

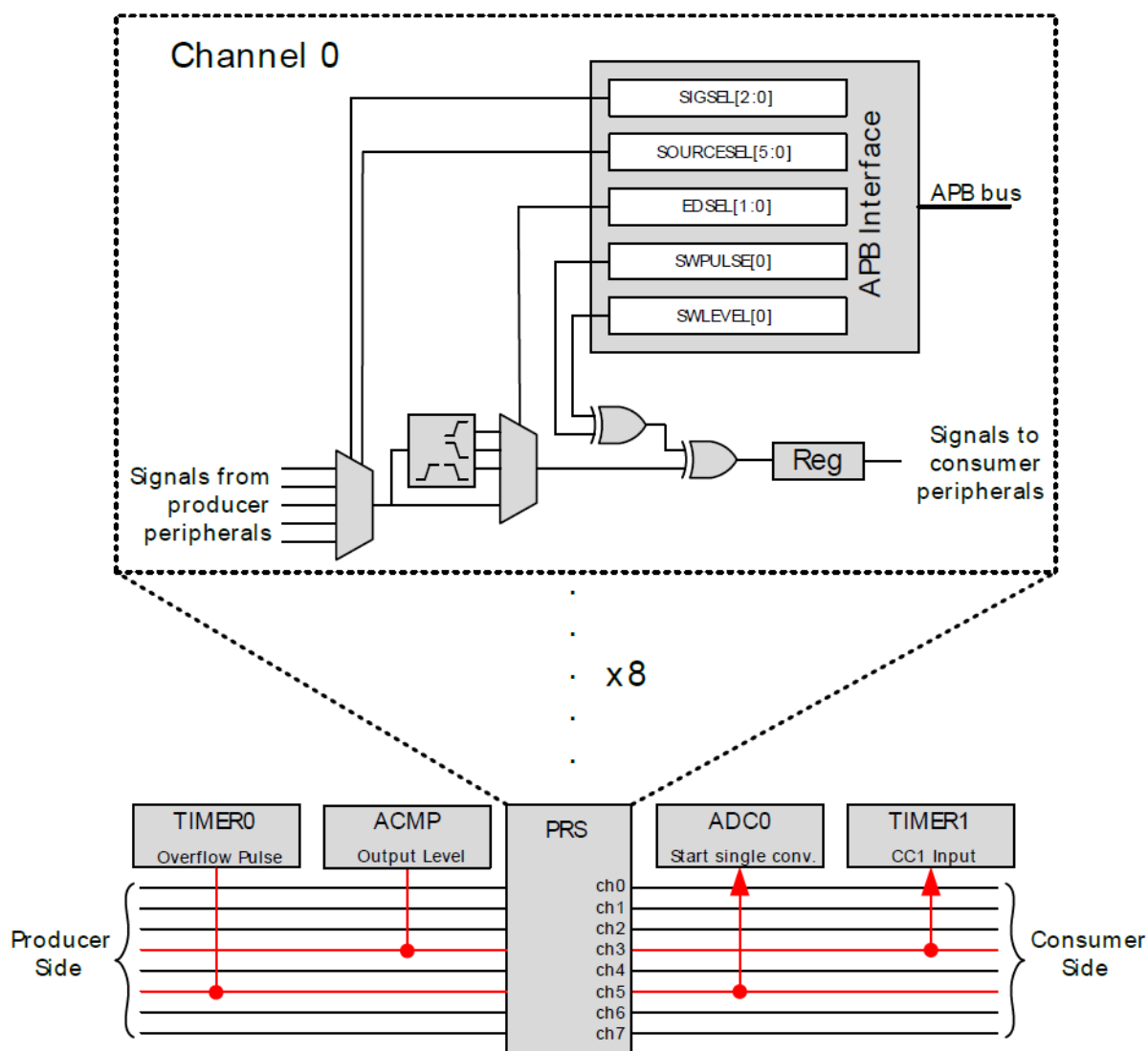
PROJEKTOWANIE ENERGOOSZCZĘDNYCH SYSTEMÓW WBUDOWANYCH

ĆWICZENIE 6

PERIPHERAL REFLEX SYSTEM

1. Peripheral Reflex System (PRS)

Peripheral Reflex System (PRS) jest to moduł sprzętowy zaimplementowany w mikrokontrolerach EFM32 służący do komunikacji pomiędzy różnymi peryferiami bez potrzeby użycia procesora. Pozwala to w pewnych przypadkach na znaczącą redukcję poboru prądu przez mikrokontroler. Schemat blokowy modułu *PRS* został przedstawiony na rysunku 1.



Rys. 1. Schemat blokowy modułu *PRS*

Moduł *PRS* tworzy sieć połączeń złożoną z 8 kanałów (*channels*) do komunikacji między peryferiami (numerowanymi 0 ÷ 7). Każdy kanał *PRS* może zostać osobno skonfigurowany. Peryferia, które mogą brać udział w komunikacji są podzielone ze względu na to jaką pełnią w danej chwili rolę. Mogą zarówno generować sygnały *PRS* – nazywane są wtedy *Producers*; jak i je odbierać – nazywamy je wtedy *Consumers*. Każdy kanał posiada logikę sterowania, dzięki której można zaprogramować wiele wariantów aktywacji peryferii. Oprócz aktywacji sygnałów przez peryferia (sprzętowej), możliwa jest również aktywacja programowa.

2. Opis ćwiczenia

Ćwiczenie ma na celu zaznajomienie z funkcjonalnością modułu *PRS* w mikrokontrolerze *EFM32* oraz możliwościami oszczędzania energii w aplikacji poprzez użycie modułu *PRS* do pewnych zadań (AN0025).

Analizując przykłady komunikacji pomiędzy peryferiami z użyciem modułu *PRS* należy stworzyć sprawozdanie zawierające analizę poboru energii mikrokontrolera EFM32 podczas pracy w każdym ćwiczeniu oraz określić tryby energetyczne w jakim znajduje się mikrokontroler.

3. Cykliczne wyzwalanie przetwornika ADC

W tej części ćwiczenia analizować będziemy wyzwalanie konwersji przetwornika ADC spowodowane przez przepełnienie licznika *TIMER0*. Wyzwalanie będzie obsługiwane przez moduł *PRS*, bez użycia procesora. Po zakończeniu przetwarzania, odpowiednia wartość zostanie pokazana na LCD.

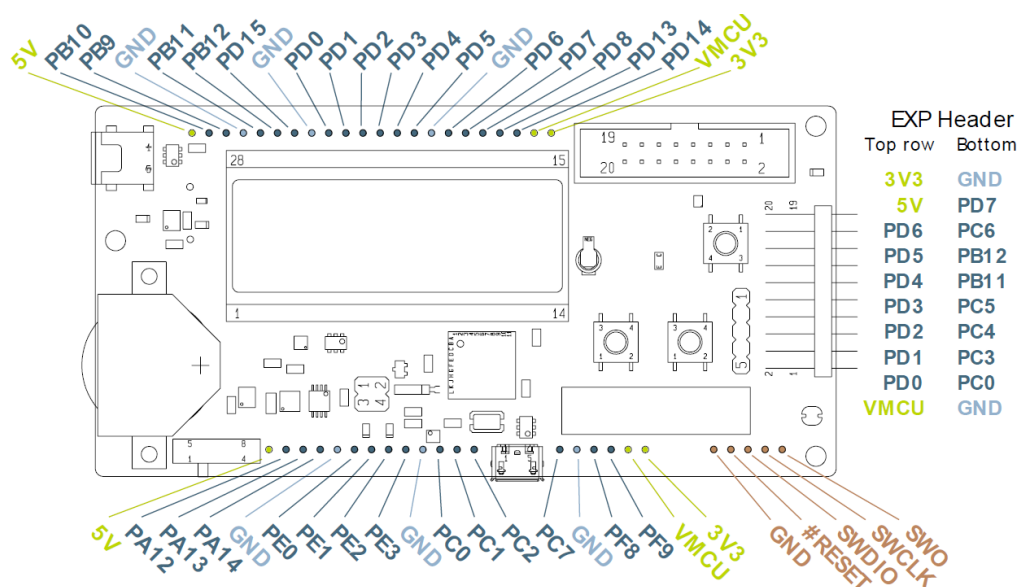
W sprawozdaniu proszę odpowiedzieć na pytania:

- Jaka jest ustawiona rozdzielczość przetwornika i jaki jest okres jego wyzwalania? Jak dokładnie go policzyć?
- W jaki sposób można jeszcze bardziej zoptymalizować energetycznie ten przykład?

4. Pomiar szerokości impulsu

W tej części ćwiczenia użyty zostanie *TIMER* oraz analogowy komparator *ACMP*. Kanał modułu *PRS* skonfigurowany jest w ten sposób, że wyjście komparatora jest sygnałem kanału *PRS*. Narastające zbocze sygnału z komparatora aktywuje zliczenie czasu poprzez moduł *TIMER*, który jest skonfigurowany w trybie *CC*. Opadające zbocze sygnału z komparatora zatrzymuje zliczanie i wartość szerokości impulsu zostaje wyświetlona na LCD.

Sterowanie sygnałem z komparatora odbywa się przez pin *PC4* na płytce *STK* (Rys.2) – podłączając go do zasilania *VMCU* (1) i do *GND* (0).



Rys. 2. Schemat wyprowadzeń płytki STK3700

5. Zadanie projektowe

Proszę dobrać się w zespoły po 2 płytki STK oraz opracować aplikację, która będzie działać następująco.

1 płytka – po naciśnięciu przycisku PB0, poprzez moduł PRS aktywowane zostaje wysłanie znaku (UART), który odpowiada liczbie naciśnień tego przycisku od resetu mikrokontrolera.

2 płytka – po odebraniu przez UART poprawnego bajtu, przez kanał PRS aktywowane zostaje jednorazowe przetwarzanie ADC. Po zakończony przetwarzaniu na wyświetlaczu LCD ma pojawić się wynik przetwarzania w mV oraz znak otrzymany przez UART. W przypadku, gdy znak jest inny niż 0-9, przez UART ma zostać wysłany znak 'X' do płytki 1.

1 płytka – po otrzymaniu przez UART znaku 'X', na wyświetlaczu LCD powinien pojawić się napis „ERROR” na czas 2 sekund.

Proszę zoptymalizować kod pod względem zużycia energii.