

Огляд характеристик ПЛІС Xilinx Versal

Ілля Тарасов

Перехід до технологічного процесу з нормами 7 нм становить істотний інтерес як для виробників електронних компонентів, які розраховують почати виготовляти нове покоління продукції, так і для розробників, які очікують від цієї елементної бази передових характеристик, що дозволять реалізувати конкурентоспроможні вироби. У статті розглядаються технічні характеристики сімейства ПЛІС Xilinx Versal.

ВСТУП

Наразі склад платформи Versal має такий вигляд (рис. 1):

- Versal AI — сімейство з матрицею процесорних ядер AI core;
- Versal Prime — сімейство зі збалансованими характеристиками;
- Versal Premium — високопродуктивні ПЛІС із піковими для серії характеристиками.

Існують такі різновиди Versal: Versal RFSOC, Versal HBM і Versal Edge. Наразі доволі важко вказати на прямі відповідності сімейств Versal попереднім поколінням ПЛІС Xilinx, проте можна умовно порівняти призначення Versal Prime з мікросхемами Kintex, що являють собою високопродуктивні платформи з відносно невисокою ціною (через відсутність великої кількості високошвидкісних приймачів-передачів). Флагманське рішення в попередніх поколіннях було представлено серією Virtex, якій відповідає цілий спектр рішень — Versal AI (з додаванням матриці процесорних ядер з архітектурою RISC), Versal Premium (великий логічний об'єм і безліч приймачів-передачів). Платформа Versal RFSOC має високу продуктивність і реалізацію в корпусі височастотних АЦП і ЦАП. Нарешті, аббревіатура HBM означає High Bandwidth Memory, що, звісно ж, стосується реалізації в корпусі ПЛІС компонента динамічної пам'яті об'ємом у кілька гігабайт. Аналогічне рішення вже було у складі продукції Xilinx, що дає змогу розширення функціональності та поліпшення характеристик у варіанті Versal HBM.

Певний інтерес становить платформа Versal Edge. Нагадаємо, що термін edge

computing перекладається як «граничні обчислення» і відповідає архітектурі системи, в якій основний масив обчислень проводиться в місці отримання даних — «на границі». Такий архітектурний підхід, який сам по собі потребує глибокого методологічного опрацювання, є серйозною альтернативою підходу, за якого ПЛІС використовують переважно як інтерфейс до зовнішніх пристроїв, виконуючи тільки введення і попереднє опрацювання даних, надсилаючи їх на ПК, який інтуїтивно уявляють розробники як потужнішу обчислювальну платформу. Однак сучасні ПЛІС мають істотно вищу продуктивність порівняно з ПК, а наявність на кристалі процесорної підсистеми ARM дає змогу додати в ПЛІС функціональність звичних процесорних систем, утворюючи гетерогенну систему на кристалі. Оскільки останнім часом платформа ARM прони-

кає у світ програмування, а інструменти розробки набувають значного поширення (та уніфікуються з інструментами для x86), розробники отримують можливість реалізувати в рамках ПЛІС увесь комплекс збирання даних, їх обробки та візуалізації для користувача, не залучаючи ПК із процесором x86 як обов'язковий елемент системи.

У попередніх поколіннях ПЛІС в ролі подібної платформи могли виступати мікросхеми — low-cost portfolio — Spartan6/7, Artix-7, і особливо Zynq-7000, що мають на кристалі процесорну систему ARM. Відносно невисока ціна давала змогу розширити спектр застосування ПЛІС без спеціального техніко-економічного обґрунтування їхнього застосування, оскільки сумарні ціни на комплектуючі для ПЛІС початкового рівня й альтернативи у вигляді системи на базі CPU перебувають у порівнянній категорії, однак ПЛІС дає змогу гнучкіше будувати системну архітектуру, залучаючи в разі потреби апаратні прискорювачі на базі логічних елементів і особливо блоків цифрового опрацювання сигналів XtremeDSP.

Порівняльні характеристики трьох сімейств ПЛІС Versal наведено в таблиці 1. У таблиці видно, що для Versal



Рис. 1. Сімейства платформи Xilinx Versal

Таблиця 1. Порівняльні характеристики сімейств ПЛІС Versal			
	Versal AI	Versal Prime	Versal Premium
Процесорна підсистема	2-ядерний Arm Cortex-A72; 48 кбайт/32 кбайт L1-кеш; 1 Мбайт L2-кеш; 2-ядерний Arm Cortex-R5F; 32 кбайт/32 кбайт L1-кеш		
Процесорних ядер AI	128–400	–	–
Логічних елементів, тис.	540–1968	329–2233	1575–7352
Блоків DSP	928–1968	464–3984	1904–14352
Пам'ять, Мбіт	90–191	54–282	198–994
Приймачі-передаччі	8–44	8–48	72–168

Таблиця 2. Характеристики ПЛІС сімейства Versal Premium							
ПЛІС	VP1102	VP1202	VP1402	VP1502	VP1552	VP1702	VP1802
Блоків DSP	1904	3984	2672	7440	7392	10896	14352
Логічних елементів, тис.	1575	1969	2233	3763	3837	5558	7352
Логічних генераторів (LUT)	719872	900224	1020928	1720448	1753448	2540672	3360896
Приймачі-передавачі GTYP 32G	8	28	8	28	68	28	28
Приймачі-передавачі GTM 58G (112G)	64 (32)	20 (10)	96 (48)	60 (30)	20 (10)	100 (50)	140 (70)
CCIX & PCIe w/DMA (CPM)	–	2x Gen5x8, CCIX	–	2x Gen5x8, CCIX			
Контролери PCI Express з CXL	2x Gen5x4			8x Gen5x4	2x Gen5x4		
100G Multirate Ethernet MAC	6	2	6	4	4	6	8
600G Ethernet MAC	4	1	8	3	1	5	7
600G Interlaken	2	0	2	1	0	2	3
400G високошвидкісний криптомодуль	3	1	5	2	2	3	4

Premium досягнуто значних абсолютних показників — наприклад, (для сімейства Virtex UltraScale+ максимальна ємність у логічних елементах дорівнює 8938). За показниками інших логічних ресурсів — блокової пам’яті та кількості блоків цифрової обробки сигналів — Versal Premium також демонструє рекордні параметри. Характеристики сімейства Versal Premium наведено в таблиці 2.

Крім переходу на технологічні норми 7 нм, сімейство Xilinx Versal відкриває і новий підклас ПЛІС — ACAP (*Adaptive Compute Acceleration Platform*). Раніше сімейства Zynq7000 і Zynq MPSoC були заявлені як такі, що належать до окремого різновиду ПЛІС — APSOC (*All-Programmable System On Chip*, тобто системи, які повністю програмуються на кристалі). Дане позиювання не скасовує той факт, що в основі платформи лежить матриця конфігурованих логічних елементів з архітектурою FPGA, однак виробнику важливо звернути увагу розробників на необхідність використання інших методологій проектування, які могли б ефективно поєднувати переваги різних підсистем на кристалі — реконфігурованої і реалізованої апаратно.

Сімейство Versal має схожі з попередніми поколіннями блоки статичної пам’яті, а підсистема процесорів ARM і приймачі-передавачі вимагають окремих публікацій для висвітлення. Тому в цій статті буде описано найцікавіші компоненти, які помітно відрізняються від реалізованих у попередніх поколіннях ПЛІС. До таких належать логічні елементи, блоки цифрової обробки сигналів і, природно, матриця RISC-ядер.

АРХИТЕКТУРА КОНФІГУРОВАНОГО ЛОГІЧНОГО БЛОКУ СІМЕЙСТВА ПЛІС XILINX VERSAL

Конфігурований логічний блок (КЛБ, CLB — *Configurable Logic Block*) —

основний будівельний елемент матриці прогамованих ресурсів FPGA. Він об’єднує деяку кількість логічних елементів (*logic cell*), які традиційно склалися з 4-вхідної таблиці істинності (LUT4) і тригера. Сучасні FPGA Xilinx мають у складі логічного елемента 6-вхідну LUT і два тригери, що зумовлює різницю між показниками «кількість логічних елементів» і «кількість LUT» у таблицях характеристик. На практиці Xilinx вказує як логічну ємність ПЛІС ту кількість стандартних (LUT4 + тригер) елементів, які знадобилися б для реалізації проекту, що заповнює всю ПЛІС цього обсягу. Ця величина, очевидно, більша, ніж кількість LUT6.

Протягом тривалого часу, починаючи з 65-нм сімейства Virtex-5, архітектура логічного елемента змінювалася незначно. Введена у Virtex-5 6-входова LUT збереглася аж до сімейства Versal. Проте низка змін щодо сімейства Virtex відбулася як на рівні логічного осередку, так і більш суттєві зміни є на рівні КЛБ. На рисунку 2 показано склад КЛБ Versal.

Відмінності КЛБ Versal від КЛБ попередніх поколінь ПЛІС Xilinx:

- КЛБ збільшено в 4 рази порівняно з попередніми поколіннями ПЛІС і

містить 32 LUT і 64 тригери (порівняно з 8 LUT і 16 тригерами);

- додано каскадні з’єднання між LUT для зменшення затримок трасування для багаторівневої логіки;
- наддовгі трасувальні лінії (SLL) тепер частина КЛБ, а не окрема колонка трасувальних ресурсів;
- у КЛБ більше немає мультиплексорів F7MUX, F8MUX, F9MUX, натомість складні логічні функції реалізуються за допомогою додаткових LUT;
- у парі LUT + тригер тепер тільки три виходи, а не чотири;
- у LUT можна реалізувати дві незалежні логічні функції з шістьма входами;
- каскадувальні мультиплексори ланцюгів перенесення можуть вибирати розряди, починаючи з 0 і 4;
- усі КЛБ тепер мають один і той самий тип, відмінність між «тільки логіка» і «логіка, пам’ять або зсувний регістр» є у LUT всередині КЛБ;
- режим розподіленої пам’яті в LUT спрощено, реалізовано підтримку для 32 і 64 елементів пам’яті, великі розміри пам’яті вимагають додаткової логіки;
- вихідні мультиплексори КЛБ введено в сімействі Versal;

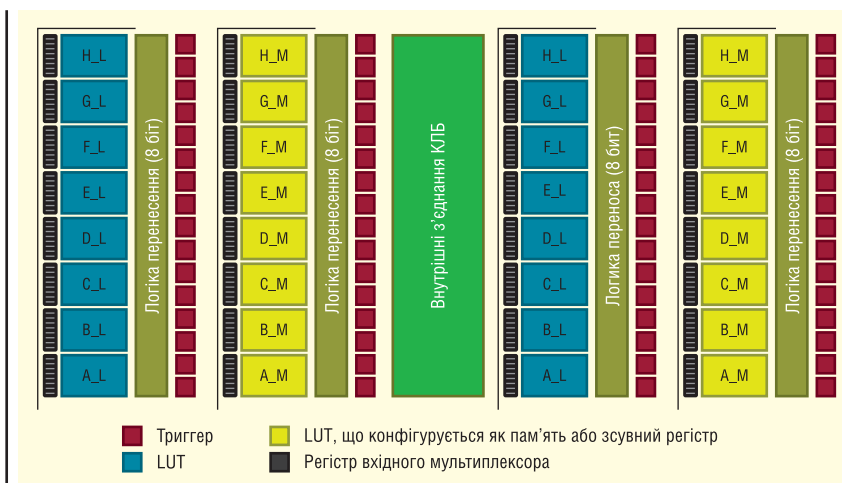


Рис. 2. Склад КЛБ ПЛІС серії Versal

- додано додаткові регістри вхідних мультиплексорів (регістри *Itux*).

Можна бачити, що в КЛБ Versal спостерігається тенденція до збільшення логічних ресурсів і зменшення їхньої різноманітності. Для попередніх поколінь FPGA окремо зазначалося, що введення додаткових апаратних ресурсів, таких як F7MUX, полегшує побудову деяких вузлів. У сімействі Versal спостерігається рух у бік простішої та уніфікованішої архітектури з додатковими зв'язками. Крім усунення апаратних мультиплексорів, що розширюють логічні функції, ту саму тенденцію видно в обмеженні режимів роботи розподіленої пам'яті.

Цікавим є пояснення до регістрів *Itux*. Крім очевидного застосування у вигляді додаткової конвеєризації критичних ланцюгів, ці компоненти можуть бути також використані для усунення порушень часу утримання сигналу (*hold violations*). Можна коротко нагадати, чому ця проблема є суттєвою. Зменшення технологічних норм під час виробництва цифрових мікросхем супроводжується зменшенням часу поширення сигналу та збільшенням технологічного розкиду параметрів реалізованих компонентів. Тому міркування, справедливі для старіших технологічних процесів, частково втрачають своє підґрунтя. Наприклад, під час передавання даних між двома тригерами можна було покладатися на те, що затримка передавання даних від виходу одного регістра до входу іншого буде свідомо більшою, ніж розсинхронізація між їхніми тактовими входами. Тому про виконання вимоги на час утримання сигналу (*hold time*) можна було говорити практично з повною упевненістю, розраховуючи на те, що спеціально прокладені тактові мережі ПЛІС забезпечують мінімальну неузгодженість затримок для тактових сигналів.

Якщо ж говорити про технологічні норми 28 нм і менше, то два тригери за їхнього невдалого розташування можуть мати надто коротку лінію даних, що їх пов'язує, але при цьому отримувати тактові сигнали з різних гілок тактового дерева. Може виникнути ситуація, коли сигнал на вході другого тригера вже почне змінюватись, але фронт тактового сигналу, за яким мало записується попереднє значення, ще тільки-но досяг тактового входу цього тригера. У такому випадку і спостерігається «порушення часу утримання сигналу» (*hold violation*), яке призводить до невизначеної роботи другого тригера.

У лінійці продуктів Xilinx боротьба з цим явищем спостерігалася поетапно.

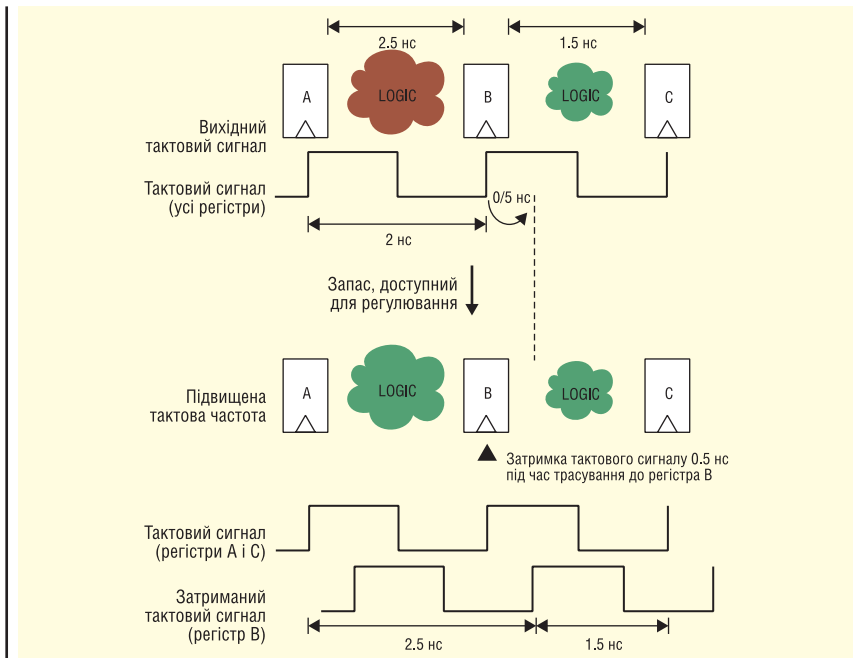


Рис. 3. Механізм *time borrowing* у сімействах UltraScale+

Перш за все, з виходом ПЛІС серії 7, що випускаються за технологічним процесом з нормами 28 нм, в САПР Vivado було додано відповідні перевірки на час утримання сигналу. Для середніх і великих ПЛІС серії 7 на практиці спостерігалися ефекти поліпшення стабільності роботи проекту під час його трасування в САПР Vivado замість попереднього покоління САПР ISE, яка також підтримувала серію 7.

Для ПЛІС серії 7 зберігалася можливість трасування глобальної тактової мережі по всьому кристалу, включно з найбільшими Virtex-7 з еквівалентною логічною ємністю 2 млн елементів. Однак уже в 20-нм ПЛІС UltraScale від цієї можливості довелося відмовитися, обмеживши трасування окремими тактовими регіонами, які зобов'язані мати незалежні тактові мережі, керовані локальними блоками синхронізації.

У 16-нм сімействі UltraScale+ з'явилася ще одна можливість — наприклад, *time borrowing* (дослівно: «запозичення часу»). Її суть проілюстровано на рисунку 3. Видно, що у вихідній немодифікованій схемі затримка поширення сигналів різняться, у результаті один із ланцюгів є критичним. Однак регулювання затримок у тактовій мережі, яке стало можливим в UltraScale+, дає змогу трохи затримати фронт тактового сигналу для регістра B. Це погіршить ситуацію для ланцюга B-C, однак цей ланцюг від самого початку перебував у кращих умовах. Тому, як показано на рисунку 3, після індивідуального підстроювання за-

тримки тактового сигналу час між фронтами на регістрах A і B становитиме 2.5 нс, а між B і C — 1.5 нс. Таким чином, період тактового сигналу в цьому прикладі вдалося зберегти рівним 2 нс.

Подібна можливість, вочевидь, корисна і для усунення проблем із часом утримання сигналу. Дійсно, підстроювання фронту тактового сигналу в невеликих межах дає змогу вивести його з часового інтервалу, у якому дані на вході мають залишатися незмінними.

Можливість захоплення сигналу в додаткових регістрах *Itux* додатково розширює можливості САПР із трасування критичних до часу поширення сигналу ланцюгів. Можна зазначити, що перераховані підходи застосовуються в САПР автоматично, без необхідності їх явної вказівки розробником.

ХАРАКТЕРИСТИКИ БЛОКІВ ЦИФРОВОЇ ОБРОБКИ СИГНАЛІВ

Блок цифрової обробки сигналів, який раніше позначали DSP48, у сімействі Versal змінив своє позначення на DSP58. Це сталося через зміну розрядності акумулятора.

Можна коротко нагадати архітектуру і порядок застосування блоків цифрової обробки сигналів у ПЛІС. Безсумнівно, ці блоки є одним із ключових компонентів ПЛІС, що зумовлюють їхнє широке застосування в задачах цифрової обробки сигналів. Центральним елементом такого блоку стає апарат-

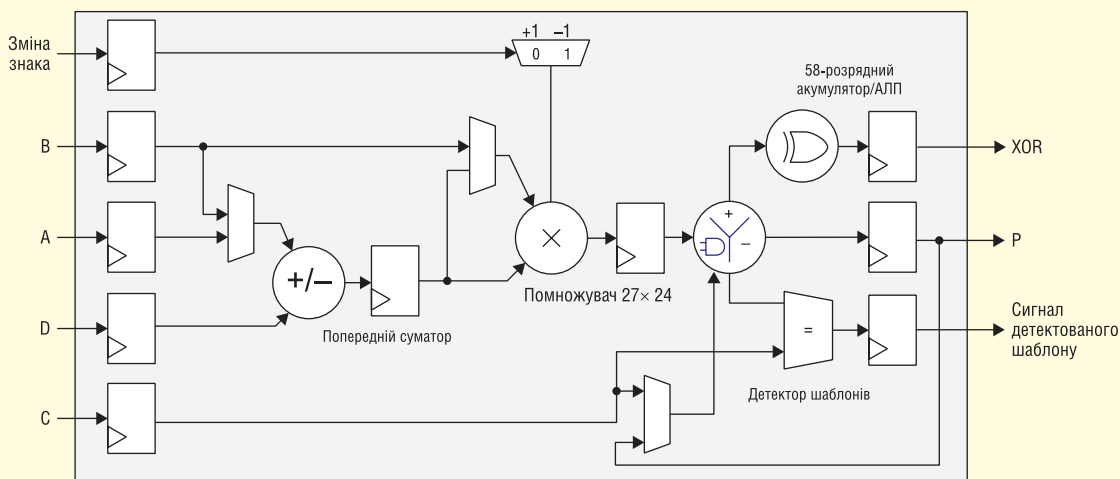


Рис. 4. Спрощена схема блока цифрової обробки сигналів у ПЛІС Versal

ний помножувач незалежних операндів. З одного боку, операція множення істотно складніша, ніж операції додавання або віднімання (а тим паче ніж побітові логічні операції), тому реалізація такої операції на базі програмованих логічних елементів призведе до появи «пухкої» логічної структури з неефективним використанням ресурсів елементів ПЛІС та невисокими показниками частоти. Тому ще в сімействі Virtex-II було додано апаратні блоки, що виконують множення 18-розрядних незалежних операндів. Це істотно покращило показники продуктивності ПЛІС під час реалізації таких поширених вузлів, як цифрові фільтри з постійними коефіцієнтами.

Архітектура блоку цифрової обробки сигналів постійно еволюціонувала, отримуючи нові функціональні можливості. Зокрема, було виявлено, що критичним компонентом для цифрових фільтрів став акумулятор, оскільки фільтрація передбачає не просто множення, а множення та підсумовування (накопичення, акумулювання) результатів. Тому в сімействі Virtex-4 було реалізовано структуру, що містила помножувач 18-розрядних операндів і 48-розрядний апаратний акумулятор. Весь блок був здатний працювати на високій тактовій частоті, реалізуючи цифрові фільтри без залучення додаткових програмованих елементів ПЛІС. Можливості блоку позначалися формулою $18 * 18 = 48$, що означало можливість перемноження 18-розрядних операндів і 48-розрядний акумулятор.

Розрядність операндів для цифрової обробки сигналів, безсумнівно, диктується практичними потребами. Однією з них є необхідність збільшення розряд-

ності, причому самі дані, одержувані з АЦП, цілком можуть бути представлені 18 розрядами (високошвидкісні АЦП часто мають 12–14 розрядів). Однак подання коефіцієнтів цифрових фільтрів цілком може вимагати більшої розрядності. Наприклад, під час моделювання фільтрів у програмному пакеті Matlab можна використовувати формат подання коефіцієнтів із плаваючою комою, але мантиса такого числа (у форматі `fp32`) потребує 25 розрядів. Перетворення такого коефіцієнта у 18-розрядний цілочисельний формат призводить до погіршення характеристик фільтра. Тому вже наступне покоління блоку DSP48 давало змогу перемножувати 18- і 25-розрядні операнди за збереження розрядності акумулятора.

Спрощену схему блока цифрової обробки сигналів сімейства Versal показано на рисунку 4.

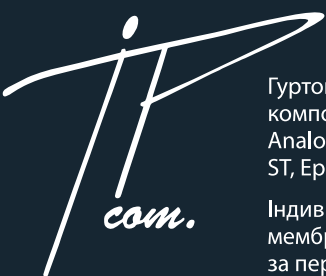
Блок отримав такі основні покращення порівняно з попередніми поколіннями ПЛІС:

- точкою `fp32` у блоці реалізовано підтримку операцій множення і додаван-

ня над числами з плаваючою комою, помножувач працює з операндами розрядності 27 і 24 біт;

- акумулятор і пов'язаний з ним арифметико-логічний пристрій мають розрядність 58 біт;
- акумулятори каскадуються для утворення 116-розрядного пристрою;
- вхід А блоку має розрядність 34 біти, з яких 27 є операндом для помножувача, а всі 34 розряди можуть бути використані для формування вихідного 58-розрядного результату;
- завдяки тому, що розрядність обох операндів стала не меншою за розрядність мантиси числа з плаваючою комою одинарної точності.

Блок цифрової обробки сигналів у сімействі Versal має зворотну сумісність із подібними блоками попередніх сімейств ПЛІС Xilinx. Використання операцій множення в HDL-описі призводить до автоматичного синтезу таких блоків у проєкті. Також зберігається можливість генерації IP-ядер цифрової обробки сигналів за допомогою вбудованих засобів САПР.



ТОВ «АЙПІ-КОМ»

Гуртове постачання широкого спектра електронних компонентів відомих брендів: Xilinx, Intel (Altera), Analog Devices, Texas Instruments, NXP, Microchip, ST, Epcos, Vishay, Sharp, KLS тощо

Індивідуальна розробка та виготовлення мембранних плівкових клавіатур різної складності за персональними вимогами замовника

☎ моб.: 098-980-48-38

📞 ill1982
 ✉ ill82@ukr.net
 🌐 ip-com.net.ua

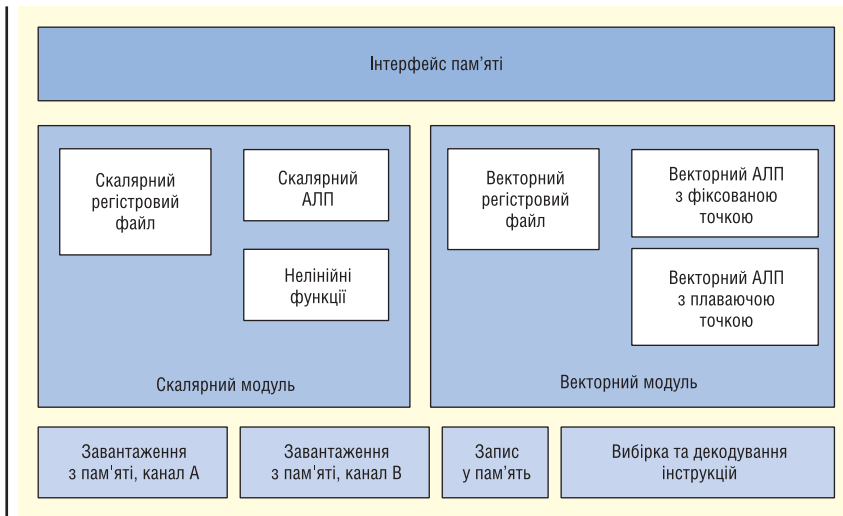


Рис. 5. Спрощена схема ядра AI Core

ХАРАКТЕРИСТИКИ ПРОЦЕСОРНИХ ЯДЕР AI CORE

Найбільш вагомим компонентом сімейства Versal є матриця процесорних ядер, реалізована в платформі Versal AI (рис. 5). З додаванням такої важливої складової стало можливим говорити про об'єднання в складі однієї мікросхеми трьох основних підходів до організації обчислень. У складі платформи Zynq можна було бачити комбінацію універсальних процесорних ядер і матриці програмованих логічних елементів, що давало крайній підходи.

Кожне ядро являє собою 32-розрядний RISC-процесор із векторним розширенням. Його функції можна перерахувати таким чином:

- регістри загального призначення, організовані у вигляді реєстрового файлу;
- нелінійні функції (квадратний корінь, \sin/\cos);
- помножувач 32-розрядних скалярних операндів;
- перетворення даних між цілочисельним форматом і поданням із плаваючою комою;
- три пристрої генерації адреси (AGU, Address Generation Unit): два для завантаження даних і один для запису в пам'ять, підтримується кілька режимів адресації, зокрема режим генерації адрес для підтримки швидкого перетворення Фур'є;
- векторне розширення операцій з фіксованою точкою.

Векторне розширення підтримує безліч форматів представлення операндів, з докладним списком яких можна ознайомитися в описі процесорів AI

Core. При використанні 8-бітних операндів векторне розширення виконує 128 операцій «множення з накопиченням» (MAC) за такт — можна окремо звернути увагу, що йдеться про одне ядро AI core, а не про всю матрицю з 128–400 процесорів. Зрозуміло, при збільшенні розрядності операндів кількість операцій зменшується, і для двох операндів у форматі fp16 кількість MAC дорівнює 32, а для fp32 — 8.

Якщо зіставити цю продуктивність із показниками блоків цифрової обробки сигналів, можна помітити, що кількість MAC, що виконуються за один такт, становить 4 096–12 800 для діапазону ПЛІС, наведених у таблиці 1. Таким чином, матриця RISC-процесорів завдяки векторному розширенню може виконати більше операцій MAC, ніж усі блоки цифрової обробки сигналів у тих самих ПЛІС (928–1 968 у сімействі Versal AI). У цьому разі проявляється ефект гетерогенної архітектури, оскільки для блоків цифрової обробки сигналів, розподілених по площі ПЛІС і оточених додатковими конфігурованими ресурсами, можуть знайтися й такі завдання, які менш ефективно виконуватимуться в матриці RISC-процесорів, що мають цілком певне матричне розміщення та задані з'єднання з оточуючими компонентами. Розробник при цьому вільний у виборі найбільш придатної підсистеми ПЛІС — елементів з блоками DSP, RISC-процесорів або підсистеми ARM.

Скалярні регістри RISC-ядра AI представлені регістрами загального призначення r0...r15, 64-розрядними конфігураційними регістрами s0–s7, і двома наборами 20-розрядних регістрів — модифікатори та покажчики.

128 біт	256 біт	512 біт	1024 біт	
VrI0	Wr0	xa	ya	–
Vrh0				
VrI1	Wr1	xb	ya	Yd (старші розряди)
Vrh1				
VrI2	Wr2	xc	–	–
Vrh2				
VrI3	Wr3	xd	–	Yd (молодші розряди)
Vrh3				
Vcl0	Wc0	xe	–	–
Vch0				
Vcl1	Wc1	xf	–	–
Vch1				
Vdl0	Wd0	yg	–	–
Vdh0				
Vdl1	Wd1	zh	–	–
Vdh1				

Векторні регістри RISC-ядра AI для зручності представлення показано в таблиці 3.

У форматі цієї статті немає практичного сенсу копіювати інформацію про всі регістри AI core, оскільки вона наведена в технічній документації. Наведені вище відомості дають змогу скласти загальне враження про клас цього процесора і його можливості. Можна додати, що процесор використовує 128-розрядну команду і належить до архітектури VLIW, оскільки в одній команді може бути задано до 6 операцій. Таким чином, сумарна продуктивність всієї підсистеми AI виявляється досить високою, і її слід зіставляти не з процесорною підсистемою ARM (яка в ПЛІС відіграє в основному допоміжну роль організації інтерфейсів і взаємодії з користувачем і зовнішнім оточенням під керуванням однієї з операційних систем), а з матрицею конфігурованих логічних ресурсів FPGA.

ВИСНОВОК

Наведені на сайті компанії-виробника відомості дають змогу оцінити можливості та загальну продуктивність ПЛІС сімейства Versal. Можна зазначити, що розробка проектів для пристроїв великого логічного об'єму на рівні RTL-відображення вже досить складна, а індивідуальне програмування кожного з декількох сотень ядер AI є не виправдано трудомістким завданням. У цьому разі є сенс розраховувати на подальший випуск САПР, що підтримують наскрізний маршрут проектування із застосуванням високорівневих описів проекту та спеціалізованих фреймворків для розв'язання завдань, які найбільше підходять для AI core. **СН**