

УДК 004.7:621.39

Акселерована реєстрація MIPI CSI відеопотоку в задачах передачі відео реального часу

Ходнев Т. А., Голуб М. С., Кужильний О. В., Лисенко О. М., Варфоломеев А. Ю.

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

E-mail: *t.khodniev@kpi.ua*

В роботі розглянуто питання зменшення затримок передачі відеопотоку в реальному часі з камер, що передбачають підключення через інтерфейс MIPI CSI. Наведено основні складові затримки реєстрації/передачі відеопотоку, проаналізовано міру їхнього внеску в сумарну затримку, дано оцінку можливості потенційного впливу на них при розробці систем реєстрації/передачі відеопотоку реального часу. Окреслено проблематику, пов'язану з застосуванням буферизації в таких системах, головним чином, вплив наявності в системі покадрової буферизації на величину сумарної затримки. Охарактеризовано обмеження реалізацій модулів MIPI, що призводять до збільшення затримок реєстрації відеопотоку з MIPI CSI камер в певних ARM-мікропроцесорах. Запропоновано структурно-функціональну організацію систем реєстрації MIPI CSI відеопотоку з застосуванням потокових цифрових шин, фрагментації кадрів відеопотоку та DMA транзакцій, що не потребує використання покадрової буферизації та, відповідно, дозволяє зменшити сумарну затримку реєстрації відеопотоку. Запропоновану структурно-функціональну організацію може бути реалізовано на базі SoC-FPGA рішень, в тому числі, з використанням існуючих IP-ядер. Наведено прагматичні особливості та відповідний оціночний вираз для визначення обмежень величини затримки при застосуванні запропонованих рішень. Для експериментальної перевірки, створено прототип системи реєстрації/передачі відеопотоку на основі SoC-FPGA Xilinx сімейства Zynq-7000 відповідно до запропонованої структурно-функціональної організації, розглянуто його специфіку та особливості реалізації. Дано оцінку отриманій швидкодії прототипу та розглянуто можливі напрямки подальшого зменшення сумарної затримки реєстрації/передачі відеопотоку. Результати роботи можуть бути використані для зменшення величини затримок реєстрації відеопотоку з MIPI CSI камер в системах відеопередачі реального часу на основі SoC-FPGA.

Ключові слова: SoC, FPGA, MIPI CSI, GigE Vision, відео, потокова передача

DOI: [10.20535/RADAP.2020.82.35-43](https://doi.org/10.20535/RADAP.2020.82.35-43)

Вступ

По мірі того, як задачі, що потребують використання відеокamer стають складнішими, все більшою стає потреба у датчиках зображення з високою роздільною здатністю. Актуальні вимоги ринку до датчиків зображення призводять до обмеженої застосовуваності типових паралельних інтерфейсів, оскільки вони відносно складно масштабуються за рахунок необхідності у збільшенні кількості сигнальних ліній. Це стало однією з передумов розробки альянсом MIPI (Mobile Industry Processor Interface) стандарту CSI (Camera Serial Interface) для забезпечення стандартизованого, надійного та високошвидкісного інтерфейсу сенсорів зображень з низьким енергоспоживанням, який би підтримував широкий спектр сучасних рішень з використанням камер [1].

В попередніх роботах [2–5] було показано, що мінімізація транспортних затримок є однією з ключових задач передачі відео в реальному часі. Успі-

шне вирішення даної задачі потребує накладення ряду обмежень як на апаратну, так і на програмну складову взаємодіючих систем. В роботі [3] було проаналізовано ефективність використання тракту зв'язку широкоживаними прикладними технологіями передачі відео реального часу — на базі RTSP та на базі GigE Vision стеків та порівняно ці конкуруючі технології між собою за критерієм ефективності використання ними тракту зв'язку топології точка-точка з Ethernet-каналом. Результати роботи [3] свідчать про те, що технологія відеопередачі на базі GigE Vision створює менше навантаження на канал передачі, призводить до меншої кількості помилок в процесі передачі (а відповідно і практично елімінує затримки, пов'язані з необхідністю повторної передачі кадрів відео чи їх складових), є більш ефективною та потенційно більш придатною для вирішення задач передачі відео в реальному часі. В роботах [3,4] було представлено та апробовано модифіковану версію програмної бібліотеки ARAVIS, що

дозволяє реалізувати сумісну зі стандартами GigE Vision трансляцію відеопотоку з камери.

Дана ж робота, в свою чергу, присвячена вирішенню задачі мінімізації затримки реєстрації та подальшої передачі відеопотоку з MIPI CSI камер за рахунок застосування акселерованих IP-модулів (Intellectual Property) реєстрації відео на базі SoC-FPGA (System on a Chip – Field Programmable Gate Array) та є логічним продовженням циклу попередніх робіт за прикладною тематикою організації високошвидкісних трактів відеопередачі реального часу.

1 Проблематика

В системах, що отримують потік відео з камери, підключеної через інтерфейс MIPI CSI, здійснюють його попередню обробку та подальшу передачу кінцевому пристрою-приймачу, сумарна затримка передачі відеопотоку може бути оцінена, виходячи з (1):

$$T_{D\Sigma} = T_{Dcam} + T_{Dif} + T_{Dpp} + T_{Dtx}, \quad (1)$$

де $T_{D\Sigma}$ – сумарна затримка передачі відеопотоку, с; T_{Dcam} – затримка реєстрації та обробки зображення камерою, с; T_{Dif} – затримка інтерфейсу (MIPI CSI), с; T_{Dpp} – затримка попередньої обробки відеопотоку, с; T_{Dtx} – затримка передачі відеопотоку кінцевому вузлу, с.

Розглянемо внесок окремих складових (1) в сумарну затримку передачі відеопотоку та можливість потенційного впливу на них при розробці відповідних апаратно-програмних рішень:

- Затримка реєстрації та обробки зображення камерою (T_{Dcam}) пов'язана з часом, необхідним для фіксації зображення матрицею камери, а також обробки зображення самою камерою перед подальшою передачею з використанням інтерфейсу MIPI CSI, є специфічною для кожного конкретного модуля камери та, як правило, наводиться у відповідній документації. Тим не менш, можна виділити ряд особливостей даної затримки, характерних для більшості MIPI CSI камер.

По-перше, це тип затвору модуля камери (rolling/global shutter). Зображення рухомих об'єктів, отримані з застосуванням глобального затвору є більш чіткими, однак, використання плаваючого затвору дозволяє досягти потенційно менших затримок реєстрації відео [6]. Варто зазначити, що спотворення, викликані застосуванням плаваючого затвору при зйомці рухомих об'єктів можуть бути частково компенсовані шляхом утилізації відповідних алгоритмів корекції на етапах попередньої обробки відеопотоку системою відеопередачі

чи обробки результуючого відеопотоку на кінцевому вузлі-приймачі [7].

По-друге, це особливості апаратної організації модуля камери та її постініціалізаційна конфігурація. Варто зазначити, що не всі модулі камер здатні передавати реєстрований потік відео безпосередньо та можуть вносити додаткові затримки в міру особливостей структури та організації їх апаратної складової. Загалом, деякі камери дозволяють розпочинати відправку кадру відео лише після його повного захоплення та буферизації. Деякі камери надають додаткові можливості обробки реєстрованих кадрів перед подальшою передачею відеопотоку через інтерфейс MIPI CSI. Зокрема, вони можуть здійснювати просторове фільтрування для подавлення шумів, поворот кадрів, стиснення у JPEG тощо [8]. Вимкнення певного функціоналу обробки, що не підтримує операцій над потоком, потребує буферизації, за наявності такої можливості, може призвести до суттєвого зменшення загальної затримки реєстрації відеопотоку.

- Затримка інтерфейсу (T_{Dif}) включає в себе час, необхідний для формування MIPI пакетів камерою, їх відправки трансівером камери, проходження сигнальних ліній C-PHY (D-PHY або M-PHY), отримання їх приймачем системи реєстрації відео, а також час, необхідний для представлення системою реєстрації результуючого відеопотоку у вигляді, придатному для його подальшої обробки [9, 10]. Характерною особливістю даної затримки є необхідність фрагментації вихідного відеопотоку та його інкапсуляції в пакети MIPI у відповідності до специфікації інтерфейсу [1]. При цьому, істотний вплив на величину даної затримки (а відповідно, і сумарної затримки передачі відеопотоку) має специфіка організації роботи системи реєстрації відео з одержаними через CSI-інтерфейс пакетами. У випадках, коли система (або окрема її складова) виконує покадрову буферизацію вхідного відеопотоку, розпочати подальшу обробку стає можливим лише після повного збереження системою останнього отриманого кадру в буфері, відповідно сумарна затримка становитиме принаймні (2):

$$T_{D\Sigma} \Big|_{FB} > T_{Dif} > \frac{1}{N_{FPS}}, \quad (2)$$

де N_{FPS} – кількість кадрів, що надходять за секунду; $|_{FB}$ – нотація виконання умови застосування покадрової буферизації.

Так, наприклад, якщо частота кадрів камери становить 25 кадрів/с, при застосуванні покадрової буферизації затримка передачі відео-

потоків системою становитиме більше 40 мс. Варто зазначити, що подібне значення затримки вважається неприйнятним при вирішенні значної кількості задач, пов'язаних з відеопередачею реального часу (таких, як системи машинного зору, системи відеотелеметрії літальних апаратів, системи автоматизованого контролю дорожньої обстановки тощо). Питання зменшення даної затримки детально розглядається в роботі надалі.

- Затримка попередньої обробки відеопотоку (T_{Dpp}) включає в себе суму затримок кожної з виконуваних операцій обробки відеопотоку (таких, як гамма-корекція, подавлення шумів тощо). Виходячи з тої ж прагматики, що була зазначена вище, необхідною умовою для зменшення даної затримки є забезпечення використання лише тих операцій попередньої обробки, які підтримують обробку відео в потоковому режимі, не потребуючи при цьому повної буферизації кожного відеокадру.

Окремо розглянемо операції компресії. При передачі відеопотоку без стиснення, збільшення роздільної здатності та частоти кадрів відео призводить до збільшення навантаження на тракт передачі. У випадках, коли обмежена смуга пропускання тракту системи відеопередачі реального часу не дозволяє передавати відео з бажаними характеристиками в нестисненому вигляді, для подолання таких обмежень доцільно використовувати алгоритми відеокомпресії. При цьому варто враховувати, що не всі алгоритми підтримують стиснення відеокadrів частками та потребують, як мінімум, наявності одного цілого кадру відео у буфері. Інші алгоритми дозволяють порційну поточкову обробку відеопотоку (наприклад, H.264 в *intra-frame* режимі) [11]. Для їх застосування, як правило, достатньою є наявність буфера в певну кількість рядків (або стовпців) відеокадру.

Варто також враховувати, що використання операцій компресії відеопотоку, в загальному випадку, призводить не лише до збільшення затримок на стороні передавача, а й на стороні кінцевого вузла, оскільки такий відеопотік потребуватиме подальшого декодування.

Іншим аспектом використання компресії відеопотоку є здатність алгоритму коректно відпрацьовувати завади у разі їх виникнення. Такі завади можуть бути викликані пошкодженням пікселів камери, викривленнями через надходження в лінії передачі сильних електромагнітних сигналів ззовні тощо.

Резюмуючи, можна стверджувати, що в міру ряду характерних особливостей, застосування алгоритмів відеокомпресії значно ускладнює

задачу створення систем потокової відеопередачі реального часу та може бути як необхідним, так і неприйнятним в конкретній системі, виходячи з вирішуваних нею задач.

- Затримка передачі відеопотоку кінцевому вузлу (T_{Dtx}) включає в себе затримку на представлення результату попередньої обробки в форматі, придатному для подальшої передачі, а також затримки, пов'язані з роботою використовуваного системою відеопередачі стеку мережевих протоколів та відповідних технологій. В раніш представлених роботах розглядалися ключові аспекти деяких широкозастосовуваних стрімінгових протоколів, що задовольняють вимогам передачі відео в реальному часі – загалом, на основі GigE Vision та RTSP [2–4].

Таким чином, виходячи з вищевказаного огляду затримок систем відеопередачі, які використовують отримання відеопотоку з MIPI CSI камер, можна зробити висновок, що затримки, викликані покадровою буферизацією мають істотний вплив на сумарну затримку передачі та, відповідно, призводять до обмежень можливості передачі відеопотоку в реальному часі. І якщо покадрова буферизація з боку модуля камери може бути усунена шляхом конфігурації існуючої камери або вибором іншого модуля камери, то усунення покадрової буферизації з боку реєструючої відеопотік системи є більш проблематичним, оскільки пов'язане з архітектурою апаратного модуля MIPI самої системи.

В ході підготовчих етапів дослідження, було проаналізовано ряд доступних та комерційно придатних ARM-мікропроцесорів на предмет потенційної доцільності створення системи реєстрації та передачі відеопотоку реального часу з MIPI CSI камер на їх базі. Встановлено, що більшість таких мікропроцесорів (наприклад, сімейства i.MX6 компанії NXP чи сімейства AM57x компанії Texas Instruments), в міру особливостей апаратної реалізації їхніх модулів MIPI CSI або взагалі не підтримують захоплення відеопотоку MIPI-камери без його буферизації в процесі, або не мають належної підтримки такого функціоналу на рівні низькорівневих програмних компонентів системи (відповідні модулі ядра Linux тощо) [12, 13].

З іншого боку, з появою систем на кристалі з вбудованим апаратним ARM-ядром та FPGA-матрицею для втілення довільної програмованої користувацької логіки (таких, як Zynq компанії Xilinx або Cyclone V компанії Intel), стає можливою гнучка реалізація апаратних компонентів систем відеореєстрації відповідно до вирішуваних задач, в тому числі, створення такого технічного рішення захоплення, яке б не потребувало покадрової буферизації і, відповідно, мало б зменшену в порівнянні з буферизуючими системами захоплення затримку реєстрації відеопотоку. Структурно-функціональна

організація та принципи побудови таких систем розглядаються в роботі надалі.

2 Запропоноване рішення

2.1 Структурно-функціональна організація системи

Для вирішення вищезазначених проблем, пов'язаних із затримкою інтерфейсу через застосування покадрової буферизації, запропоновано структурно-функціональну організацію системи реєстрації відеопотоку без покадрового буфера на базі FPGA-SoC. На рис. 1 наведено запропоноване рішення, нижче представлено відповідний опис структурно-функціональної організації системи та її окремих компонентів.

В основі запропонованого рішення лежить застосування MIPI-модуля реєстрації відеопотоку, що підтримує подальшу передачу отримуваних з камери даних через високопродуктивну цифрову поточкову шину (наприклад, AXI4-Stream [14]). На відміну від розглянутих у попередньому розділі існуючих та розповсюджених реалізацій MIPI-модулів, даний модуль не містить в своєму складі вбудованого кадрового буфера та не виконує будь-яких затратних (в сенсі внесеної затримки) операцій над вхідним відеопотоком, натомість, делегуючи всі подальші задачі обробки іншим компонентам системи реєстрації відеопотоку. Ключовою особливістю поточкових цифрових шин є можливість забезпечення безпосередньої передачі байт (або групи байт), без необхідності у таких операціях, як перетворення формату даних, їх попереднє збирання у пакети шини, обов'язкова адресація чи підтримка арбітрації великої кількості паралельних запитів доступу до шини тощо [14, 15]. В контексті запропонованого рішення це дозволяє, по-перше, елімінувати додаткові затримки за рахунок спрощення набору підтримуваних операцій шини, а по-друге, забезпечити можливість коректного відпрацювання більшості з вказаних в попередньому розділі завад передачі відеопотоку камери. За рахунок наявності значної кількості вже готових програмних компонентів, необхідних для створення систем реєстрації/передачі відеопотоку (операційні системи реального часу, стеки мережевих протоколів, бібліотеки обробки відеопотоку та ін.), з врахуванням наявності інтегрованих апаратних процесорних ядер в SoC-FPGA рішеннях таких виробників, як Intel та Xilinx, що мають більшу швидкодію, ніж аналогічні софт-ядра на базі FPGA [16], автори вважають програмну реалізацію відповідних алгоритмів відеообробки та передачі більш простою, виправданою та менш затратною у порівнянні з реалізацією на базі лише цифрової логіки. Представлена на рис. 1 структурно-функціональна організація враховує особливості функціонування вбудованих

мікропроцесорних ядер SoC-FPGA з метою зменшення можливих затримок. Розглянемо дані особливості. Головним чином, мікропроцесорні ядра ARM оптимізовані для обробки даних, що знаходяться в оперативній пам'яті [17]. Разом із використанням вбудованих операційних систем, це дозволяє досягти водночас як достатньо швидкої реакції ядра на зовнішні події (наприклад, переривання щодо надходження нової порції вхідних даних), так і високопродуктивного виконання пріоритетних обчислювальних процедур, так і виконання ряду фонових, менш пріоритетних задач, а також дозволяє зменшити енергоспоживання мікропроцесорного ядра під час очікування. І хоча виробники SoC-FPGA мікросхем надають можливість з'єднання FPGA-логіки з ARM-ядром через спеціалізовані інтерфейси вводу-виводу з низькою затримкою (наприклад, EMIO в Xilinx Zynq-7000 [18]), такий підхід в значній мірі ускладнює задачу синхронізації доступу до даних, вочевидь потребує застосування операційних систем строгого реального часу з заздалегідь визначеним максимальним часом реакції на зовнішні події [19] та накладає обмеження на застосування високопродуктивних наборів інструкцій (типу SIMD) для обробки відеопотоку [17]. Виходячи з наведених обмежень, прийнято рішення про використання принципу порційного доступу до даних відеопотоку мікропроцесорним ядром із застосуванням DMA-контролера (Direct Memory Access), фрагментатора відеопотоку та механізму переривань.

Відповідно до запропонованого рішення (рис. 1), після отримання від MIPI-камери відеопотоку, через цифрову поточкову шину дані надходять до блоку фрагментатора потоку. Фрагментатор збирає складові відеокадру у фрагменти та направляє їх контролеру DMA. Такими складовими можуть бути один чи декілька рядків/стовпців кадру відео або група пікселів одного рядку (стовпця), в залежності від вимог до системи реєстрації відеопотоку та відповідних обмежень максимально допустимої затримки. DMA-контролер, в свою чергу, забезпечує направлення фрагменту до оперативної пам'яті системи та подачу запиту контролеру переривань системи після успішного запису фрагменту у пам'ять. Контролер переривань встановлює переривання, що сигналізує мікропроцесорному ядру про доступність отриманого фрагменту відеопотоку для подальшої обробки.

Таким чином, при реалізації запропонованої структурно-функціональної організації системи, теоретичне значення мінімальної величини затримки інтерфейсу (T_{Dif}), за грубої оцінки, обмежується знизу відповідно до (3):

$$T_{D\Sigma} \Big|_{SF} > T_{Dif} > \frac{1}{N_{SF\Sigma} \cdot N_{FPS}}, \quad (3)$$

де $N_{SF\Sigma}$ – кількість фрагментів, на які поділяється кадр відео; N_{FPS} – кількість кадрів, що надходять

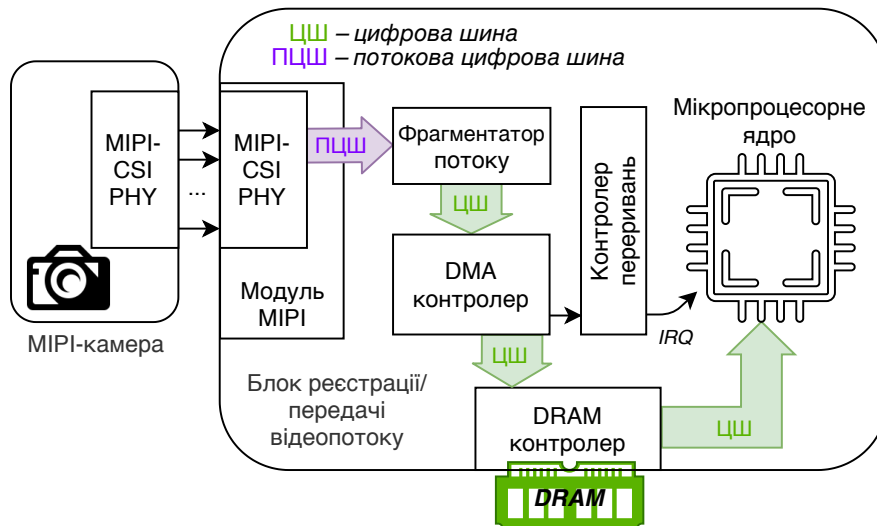


Рис. 1. Структурно-функціональна організація системи

за секунду; $|_{SF}$ – нотація виконання умови застосування запропонованої структурно-функціональної організації (з фрагментацією кадрів відеопотоку).

Розглядаючи наявні засоби для реалізації запропонованого рішення, варто зазначити про потенційну можливість застосування існуючих IP-ядер (за відповідності вимогам до створюваних систем реєстрації відеопотоку). Оскільки IP-ядра є готовими та протестованими компонентами, їх використання дозволяє в істотній мірі спростити задачу створення таких систем.

2.2 Експериментальна перевірка

На основі запропонованої структурно-функціональної організації, було створено експериментальний прототип системи реєстрації/передачі відеопотоку та представлено на конкурсі Digilent Design Contest 2019 [20]. Створений прототип реалізує захоплення відеопотоку з MIPI CSI камери (модуль Digilent Pcam 5C на базі сенсора OV5640 [8, 21]), підключеної до плати Zybo Z7-10 (на базі SoC-FPGA Xilinx XC7Z010 сімейства Zynq-7000 [18, 22]) та подальшу його передачу кінцевому вузлу з застосуванням сумісної зі стандартами реалізації стеку протоколів GigE Vision на основі модифікованої програмної бібліотеки ARAVIS [3, 23].

Для створення компонентів програмованої логіки системи використано IP-ядра Xilinx, синтез проводився в середовищі Vivado Design Suite версії 2018.2 [24]. Загалом, для отримання потоку з камери, використано IP-ядро MIPI CSI-2 Receiver Subsystem [25]; в якості цифрової потокової шини (відповідно до рис. 1) застосовано AXI4-Stream [14]; для реалізації фрагментатора потоку використано набір IP-ядер AXI4-Stream Infrastructure [15].

Для побудови програмної складової системи, в якості операційної системи використано спеціалізований embedded Linux дистрибутив на базі Petalinux версії 2017.4, для збірки якого засто-

совано інструментарій Yocto [26, 27]. До базових мета-шарів (в термінології Yocto), додано шари підтримки Linux-ядром використовуваних системою IP-ядер з автоматично-генерованою конфігурацією BSP (Board Support Package), а також ряд користувачьких шарів, що забезпечують підтримку роботи бібліотеки ARAVIS, загалом, meta-aravis [28]. Для передачі відеопотоку використано представлену в попередніх роботах модифікацію ArvFakeCamera компоненту ARAVIS [3, 4], до якого в подальшому було заплановано внесення змін з метою отримання відеопотоку в режимі DMABUF програмного інтерфейсу V4L2 (Video 4 Linux 2) підсистеми ядра Linux [29]. Даний режим дозволяє отримувати відеопотік в просторі користувача (userspace) напряму з DMA-буферів, що виділяються драйвером відеопристрою, тим самим мінімізуючи затримку доступу до даних.

В ході роботи над прототипом, було виявлено ряд помилок в реалізації програмних компонентів системного рівня підтримки IP-ядер (некоректне рапортування підсистемі V4L2 формату кадру відеопотоку, помилки синхронізації DMA транзакцій, відносно неефективна реалізація роботи з багатопланарними DMA-буферами в драйвері відеопристрою), що в значній мірі обмежило можливість більш оптимізованої реалізації проекту. Загалом, при надходженні потоку з використанням DMA, для його захоплення було використано режим MMIO підсистеми V4L2 замість дещо більш ефективного режиму DMABUF [29]. Задача виправлення виявлених помилок виходить за рамки даної роботи, оскільки, з одного боку, підтримка таких програмних компонентів покладена на компанію-розробника, а з іншого боку, пов'язана з дотриманням встановленого порядку прийняття змін до відповідних підсистем Linux-ядра.

Тим не менш, для створеного прототипу системи реєстрації/передачі відеопотоку, в ході оцінки результатів, при передачі відеопотоку з роздільною здатністю 1920x1080 пікселів в форматі YUV 4:2:2,

25 кадрів/с, досягнуто скорочення затримки отримання відеопотоку в 11.7 мс, в порівнянні прототипу створеної системи з реалізацією, представленою у референс-дизайнах компанії Digilent [22]. Проміжні результати проекту створеної системи розміщено в GitHub-репозиторії [30].

Розглядаючи можливості подальшого скорочення затримок в системах на базі SoC-FPGA Xilinx, варто зазначити про можливість застосування IP-ядер AXI VDMA (AXI Video Direct Memory Access) та VFB R/W (Video Frame Buffer Read/Write) [31, 32]. Ключовою особливістю IP-ядра VFB R/W є забезпечення порівняно швидкого захоплення кадру потоку з AXI4-Stream та подальшої буферизації. У разі, якщо замість цілого кадру, передавати на вхід даного ядра фрагмент кадру, існує потенційна можливість подальшого зменшення затримок за рахунок усунення інших проміжних IP-ядер з шляху передачі даних, кожне з яких вносить свою затримку. Проте, такий підхід вочевидь потребуватиме створення власних специфічних модулів програмованої логіки.

3 Обговорення результатів

Виходячи з концепції запропонованої структурно-функціональної організації системи та отриманих результатів її експериментальної реалізації, можна стверджувати, що задача мінімізації затримок захоплення відеопотоку в створюваних системах реєстрації/передачі відео реального часу потребує комплексного підходу як до архітектури таких систем, так і до вибору апаратної складової та відповідних програмних компонентів користувачького та системного рівнів. При цьому, варто зазначити про можливість створення таких систем у вигляді програмно-апаратних комплексів (наприклад, на основі SoC-FPGA з використанням процесорних ядер) або у вигляді суто апаратних рішень (наприклад, на основі FPGA без використання процесорних ядер). Застосування суто апаратного підходу може дозволити досягти менших величин затримок, проте, є на порядок більш складним у реалізації та подальшій довгостроковій підтримці створюваних систем.

При застосуванні програмно-апаратного підходу до створення систем реєстрації/передачі відеопотоку, важливим фактором є необхідність забезпечення програмною складовою системи обмежень реального часу відповідно до вирішуваної задачі. Для системного рівня програмної складової, виправданим є використання операційних систем реального часу (RTOS — Real-Time Operating System). Варто зазначити, що ядро Linux є ядром операційної системи широкого призначення та не вважається RTOS-ядром, хоча і частково підтримує певний функціонал реального часу. При цьому, еко-система вбудованого Linux має значну кількість

вже готових й протестованих програмних компонентів та інструментів, доступних розробнику системи для перевикористання, що дозволяє в істотній мірі скоротити час, необхідний для вирішення задач розробки. Отже, з одного боку, при створенні систем реєстрації/передачі відеопотоку, реалізація програмної складової системи на основі Linux є досить доцільною, а з іншого боку, призводить до обмежень реального часу. Для подолання таких обмежень, до Linux-ядра може бути застосовано набір патчів реального часу (наприклад, RT-Preempt), що в свою чергу може потребувати внесення змін до драйверів пристроїв для їх коректної роботи. Іншою альтернативою є використання спеціалізованих Linux-сумісних операційних систем реального часу, таких як QNX. QNX запускає ядро Linux як привілейовану задачу власного планувальника реального часу. В порівнянні з вищезазначеними патчами Linux, це дозволяє досягти більш строгих обмежень реального часу. Тим не менш, це потребує реалізації відповідних драйверів, що повинні виконуватися в реальному часі, у вигляді драйверів QNX (з можливістю делегації Linux певних не критичних операцій). Окрім цього, на відміну від Linux, QNX розповсюджується на комерційній основі за пропріетарною ліцензією. Останньою з зазначених альтернатив, є використання окремих RTOS (наприклад, FreeRTOS), що дозволяють досягти строгих обмежень реального часу. Але недоліком такого підходу є більші витрати часу на розробку через відсутність значної кількості вже готових програмних компонентів.

Представлена в роботі структурно-функціональна організація систем реєстрації/передачі відеопотоку призначена для реалізації відповідно до програмно-апаратного підходу, проте, може бути адаптована для суто апаратного підходу у разі необхідності.

Висновки

В роботі розглянуто питання зменшення затримок в системах реєстрації/передачі відеопотоку з MIPI-CSI камер, наведено аналіз окремих складових сумарної затримки реєстрації/передачі відеопотоку, їхніх характерних особливостей, відповідних обмежень та можливостей потенційного впливу на кожну з цих складових з метою зменшення сумарної затримки при створенні таких систем. Головним чином, окреслено проблематику, пов'язану із використанням в системах реєстрації/передачі відеопотоку покадрової буферизації, що призводить до обмеження мінімально досяжної затримки захоплення.

Запропоновано структурно-функціональну організацію системи реєстрації/передачі відеопотоку без покадрової буферизації, що дозволяє досягти зменшеної затримки реєстрації відеопотоку з MIPI CSI камер. Запропоноване рішення може бути реалізовано на базі SoC-FPGA, в тому числі з використа-

нням вже готових IP-ядер. В основі запропонованої структурно-функціональної організації лежить застосування цифрової потокової шини, фрагментатора відеопотоку та модуля DMA. Такий підхід дозволяє реалізувати операції попередньої обробки вхідного відеопотоку програмно, на наявних в SoC-FPGA мікропроцесорних ядрах. Для експериментальної перевірки запропонованого рішення, на його основі було створено прототип системи реєстрації/передачі відеопотоку на базі SoC-FPGA Xilinx XC7Z010 сімейства Zynq-7000. Не зважаючи на виявлені в процесі роботи над прототипом обмеження підтримуваних виробником відповідних IP-модулів Xilinx програмних компонентів системного рівня, в ході експерименту досягнуто зменшення затримки отримання відеопотоку в 11.7мс, що доводить потенційну придатність запропонованих рішень.

Результати роботи можуть бути використані для зменшення сумарної затримки захоплення відеопотоку з MIPI CSI камер в системах реєстрації/передачі відеопотоку, що потребують дотримання обмежень реального часу та допускають реалізацію з використанням SoC-FPGA.

Подяка

Роботу було виконано в рамках держбюджетної теми «Багатоканальний тепловізіо-телевізійний комплекс пошуку-виявлення із заводстийким швидкісним інтерфейсом передачі даних» (д/р №0118U003751) за підтримки Міністерства освіти і науки України.

Перелік посилань

1. MIPI Camera Serial Interface 3 (MIPI CSI-3) / [MIPI Alliance](#)
2. Ходнев Т. А. Оценка эффективности использования тракта связи протоколами RTSP-видеовещания в задачах передачи видеопоследовательностей реального времени / Т. А. Ходнев, А. Ю. Варфоломеев // *XII Міжнародно науково-технічна конференція "Проблеми телекомунікацій" ПТ-2018: Збірник матеріалів конференції*. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 500 с. – с. 332-335.
3. Khodniev T. A., Varfolomeiev A. Y., Lysenko O. M., Antonyuk O. I. (2018) Comparison of RTSP and GigE Vision video streaming technologies in terms of communication path utilization efficiency: an experimental approach. *2018 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*. Одеса, с. 1-4. – DOI:10.1109/UkrMiCo43733.2018.9047531.
4. Марченко В. І. Програмно-апаратна реалізація відеокамери, сумісної зі стандартом GigE Vision / В. І. Марченко, Т. А. Ходнев, А. Ю. Варфоломеев // *Мікросистеми, Електроніка та Акустика*. – 2018. – Т. 23, № 5. – с. 32-37. DOI : 10.20535/2523-4455.2018.23.5.147686.
5. Ходнев Т. А. Поуровнево-декомпозиционная модель оценки интегральной эффективности использования тракта связи с учетом помех / Т. А. Ходнев, А. И. Антонюк, А. Ю. Варфоломеев, А. Н. Лысенко // *Мікросистеми, Електроніка та Акустика*. – 2018. – Т. 23, № 6. – с. 29-33. DOI : 10.20535/2523-4455.2018.23.6.154720.
6. *Rolling Shutter vs. Global Shutter* / Teledyne QImaging
7. Liang C. K. Analysis and compensation of rolling shutter effect / C. K. Liang, L. W. Chang, H. H. Chen // *IEEE Transactions on Image Processing*. – 2008. – Т. 17, № 8. – с. 1323-1330. – DOI : 10.1109/TIP.2008.925384
8. OV5640: color CMOS QSXGA (5 megapixel) image sensor with OmniBSI technology / *OmniVision Technologies Inc.*
9. Low-latency design considerations for video-enabled drones (SPRY301) / *Texas Instruments Inc.*
10. Ahmad J. FPGA based Deterministic Latency Image Acquisition and Processing System for Automated Driving Systems / J. Ahmad, A. Warren // *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. – 2018. DOI : 10.1109/iscas.2018.8351472.
11. Understanding and Reducing Latency in Video Compression Systems / *CAST Inc.*
12. MIPI-CSI2 Peripheral on i.MX6 MPUs (AN5305) / *NXP Semiconductors Inc.*
13. Camera Abstraction Layer - Processor SDK Linux Documentation / *Texas Instruments Inc.*
14. AMBA 4 AXI4-Stream Protocol / *ARM Ltd.*
15. AXI4-Stream Infrastructure IP Suite v3.0 (PG085) / *Xilinx Inc.*
16. Jayakrishnan V. Embedded Processors on FPGA: Soft vs Hard / V. Jayakrishnan, C. Parikh // *Proceedings of the 2019 ASEE North Central Section Conference*. – 2019
17. ARM A-Profile Architecture Specifications / *ARM Ltd.*
18. Zynq-7000 SoC Data Sheet: Overview (DS190) / *Xilinx Inc.*
19. Siewert S. Real-Time Embedded Components And Systems Using Linux And RTOS / Siewert S. and Pratt J. – 2-ге вид. – Dulles, VA : Mercury Learning and Information, 2016. – 500 с. – ISBN: 1942270046
20. Digilent Design Contest 2019 EU Region Finalists / *Digilent Inc.*
21. Pcam 5C: 5 MP Color Camera Sensor / *Digilent Inc.*
22. Zybo Z7: Zynq-7000 ARM/FPGA SoC Development Board / *Digilent Inc.*
23. ARAVIS: A vision library for genicam based cameras / [Github](#)
24. Vivado Design Suite / *Xilinx Inc.*
25. MIPI CSI-2 Receiver Subsystem v4.1 IP (PG232) / *Xilinx Inc.*
26. PetaLinux Tools / *Xilinx Inc.*
27. Yocto Project / [Linux Foundation](#)
28. Meta-aravis: Yocto layer for the Aravis application / [Github](#)
29. V4L2 API Input/Output: Streaming I/O (DMA buffer importing) / *Linux Kernel Organization Inc.*

30. Т. А. Ходнев, М. С. Голуб, О. В. Кужильный. *Gigevision-xilinx: GigE Vision compatible video streaming from MIPI-CSI camera with Zybo Z7-10 board*
31. AXI Video Direct Memory Access v6.3 (PG020) / [Xilinx Inc.](#)
32. Video Frame Buffer Read and Video Frame Buffer Write v2.1 (PG278) / [Xilinx Inc.](#)
- [13] Camera Abstraction Layer - Processor SDK Linux Documentation. [Texas Instruments Inc.](#), developer's guide, viewed 18 Jun 2020.
- [14] AMBA 4 AXI4-Stream Protocol. [ARM Ltd.](#), specification, viewed 18 Jun 2020.
- [15] AXI4-Stream Infrastructure IP Suite v3.0 (PG085). [Xilinx Inc.](#), product guide, viewed 18 Jun 2020.
- [16] Jayakrishnan V. and Parikh C. (2019) Embedded Processors on FPGA: Soft vs Hard. *Proceedings of the 2019 ASEE North Central Section Conference*.

References

- [1] MIPI Alliance, *MIPI Camera Serial Interface 3 (MIPI CSI-3)*, standard, viewed 18 Jun 2020.
- [2] Khodniev T. A. and Varfolomieiev A. Y. (2018) Evaluating the efficiency of communication path utilization by RTSP broadcasting protocols in tasks of real-time video sequences transmission. *Conference materials of XIIth International Scientific Conference "Modern challenges in telecommunications"*, pp. 332-335, in Russian.
- [3] Khodniev T. A., Varfolomieiev A. Y., Lysenko O. M. and Antonyuk O. I. (2018) Comparison of RTSP and GigE Vision video streaming technologies in terms of communication path utilization efficiency: an experimental approach. *2018 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*, pp.1-4. DOI: 10.1109/ukrmi-co43733.2018.9047531.
- [4] Marchenko V. I., Khodniev T. A. and Varfolomieiev A. Y. (2018) Software and Hardware Implementation of Video Camera, Compatible with GigE Vision Standard. *Microsystems, Electronics and Acoustics*, Vol. 23, Iss. 5, pp. 32-37. DOI: 10.20535/2523-4455.2018.23.5.147686.
- [5] Khodniev T. A., Antoniuk O. I., Varfolomieiev A. Y. and Lysenko O. M. (2018) By-Layer Decomposition Model for Evaluating the Integral Communication Path Utilization Efficiency with Account for Errors. *Microsystems, Electronics and Acoustics*, Vol. 23, Iss. 6, pp. 29-33. DOI: 10.20535/2523-4455.2018.23.6.154720.
- [6] Teledyne QImaging, *Rolling Shutter vs. Global Shutter*, technical note, viewed 18 Jun 2020.
- [7] Liang C., Chang L. and Chen H. (2008) Analysis and Compensation of Rolling Shutter Effect. *IEEE Transactions on Image Processing*, Vol. 17, Iss. 8, pp. 1323-1330. DOI: 10.1109/tip.2008.925384.
- [8] OV5640: color CMOS QSXGA (5 megapixel) image sensor with OmniBSI technology. [OmniVision Technologies Inc.](#), datasheet, archived, viewed 18 Jun 2020.
- [9] Low-latency design considerations for video-enabled drones (SPRY301). [Texas Instruments Inc.](#), application note, viewed 18 Jun 2020.
- [10] Ahmad J. and Warren A. (2018) FPGA based Deterministic Latency Image Acquisition and Processing System for Automated Driving Systems. *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. DOI: 10.1109/iscas.2018.8351472
- [11] Understanding and Reducing Latency in Video Compression Systems. [CAST Inc.](#), viewed 18 Jun 2020.
- [12] MIPI-CSI2 Peripheral on i.MX6 MPUs (AN5305). [NXP Semiconductors Inc.](#), application note, viewed 18 Jun 2020.
- [17] ARM A-Profile Architecture Specifications. [ARM Ltd.](#), reference manual, viewed 18 Jun 2020.
- [18] Zynq-7000 SoC Data Sheet: Overview (DS190). [Xilinx Inc.](#), product specification, viewed 18 Jun 2020.
- [19] Siewert S. and Pratt J. (2016) *Real-Time Embedded Components And Systems Using Linux And RTOS*, 2nd ed., Dulles, Virginia: Mercury Learning and Information, ISBN: 1942270046
- [20] Digilent Design Contest 2019 EU Region Finalists. [Digilent Inc.](#)
- [21] Pcam 5C: 5 MP Color Camera Sensor. [Digilent Inc.](#), product page, viewed 18 Jun 2020.
- [22] Zybo Z7: Zynq-7000 ARM/FPGA SoC Development Board. [Digilent Inc.](#), product page, viewed 18 Jun 2020.
- [23] ARAVIS: A vision library for genicam based cameras. [GitHub, software library repository](#), viewed 18 Jun 2020.
- [24] Vivado Design Suite. [Xilinx Inc.](#), product page, viewed 18 Jun 2020.
- [25] MIPI CSI-2 Receiver Subsystem v4.1 IP (PG232). [Xilinx Inc.](#), product guide, viewed 18 Jun 2020.
- [26] PetaLinux Tools. [Xilinx Inc.](#), product page, viewed 18 Jun 2020.
- [27] *Yocto Project*, Linux Foundation project home, viewed 18 Jun 2020.
- [28] Meta-aravis: Yocto layer for the Aravis application. [GitHub, software repository](#), viewed 18 Jun 2020.
- [29] V4L2 API Input/Output: Streaming I/O (DMA buffer importing). [Linux Kernel Organization Inc.](#), Linux kernel v. 4.9 documentation, viewed 18 Jun 2020.
- [30] Khodniev T. A., Holub M. S. and Kuzhylnyi O. V. (2020) Gigevision-xilinx: GigE Vision compatible video streaming from MIPI-CSI camera with Zybo Z7-10 board. [GitHub, project software repository](#), viewed 18 Jun 2020.
- [31] AXI Video Direct Memory Access v6.3 (PG020). [Xilinx Inc.](#), product guide, viewed 18 Jun 2020.
- [32] Video Frame Buffer Read and Video Frame Buffer Write v2.1 (PG278). [Xilinx Inc.](#), product guide, viewed 18 Jun 2020.

Акселерированная регистрация MIPI CSI видеопотока в задачах передачи видео реального времени

*Ходнев Т. А., Голуб М. С., Кужильный О. В.,
Лысенко А. Н., Варфоломеев А. Ю.*

В работе рассмотрены вопросы уменьшения задержек передачи видеопотока в реальном времени с камер, предусматривающих подключение через интерфейс MIPI CSI. Приведены основные составляющие задержки регистрации/передачи видеопотока, проанализирована степень их вклада в суммарную задержку, дана оценка возможности потенциального воздействия на них при разработке систем регистрации/передачи видеопотока реального времени. Обозначена проблематика, связанная с применением покадровой буферизации в таких системах, главным образом, воздействие наличия покадровой буферизации в системе на величину суммарной задержки. Охарактеризованы ограничения реализаций модулей MIPI, приводящие к увеличению задержек регистрации видеопотока с MIPI CSI камер в некоторых ARM-микропроцессорах.

Предложена структурно-функциональная организация систем регистрации MIPI CSI видеопотока с использованием потоковых цифровых шин, фрагментации кадров видеопотока и DMA транзакций, которая не требует использования покадровой буферизации и, соответственно, позволяет уменьшить суммарную задержку регистрации видеопотока. Предложенная структурно-функциональная организация может быть реализована на основе SoC-FPGA решений, в том числе, с использованием существующих IP-ядер. Приведены прагматические особенности и соответствующее оценочное выражение для определения ограничений величины задержки при использовании предложенных решений.

Для экспериментальной проверки, создан прототип системы регистрации/передачи видеопотока на основе SoC-FPGA Xilinx семейства Zynq-7000, в соответствии с предложенной структурно-функциональной организацией, рассмотрена его специфика и соответствующие особенности реализации. Дана оценка полученному быстродействию прототипа и рассмотрены возможные направления дальнейшего уменьшения суммарной задержки регистрации/передачи видеопотока.

Результаты работы могут быть использованы для уменьшения задержек регистрации видеопотока с MIPI CSI камер в системах видеопередачи реального времени на основе SoC-FPGA.

Ключевые слова: SoC; FPGA; MIPI CSI; GigE Vision; видео; потоковая передача

Accelerated MIPI CSI video stream acquisition in tasks of real-time video streaming

*Khodniev T. A., Holub M. S., Kuzhylnyi O. V.,
Lysenko O. M., Varfolomeiev A. Y.*

In present study the challenges of reducing transmission latencies of a real-time video stream acquired from cameras connected via the MIPI CSI interface were addressed. In the study, the main components of the video stream acquisition/transmission latency are given, the degree of their contribution to the total latency was analyzed, the assessment on the potential ability to influence them when developing a real-time video stream acquisition/transmission systems was given. The issues connected with using the frame buffering in such systems are designated, primarily the impact on the total latency value when having a framebuffer in such a system. The limitations of the existing MIPI module implementations of some ARM microprocessors resulting in latency increase for MIPI CSI camera video stream acquisition were characterized.

The structural and functional organization based on the use of digital streaming buses, fragmentation of video frames and DMA transactions for MIPI CSI video stream acquisition systems was proposed, which does not require the use of framebuffers and, as a result, provides the possibility of reducing the overall video stream acquisition latency. The proposed structural and functional organization could be implemented based on SoC-FPGA solutions, including the use of the existing IP-cores. Pragmatic peculiar features were described and the corresponding expression for estimating the limiting value of the latency for the proposed structural and functional organization was given.

For experimental verification, a prototype of the video stream acquisition/transmission system, based on the Zynq-7000 SoC-FPGA family of Xilinx following the proposed structural and functional organization was created. Its specifics and corresponding features of its implementation were discussed in the paper. The performance of the obtained prototype was estimated, and the possible directions towards further reduction of the overall latency of video stream acquisition/transmission were considered.

The results of the study may prove useful to reduce the latencies of the video streams acquired from MIPI CSI cameras in real-time video stream transmission systems based on SoC-FPGA.

Key words: SoC; FPGA; MIPI CSI; GigE Vision; video; streaming