підтримки часу виконання, модель пам'яті, а також спеціальні розширення для збільшення пропускну здатності. Потім приступають до написання коду, профілізації, оптимізації, тощо.

В результаті використання *SDK* і *BSP* виходить єдиний файл конфігурації (бітовий потік, *bitstream*), яким конфігурується ІМС *FPGA* і виходить кінцева *SOC* під конкретну обчислювальну або керуючу задачу. Результатом програмування є мікропрограма для рішення цієї задачі (наприклад, розрахунок матриці рівнянь, перетворення відео-форматів і т.д.). Така мікропрограма називається *FPGA*-образом (*FPGA Image*). Досить часто замість терміна «образ» використовується термін «*IP*-ядро».

### **Складнощі доступу до технології FPGA для клієнтів**

Незважаючи на привабливу концепцію, – «найвища продуктивність під конкретну обчислювальну задачу», поширенню *FPGA* заважають два об'єктивних чинника. Це висока вартість адаптера (прискорювача) з ІМС *FPGA* і дефіцит розробників, які мають практичний досвід програмування і налагодження *FPGA*-ядер.

Крім прискорювача, необхідно також обзавестися ліцензією на *Intel OpenCL SDK*, без якої можливий тільки запуск скомпільованих ядер, але неможлива їх компіляція. Вимоги до комп'ютера розробника теж досить високі: це в тому числі рекомендації до обсягу ОЗУ в 18-48 ГБ. На машині з 8-ядерним *CPU* і 32 ГБ пам'яті компіляція ядра для обчислення безлічі Мандельброта займає близько 2-х годин. Якщо утилізація процесора перевищує 90%, то компіляція може займати добу і навіть більше. При обсязі пам'яті менш 16 ГБ компіляція може виявитися нездійсненним.

Тому потенційні клієнти активно цікавляться цією технологією, але не поспішають з придбанням *FPGA*-прискорювачів. Також це пов'язано з побоюваннями, що витрати на прискорювач будуть істотними для їх бюджету, а *in-house* команда не зможе на належному рівні освоїти програмування і налагодження *FPGA*-образів.

### **«Хмарні» обчислення на *FPGA***

«Хмарні» сервіси *FPGA* з'явилися як відповідь на високу вартість плат прискорювачів з ІМС *FPGA*. В цьому випадку клієнтам пропонуються в оренду фізичні та/або віртуальні сервери з встановленими в них *FPGA*-прискорювачами. Як правило, це партнерський продукт від виробника (наприклад, *Intel*) і дата-центру як провайдера *IaaS*-послуг.

Одним з рішень проблеми доступності технології для масового застосування бачиться також можливість оренди обчислювальних потужностей на базі *FPGA*. Наприклад, послуга передбачає отримання доступу до сервера з встановленим *FPGA*-прискорювачем на базі ІМС *Intel Arria 10 FPGA*. На сервері розгорнуті

необхідні *SDK* і *BSP* для розробки, налагодження і компіляції *OpenCL*-ядер, засоби розробки для написання хост-додатків (*Visual Studio*).

Деякі компанії проводять для всіх бажаючих навчальні курси з програмування на *OpenCL* для *FPGA*. Подібні курси розробляються спеціально інженерів, науковців, студентів технічних ВНЗ. Зазвичай вони вбирають в себе матеріал офіційних тренінгів *Intel* і дають можливість покрокового вивчення технології від збірки найпростішого додатка до застосування специфічних методів оптимізації, необхідних для досягнення майже максимальної швидкодії.

У такому вигляді *FPGA*-технологія стає більш привабливою для клієнтів, так як вже не потрібно купувати безпосередньо «залізо», а капітальні витрати замінюються операційними. Відповідно, значно розширюється коло компаній, які можуть дозволити собі використання розрахунків на *FPGA*-прискорювачах для своїх проектів.

### **Концепція магазину образів *FPGA***

Створення ефективно працюючого *FPGA*-образу (*IP* ядра) для певної прикладної задачі досить трудомістка. Навіть у добре злагодженої команди на програмування образу може піти до кількох місяців.

Тому сама собою напрошується концепція магазину образів (*IP* ядер), – за аналогією з існуючими магазинами додатків для таких платформ як *MacOS*, *Windows* або *Android*. Розробники могли б передавати туди працездатні образи, створені ними для різних задач, а клієнти – купувати їх для завантаження на свої сервери з *FPGA*-прискорювачами, якщо ці образи відповідають їх задачам. Тим самим для клієнтів значно прискорився цикл розробки нових проектів, а програмісти отримували дохід від раніше зробленого й були захищені від піратського поширення створених ними образів.