Удосконалення вбудованих систем за допомогою логіки та гнучкого периферійного обладнання

Роберт Перкел (Robert Perkel)
Переклад та редагування: Ірина Приходько, к.т.н., доцент кафедри ПРЕ, РТФ, КПІ ім. Ігоря Сікорського

У цій статті буде розглянуто периферійні пристрої з конфігурованою логікою (*Configurable Logic Cell, CLC*), конфігурованою користувацькою логікою (*Configurable Custom Logic, CCL*), системою обробки подій (*Event System, EVSYS*) та портом маршрутизації сигналів (*Signal Routing, SR*), а також те, як їх можна використовувати для підвищення ефективності вашого проєкту.

Апаратна периферія на кристалі добре відома своєю здатністю знижувати енергоспоживання, підвищувати продуктивність, збільшувати можливості пристрою та зменшувати розмір коду. Існує широкий спектр периферійних пристроїв: від операційних підсилювачів і аналого-цифрових перетворювачів (*Analog to Digital Converter, ADC*) (АЦП) до широтно-імпульсних модуляторів (*Pulse Width Modulators, PWM*) (ШІМ) і універсальних таймерів (*Universal Timer, UTMR*) (УТМР).

Одними з найпотужніших типів цих периферійних пристроїв є ті, які можуть реалізовувати дискретну логіку або можуть з'єднувати інші периферійні пристрої разом.

CLC/CCL

Периферійні пристрої — елементи конфігурованої логіки (CLC) та конфігурованої користувацької логіки (CCL) є програмованими таблицями пошуку (*Look-Up Tables, LUT*), кожна з яких фактично еквівалентна одній комірці ПЛІС (FPGA). Логічна функція, сконфігурована всередині кожного периферійного пристрою, визначається під час виконання програми. CLC/CCL можуть працювати незалежно від центрального процесора, що дозволяє їм замінити дискретні логічні мікросхеми в структурі. Щодо різниці між CLC і CCL, то існують незначні відмінності в реалізації. CLC використовується для мікроконтролерів PIC®, а CCL — для мікроконтролерів AVR®. Основні операції кожного периферійного пристрою залишаються незмінними.

УСУНЕННЯ БРЯЗКОТУ КОНТАКТІВ КНОПОК І ПЕРЕМИКАЧІВ

Одним із найпоширеніших варіантів використання CLC/CCL є реалізація усунення брязкоту на апаратному рівні в поєднанні з таймером/осцилятором. В *Application Note 2805 (AN2805)* представлено три способи реалізації усунення брязкоту за допомогою CLC. З них дві версії CLC (із трьох) забезпечують хороший баланс між використанням апаратних ресурсів і продуктивністю. Вихідні коди всіх трьох версій доступні на *Github*.

Для реалізації функції усунення брязкоту, один із CLC працює як D-тригер для фіксації значення від кнопки чи перемикача. Друга CL C виконує операцію логічного «І» між фіксованим значенням з попереднього кроку та безпосередньо введеним значенням, а потім фіксує результат. Джерело тактового сигналу для обох тригерів є низькочастотним, що створено на основі таймера або генератора на пристрої. Реалізація представлена на рисунку 1.

Для CCL на мікроконтролері AVR схема ще простіша. CCL мають опцію

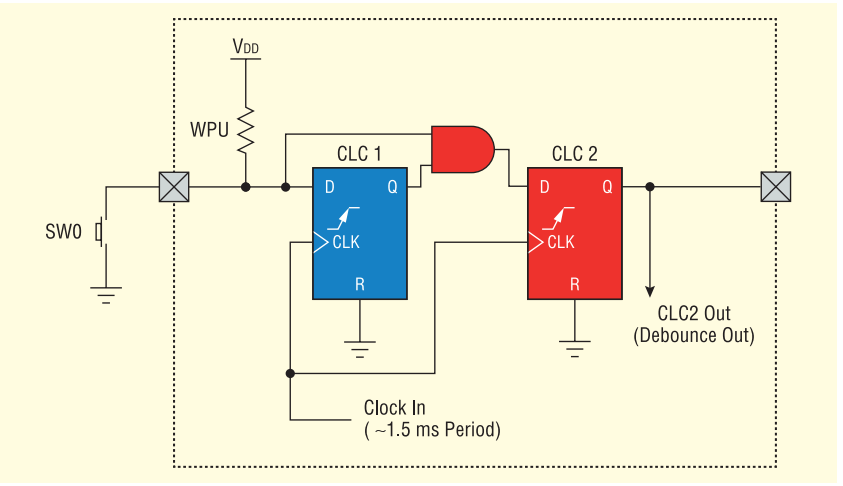

Рис. 1. Усунення брязкоту контактів за допомогою двох CLC

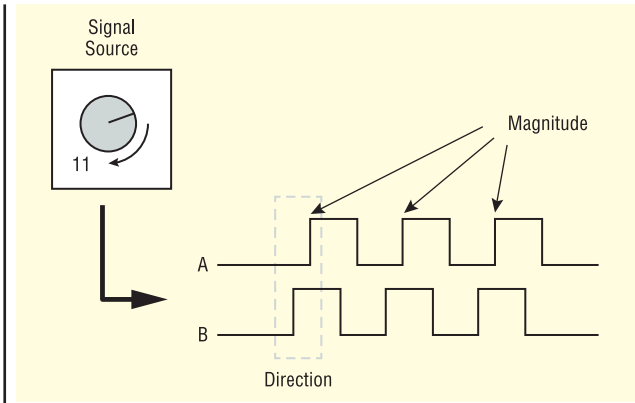

Рис. 2. Приклад сигналу квадратурного декодера

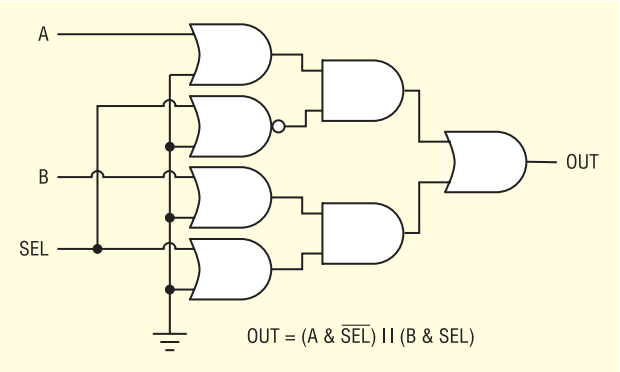

Рис. 3. Мультиплексор 2:1 (реалізовано в логіці CLC; невикористані входи приховано)

фільтрації вхідних сигналів, яка ефективно виконує ту саму двоетапну фільтрацію, що була реалізована на CLC. Крім того, CCL можна тактувати від генератора пристрою з частотою 1 кГц, що є достатньо повільним для усунення брязкоту.

КВАДРАТУРНЕ ДЕКОДУВАННЯ

Інший варіант застосування інтегральних схем CLC — це виконання квадратурного декодування. Інкрементальні квадратурні декодери генерують два прямокутних сигнали, де фаза одного випереджає інший на 90 градусів. Кількість обертів визначається кількістю переходів рівня сигналу (числом імпульсів) обох сигналів, тоді як фаза сигналу вказує на напрямок обертання. На рисунку 2 наведено приклад сигналу квадратурного декодера.

Для декодування цього сигналу CLC перетворюють його на два вихідних сигнали: імпульси обертання за годинниковою стрілкою та проти неї. Два таймери в мікроконтролері підраховують кількість отриманих імпульсів кожного типу. Коли мікроконтролеру потрібно визначити загальну зміну положення, виконується проста математична операція над значеннями лічильників обох таймерів. Це дозволяє визначити абсолютне обертання з моменту останнього зчитування.

СИСТЕМА ОБРОБКИ ПОДІЙ

Система обробки подій (EVSYS) — це функція, що наявна тільки в мікроконтролерах AVR. Вона призначена для вибору вихідного сигналу від одного периферійного пристрою та його маршрутизації до інших периферійних пристроїв вбудованих в мікроконтролер. Ця взаємодія може відбуватися незалежно від центрального процесора (CPU), що дозволяє економити енергію під час режиму сну або простою, та підвищувати продуктивність.

ПОРТ SR

Мікроконтролери PIC мають периферійний пристрій, відомий як порт маршрутизації сигналів (SR). За своєю структурою він схожий на вихідний порт вводу/виводу (I/O), але знаходиться всередині мікроконтролера. Програмне забезпечення може встановлювати або скидати біти всередині нього, як у стандартному регістрі виводу I/O, але він також підтримує сигнали виводу периферійних пристроїв та функціональність регістра зміщення.

Порт маршрутизації сигналу (SR) також добре працює з функцією вибору периферійних виводів (*Peripheral Pin Select, PPS*), яка також присутня в мікроконтролерах PIC. PPS надає розробнику гнучкість у призначенні портів вводу/виводу, дозволяючи переміщувати цифрові сигнали вводу/виводу на різні виводи мікроконтролера. Аналогічно, PPS дозволяє периферійним пристроям окремо обирати «виводи» порту SR як входи. Це дозволяє створювати та керувати складними автоматами станів за допомогою цього периферійного пристрою.

ON-DEVICE SIGNAL SELECT

Порт маршрутизації сигналу (SR) та інтегральні схеми CLC можуть використовуватися для реалізації внутрішнього мультиплексора вибору сигналу. Це корисно для внутрішнього самотестування або для вибору одного з декількох N сигналів для обробки. Для реалізації цього використовується CLC для створення мультиплексора 2:1. Також можливе використання мультиплексора 4:1, але він потребує трьох CLC та двох бітів з порту маршрутизації сигналу (SR). Для керування мультиплексором використовується один біт з порту SR як лінія вибору. Логічна схема реалізації наведена на рисунку 3.

Одна з переваг цієї реалізації за допомогою CLC над PPS полягає в швидкості та гнучкості. PPS можна заблокувати, щоб запобігти випадковим змінам під час виконання програми. Крім того, можна встановити конфігураційний біт, щоб гарантувати, що розблокування PPS можливе лише один раз. На відміну від цього, налаштування мультиплексора CLC дозволяє програмі змінювати вхідні сигнали без необхідності щоразу проходити процедуру розблокування.

Цей підхід використовується в демонстраційній програмі *кода Морзе* (рис. 4) для *сімейства мікроконтролерів PIC18F56Q71 від Microchip*. Ця програма реалізує простий передавач і приймач кода Морзе. Мультиплексор використовується для вибору між вихідним сигналом передавача та зовнішнім вхідним сигналом для прийому.

В програмі для покращення зручності читання коду визначені макроси для встановлення, скидання та перемикання окремих бітів портів маршрутизації сигналу (SR).

```
//Select input to decoder
#define SELECT_TX_DECODE() do { RW0_SetLow(); } while(0)
#define SELECT_USER_DECODE() do { RW0_SetHigh(); } while(0)
#define SWITCH_DECODE_SOURCE() do { RW0_Toggle(); } while(0)
```

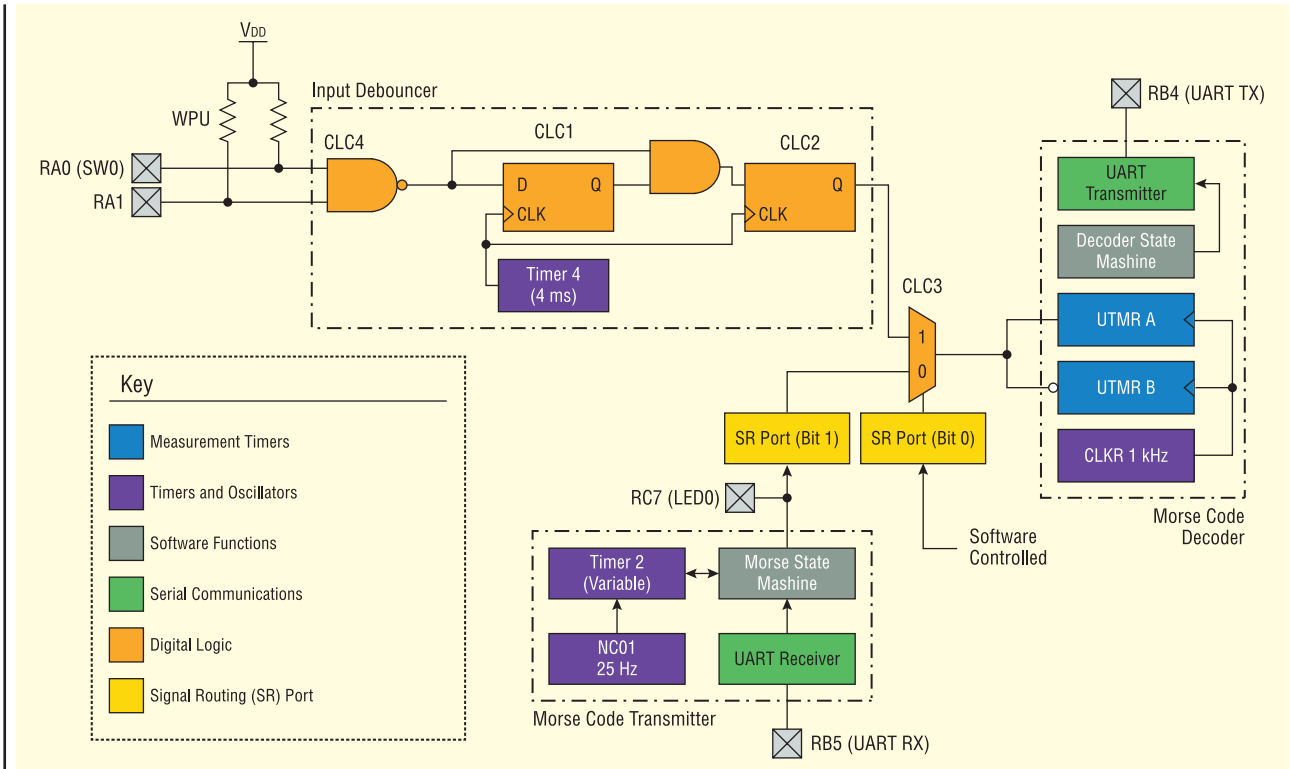

Рис. 4. Структурна схема демонстраційної програми кода Морзе

Наведений фрагмент коду відповідає за перемикання між вхідними джерелами. Якщо передавач і приймач е активні, а користувач надсилає символ '#' в термінал, код перемикає вхідні джерела.

```
if (morseTx_isSwitchRequested() && morseRx_isIdle() &&
morseTx_isIdle())
   {
     //Request to switch input sources
     SWITCH_DECODE_SOURCE();
      morseTx_clearSwitchRequest();
     if (IS_USER_INPUT_ACTIVE())
     {
       //User Input
       printf(«User input is now active.\r\n»);
     }
     else
     {
        //TX Input
       printf(«Transmitter input is now active.\r\n»);
     }
   }
```

Вихідний код цієї програми доступний на *Github*.

АПАРАТНЕ ПРИСКОРЕННЯ ОБЧИСЛЕННЯ ПАРНОСТІ

Іноді під час передачі або обміну даними необхідно генерувати біт парності. Обчислення парності за допомогою програмного забезпечення є досить простою операцією, але виконується повільніше, ніж апаратне рішення. Далі наведено приклад простої функції для обчислення парності.

Примітка: *Для цих простих прикладів тестовий шаблон, що обчислюється, зберігається глобально.*

```
bool isOdd_SW(void)
{
   bool isOdd = false;
   uint8_t temp;
   //Byte Scan
   for (uint8_t byIndex = 0; byIndex < DATA_SCAN_LENGTH;
byIndex++)
   {
     //Bit Scan
      temp = data[byIndex];
     for (uint8_t biIndex = 0; biIndex != 8; biIndex++)
     {
       if (temp & 0b1)
       {
          //Count
         isOdd = !isOdd;
       }
       //Shift bits
       temp >>= 1;
     }
   }
    return (isOdd);
}
```

Для прискорення обчислень периферійний SPI-модуль можна використовувати разом з лічильником циклів (CLC) для створення апаратного калькулятора парності. Апаратне забезпечення SPI містить послідовний регістр зміщення для передачі та прийому даних.

Вихідний сигнал апаратного забезпечення SPI (регістр зміщення) можна подати на CLC для створення калькулятора

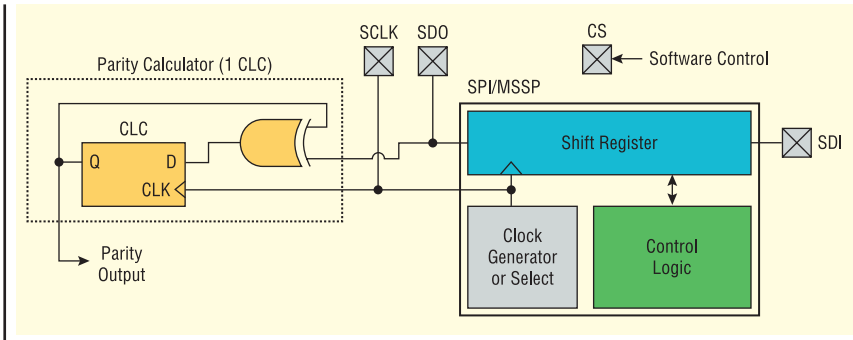

Рис. 5. Апаратна реалізація

Таблиця 1. Відмінності продуктивності між програмною та апаратною реалізацією, послідовність з 10 байт

Рівень оптимізації	Час програмного забезпечення	Апаратний час	Різниця в продуктивності, %
Рівень 0 (відсутня)	1726	552	+313%
Рівень 1	1371	533	+257%
Рівень 2	1371	533	+257%
Рівень 3 (Швидкість)	1019	465	+219%
Рівень S (Розмір)	1019	465	+219%

парності, який працює значно швидше, ніж програмна версія (рис. 5).

Ця реалізація також може використовувати вже наявний периферійний SPI-пристрій, що додатково знижує вартість впровадження. Це можливо тому що зазвичай для зв'язку з іншими SPI-пристроями необхідно активувати лінію вибору кристала (*Chip Select, CS*). Але якщо її не активувати, то зв'язок буде просто ігноровано.

Для реалізації апаратного прискореного обчислення парності, спочатку потрібно захопити поточне значення, зафіксоване в CLC. Потім передайте дані, для яких потрібно обчислити парність, з деактивованою лінією вибору кристала. Після цього слід перевірити нове значення, зафіксоване в CLC.

Якщо нове значення дорівнює попередньому, значить було пораховано парну кількість одиниць.

Якщо значення не співпадають, значить кількість одиниць непарна. Нижче наведено програму для обчислення парності або непарності.

```
bool isOdd_HW(void)
{
    bool initialState = CLC3_
OutputStatusGet();
    SPI1_BufferWrite(&data[0], DATA_
SCAN_LENGTH);
    return (!(CLC3_OutputStatusGet()
== initialState));
}
```

Для демонстрації переваги у швидкості було створено невелику демонстраційну програму на базі *сімейства мікроконтролерів PIC16F18146 від Microchip*. Один із внутрішніх таймерів, Таймер 1, використовується для підрахунку кількості тактових циклів (FOSC/4), необхідних для виконання обчислення парності за допомогою апаратного та програмного забезпечення. Результати виводяться на послідовному терміналі для аналізу. В таблиці 1 наведена копія результатів тестування на різних рівнях оптимізації компілятора.

Продемонстроване покращення продуктивності в цьому прикладі залежить від тактової частоти мікроконтролера та периферійного пристрою SPI. Вихідний код цієї програми доступний на *Github*.

ВИСНОВКИ

Апаратні периферійні пристрої є важливою функцією вбудованих систем. Завдяки їм мікроконтролери можуть стати більш потужними, ефективними та функціональними. Логічні периферійні пристрої та гнучкі периферійні пристрої — це особливо потужні інструменти для виконання простих завдань, таких як усунення брязкоту контактів або квадратурне декодування. Творче використання апаратних периферійних пристроїв покращує дизайн і розширює межі того, що можливо на мікроконтролері.

Співпраця з компанією Microchip Technology дозволяє її клієнтам створювати оптимальні рішення, при цьому зменшують ризики, знижуючи їх загальну вартість та час виведення на ринок. CN

ПОЛЕГШЕННЯ РОЗРОБКИ БОРТОВОГО ЗАРЯДНОГО ПРИСТРОЮ

Ринок акумуляторних електромобілів (Battery Electric Vehicles, BEV) та гібридних електромобілів (Plug-in Hybrid Electric Vehicles, PHEV) продовжує зростати, оскільки поштовх до декарбонізації вимагає стійких рішень для зменшення викидів. Важливим елементом електромобілів є бортовий зарядний пристрій, який перетворює змінний струм у постійний для підзаряджання високовольтної батареї транспортного засобу. Компанія ***Microchip Technology*** *оголосила про випуск бортового зарядного пристрою (БЗП, англ. On-Board Charger, OBC), в якому використовується низка цифрових, аналогових, комунікаційних та силових пристроїв, призначених для автомобільної промисловості, в тому числі контролер цифрових сигналів dsPIC33C, драйвер ізольованого SiC-затвора MCP14C1 та MOSFETs mSiC™ в стандартному корпусі D2PAK-7L XL.*

Це рішення призначене для підвищення ефективності та надійності системи БЗП завдяки розширеним функціям керування dsPIC33 (Digital Signal Controller, DSC), високовольтній посиленій ізоляції драйвера вентилів MCP14C1 з надійною завадостійкістю, а також зменшеним втратам на перемикання та покращеним можливостям терморегуляції mSiC MOSFET. Щоб ще більше спростити ланцюжок постачання для клієнтів, компанія Microchip надає ключові технології, які підтримують інші функції БЗП, в тому числі комунікаційні інтерфейси, безпеку, датчики, пам'ять і синхронізацію.

Для прискорення розробки і тестування систем Microchip пропонує гнучке програмоване рішення з готовими програмними модулями для корекції коефіцієнта потужності (Power Factor Correction, PFC), DC-DC перетворення, зв'язку і діагностичних алгоритмів. Програмні модулі в dsPIC33 DSC розроблені для оптимізації продуктивності, ефективності та надійності, пропонуючи при цьому гнучкість для кастомізації та адаптації до конкретних вимог OEM-виробників.

Як провідний постачальник, компанія Microchip пропонує комплексні рішення для оптимізації процесу розробки, включаючи продукцію, що відповідає вимогам автомобільної промисловості, еталонні зразки, програмне забезпечення та глобальну технічну підтримку.

www.microchip.com