

**Akademia Górniczo-Hutnicza  
im. Stanisława Staszica w Krakowie**

---

Wydział Fizyki i Informatyki Stosowanej  
KATEDRA ODDZIAŁYWAŃ I DETEKCJI CZĄSTEK



**PRACA INŻYNIERSKA**

**ŁUKASZ CISOWSKI**

**POZYCJOMETR INERCYJNY**

PROMOTOR:  
dr inż. Krzysztof Świątek

Kraków 2014

## **OŚWIADCZENIE AUTORA PRACY**

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

.....

PODPIS

**AGH**  
**University of Science and Technology in Krakow**

---

Faculty of Physics and Applied Computer Science  
DEPARTMENT OF PARTICLE INTERACTION AND DETECTION TECHNIQUES



**BACHELOR'S THESIS**

**ŁUKASZ CISOWSKI**

**INERTIAL POSITION METER**

SUPERVISOR:  
Dr Eng. Krzysztof Świentek

Krakow 2014

Serdecznie dziękuję mojemu promotorowi za pomoc i nieocenione uwagi, które przyczyniły się do znacznej poprawy merytorycznej mojej pracy.

## Spis treści

<b>1. Wprowadzenie</b>	7
<b>2. Akcelerometry</b>	8
2.1. Przetwornik sejsmiczny	9
2.2. Akcelerometr piezorezystancyjny	13
2.3. Przetworniki piezoelektryczne	15
2.3.1. Zasada działania	15
2.3.2. Własności dynamiczne	17
2.3.3. Akcelerometr piezoelektryczny	20
<b>3. Żyroskopy i ich zastosowanie w elektronice użytkowej</b>	22
3.1. Budowa i zasada działania	22
3.2. Podział żyroskopów	24
3.3. Żyroskopy w elektronice użytkowej	24
3.4. Żyroskopy wieloosiowe	25
3.5. Zastosowania	26
<b>4. Współpraca mikrokontrolerów z urządzeniami zewnętrznymi</b>	28
4.1. Wiadomości ogólne	28
4.1.1. Transmisja szeregową i równoległą	30
4.1.2. Transmisja asynchroniczna i synchroniczna	31
4.1.3. Transmisja równoległą	32
4.2. Interfejs SPI	32
4.2.1. Przesył danych	35
4.2.2. Faza i polaryzacja sygnału zegarowego	36
4.2.3. Współpraca z kilkoma urządzeniami podrzędnymi	36
4.2.4. Zalety i wady interfejsu SPI	37
4.2.5. Zastosowania	38
4.3. Interfejs UART	39
4.3.1. Transmisja	39
4.3.2. Ramka danych	40

4.3.3. Odbiór danych.....	40
4.3.4. Nadawanie danych .....	41
4.3.5. Własności .....	41
4.3.6. Transmisja w trybie synchronicznym .....	41
4.3.7. Stany błędne .....	41
<b>5. Mikrokontrolery ARM Cortex M3 .....</b>	<b>43</b>
5.1. Firma ARM i jej działalność .....	43
5.2. Seria Cortex .....	43
5.3. Rdzeń Cortex-M3.....	44
5.4. Rejestry ogólnego przeznaczenia.....	46
5.5. Przestrzeń adresowa.....	47
5.6. Obszary o dostępie bitowym.....	47
5.7. Kontroler przerwań NVIC .....	48
5.8. Lista instrukcji Thumb-2 .....	50

## **1. Wprowadzenie**

## 2. Akcelerometry

Akcelerometr, zwany także przyspieszeniomierzem, jest czujnikiem służącym do pomiarów przyspieszeń liniowych oraz kątowych. Przyspieszenie liniowe jest wielkością fizyczną, określoną następującym wzorem:

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} \quad (2.1)$$

gdzie:

- $\mathbf{a}$  - wektor przyspieszenia,
- $\mathbf{v}$  - wektor prędkości,
- $t$  - czas.

Rozkładając wektor  $\mathbf{v}$  na składowe  $v_x$ ,  $v_y$ ,  $v_z$  wzdłuż odpowiednio osi  $x$ ,  $y$ ,  $z$  kartezjańskiego układu współrzędnych, wzór 2.1 można zapisać w następujący sposób:

$$\mathbf{a} = \frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k} = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k} \quad (2.2)$$

gdzie:

- $v_x$ ,  $v_y$ ,  $v_z$  - składowe wektora prędkości wzdłuż odpowiednio osi  $x$ ,  $y$  i  $z$ ,
- $a_x$ ,  $a_y$ ,  $a_z$  - składowe wektora przyspieszenia wzdłuż odpowiednio osi  $x$ ,  $y$ ,  $z$ ,
- $\mathbf{i}$ ,  $\mathbf{j}$ ,  $\mathbf{k}$  - wersory osi (odpowiednio)  $x$ ,  $y$ ,  $z$ .

Obecnie produkowane akcelerometry potrafią mierzyć przyspieszenia w jednej (akcelerometry jednoosiowe), dwóch (dwuosiowe) bądź trzech osiach (trójosiowe). W przypadku układów wieloosiowych, często spotyka się rozwiązania, które łączą w sobie kilka przyspieszeniomierzy jednoosiowych, mierzących przyspieszenia wzdłuż trzech prostopadłych osi  $x$ ,  $y$ ,  $z$ .

Akcelerometry znalazły szerokie zastosowanie w technice i są używane jako części składowe wielu urządzeń elektronicznych i mechanicznych. Do przykładów ich użycia można zaliczyć:

- układy stabilizacji optycznej w aparatach fotograficznych,



- zabezpieczenia dysków twardych, chroniące przed skutkami upadków i utratą danych,
- urządzenia mobilne, których aplikacje wykorzystują informacje o nachyleniu i przemieszczeniu urządzenia,
- konsole i kontrolery gier,
- układy wyzwalania poduszek powietrznych w samochodach,
- układy automatyki i robotyki (identyfikatory położenia),
- detektory ruchu w systemach zabezpieczeń,
- protezy kończyn,
- detektory zderzeń.

Istniejące obecnie na rynku konstrukcje akcelerometrów cechują się następującymi parametrami:

- napięcie zasilania (zwykle 3.3 V lub 5 V),
- charakter sygnału wyjściowego (zwykle analogowy, rzadziej cyfrowy),
- pasmo przenoszenia,
- zakres pomiarowy (podawany zwykle w krotnościach przyspieszenia ziemskiego  $g$ ),
- ilość osi pomiarowych,
- interfejs komunikacyjny (najczęściej SPI lub I<sup>2</sup>C).

Istotnym aspektem odróżniającym akcelerometry jest wykorzystywane w nich zjawisko fizyczne, pozwalające na pomiar przyspieszenia. W dalszej części omówiono najczęściej spotykane rodzaje akcelerometrów oraz zasadę ich działania.

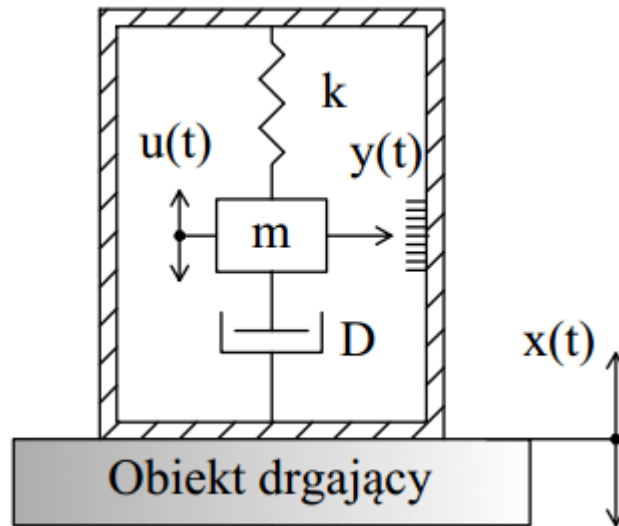
## 2.1. Przetwornik sejsmiczny

Na rysunku 2.1 pokazano model przetwornika sejsmicznego. Jego podstawowymi elementami są:

- masa sejsmiczna  $m$ ,
- stała sprężystości  $k$ ,
- stała tłumienia  $D$ .

Z uwagi na stałość parametrów  $k$ ,  $D$  przyjmujemy założenie co do liniowości przetwornika. Znaczenie wielkości  $x$ ,  $u$ ,  $y$  jest następujące:

- $x(t)$  – bezwzględne przemieszczenie obudowy przetwornika,



Rysunek 2.1: Model przetwornika sejsmicznego (Źródło: [8])

- $y(t)$  – przemieszczenie masy sejsmicznej względem obudowy,
- $u(t)$  – bezwzględne przemieszczenie masy sejsmicznej.

Równanie ruchu masy sejsmicznej ma następującą postać:

$$mu'' = -ky - Dy' \quad (2.3)$$

Ponadto można zauważyć, że:

$$u - x = y \quad (2.4)$$

Po przekształceniach otrzymujemy zależność:

$$my'' + Dy' + ky = -mx'' \quad (2.5)$$

Równanie 2.5 wiąże bezwzględne przemieszczenie obudowy z względnym przemieszczeniem masy sejsmicznej. Po dokonaniu podstawień:

$$\omega_0 = \sqrt{\frac{k}{m}} \quad (2.6)$$

$$\xi = \frac{D}{2\sqrt{km}} \quad (2.7)$$

omawiane równanie różniczkowe może być zapisane jako:

$$\frac{1}{\omega_0^2} y'' + \frac{2\xi}{\omega_0} y' + y = -\frac{1}{\omega_0^2} x'' \quad (2.8)$$

W celu wyznaczenia poprawnych warunków pracy przetwornika, należy posłużyć się analizą częstotliwościową. W tym celu równanie 2.8 poddajemy obustronnie transformacji Laplace'a (zakładamy zerowe warunki początkowe):

$$\frac{1}{\omega_0^2} s^2 Y + \frac{2\xi}{\omega_0} s Y + Y = -\frac{1}{\omega_0^2} s^2 X \quad (2.9)$$

gdzie  $X$  i  $Y$  są transformatami sygnałów odpowiednio  $x$  i  $y$ :

$$X(s) = \mathcal{L}\{x(t)\} \quad (2.10)$$

$$Y(s) = \mathcal{L}\{y(t)\} \quad (2.11)$$

gdzie  $\mathcal{L}\{f(t)\}$  oznacza transformację Laplace'a funkcji  $f$ , określoną wzorem:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{+\infty} e^{-st} f(t) dt \quad (2.12)$$

Zmienna  $t$  jest zmienną, po której całkujemy i najczęściej oznacza czas, zaś  $s$  jest zmienną zespoloną, będącą zmienną niezależną transformaty. Sygnały poddawane przekształceniu Laplace'a przyjęło się nazywać w literaturze oryginałami.

Zgodnie z literaturą [8], opisywane przetwornik sejsmiczny może służyć do pomiaru przemieszczenia lub przyspieszenia. Jeżeli za jego transmitancję operatorową przyjmiemy iloraz transformaty względnego przemieszczenia  $Y$  oraz transformaty bezwzględnego przyspieszenia obudowy  $s^2 X$ , wówczas taki przetwornik nazywamy akcelerometrem. Omawiana transmitancja zgodnie ze wzorem 2.9 ma postać:

$$G(s) = -\frac{\frac{1}{\omega_0^2}}{\frac{1}{\omega_0^2} s^2 + \frac{2\xi}{\omega_0} s + 1} \quad (2.13)$$

W celu zbadania własności częstotliwościowych przetwornika należy posłużyć się transmitancją widmową. W tym celu do zależności 2.13 podstawiamy  $s = j\omega$ , gdzie  $j$  jest jednostką urojoną a  $\omega$  pulsacją. Otrzymana transmitancja ma postać:

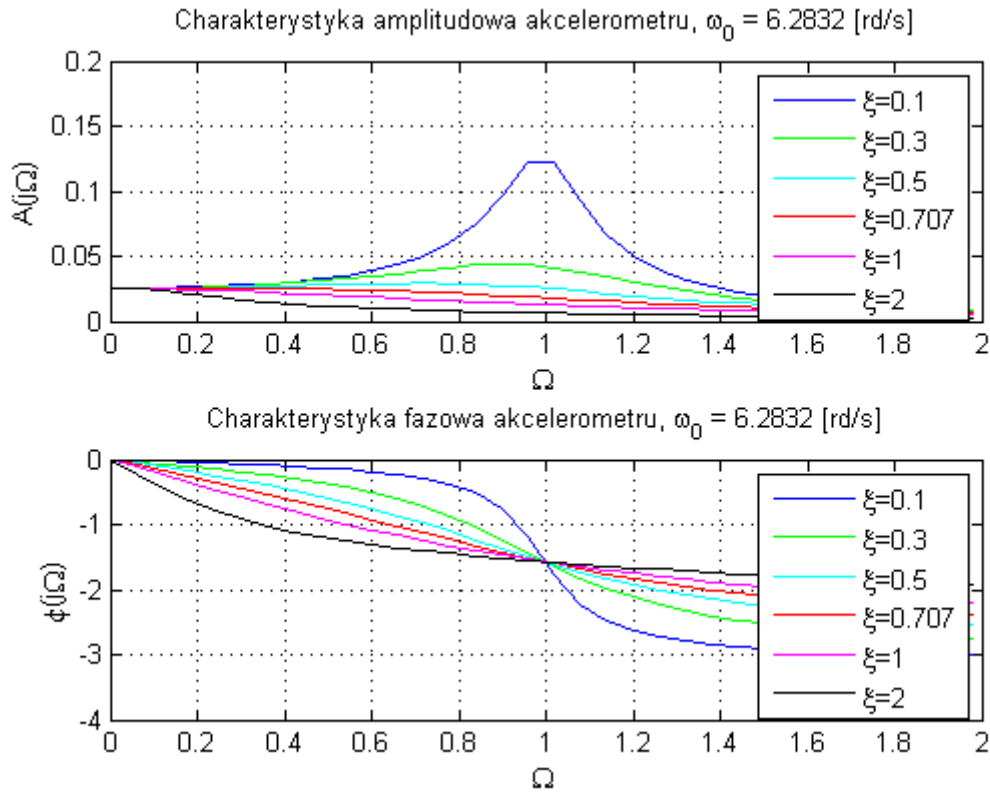
$$G(j\omega) = -\frac{\frac{1}{\omega_0^2}}{-\left(\frac{\omega}{\omega_0}\right)^2 + j2\xi\frac{\omega}{\omega_0} + 1} \quad (2.14)$$

Wygodniej posłużyć się transmitancją widmową w funkcji pulsacji względnej:

$$G_r(j\Omega) = -\frac{\frac{1}{\omega_0^2}}{-\Omega^2 + j2\xi\Omega + 1} \quad (2.15)$$

gdzie  $\Omega = \frac{\omega}{\omega_0}$ .

Na podstawie zależności 2.15 można wyznaczyć charakterystyki amplitudowe  $A$  i fazowe  $\phi$  akcelerometru [8]:



Rysunek 2.2: Charakterystyki częstotliwościowe akcelerometru (Źródło: [8])

$$A(j\Omega) = |G_r(j\Omega)| = \frac{1}{\omega_0^2} \frac{1}{\sqrt{(1 - \Omega^2)^2 + 4\xi^2\Omega^2}} \quad (2.16)$$

$$\phi(j\Omega) = \arg\{G_r(j\Omega)\} = -\arctg\left(\frac{2\xi\Omega}{1 - \Omega^2}\right) \quad (2.17)$$

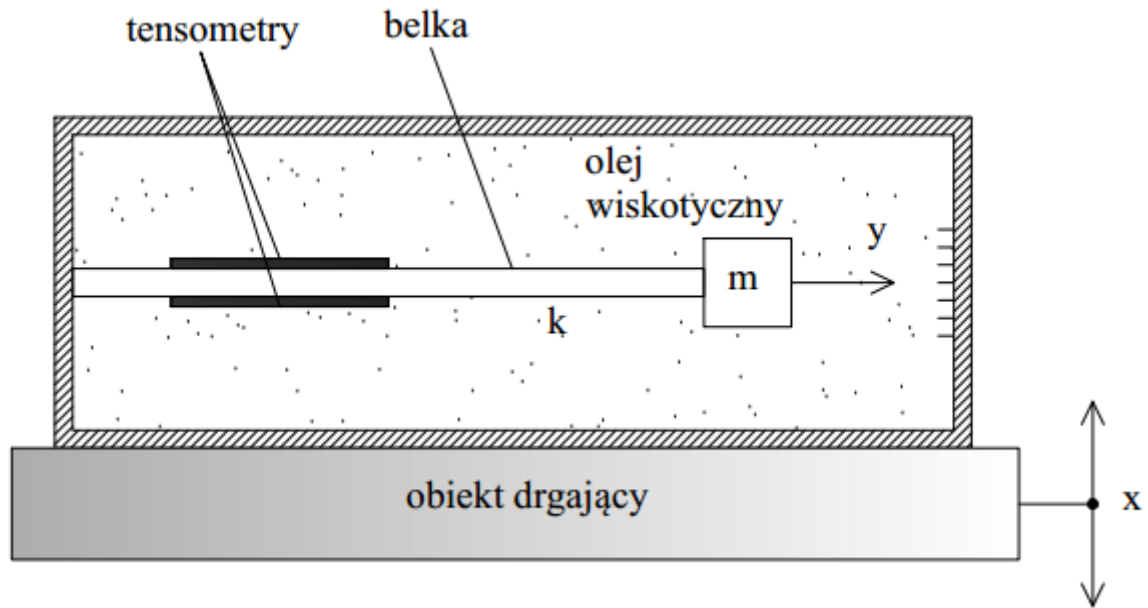
Na rysunku 2.2 przedstawiono przebiegi odpowiadające wzorom 2.16 oraz 2.17. Można zauważyć, że dla dostatecznie małych tłumień  $\xi$  charakterystyki amplitudowe omawianego akcelerometru posiadają wyraźne maksimum, dla częstotliwości zbliżonych do częstości drgań własnych. W przypadku granicznym, gdy tłumienie wynosi zero, amplituda wzmocnienia dąży do nieskończoności. Omawiane zjawisko nosi nazwę rezonansu. Aby rezonans mógł wystąpić, musi być spełniony warunek [8]:

$$\xi < \frac{1}{\sqrt{2}} \approx 0.707 \quad (2.18)$$

Maksimum funkcji modułu transmitancji widmowej jest osiągane dla pulsacji względnej:

$$\Omega_r = \sqrt{1 - 2\xi^2}$$

Z literatury [8] wiadomo, że warunkiem niezniekształcającego przetwarzania sygnału jest stałość charakterystyki amplitudowej i liniowość charakterystyki fazowej. Biorąc pod uwagę rysunek 2.2 można dojść do wniosku, że warunki te spełnione są, gdy:



Rysunek 2.3: Akcelerometr z piezorezystancyjnym czujnikiem tensometrycznym (Źródło: [8])

$$\omega < \omega_g \approx 0.5\omega_0 \quad (2.19)$$

dla tłumienia ograniczonego warunkiem:

$$\xi \in (0.6, 0.707) \quad (2.20)$$

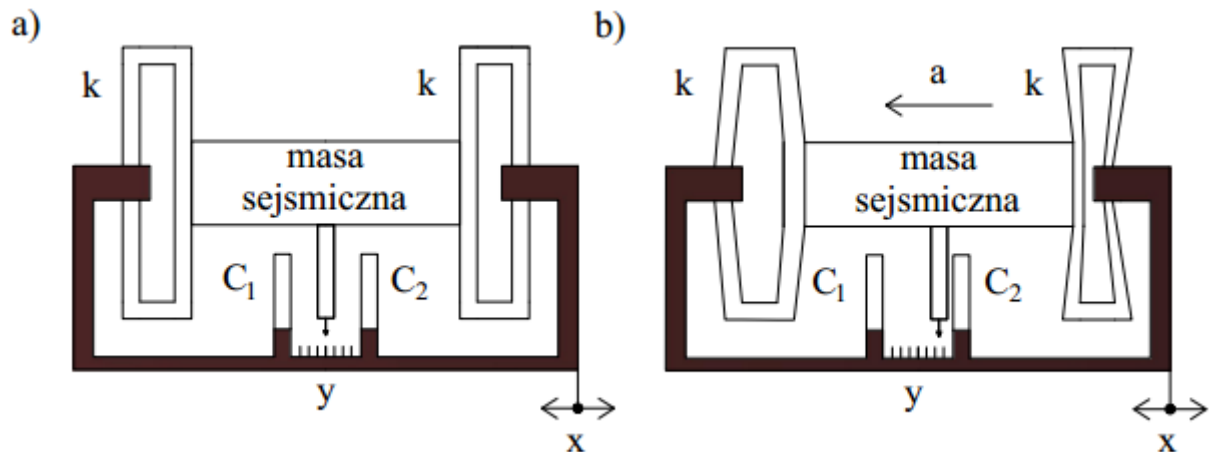
W przypadku pulsacji bliskich  $\omega_g$  należy liczyć się z większym błędem przetwarzania wynikającym ze spadku wzmocnienia i z błędów fazowego [8]. Z przebiegu charakterystyk częstotliwościowych wynika, że pasmo przenoszenia będzie tym szersze im większa będzie pulsacja drgań własnych. Można to uzyskać zgodnie z zależnością 2.6 poprzez zwiększenie stałej  $k$  i zmniejszenie masy sejsmicznej  $m$ . Wzrost sprężystości prowadzi również do spadku tłumienia (zależność 2.7). Pozwala to na uproszczenie równania (wzór) do postaci [8]:

$$y = -\frac{1}{\omega_0^2} x'' = -\frac{1}{\omega_0^2} a \quad (2.21)$$

z której wynika, że względne wychylenie masy przetwornika jest proporcjonalne do przyspieszenia bezwzględnego. Współczynnik proporcjonalności łączący te dwie wielkości jest jednak odwrotnie proporcjonalny do kwadratu pulsacji drgań własnych. Wynika z tego, że żądania uzyskania szerokiego pasma przepustowego i dużej czułości przetwarzania są sprzeczne.

## 2.2. Akcelerometr piezorezystancyjny

Rysunek 2.3 przedstawia schemat akcelerometru z piezorezystancyjnym czujnikiem tensometrycznym. Zasadniczym elementem przetwornika jest masa sejsmiczna znajdująca się na końcu sprężystej



Rysunek 2.4: Zasada działania akcelerometru pojemnościowego: a) układ znajdujący się w spoczynku, b) układ poddawany działaniu przyspieszenia  $a$  (Źródło: [8])

belki, zamocowanej jednostronnie do obudowy. Drgania masy na końcu belki zachodzą wewnątrz obudowy wypełnionej olejem wiskotycznym.

Mierzone przyspieszenie bezwzględne jest przekształcane kolejno na:

- względne przemieszczenie  $y$  masy  $m$  względem obudowy,
- odkształcenie  $\varepsilon$  względne belki,
- względną zmianę rezystancji  $\varepsilon_R$  czujnika tensometrycznego,
- względną zmianę napięcia nierównowagi  $\frac{\Delta U}{U_Z}$  mostka ( $\Delta U$  oznacza przyrost napięcia nierównowagi mostka, zaś  $U_Z$  napięcie zasilające mostek).

Ponieważ częstotliwość drgań własnych tego typu przetworników są stosunkowo niskie (od kilkuset herców do kilku kiloherców), niewielkie jest również ich pasmo przenoszenia. Do zalet tego typu konstrukcji można zaliczyć możliwość pomiaru stałych przyspieszeń [8]. Czułość przetwornika można wyznaczyć z następującej zależności [8]:

$$S_a = \frac{\frac{\Delta U_{-g}^g}{U_Z}}{2g} \quad (2.22)$$

gdzie  $\Delta U_{-g}^g$  oznacza przyrost napięcia nierównowagi mostka przy zmianie pomiaru przyspieszenia z  $g$  na  $-g$ .

Układy podobne do powyższego są wytwarzane w technice mikromaszynowej MEMS, pozwalającej na znaczne zmniejszenie rozmiaru przetwornika. W akcelerometrach takich do przetwarzania zmiany odkształcenia na zmianę rezystancji wykorzystuje się tensometry piezorezystywne.

Wspomniana technika MEMS pozwala również na produkcję przyspieszeniomierzy pojemnościowych, stosowanych bardzo często jako elementy układów bezpieczeństwa czynnego w samochodach (wyzwalacze poduszek powietrznych). Przykład konstrukcji takiego akcelerometru pokazano na rysunku 2.4.

Podstawowym elementem przetwornika jest płyta, stanowiąca masę sejsmiczną, której końce zamocowane są poprzez sprężyste uchwyty do obudowy. Z płytą połączona jest elektroda, znajdująca się pomiędzy dwiema innymi, ale nieruchomymi (względem obudowy) elektrodami, połączonymi z obudową. Całość tworzy różnicowy układ dwóch kondensatorów  $C_1$  i  $C_2$ , których pojemności zależą od położenia środkowej okładki, które z kolei zależą od wychylenia masy sejsmicznej. Do okładek nieruchomych przykłada się prostokątne napięcia znajdujące się w przeciwfazie.

Jeżeli układ nie jest poddawany działaniu przyspieszenia, pojemności  $C_1$  i  $C_2$  są równe, a napięciowy sygnał wyjściowy ze środkowej elektrody ma wartość zerową. Działanie przyspieszenia powoduje wychylenie masy sejsmicznej oraz środkowej okładki, co prowadzi do zmniejszenia jednej z pojemności i zwiększenia drugiej. Niezerowy już sygnał z ruchomej elektrody jest demodulowany i przetwarzany przez wzmacniacz buforowy [8]. Cały układ pomiarowy posiada pętlę ujemnego sprzężenia zwrotnego, utrzymującego środkową elektrodę w położeniu równowagi. Miarą przyspieszenia jest wartość napięcia wyjściowego akcelerometru, niezbędna do utrzymania środkowej okładki w położeniu równowagi [8].

## 2.3. Przetworniki piezoelektryczne

### 2.3.1. Zasada działania

Działanie tego typu przetworników polega na wykorzystaniu zjawiska piezoelektrycznego. Polega ono na pojawianiu się ładunku elektrycznego na powierzchni ładunku pod wpływem istniejących naprężeń mechanicznych. Kryształy, które cechują się omawianymi własnościami nazywane są piezoelektrykami. W naturze istnieje również zjawisko odwrotne (nazywane zjawiskiem piezoelektrycznym odwrotnym, w odróżnieniu od prostego), polegające na zmianie wymiarów kryształu pod wpływem przyłożonego pola elektrycznego.

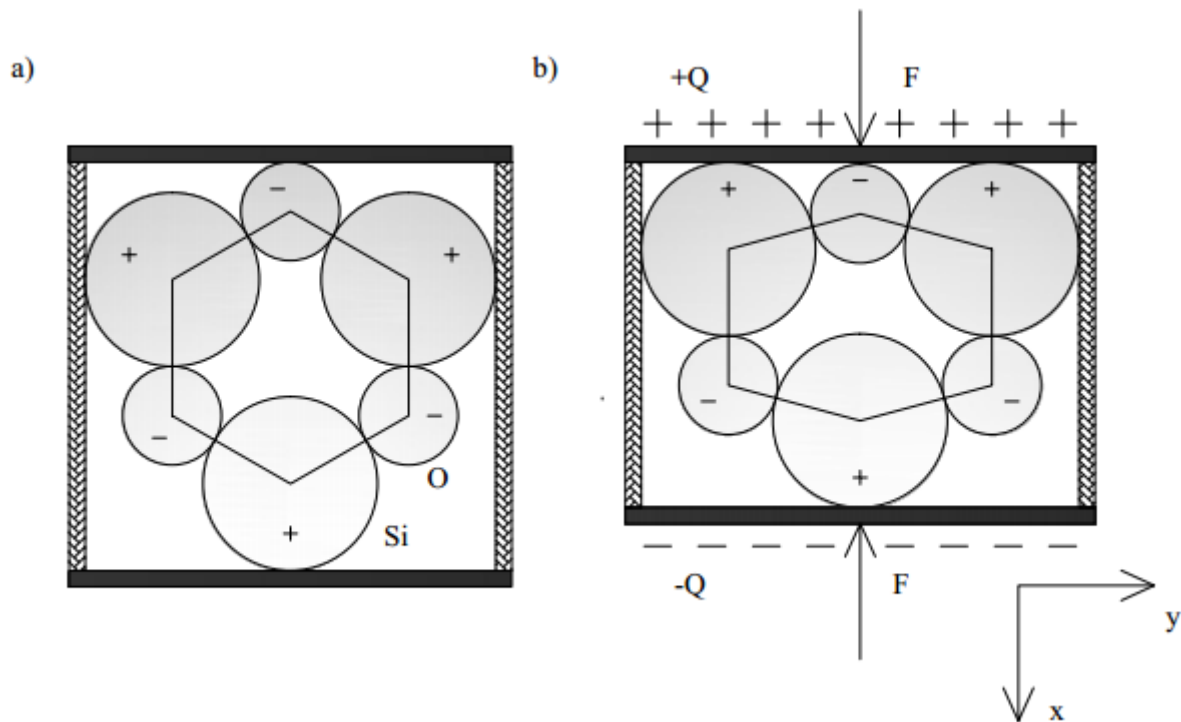
W praktyce jako piezoelektryka najczęściej używa się krzemu ( $\text{SiO}_2$ ). Do jego zalet można zaliczyć stosunkowo dużą wytrzymałość mechaniczną, znaczną rezystywność i stałą piezoelektryczną i niewielką zależność zjawiska piezoelektrycznego od temperatury.

Na rysunku przedstawiono schematycznie budowę kryształu kwarcu. Jak widać jego komórka elementarna ma postać układu heksagonalnego. Przyłożenie do płytki kwarcowej siły powoduje przemieszczenie się jonów krzemu i tlenu, co skutkuje pojawieniem się ładunków na jej powierzchni oraz pola elektrycznego w jej wnętrzu. Kierunek pola zależy od kierunku działania siły oraz od sposobu wycięcia płytki względem osi krystalograficznych kwarcu, zaś jego wartość od wartości działającej siły.

Do osi krystalicznych kwarcu można zaliczyć [8]:

- trzy osie elektryczne, łączące przeciwległe wierzchołki sześciokąta,
- trzy osie mechaniczne, łączące środki przeciwległych boków sześciokąta,
- oś optyczną, prostopadłą do płaszczyzny sześciokąta.

Jeśli siła  $F$  działa wzdłuż osi  $x$ , to ładunki generowane są na powierzchniach do niej prostopadłych. Odkształcanie kryształu wzdłuż osi  $y$  powoduje powstawanie ładunków również na tych samych powierzch-



Rysunek 2.5: Ilustracja efektu piezoelektrycznego wzdłużnego na przykładzie kryształu kwarcu: a) kryształ nieobciążony, b) kryształ poddany działaniu siły  $F$  (Źródło: [8])

niach. W pierwszym przypadku mamy do czynienia ze zjawiskiem piezoelektrycznym wzdłużnym, zaś w drugim z poprzecznym.

W przypadku zjawiska wzdłużnego, gęstość  $q$  ładunku wygenerowanego na powierzchni  $A_x$  prostopadłej do osi elektrycznej  $x$ , wzdłuż której działa siła  $F$  wynosi [8]:

$$q = k_p \sigma \quad (2.23)$$

gdzie:

- $k_p$  – stała (tzw. moduł piezoelektryczny),
- $\sigma$  - naprężenie powstałe w płytce, w wyniku działania siły  $F$ .

Naprężenie występujące w ostatnim wzorze może być wyznaczone w następujący sposób:

$$\sigma = \frac{F}{A_x} \quad (2.24)$$

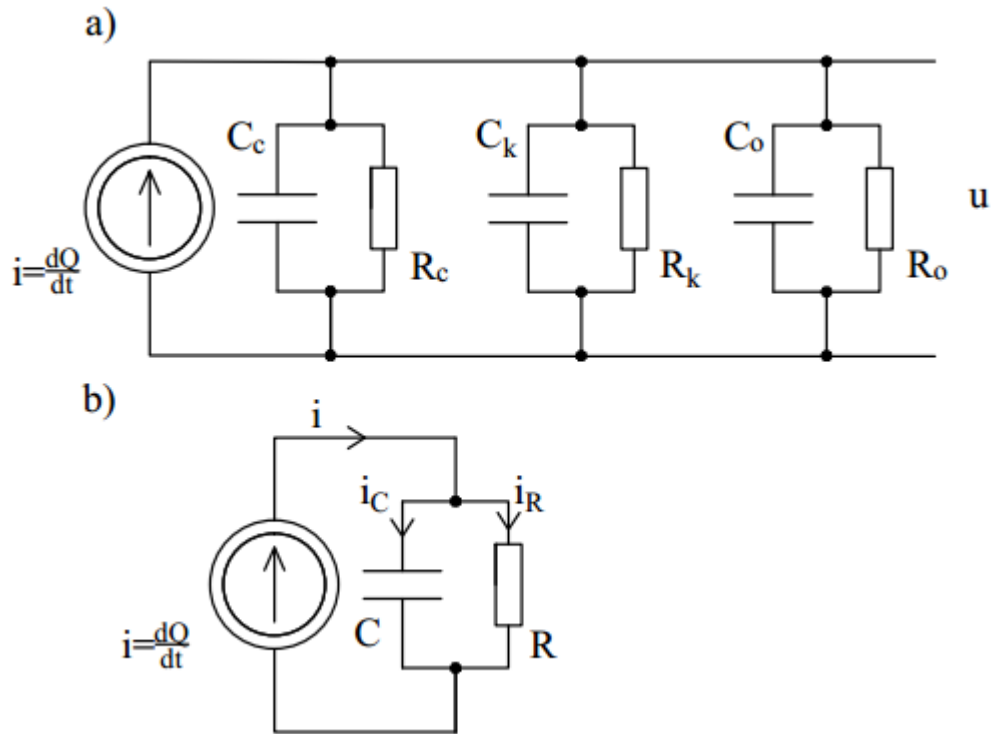
Wartość ładunku  $Q$  zgromadzonego na powierzchni krzemu wyniesie zatem:

$$Q = q A_x = k_p F \quad (2.25)$$

a napięcie  $U$  między powierzchniami płytki będzie miało wartość:

$$U = \frac{Q}{C} \quad (2.26)$$





Rysunek 2.6: Własności toru pomiarowego z przetwornikiem piezoelektrycznym w zakresie niskich częstotliwości: a) model toru, b) schemat zastępczy (Źródło: [8])

gdzie  $C$  jest sumą pojemności płytki i obwodu pomiarowego [8]. Czułość ładunkowa wynosi:

$$S_Q = \frac{dQ}{dF} = k_p \quad (2.27)$$

zaś napięciowa:

$$S_U = \frac{dU}{dF} = \frac{1}{C} S_Q = \frac{k_p}{C} \quad (2.28)$$

Jak widać z powyższych wzorów, czułość ładunkowa piezoelektryka jest wielkością stałą, zaś czułość napięciowa zależy od pojemności przewodów i wzmacniacza pomiarowego.

### 2.3.2. Własności dynamiczne

Płytkę piezoelektryka posiada częstotliwość rezonansową drgań zależną od jej masy  $m$ , i sprężystości  $k$ . Jeśli uwzględnimy, że płytkę taką jest obiektem o parametrach rozłożonych, a nie skupionych (jak w omawianym poprzednio modelu), to należy pamiętać, że będzie ona mogła posiadać kilka częstotliwości rezonansowych. Najniższa z nich ogranicza pasmo przetwarzania przetwornika od góry. W celu określenia dolnego ograniczenia, można posłużyć się modelem toru pomiarowego ukazanego na rysunku 2.6, wykorzystującego przetwornik piezoelektryczny. Tor składa się z następujących elementów:

- płytki piezoelektrycznej o rezystancji  $R_c$  i pojemności  $C_c$ ,
- przewodów łączeniowych o rezystancji  $R_k$  i pojemności  $C_k$ ,

- wzmacniacza pomiarowego o rezystancji wejściowej  $R_o$  i pojemności wejściowej  $C_o$ ,
- przetwornika piezoelektrycznego pełniącego rolę generatora ładunku.

Dla obwodów z rysunku 2.6 można napisać następujące zależności:

$$C = C_c + C_k + C_o \quad (2.29)$$

$$\frac{1}{R} = \frac{1}{R_c} + \frac{1}{R_k} + \frac{1}{R_o} \quad (2.30)$$

$$i = i_C + i_R \quad (2.31)$$

$$\frac{dQ}{dt} = C \frac{du}{dt} + \frac{u}{R} \quad (2.32)$$

Ostatnie równanie można zapisać w postaci operatorowej (zakładamy zerowe warunki początkowe):

$$sQ(s) = sCU(s) + \frac{1}{R}U(s) \quad (2.33)$$

Traktując  $Q(t)$  jako sygnał wejściowy, zaś napięcie  $u(t)$  jako sygnał wyjściowy, możemy wyznaczyć transmitancję  $K_p(s)$  operatorową toru:

$$K_p(s) = \frac{Rs}{RCs + 1} \quad (2.34)$$

Transmitancja widmowa ma postać:

$$K_p(j\omega) = \frac{jR\omega}{jRC\omega + 1} \quad (2.35)$$

Na tej podstawie wyznaczamy charakterystykę amplitudową:

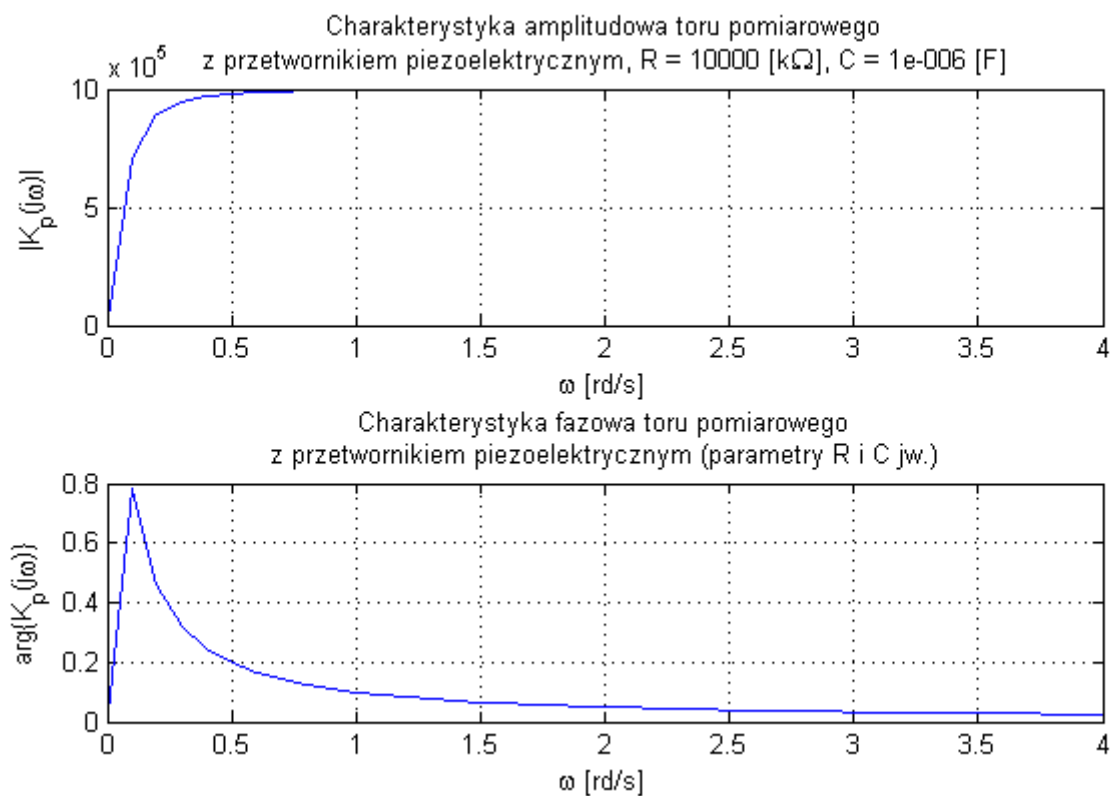
$$|K_p(j\omega)| = \frac{R\omega}{\sqrt{1 + (RC\omega)^2}} \quad (2.36)$$

oraz fazową:

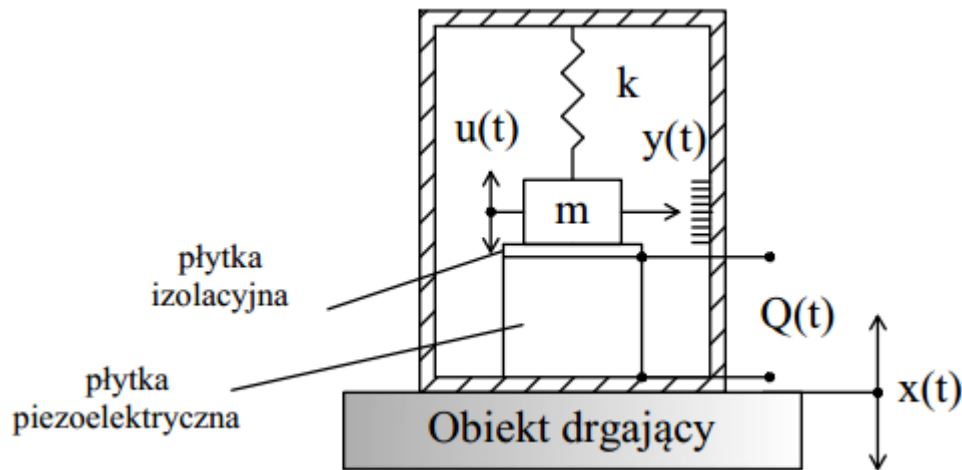
$$\phi(\omega) = \arg\{K_p(j\omega)\} = \arg\left\{\frac{jR\omega}{jRC\omega + 1}\right\} \quad (2.37)$$

Obie charakterystyki pokazano na rysunku 2.7. Jak widać układ nie przenosi niskich częstotliwości. W szczególności nie jest możliwy pomiar sygnałów stałych. Wynika to z tego, że gromadzony na powierzchni piezoelektryka ładunek, rozładowuje się przez skończoną rezystancję toru pomiarowego. Dolna pulsacja graniczna toru wynosi [8]:

$$\omega_d = \frac{1}{RC} \quad (2.38)$$



Rysunek 2.7: Charakterystyki częstotliwościowe toru pomiarowego z przetwornikiem piezoelektrycznym



Rysunek 2.8: Model akcelometru piezoelektrycznego (Źródło: [8])

i rośnie ze wzrostem rezystancji  $R$  i pojemności  $C$ . Aby poszerzyć pasmo przenoszenia, omawiana częstość musi być jak najmniejsza. Zwiększanie pojemności układu jest niekorzystne ze względu na spadek czułości napięciowej 2.28 oraz wartości napięcia 2.26. Należy więc zapewnić jak największą wartość rezystancji zastępczej  $R$ , co nakłada duże wymagania jakościowe na wzmacniacz pomiarowy oraz przewody łączące go z czujnikiem piezoelektrycznym. Zwykle rezystancja kwarcowego przetwornika piezoelektrycznego osiąga wartości rzędu  $10^{15} \Omega$  [8].

### 2.3.3. Akcelerometr piezoelektryczny

Zasada działania akcelometru piezoelektrycznego polega na wytwarzaniu ładunku elektrycznego na powierzchni piezoelektryka, w wyniku oddziaływania masy sejsmicznej poddawanej działaniu siły bezwładności. Wartość ładunku zależy od wartości siły, zaś znak od tego czy kryształ jest ściskany czy rozciągany.

Na rysunku 2.8 pokazano model akcelometru piezoelektrycznego. Oddziaływanie z płytką piezoelektryka zachodzi wzdłuż jego osi elektrycznej, zaś ładunek generowany jest na powierzchniach prostopadłych do niej. Sprężyna wywiera wstępny nacisk masy, którego wartość zmienia się w zależności od mierzonego przyspieszenia.

Siłę bezwładności, z jaką masa sejsmiczna działa na przetwornik, może być wyznaczona z zależności:

$$F_b = mu'' \quad (2.39)$$

Uwzględniając zależność 2.25 łączącą siłę z ładunkiem, można napisać że:

$$mu'' = \frac{1}{k_p} Q \quad (2.40)$$

Przy założeniu zerowych warunków początkowych, transformata napięcia  $u(t)$  ma postać:

$$U(s) = \frac{1}{k_p m s^2} Q(s) \quad (2.41)$$

Równanie ruchu masy można zapisać jako:

$$mu'' + cy' + ky = 0 \quad (2.42)$$

co po uwzględnieniu zależności między przemieszczeniami prowadzi do związku:

$$mu'' + cu' + ku = cx' + kx \quad (2.43)$$

Mając na uwadze podstawienia 2.6 i 2.7, można powyższą zależność zapisać operatorowo w następujący sposób:

$$\frac{1}{\omega_0^2} s^2 U(s) + \frac{2\xi}{\omega_0} s U(s) = \frac{2\xi}{\omega_0} s X(s) + X(s) \quad (2.44)$$

Transmitancja operatorowa, wiążąca bezwzględne przemieszczenie masy  $u$  z bezwzględnym przemieszczeniem  $x$  obiektu ma postać:

$$G_U(s) = \frac{U(s)}{X(s)} = \frac{\frac{2\xi}{\omega_0} s + 1}{\frac{1}{\omega_0^2} s^2 + \frac{2\xi}{\omega_0} s + 1} \quad (2.45)$$

Jeśli za sygnał wyjściowy przyjmiemy nie napięcie lecz ładunek, wówczas transmitancja przyjmie postać:

$$G_Q(s) = m k_p s^2 \frac{\frac{2\xi}{\omega_0} s + 1}{\frac{1}{\omega_0^2} s^2 + \frac{2\xi}{\omega_0} s + 1} \quad (2.46)$$

Ponieważ w akcelerometrach piezoelektrycznych pulsacja drgań własnych ma dużą wartość (mała masa sejsmiczna i duża sztywność sprężyny) a tłumienie niewielką (ok. 0.01) [8], więc ostatnie równanie może być zapisane w prostszy sposób:

$$G_Q(s) = m k_p s^2 \quad (2.47)$$

z czego wynika, że w dziedzinie czasu zachodzi relacja:

$$Q = m k_p x'' \quad (2.48)$$

a zatem generowany ładunek jest proporcjonalny do przyspieszenia drgającego obiektu.

### 3. Żyroskopy i ich zastosowanie w elektronice użytkowej

#### 3.1. Budowa i zasada działania

Żyroskop jest urządzeniem służącym do pomiaru lub utrzymywania położenia kąтового. Działa w oparciu o zasadę zachowania momentu pędu. Zwykle jego zasadniczym elementem jest ruchomy krążek, który wprawiony w szybki ruch obrotowy zachowuje położenie swojej osi obrotu, z niewielkimi ruchami precesyjnymi, które mogą zostać wyeliminowane dzięki zastosowaniu tłumienia. Aby poprawna praca żyroskopu była możliwa, konieczne jest zapewnienie szybkiej prędkości obrotowej krążka oraz dobrych warunków łożyskowania. Wymagają one zminimalizowania tarcia, co można osiągnąć poprzez łożyskowanie na strumieniu powietrza, lub zawieszeniu w polu elektrostatycznym (lub magnetycznym) w próżni. Przedmiotem znanym z życia codziennego, który wprawiony w szybki ruch obrotowy zachowuje się jak żyroskop jest bąk.

Na rysunku 3.1 pokazano przykładową konstrukcję żyroskopu. W jego centralnym miejscu znajduje się krążek zawieszony na podwójnej ramce. Jeśli nadamy mu dużą prędkość obrotową, to zachowa on położenie osi obrotu, choć możliwe będzie wystąpienie ruchów precesyjnych.

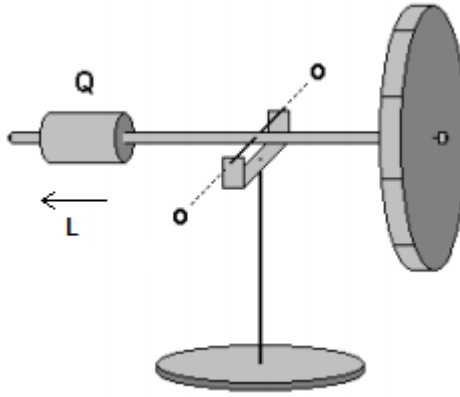
Dla ruchu obrotowego prawdziwa jest zależność:

$$\vec{M} = I \vec{\epsilon} \quad (3.1)$$

gdzie:



Rysunek 3.1: Jedna z możliwych konstrukcji żyroskopu (Źródło: [5])



Rysunek 3.2: Inna konstrukcja żyroskopu (Źródło: [6])

- $\vec{M}$  - wektor momentu siły przyłożonego do bryły,
- $I$  - moment bezwładności bryły sztywnej,
- $\vec{\omega}$  - wektor przyspieszenia kąowego (bryły).

Moment siły powoduje zmianę momentu pędu:

$$\vec{M} = \frac{d\vec{L}}{dt} \quad (3.2)$$

gdzie  $\vec{L}$  jest wektorem momentu pędu.

Wielkość  $\vec{L}$  wyznacza się z następującego wzoru:

$$\vec{L} = I\vec{\omega} \quad (3.3)$$

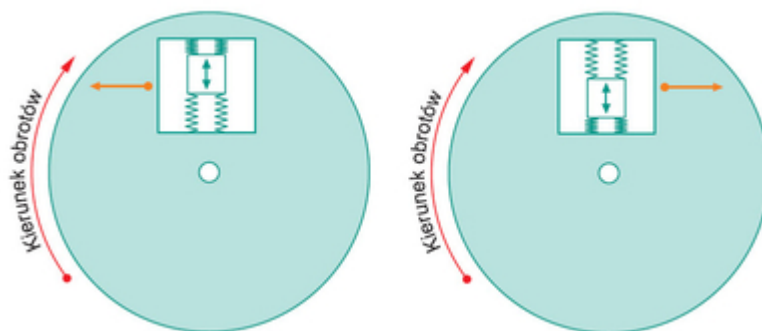
gdzie  $\vec{\omega}$  jest prędkością kątową bryły sztywnej.

Zjawisko precesji występujące w żyroskopach można opisać na przykładzie konstrukcji z rysunku 3.2. Jeżeli ciężar moment siły pochodzący od ciężaru  $Q$  równoważy moment siły pochodzący od ciężaru krążka, wówczas żyroskop wykonyuje ruch obrotowy wokół swojej osi (prostej prostopadłej do płaszczyzny krążka i przechodzącej przez jego środek). Jeśli zwiększymy ciężar  $Q$  (lub przesuniemy ciężarek w lewo lub prawo), pojawi się dodatkowy moment siły działający prostopadle do  $\vec{L}$ . Spowoduje on, że dotychczasowa oś obrotu krążka zacznie się obracać wokół dotychczasowego położenia, zakreślając przy tym powierzchnię boczną stożka. Ruch ten nazywa się precesyjnym, zaś częstość precesji jest wprost proporcjonalna do momentu, który ją wywołuje i odwrotnie proporcjonalna do momentu bezwładności i częstości ruchu obrotowego krążka:

$$f_p = \frac{M_p}{4\pi^2 I f_r} \quad (3.4)$$

gdzie:

- $f_p$  - częstość precesji,



Rysunek 3.3: Żyroskop MEMS (Źródło: [7])

- $M_p$  - moment siły powodujący wystąpienie precesji (prostopadły do pierwotnego momentu pędu  $\vec{L}$ ),
- $I$  - moment bezwładności,
- $f_r$  - częstość ruchu oobrotowego krążka.

Opisane zjawisko określa się mianem precesji wymuszonej, gdyż wywołana jest przez moment  $\vec{M}_p$ . Prócz tego możliwa jest także precesja swobodna, występująca gdy oś wokół której obraca się swobodnie bryła, nie pokrywa się z żadną z osi głównych tensora momentu bezwładności tej bryły.

### 3.2. Podział żyroskopów

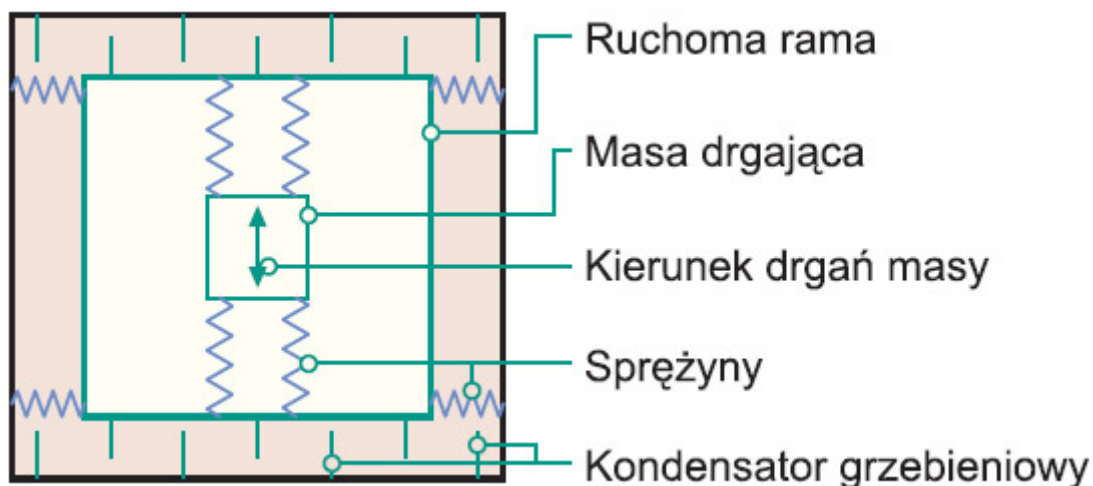
Żyroskopy można podzielić na dwie grupy - kierunkowe oraz prędkościowe. W przypadku tych pierwszych, zasadniczym elementem jest szybko wirujący obiekt, mający zwykle postać dysku, zawieszony w konstrukcji pozwalającej na swobodny obrót wokół układu odniesienia (np. ciała do którego umocowany jest żyroskop). Można to uzyskać dzięki zastosowaniu odpowiednich przegubów o osi obrotu prostopadłej do osi obrotu wirującej bryły. Takie rozwiązanie zmniejsza wpływ obrotów ciała na żyroskop, który dąży do zachowania stałego położenia.

Przeznaczeniem żyroskopów prędkościowych jest wskazywanie prędkości kątowej obiektu, na którym są umiejscowione. Zaliczamy do nich konstrukcje mechaniczne (których wadą jest ograniczona swoboda w ruchu obrotowym) a także żyroskopy optyczne (laserowe i światłowodowe) oraz konstrukcje wykorzystujące zjawisko Coriolisa.

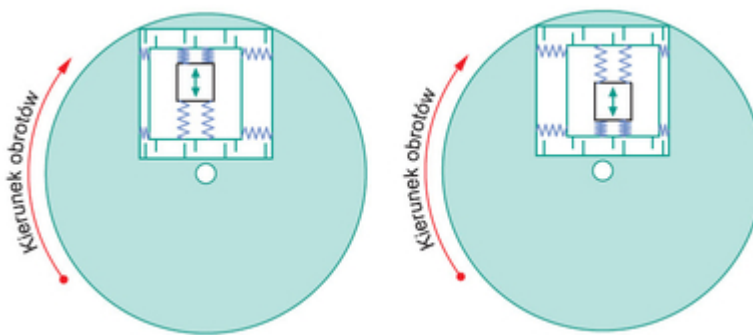
### 3.3. Żyroskopy w elektronice użytkowej

Żyroskopy są elementem wielu urządzeń elektrycznych, w których należy mierzyć lub rejestrować zmiany położenia (np. wskaźniki komputerowe, myszki, smartfony). Mogą być one wytwarzane w technologii MEMS (o której wspomniano w rozdziale o akcelerometrach), przy czym dotyczy to głównie żyroskopów prędkościowych. Konstrukcje takie wykorzystują efekt Coriolisa, a ich poglądowy schemat pokazano na rysunku 3.3. Zasadniczym elementem jest masa, wytrawiana w polikrzemie i przytwierdzona do krzemowej ramy tak, aby mogła wykonywać ruchy tylko w jednym kierunku. Mają one postać





Rysunek 3.4: Czujnik z kondensatorem grzebieniowym (Źródło: [7])

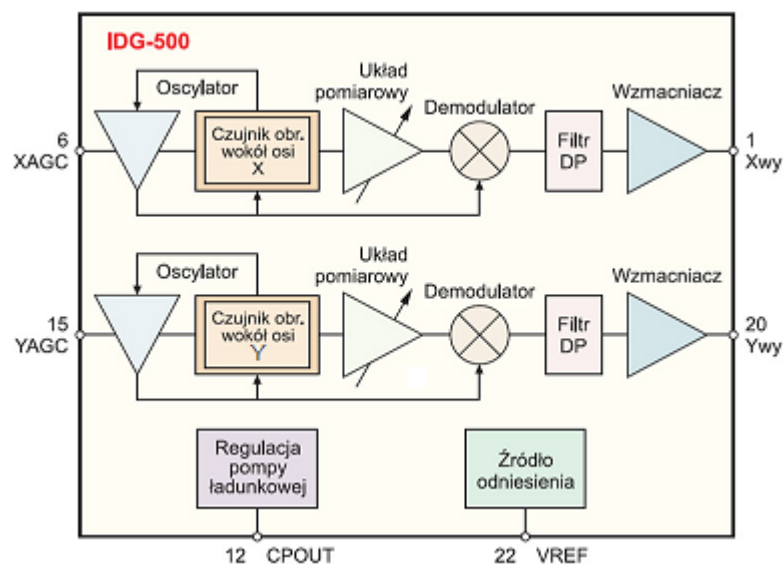


Rysunek 3.5: Żyroskop wibracyjny MEMS z kondensatorem grzebieniowym (Źródło: [7])

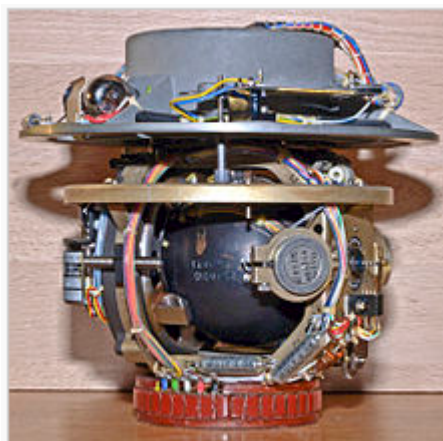
drgań możliwych, dzięki zawieszeniu masy na sprężynach. Całość jest umieszczona na platformie wykonującej ruch obrotowy. Gdy masa porusza się w kierunku brzegu tarczy, działa na nią siła skierowana w prawo. Jednocześnie sama masa oddziałuje na ramę siłą skierowaną w lewo. W trakcie przemieszczania się w przeciwnym kierunku, tj. ku środkowi tarczy, siła oddziaływania masy na ramę zmienia zwrot na przeciwny. Aby pozwolić na pomiar przyspieszenia Coriolisa, ramę wiesza się na sprężynach pod kątem prostym do kierunku przemieszczania się masy (rysunek 3.5). Rysunek 3.4 pokazuje czujnik z kondensatorem grzebieniowym, którego część elektrod jest związana z ruchomą ramą a pozostała część z nieruchomym podłożem. Wskutek siły wywieranej przez masę, dochodzi do zmiany odległości między elektrodami i w konsekwencji do zmiany pojemności kondensatora.

### 3.4. Żyroskopy wieloosiowe

Na rynku dostępne są czujniki mierzące prędkość obrotową wokół dwóch lub trzech osi. Wcześniej chcąc uzyskać taki efekt należało zastosować kilka niezależnych żyroskopów jednoosiowych. W wielu zastosowaniach takie podejście się sprawdzało, lecz mogło nie być optymalne biorąc pod uwagę koszty oraz miejsce zajmowane przez cały układ pomiarowy. Dziś problem ten rozwiązują żyroskopy wieloosiowe. Przykładowy schematu blokowy takiego czujnika pokazano na rysunku 3.6. Prócz właściwych



Rysunek 3.6: Schemat blokowy dwuosiowego czujnika żyroskopowego (Źródło: [7])



Rysunek 3.7: Żyrokompas (Źródło: [5])

żyroskopów zawiera on także:

- oscylatory, odpowiedzialne za wprowadzenie masy w ruch drgający,
- układ regulacji amplitudy drgań masy,
- demodulator synchroniczny,
- filtr dolnoprzepustowy,
- pompa ładunkowa odpowiedzialna za wytworzenie wysokiego napięcia potrzebnego do zasilenia układu rezonatora pobudzającego masę do drgań.

### 3.5. Zastosowania

Poniżej wymieniono niektóre z praktycznych zastosowań żyroskopów:

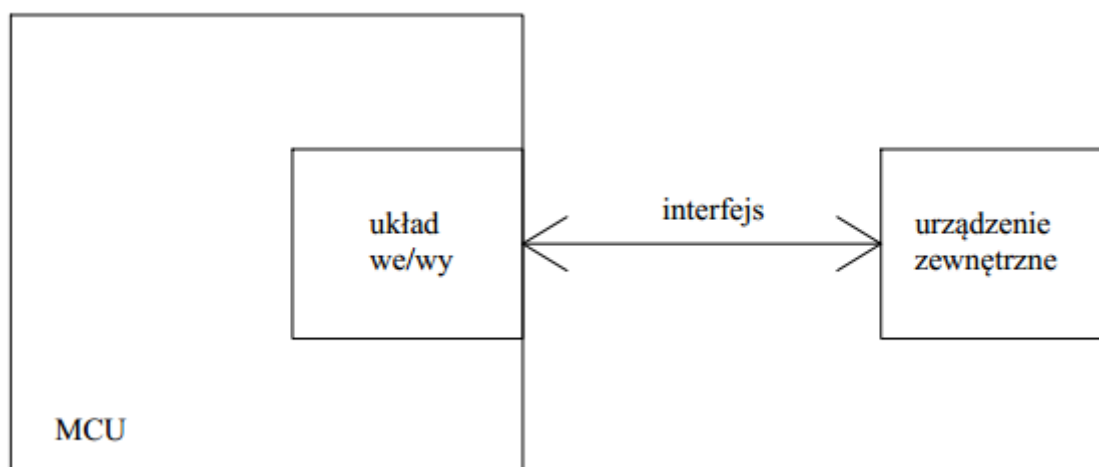
- wskaźniki komputerowe, myszki, itp.,
- żyrokomпасы używane w nawigacji morskiej,
- systemy autopilotażu samolotów,
- układy utrzymywania orientacji satelitów,
- kontrolery gier,
- układy stabilizacji jednostek pływających,
- układy detekcji położenia i orientacji w smartfonach i tabletach.

## 4. Współpraca mikrokontrolerów z urządzeniami zewnętrznymi

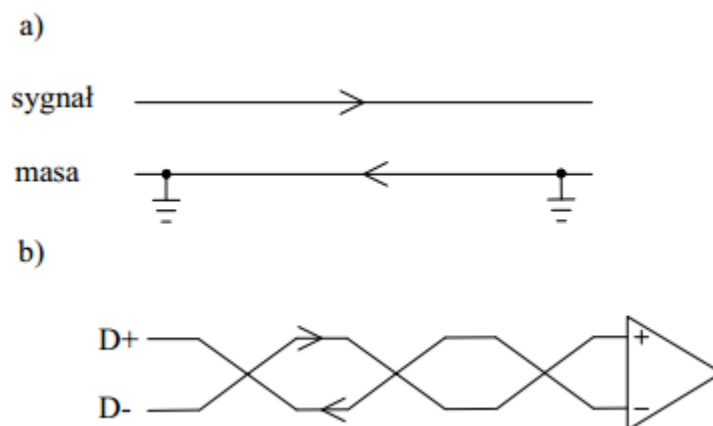
### 4.1. Wiadomości ogólne

Współpraca mikrokomputerów z urządzeniami peryferyjnymi odbywa się za pomocą tzw. interfejsów. Przez interfejs rozumie się urządzenie, które pozwala na komunikację i wymianę danych pomiędzy dwoma innymi urządzeniami. Istnieją różne kryteria podziału interfejsów:

- ze względu na istnienie fizycznego połączenia między komunikującymi się urządzeniami:
  - przewodowe (np. UART, I2C, FireWire (IEEE 1394), RS-232), w których rolę sygnałów pełnią napięcie lub prąd, zaś rolę medium przewody,
  - bezprzewodowe (np. Bluetooth, IrDA, WiFi) , w których sygnałami są fale elektromagnetyczne, zaś medium najczęściej powietrze,
- ze względu na ilość danych (bitów) przesyłanych w trakcie elementarnego cyklu transmisji:
  - szeregowe (np. SPI, UART, I2C), w których dane są transmitowane bit po bicie,
  - równoległe (np. GPIB, SCSI, ATA oraz używany kiedyś w drukarkach interfejs Centronics),



Rysunek 4.1: Współpraca mikrokontrolera z urządzeniem zewnętrznym (Źródło: [9])



Rysunek 4.2: Linie transmisyjne: a) niesymetryczna, b) symetryczna (Źródło: [9])

- charakter sygnałów:
  - analogowe (np. Jack, S-Video),
  - cyfrowe (np. SPI, I2C, USB).

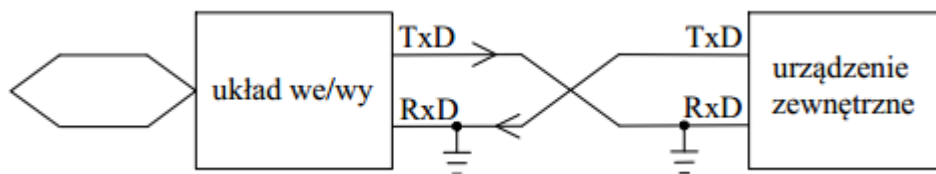
Pod pojęciem interfejsu przewodowego rozumie się złącza oraz przewody łączące urządzenie zewnętrzne z mikrokomputerem wraz z ich specyfikacją mechaniczną (kształt, rozmiar, ilość) oraz charakterystykami elektrycznymi sygnałów (poziomy napięcie i prądów, relacje czasowe). Aby przesył sygnałów był możliwy, związany z nimi obwód elektryczny musi być zamknięty., dlatego składa się z dwóch przewodów tworzących linię. Istnieją linie symetryczne i niesymetryczne. Schemat obu pokazano na rysunku 4.2.

Linia niesymetryczna składa się z przewodu sygnałowego (zwanego także gorącym) oraz masowego, stanowiącego odniesienie dla sygnału po obu stronach toru transmisyjnego. W przypadku większej liczby sygnałów możliwe jest zastosowanie wspólnego przewodu powrotnego [9]. Największą wadą linii niesymetrycznej jest jej podatność na wpływ obcych pól elektromagnetycznych. Powodują one indukowanie się zakłóceń w przewodzie sygnałowym i powrotnym, które dodają się lub odejmują od właściwego sygnału na wejściu odbiornika. Z tego powodu linie tego typu stosuje się do przesyłu informacji na nieduże odległości. W celu zmniejszenia wpływu zewnętrznych pól, stosuje się następujące zabiegi:

- skręcanie przewodów,
- ekranowanie linii.

Linia symetryczna składa się z dwóch przewodów wiodących sygnały znajdujące się w przeciwfazie oraz odbiornika z wejściem różnicowym. Dzięki temu, sygnały o takich samych przebiegach indukowane w poszczególnych przewodach znoszą się po odjęciu na wejściu wzmacniacza. W celu zapewnienia lepszej ochrony przed zakłóceniami możliwe jest również skręcanie przewodów. Zasięg linii symetrycznych jest dużo większy niż w przypadku linii niesymetrycznych i osiąga rząd 1 km [9].

Najbardziej odpornym na zakłócenia elektromagnetyczne medium jest światłowód. Pozwala on na transmisję na bardzo duże odległości (od kilku do kilkudziesięciu kilometrów). Tor światłowodowy wymaga zastosowania nadajnika przekształcającego sygnał elektryczny na impulsy świetlne, właściwego



Rysunek 4.3: Układ transmisji szeregowej (Źródło: [9])

światłowodu (wykonanego zwykle z włókna szklanego) oraz odbiornika realizującego odwrotną konwersję sygnałów.

#### 4.1.1. Transmisja szeregową i równoległą

Transmisja szeregową polega na przesyłaniu informacji bit po bicie. Oznacza to, że w elementarnym cyklu transmisji nadajnik wysyła jeden bit, zaś odbiornik go odbiera. Zaletą takiego rozwiązania jest stosunkowo niewielka ilość użytych przewodów. Liczba ta jest większa w przypadku transmisji równoległej, w której na raz możliwe jest przesyłanie większej ilości bitów (każdy swoim torem). Teoretycznie przyspiesza to szybkość transferu danych, jednak w praktyce jest ona ograniczona ze względu na zakłócanie sygnałów przez pola pochodzące od sąsiednich linii danych. Inną wadą tego typu komunikacji jest także stosunkowo skomplikowana budowa (a co zatem idzie większy koszt) interfejsów równoległych. Omawiana fakty uzasadniają dlaczego większość współczesnych urządzeń cyfrowych wykorzystuje interfejsy i transmisję szeregową.

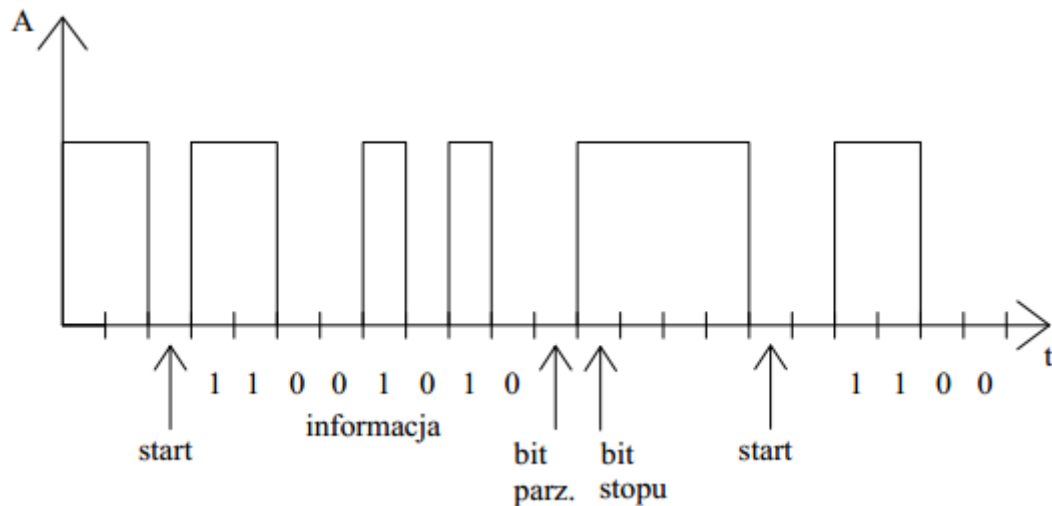
Rysunek 4.3 przedstawia uproszczony schemat układu transmisji szeregowej. Dzięki obecności dwóch przewodów sygnałowych, możliwa jest komunikacja w trybie duplex (ang. *full duplex*), pozwalająca na jednoczesną wymianę danych przez komunikujące się urządzenia. Sygnał nadawany (TxD) wysyłany jest do portu odbierającego dane (RxD). Urządzenia uczestniczące w komunikacji wyposażone są zarówno w nadajnik jak i odbiornik.

Prócz transmisji duplexowej, w układach cyfrowych możliwe są również jeszcze dwa inne tryby wymiany danych:

- półdupleks (ang. *half duplex*) - pozwalający na obustronną, lecz nie jednoczesną wymianę danych,
- simpleks (ang. *simplex*) - pozwalający na przesył danych tylko w jednym kierunku.

Dane wysyłane szeregowo są często przechowywane w rejestrach. Aby umożliwić transmisję bit po bicie stosuje się rejestry przesuwne lub multipleksery. W obu przypadkach należy określić czas trwania bitu (tzw. odcinek bitowy) [9]. Służy do tego sygnał zegarowy nadajnika. Odbiornik określa wartość przesyłanych bitów w środkach poszczególnych odcinków bitowych. Także w tym przypadku używany jest sygnał zegarowy (zegar odbiornika). Istnieją dwa rodzaje transmisji szeregowej, różniące się zależnościami sygnałów zegarowych nadajnika i odbiornika:

- transmisja asynchroniczna,
- transmisja synchroniczna.



Rysunek 4.4: Przebieg sygnału w transmisji szeregowej asynchronicznej (Źródło: [9])

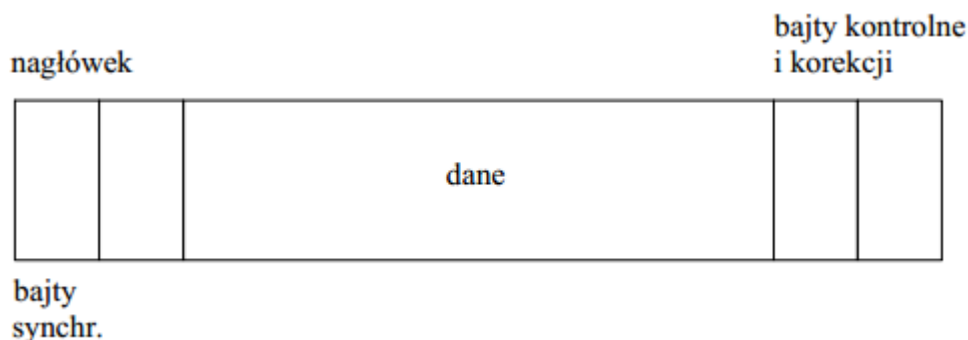
#### 4.1.2. Transmisja asynchroniczna i synchroniczna

W przypadku transmisji asynchronicznej zegary nadajnika i odbiornika nie są synchronizowane. Dane nie muszą być przesyłane w sposób ciągły, tzn. między kolejnymi transmisjami, linia danych znajduje się w stanie spoczynku przez tzw. czas martwy. Dane przesyłane noszą w literaturze nazwę znaku i składają się zwykle od pięciu do ośmiu bitów. Aby odbiornik mógł wykryć początek oraz koniec nadawania danych, każdy znak poprzedzony jest tzw. bitem startu oraz zakończony tzw. bitami stopu (od jednego do dwóch). Dodatkowo w celu zapewnienia poprawności transmisji stosuje się również tzw. bit parzystości. Wadą transmisji asynchronicznej jest istnienie czasu martwego, zmniejszającego efektywność przesyłu danych. Suma liczby jedynek tworzących znak wraz z bitem parzystości powinna tworzyć liczbę parzystą.

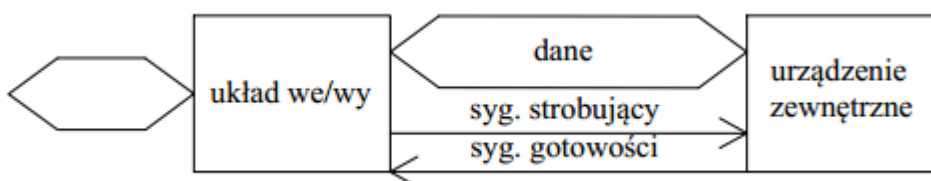
Przed właściwą komunikacją należy ustalić jej następujące parametry:

- częstotliwość zegarów,
- długość znaku,
- bit parzystości (obecny, nieobecny lub zanegowany),
- ilość bitów stopu.

W transmisji synchronicznej zegary nadajnika i odbiornika są synchronizowane (stąd nazwa), dzięki czemu nie są konieczne bity startu i stopu. Dane przesyłane grupowane są w bloki zwane ramkami (ang. *frame*). Każdą ramkę rozpoczyna nagłówek (rysunek 4.5), zawierający bajty synchronizacji potrzebne do zsynchronizowania zegarów nadajnika i odbiornika. W celu zapewnienia poprawnego transferu, synchronizacja ta powinna być utrzymywana podczas całego procesu komunikacji. Prócz wspomnianych bajtów, nagłówek zawiera także dodatkowe informacje, np. ilość przesyłanych danych. Blok danych zawiera właściwe dane zgrupowane jeden za drugim. W celu zapewnienia poprawności transmisji, są one zakończone tzw. bajtami kontrolnymi i ewentualnymi bajtami korekcyjnymi pozwalającymi na naprawę uszkodzonych danych.



Rysunek 4.5: Ramka transmisji szeregowej synchronicznej (Źródło: [9])



Rysunek 4.6: Układ transmisji równoległej (Źródło: [9])

#### 4.1.3. Transmisja równoległa

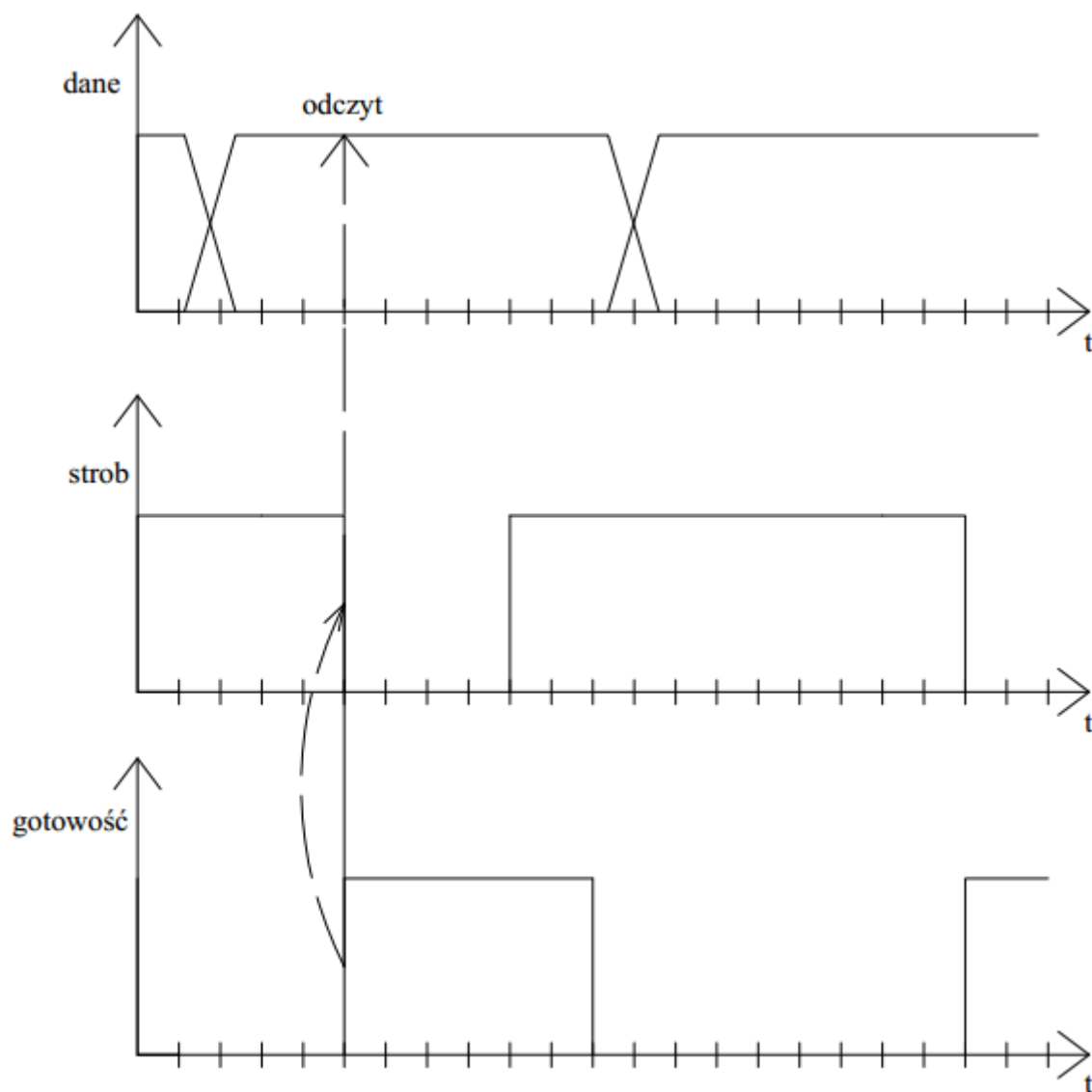
W transmisji równoległej grupy bitów wysyłane są w tym samym czasie niezależnymi liniami danych. Można powiedzieć, że w tym przypadku przesyłane są szeregowo grupy bitów (zwykle bajty lub tzw. słowa). Dane wysyłane są w takt sygnału zegarowego, zwanego sygnałem strobowym. Odbiornik wykorzystuje ten sygnał do zapisu odbieranych danych. Zwykle transmisja tego typu jest realizowana jednokierunkowo [9]. W przypadku powolnych odbiorników (np. drukarki) stosuje się dodatkowe bufory niezależniące transmisję od czasu przetwarzania. Aby jednak dane mogły być przesyłane, odbiornik musi zasygnalizować nadajnikowi gotowość do ich odbioru przy pomocy sygnału gotowości (rysunek 4.6).

Na rysunku 4.7 pokazano przebiegi sygnałów towarzyszących transmisji równoległej. Sygnał strobu (aktywny poziomem niskim) jest ustawiany w sytuacji aktywnego sygnału gotowości oraz po ustabilizowaniu się stanu magistrali danych. Urządzenie zewnętrzne (odbiornik) wykorzystuje jego zbocze opadające do zapisu odbieranych danych. W istniejących interfejsach, zarówno szeregowych jak i równoległych wykorzystuje się także inne sygnały sterujące komunikacją.

## 4.2. Interfejs SPI

Interfejs SPI (ang. *Serial Peripheral Interface*) został zaprojektowany i po raz pierwszy zastosowany przez firmę Motorola. Jest jednym z najczęściej używanych interfejsów komunikacyjnych między układami mikroprocesorowymi a peryferiami takimi jak przetworniki ADC/DAC, układy RTC, pamięci EEPROM, pamięci flash, karty MMC/SD, sterowniki ekranów dotykowych itp. W świetle omówionych wcześniej cech interfejsów, SPI może być scharakteryzowany jako:





Rysunek 4.7: Przebiegi sygnałów w transmisji równoległej (Źródło: [9])

- szeregowy,
- synchroniczny,
- komunikujący się w trybie *full duplex*,
- przewodowy.

Komunikacja z urządzeniami wykonywana jest w oparciu o model wymiany danych master/slave. Do przesyłu danych wykorzystywane są zwykle 4 przewody, choć interfejs może również funkcjonować w postaci trójprzewodowej. SPI jest standardem *de facto*, tzn. jest powszechnie stosowany przez wielu producentów sprzętu elektronicznego, lecz w przeciwieństwie do standardu *de jure* nie istnieje jego formalna specyfikacja zaakceptowana przez komitety standaryzacyjne typu IEEE, ANSI lub ISO. Interfejs SPI jest też czasem określany jako czteroprzewodowa magistrała szeregową lub SSI (ang. *Synchronous Serial Interface*).

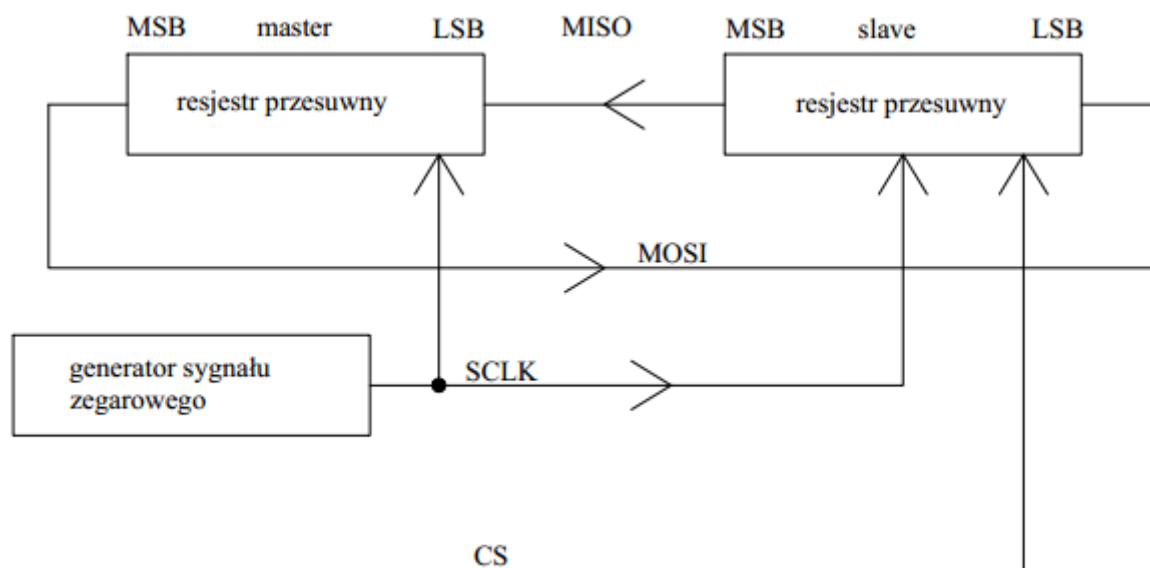
Omawiany interfejs obejmuje cztery sygnały:

- sygnał zegarowy SCLK (wyjście urządzenia *master*),
- sygnał MOSI (ang. *Master Output Slave Input*) - wyjście urządzenia *master* i wejście urządzenia *slave*,
- sygnał MISO (ang. *Master Input Slave Output*) - wejście urządzenia *master* i wyjście urządzenia *slave*,
- sygnał CS (ang. *Chip Select*) lub SS (ang. *Slave Select*) - sygnał wyboru urządzenia (wyjście urządzenia *master*).

Wymienione powyżej sygnały posiadają także alternatywne nazwy:

- SCLK - SCK, CLK,
- MOSI - SIMO, SDO, DO,
- MISO - SOMI, SDI, DI,
- CS - nCS, nSS, CSB, STE.

Z pojedynczym układem *master* może współpracować jeden lub więcej układów typu *slave*. W ostatnim przypadku należy zapewnić odpowiednią (równa liczbie urządzeń *slave*) ilość niezależnych sygnałów wyboru płytki (CS). Ponieważ układ transmisji SPI wykorzystuje rejestr przesuwany, innym rozwiązaniem może okazać się także połączenie układów podrzędnych, tak aby utworzyły jeden rejestr przesuwany. Ponadto współpraca z wieloma urządzeniami *slave* wymaga, aby w trakcie ich nieaktywności ich wyjścia MISO wchodziły w stan wysokiej impedancji.



Rysunek 4.8: Typowa realizacja transmisji SPI przy pomocy cyklicznego rejestru przesuwanego (Źródło: [9])

#### 4.2.1. Przesył danych

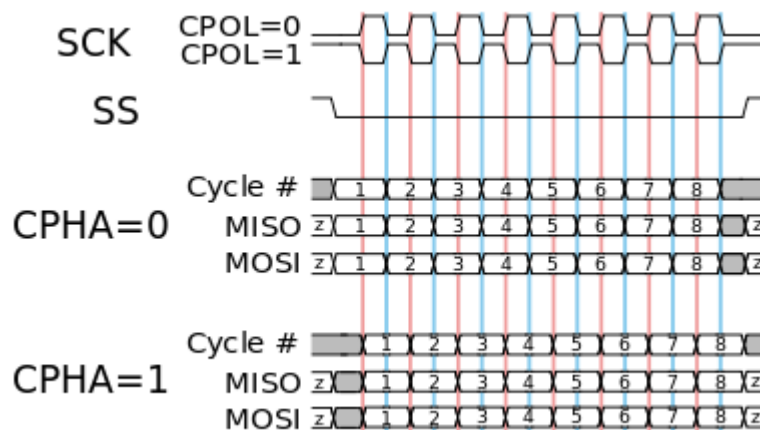
Przed dokonaniem transmisji danych, urządzenie *master* musi określić odpowiednią częstotliwość sygnału zegarowego. Nie może być ona większa od żadnej z maksymalnych dopuszczalnych częstotliwości wspieranych przez urządzenia *slave*. Następnie sygnał wyboru urządzenia (CS) przechodzi w stan aktywny (zwykle logiczne 0). Jeśli należy poczekać, aby współpracujące urządzenie podrzędne dokonało przetwarzania danych (tak jest np. w przypadku przetworników analogowo-cyfrowych), urządzenie nadrzędne musi wstrzymać się z generacją sygnału zegarowego.

Z każdym taktom zegara związany jest cykl komunikacji *full duplex*:

- urządzenie master wysyła linią MOSI kolejny bit, zaś slave go odbiera,
- urządzenie slave wysyła kolejny bit danych linią MISO, zaś master go odbiera.

W transmisji wykorzystywane są zwykle dwa rejestry przesuwne (po jednym dla urządzenia master i slave) połączone ze sobą tworząc rejestr cykliczny (rysunek 4.8). Zwykle wysyłany jest najbardziej znaczący bit MSB (ang. *most significant bit*) a odbierany najmniej znaczący LSB (ang. *least significant bit*). Po wypełnieniu rejestrów następuje obróbka zawartych w nich danych, tj. zapis do pamięci, przetwarzanie itp. W celu wymiany większej ilości informacji, wspomniany proces się powtarza. W trakcie pojedynczego cyklu komunikacji przesyłany jest zwykle bajt danych (8 bitów), choć nie jest to regułą; istnieją układy w których przesyła się 12, 16 lub 24 bity. Niektóre urządzenia *slave* nie pozwalają na odbieranie danych w ciągu większej od zadanej liczby taktów sygnału zegarowego; w innych rozwiązaniach konstrukcyjnych dane nadmiarowe są ignorowane.

Oprócz wymienionych sygnałów sterujących komunikacją, w urządzeniu *slave* możliwe jest również wyprowadzenie sygnału przerwania. Przykładowo może to być sygnał informujący o dotknięciu ekranu dotykowego, sygnał alarmowy generowany przez czujnik temperatury lub zegar czasu rzeczywistego.



Rysunek 4.9: Przebiegi odpowiadające różnym polaryzacji i fazom sygnału taktującego (Źródło: [1])

stego. Standard SPI nie przewiduje użycia przerwań, więc przeznaczanie dla nich oddzielnego pinu w urządzeniu *slave* nie jest ani zakazane ani obowiązkowe.

Każde z urządzeń *slave*, które nie zostało aktywowane sygnałem CS nie powinno reagować na sygnał taktujący ani sterować wspólną magistralą. Urządzenie *master*, wybiera zaś do wymiany informacji tylko jedno urządzenie podrzędne.

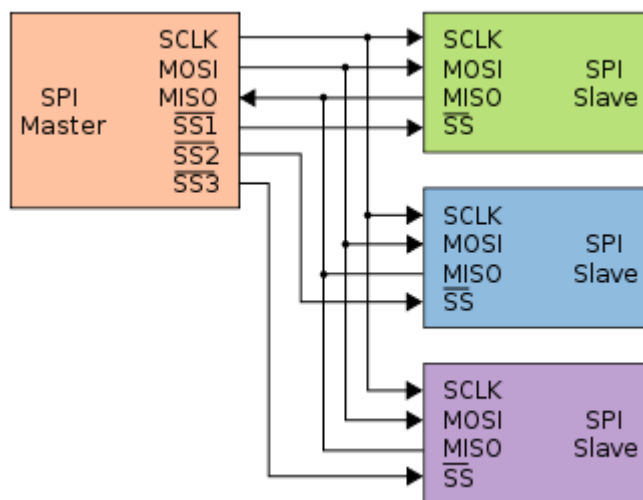
#### 4.2.2. Faza i polaryzacja sygnału zegarowego

Poza określeniem częstotliwości sygnału taktującego, urządzenie nadrzędne musi także zdefiniować polaryzację i fazę tego sygnału. Zwykle jest to związane z wpisaniem do odpowiednich rejestrów konfiguracyjnych wartości 0 lub 1 na odpowiednich pozycjach. Bity na tych pozycjach zwykle nazywane są CPOL (dla określenia polaryzacji) oraz CPHA (dla określenia fazy); ich znaczenie jest następujące:

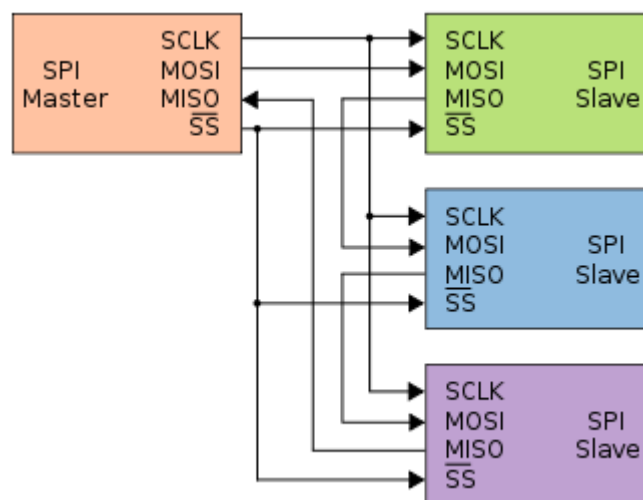
- jeśli CPOL = 0, to stanem nieaktywnym sygnału zegara jest stan niski oraz:
  - jeśli CPHA = 0, to dane zapisywane są przy zboczu narastającym zegara, zaś przesyłane przy opadającym,
  - jeśli CPHA = 1, to dane zapisywane są przy zboczu opadającym zegara, zaś przesyłane przy narastającym,
- jeśli CPOL = 1, to stanem nieaktywnym sygnału taktującego jest stan wysoki oraz:
  - jeśli CPHA = 0, to dane zapisywane są przy zboczu opadającym zegara, zaś przesyłane przy narastającym,
  - jeśli CPHA = 1, to dane zapisywane są przy zboczu narastającym zegara, zaś przesyłane przy opadającym.

#### 4.2.3. Współpraca z kilkoma urządzeniami podrzędnymi

Jak wspomniano wcześniej, interfejs SPI pozwala na współpracę jednego urządzenia nadrzędnego z kilkoma urządzeniami *slave*. Jednym ze sposobów zorganizowania takiego układu transmisji danych



Rysunek 4.10: Typowa realizacja komunikacji przy pomocy interfejsu SPI z wieloma urządzeniami *slave* (Źródło: [2])



Rysunek 4.11: Komunikacja przy pomocy interfejsu SPI w konfiguracji *daisy chain* (Źródło: [3])

jest zastosowanie niezależnych sygnałów CS dla każdego z urządzeń podrzędnych. Przykład takiego rozwiązania pokazano na rysunku 4.10. Ponieważ wszystkie wyjścia MISO układów *slave* są połączone do wspólnej linii, powinny być one wyjściami trójstanowymi.

Niektóre urządzenia wspierające SPI są zaprojektowane z myślą o zastosowaniu połączenia łańcuchowego (znanego w literaturze anglojęzycznej jako *daisy chain*). W takiej konfiguracji wyjście jednego z urządzeń *slave* jest połączone do wejścia drugiego *slave'a* itd. Port SPI takiego urządzenia podrzędnego w danym cyklu transmisji odbiera dane od poprzedniego urządzenia, które te same dane odebrało w poprzednim cyklu. Cała kaskada działa jak jeden duży rejestr przesuwany. Zaletą takiego rozwiązania jest to, że urządzenie typu *master* może wykorzystywać tylko jeden sygnał CS.

#### 4.2.4. Zalety i wady interfejsu SPI

Do zalet interfejsu SPI możemy zaliczyć:

- komunikacja w trybie *full duplex*,
- ilość przesyłanych danych nie jest ograniczona do 8 bitów,
- prosty do zrozumienia i implementacji protokół,
- stosunkowo prosta budowa:
  - relatywnie mało skomplikowane obwody elektryczne, mniejsze zużycie energii,
  - urządzenia *slave* używają wspólnego sygnału zegarowego generowanego przez układ *master* (dodatkowe precyzyjne oscylatory i pętle PLL są niepotrzebne),
  - urządzenia *slave* nie wymagają unikalnych adresów (w przeciwieństwie do I2C, GPIB czy SCSI),
  - nie są wymagane transceivery,
- wystarczają tylko 4 piny,
- większość linii sygnałowych może być współdzielonych przez różne urządzenia *slave*.

Wśród wad interfejsu SPI można wymienić:

- większą w porównaniu do I2C ilość pinów obudowy IC,
- brak sprzętowej kontroli przepływu danych przez urządzenia *slave*,
- brak sprzętowego potwierdzania obecności urządzenia *slave* (*master* może “rozmawiać” z niczym),
- brak wsparcia dla architektury multi-master,
- brak oficjalnego standardu,
- brak protokołu obsługi błędów,
- brak wsparcia dla dynamicznego podłączania urządzeń *slave*,
- działanie na stosunkowo nieduże odległości.

#### 4.2.5. Zastosowania

Dużo mniejsza w porównaniu z interfejsami równoległymi liczba niezbędnych pinów powoduje, że SPI jest bardzo często wykorzystywany w systemach wbudowanych. Jest on elementem wielu mikrokontrolerów z rodziny ARM, AVR, PIC oraz MSP. Niektóre mikrokontrolery AVR mogą być programowane z użyciem interfejsu SPI. Często używa się go także do komunikacji z takimi układami peryferyjnymi jak:

- czujniki temperatury, ciśnienia, przyspieszenia,

- przetworniki ADC oraz DAC,
- kontrolery ekranów dotykowych oraz gier,
- pamięci flash oraz EEPROM,
- zegary RTC,
- karty MMC oraz SD.

### 4.3. Interfejs UART

UART (ang. *Universal Asynchronous Receiver and Transmitter*) jest urządzeniem umożliwiającym asynchroniczne odbieranie i nadawanie informacji przy pomocy portu szeregowego. Składa się z trzech zasadniczych elementów:

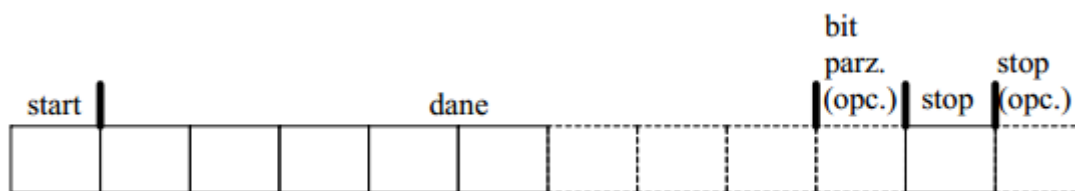
- konwertera równoległo-szeregowego (ang. *parallel to serial*), pozwalającego na przesyłanie danych z komputera,
- konwertera szeregowo-równoległego, umożliwiającego odbiór danych pochodzących z komputera przez urządzenie zewnętrzne,
- bufora danych, służącego do tymczasowego przechowywania danych w przypadku szybkiej transmisji.

UART używany jest często w połączeniu z takim standardami komunikacyjnymi jak RS-232, EIA czy RS-485. Jego uniwersalność polega na tym, że szybkość transmisji oraz format danych są konfigurowalne. Poziomy elektryczne sygnałów elektrycznych oraz ich postać (np. sygnał różnicowy) są przez zewnętrzny układ sterownika. Omawiane urządzenia występują jako układy scalone (bądź ich komponenty) będące składnikami mikrokontrolerów. Istnieją konstrukcje łączące dwa (DUART) lub osiem układów UART w jednej obudowie. Istnieją również chipy pozwalające na komunikację synchroniczną, określane w skrócie jako USART (ang. *Universal Synchronous and Asynchronous Receiver and Transmitter*).

#### 4.3.1. Transmisja

Uniwersalny asynchroniczny odbiornik/nadajnik przesyła bajty danych bit po bicie. UART znajdujący się po drugiej stronie odbiera poszczególne bity łącząc je w bajty. Konwersję z równoległej postaci danych na szeregową uzyskuje się zwykle przy pomocy rejestrów przesuwnych będących składnikami UARTów. Użycie szeregowej transmisji danych zamiast równoległej (bardziej naturalnej dla urządzeń cyfrowych) pozwala na zmniejszenie kosztów oraz problemów związanych z zakłóceniami elektromagnetycznymi.

UART zwykle nie generuje ani nie odbiera bezpośrednio sygnałów odpowiedzialnych za komunikację. W tym celu używa się dodatkowych interfejsów, konwertujących sygnały używane przez UART



Rysunek 4.12: Ramka danych wysyłana lub odbierana przez UART

na właściwe przebiegi odpowiadające za transmisję danych. Często używanymi standardami są RS-232, RS-485 oraz EIA. Komunikacja może odbywać się w trybie *simplex*, *half duplex* oraz *full duplex*.

#### 4.3.2. Ramka danych

Rysunek 4.12 przedstawia typową ramkę używaną w trakcie transmisji przy pomocy układu UART. W stanie bezczynności (gdy nie przesyłane są żadne dane), linia danych jest zwykle w stanie wysokim. Wysyłanie właściwych danych (zwanymi znakiem) sygnalizowane jest bitem startu, tj. przejściem linii danych do stanu niskiego. Znak składa się zazwyczaj z 5 - 8 bitów. W celu zapewnienia kontroli poprawności danych, może być zakończony tzw. bitem parzystości. W celu zasygnalizowania odbiornikowi zakończenia transmisji, przesyłany jest bit stopu, który może być zdublowany.

#### 4.3.3. Odbiór danych

Wszystkie operacje w układzie UART sterowane są sygnałem zegarowym o częstotliwości będącej wielokrotnością częstości używanej do przesyłania danych. Odbiornik próbkuję linię danych w oczekiwaniu na pojawienie się bitu startu. Jeśli linia danych przejdzie ze stanu spoczynkowego do aktywnego na czas wynoszący co najmniej połowę okresu trwania bitu, zostanie to zinterpretowane jako bit startu. W przeciwnym wypadku zarejestrowany impuls zostanie zignorowany. Następnie po upływie niewielkiego czasu następuje dalsze próbkowanie linii danych połączone z wczytywaniem znaku do odbiorczego rejestru przesuwanego. Po wczytaniu określonej liczby bitów, UART ustawia odpowiednią flagę lub generuje sygnał przerwania informujący mikroprocesor o odebraniu danych.

Komunikujące się układy UART nie posiadają wspólnego sygnału zegarowego. Każdy z nich posiada własne zegary, których synchronizacja dokonuje się w oparciu o sygnał na linii danych. Zwykle synchronizacja ma miejsce w chwili zmiany stanu linii danych, o ile zmiana ta może być uznana za ważną. Uproszczone układy synchronizują się w momencie detekcji opadającego zbocza bitu startu, zaś odczyt kolejnych bitów następuje w połowach czasów ich trwania. Takie rozwiązanie działa poprawnie, o ile szybkość transmisji pozwala na poprawny odczyt bitów stopu.

Standardową cechą układów UART jest pobieranie kolejnego znaku, gdy poprzedni został zapisany w rejestrze. Zastosowane w ten sposób podwójne buforowanie pozwala mikroprocesorowi na odczyt poprzedniego znaku, przed odebraniem następnego. Jeśli centralna jednostka przetwarzająca po stronie odbiornika potrzebuje jeszcze więcej czasu na odbiór, można zastosować buforowanie danych z użyciem kolejek FIFO.



#### 4.3.4. Nadawanie danych

Po umieszczeniu znaku w rejestrze przesuwym nadajnika, UART wysyła na linię danych bit startu, kolejne bity danych, bit parzystości (jeśli jest używany) oraz bit (lub dwa bity) stopu. Ponieważ transmisja pojedynczego znaku zajmuje pewien czas, układ nadawczy ustawia odpowiednią flagę zajętości, która informuje mikroprocesor, żeby nie umieszczał kolejnego znaku w rejestrze przesuwym. Zamiast flagi może być wygenerowane odpowiednie przerwanie. W przypadku trybu *full duplex* oba układy UART używają dwóch rejestrów przesuwnych.

#### 4.3.5. Własności

Oba układy UART, zarówno nadawczy jak i odbiorczy muszą mieć zgodne następujące parametry:

- szybkość transmisji danych,
- ilość bitów danych,
- obecność (lub jej brak) bitu parzystości,
- ilość bitów stopu.

W przypadku niezgodności ww. parametrów, układ odbiorczy może ustawić odpowiednią flagę błędu. Typowo porty szeregowie komputerów wykorzystują osiem bitów danych, bit parzystości i jeden bit stopu.

#### 4.3.6. Transmisja w trybie synchronicznym

Jak wspomniano wcześniej istnieją układy typu USART potrafiące także pracować w trybie transmisji synchronicznej. W tym wypadku przebieg sygnału zegarowego jest pozyskiwany na podstawie sygnału na linii danych. Dzięki obecności sygnału synchronizacyjnego niepotrzebne stają się bity startu i stopu, co pozwala na lepsze wykorzystanie linii danych i w efekcie bardziej efektywną komunikację. W przypadku trybu asynchronicznego, gdy nie ma danych do przesłania, linia danych znajduje się w stanie spoczynku. W trybie transmisji synchronicznej konieczne jest przesyłanie specjalnych znaków w celu utrzymania synchronizacji.

#### 4.3.7. Stany błędne

Poniżej opisano możliwe błędy towarzyszące komunikacji z wykorzystaniem urządzeń UART.

- Błąd przepełnienia (ang. *overrun error*) - występuje, gdy odbiorca danych nie może przetworzyć danej przed pojawieniem się kolejnej. Różne urządzenia dysponują różnymi pojemnościami buforów służących do przechowywania odebranych danych. Jednostka przetwarzająca dane musi to robić na tyle szybko, aby zwolnić miejsce dla kolejnych znaków - w przeciwnym razie dochodzi do nadpisania nie przetworzonych jeszcze informacji.

- Błąd niedopełnienia (ang. *underrun error*) - ma miejsce, gdy po wysłaniu danych przez nadajnik, bufor nadawczy jest pusty. W trybie asynchronicznym sytuacja taka traktowana jest raczej jako brak danych do wysłania niż stan błędny. W trybie synchronicznym jest to poważny błąd.
- Błąd ramki (ang. *framing error*) - pojawia się, gdy bity startu i stopu nie są wykrywane. Bit startu sygnalizujący początek znaku stanowi odniesienie dla pozostałych bitów. Jeśli linia danych nie znajdzie się we właściwym stanie, gdy spodziewany jest bit stopu sygnalizowany jest błąd ramki.
- Błąd parzystości (ang. *parity error*) - występuje, gdy liczba jedynek w przesyłanym znaku wraz z bitem parzystości jest liczbą parzystą. Istnienie omawianego bitu w ramce jest opcjonalne.

## 5. Mikrokontrolery *ARM Cortex M3*

Od kilku lat widać duży wzrost popularności mikrokontrolerów *ARM* w systemach wbudowanych. Producenci wprowadzają na rynek coraz tańsze i lepiej wyposażone układy. Zjawisko to zostało zainicjowane w 2003 [11] roku przez firmę *Philips Semiconductor*, która spopularyzowała 32 bitowe wersje mikrokontrolerów *ARM*. W niniejszym rozdziale opisane zostaną, bardzo powszechne dzisiaj, układy z rdzeniami *Cortex*. Szczegółowa uwaga zostanie poświęcona także serii *M3*, gdyż jej dotyczy temat niniejszej pracy.

### 5.1. Firma *ARM* i jej działalność

Firma *ARM* powstała ok. roku 1990 [10] dzięki porozumieniu kilku przedsiębiorstw (m. in. *Apple Computer* oraz *VLSI Technology*) jako *Advanced RISC Machines*. W 1998 roku zmieniono nazwę i obecnie funkcjonuje ona jako *ARM Holdings*. Działalność firmy skupia się na układach cyfrowych, przy czym nie jest to stricte produkcja półprzewodników, lecz projektowanie i opracowywanie tzw. bloków własności intelektualnej IP (ang. *Intellectual Property*). Flagowym produktem przedsiębiorstwa są rdzenie mikrokontrolerów, z których najbardziej popularnymi były *ARM7*, *ARM9* i *ARM11* zastępowane obecnie przez układy serii *Cortex*.

Przeglądając dokumentacje konkretnych urządzeń cyfrowych wykorzystujących rdzenie *ARM* i porównując ją z treścią zawartą na stronie firmy *ARM* można zauważyć rozbieżność [10] nazw. Wynika ona z tego, że producenci sprzętu używają nazwy rdzenia, podczas gdy firma *ARM* odnosi się do nazw architektur (5.1).

### 5.2. Seria *Cortex*

Seria *Cortex* obejmuje trzy podrodziny dostosowane pod kątem konkretnych zastosowań [10]:

Nazwa architektury	Nazwa rdzenia
ARMv4	ARM7
ARMv5	ARM9
ARMv6	ARM11
ARMv7	Cortex

Tablica 5.1: Nazwy rodzin rdzeni *ARM* i odpowiadające im oznaczenia architektur (Źródło: [10])

- *Cortex-Ax* - przeznaczona dla aplikacji pracujących pod kontrolą systemów operacyjnych takich jak *Symbian*, *Linux* oraz *Windows Embedded*, wymagających dużej mocy obliczeniowych, układu zarządzania pamięcią (MMU) lub implementacji maszyny wirtualnej Javy,
- *Cortex-Rx* - przeznaczona dla systemów czasu rzeczywistego, w których krytyczny jest czas odpowiedzi na zdarzenia (np. układy bezpieczeństwa biernego w samochodach),
- *Cortex-Mx* - przeznaczona dla zastosowań przemysłowych i konsumenckich; stanowi próbę osiągnięcia kompromisu między dużą wydajnością i niskimi kosztami.

Litera *x* w każdym z powyższych przypadków oznacza liczbę, precyzującą wersję rdzenia.

Oprócz wspomnianych różnic, rdzenie *Cortex-Mx* obsługują wyłącznie rozkazy z listy określonej w literaturze [10] jako *Thumb-2*, w odróżnieniu od pozostałych podrodzin *Cortex*.

Najprostszym i najbardziej energooszczędnym rdzeniem z serii *M*, jest układ oznaczony jako *M0*. W założeniach miał być jednostką 32 bitową stanowiącą poważną konkurencję dla mikrokomputerów ośmio i szesnastobitowych w mniej zaawansowanych zastosowaniach. Rdzeń *Cortex-M1* projektowano z myślą o zastosowaniu w układach logiki programowalnej (FPGA). Linia *M3* zostanie szczegółowiej opisana w kolejnym paragrafie. Najsilniejszym, pod względem mocy i możliwości obliczeniowych, przedstawicielem podrodziny *Cortex-Mx* są układy sygnowane jako *M4*. Ich zastosowaniem mają być w założeniach obszary podobne do tych, w których używa się procesorów DSP, a więc cyfrowe przetwarzanie sygnałów. W tym celu mikrokontrolery *Cortex-M4* wyposażono w sprzętowe bloki (np. koprocesory) pozwalające realizować obliczenia charakterystyczne dla procesorów sygnałowych.

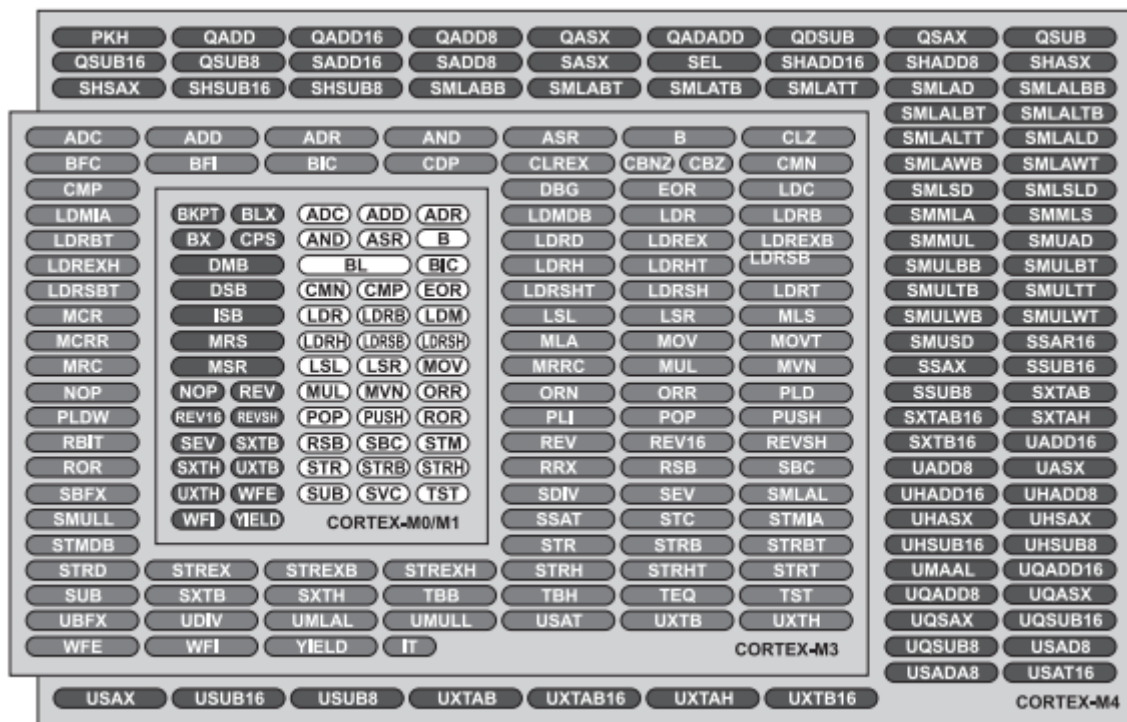
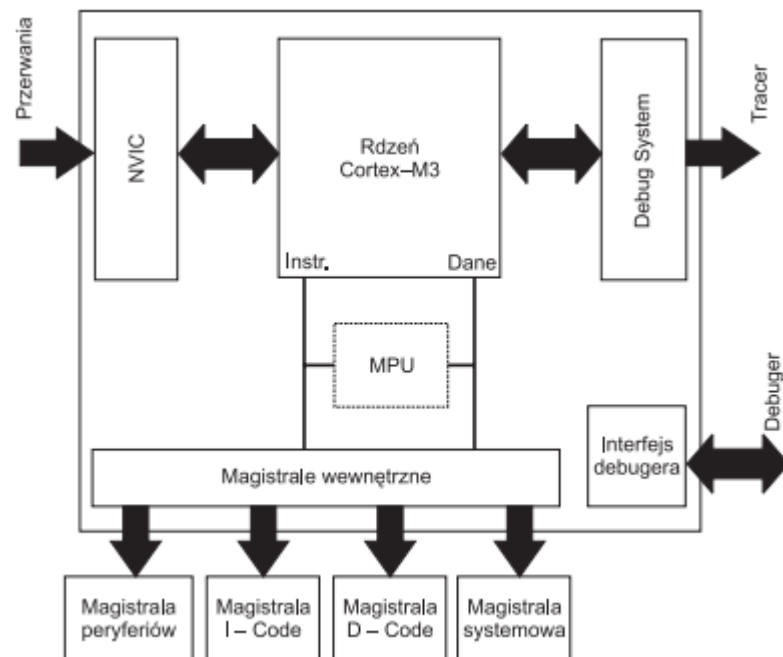
Z punktu widzenia programisty najistotniejszą różnicą między poszczególnymi rdzeniami *Cortex-Mx* jest lista rozkazów (rysunek 5.1).

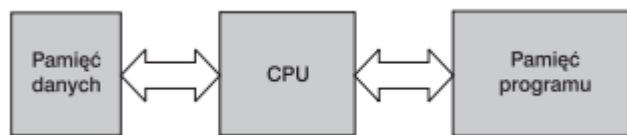
### 5.3. Rdzeń Cortex-M3

Na rysunku 5.2 przedstawiono w sposób uproszczony budowę rdzenia *Cortex-M3*. Obsługuje on listę rozkazów *Thumb-2*, która pozwala realizować operacje zarówno na liczbach 16- jak i 32-bitowych. Dzięki temu uzyskuje się większą (w porównaniu do rozkazów *ARM*) gęstość upakowania poleceń, mniejsze zapotrzebowanie na pamięć programu (flash) oraz szybsze wykonywanie rozkazów w stosunku do programu zapisanego przy użyciu listy *Thumb*. Pisząc program, nie ma zatem potrzeby przełączania się między 32-bitowym trybem *ARM* i 16-bitowym trybem *Thumb*.

Rdzeń *Cortex-M3* jest zaprojektowany zgodnie z pełną architekturą harwardzką. Oznacza to rozdzielenie pamięci programu i danych oraz magistrali danych i rozkazów (rysunek 5.3). Zaletą takiej architektury jest możliwość dostępu do pamięci danych i programu w tym samym czasie. Ponieważ jednak przestrzeń adresowa dla obu z nich jest wspólna, nie mogą one wykorzystywać tej przestrzeni w pełni, lecz tylko w części. Teoretycznie architektura harwardzka pozwala zmniejszyć wpływ dostępu do danych na szybkość wykonywania się programu. W praktyce stosunkowo powolne taktowanie pamięci flash (do 24 Mhz [10]) może spowolnić wykonywanie programu.

Mikrokontrolery z rdzeniem *Cortex-M3* wspierają dostęp do danych z uwzględnieniem dwóch sposobów ułożenia bajtów, tj. *little endian* i *big endian*. W pierwszym przypadku młodsze bajty zapisywane

Rysunek 5.1: Rozkazy wykonywane przez rdzenie rodziny *Cortex-Mx* (Źródło: [10])Rysunek 5.2: Schemat rdzenia *Cortex-M3* (Źródło: [10])



Rysunek 5.3: Architektura Harvard (Źródło: [10])

są pod niższymi adresami, zaś starsze pod wyższymi. W przypadku big endian jest na odwrót. Dzięki temu można dostosować i zoptymalizować projektowaną aplikację pod kontem własnych potrzeb. Przykładowo wykorzystanie w programie komunikacji sieciowej przy pomocy interfejsu Ethernet, nie będzie wymagało zmiany kolejności bajtów na big endian.

Bardziej złożone rdzenie *Cortex-M3* posiadają jednostkę ochrony pamięci MPU (ang. *memory protection unit*). Ponadto architektura Cortex przewiduje obecność bloków debugowania z obsługą pułapek (ang. *breakpoint*) oraz punktów podglądowych (ang. *watchpoint*).

Dla rdzeni rodziny *Cortex* przewidziane są dwa tryby pracy - uprzywilejowany oraz użytkownika. W pierwszym z nich aplikacja ma pełny dostęp do wszystkich zasobów rdzenia. W trybie użytkownika, niektóre zasoby nie mogą być używane przez aplikacje. Takie podejście pozwala na tworzenie bezpiecznych programów, w tym systemów operacyjnych, których jądra pracują w trybie uprzywilejowanym, zaś programy użytkownika w trybie nieuprzywilejowanym.

## 5.4. Rejestry ogólnego przeznaczenia

Rdzeń *Cortex M3* posiada 16 rejestrów podstawowych (R0 - R15), z których trzynaście (R0 - R12) stanowią rejestry ogólnego przeznaczenia (rysunek 5.4). Większość rozkazów 16-bitowych może używać jedynie rejestrów R0 - R7. Rejestr R13 pełni funkcje wskaźnika stosu SP (ang. *stack pointer*) i w rzeczywistości składa się z dwóch rejestrów bankowanych [10], z których w danej chwili widoczny jest jeden:

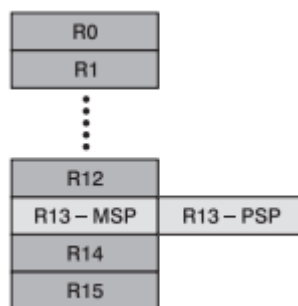
- głównego wskaźnika stosu MSP (ang. *main stack pointer*) - używanego domyślnie przez przerwa-  
nia i jądra systemów operacyjnych pracujących w trybie uprzywilejowanym,
- procesowego wskaźnika stosu PSP (ang. *process stack pointer*) - używanego przez programy użyt-  
kownika uruchamiane pod kontrolą systemu operacyjnego.

Wykorzystanie dwóch stosów jest korzystne dla tworzenia bezpiecznych aplikacji, gdyż uniemożliwia dostęp do systemowego stosu, co groziłoby naruszeniem stabilności systemu.

Pozostałe dwa rejestry podstawowe to:

- R14 - zawierający adres powrotu z podprocedury,
- R15 - licznik programu PC (ang. *program counter*) znany też jako licznik instrukcji IC (ang. *instruction counter*) - zawiera adres aktualnie wykonywanego rozkazu.

Oprócz wymienionych rejestrów podstawowych, rdzeń *Cortex-M3* zawiera także:



Rysunek 5.4: Rejestry podstawowe rdzenia *Cortex-M3* (Źródło: [10])

- rejestr stanu (ang. *program status register*) - zawierający tzw. flagi,
- rejestr maski przerwań (ang. *interrupt mask register*),
- rejestr sterowania (ang. *control register*).

Rejestry służą do sterowania rdzeniem oraz do kontroli przebiegu wykonywania programu. Mogą być modyfikowane jedynie za pomocą specjalnych rozkazów i nie może to mieć miejsca w trakcie normalnej pracy mikrokontrolera [10].

## 5.5. Przestrzeń adresowa

Rdzeń Cortex-M3 jest w stanie zaadresować przestrzeń 4 GB pamięci [9]. Obejmuje ona tzw. segmenty, m. in. pamięć programu, SRAM, zewnętrzną pamięć RAM. Odpowiednią mapę pamięci pokazano na rysunku 5.5.

## 5.6. Obszary o dostępie bitowym

Przestrzeń adresowa rdzenia *Cortex-M3* zawiera dwa obszary o dostępie bitowym (określanych jako *bit-band*) - jeden w regionie pamięci RAM, a drugi w regionie urządzeń peryferyjnych. W pierwszym przypadku obszar rozpoczyna się od adresu 0x20000000, zaś drugi od 0x40000000 [źródło]. Takie rozwiązanie pozwala na optymalną pracę rdzenia. Normalnie zaby zmienić jeden bit, należy odczytać wartość z właściwej komórki pamięci, ustawić w niej wartość odpowiedniego bitu, a następnie tak zmodyfikowaną wartość zapisać w pierwotnej komórce. Dzięki wykorzystaniu dostępu bitowego, ten sam rezultat można uzyskać dzięki dostępowi do właściwej komórki pamięci w obszarze mapowania bitów regionu *bit-band*. Jej odpowiedni adres można wyznaczyć dzięki poniższemu formułom [4]:

$$bit\_word\_offset = (byte\_offset \cdot 32) + (bit\_number \cdot 4) \quad (5.1)$$

$$bit\_word\_addr = bit\_band\_base + bit\_word\_offset \quad (5.2)$$

gdzie:

Obszar systemowy 0,5 GB	0xFFFFFFFF
Prywatne peryferia (NVIC, MPU itp.)	0xE0000000 0xDFFFFFFF
Urządzenia zewnętrzne 1 GB	0xA0000000 0x9FFFFFFF
Zewnętrzna RAM 1 GB	0x60000000 0x5FFFFFFF
Peryferia 0,5 GB	0x40000000 0x3FFFFFFF
SRAM 0,5 GB	0x20000000 0x1FFFFFFF
Program 0,5 GB	0x00000000

Rysunek 5.5: Mapa pamięci rdzenia Cortex-M3 (Źródło: [10])

- *bit\_word\_offset* - pozycja bitu w obszarze pamięci *bit-band*,
- *byte\_offset* - numer bajtu w obszarze *bit-band* zawierającego żądany bit,
- *bit\_number* - pozycja bitu w bajcie,
- *bit\_word\_addr* - adres bajtu mapującego żądany bit w obszarze pamięci mapowania bitów (*alias memory region*),
- *bit\_band\_base* - adres początku regionu mapowania bitów (dla obszaru pamięci RAM wynosi on 0x22000000 [10]).

Przykładowo, dla piątego bitu w bajcie o adresie 0x20000007, wymienione powyżej parametry wynoszą:

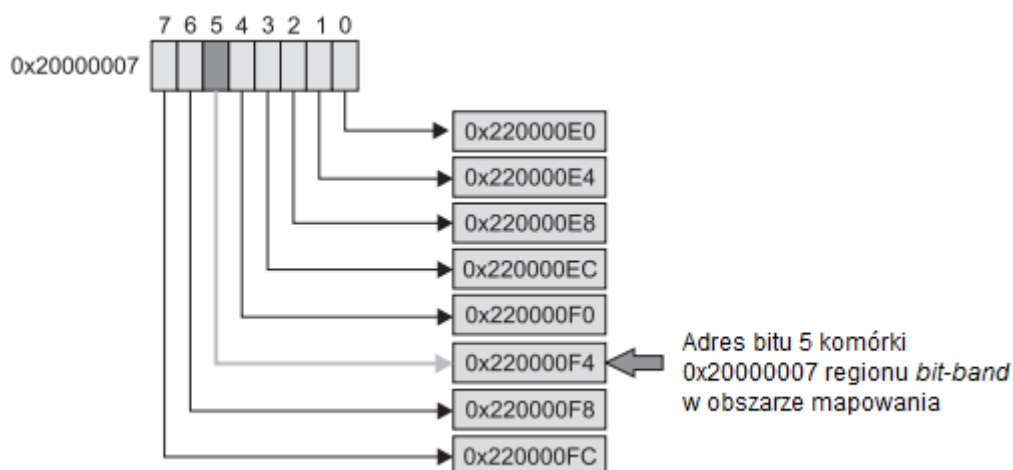
- *bit\_number* = 5;
- *byte\_offset* = 7 (0x20000007 - 0x20000000),
- *bit\_word\_offset* =  $7 \cdot 32 + 5 \cdot 4 = 244 = 0x000000F4$ ,
- *bit\_word\_addr* =  $0x22000000 + 0x000000F4 = 0x220000F4$ .

Aby skasować lub ustawić określony bit w regionie bit-band, należy wpisać odpowiednio 0 lub 1 do właściwej komórki w obszarze mapowania.

## 5.7. Kontroler przerwań NVIC

Aby mikroprocesor (lub centralna jednostka przetwarzająca mikrokontrolera) mógł się komunikować z urządzeniami peryferyjnymi poprzez układy wejścia/wyjścia, stosuje się dwie techniki:





Rysunek 5.6: Mapowanie piątego bitu komórki o adresie 0x20000007 z regionu bit-band (Źródło: [10])

Numer	Wyjątek	Priorytet
1	Reset	–3
2	NMI	–2
3	Hard Fault	–1
4	MemManage Fault	Programowany
5	Bus Fault	Programowany
6	Usage Fault	Programowany
7...10	Reserved	–
11	SVCall	Programowany
12	Debug Monitor	Programowany
13	Reserved	–
14	PendSV	Programowany
15	SYSTICK	Programowany
16...255	External Interrupt	Programowany

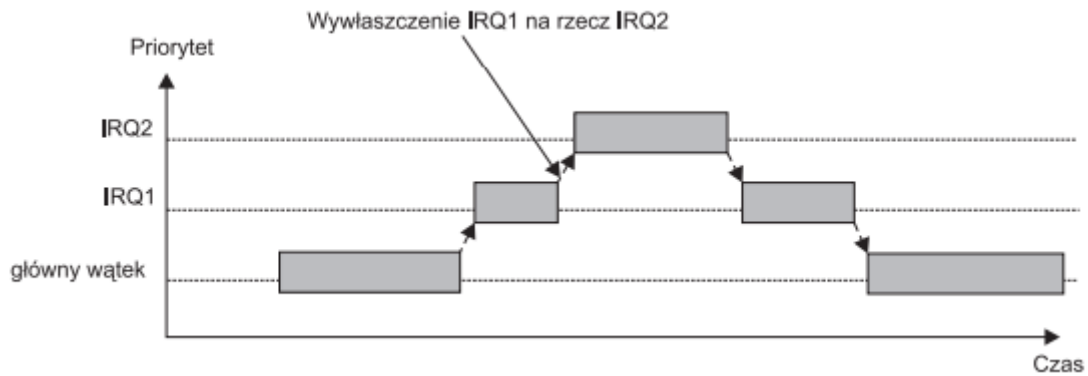
Rysunek 5.7: Przerwania obsługiwane przez rdzeń *Cortex-M3* i ich priorytety (Źródło: [10])

- przeglądanie (ang. *polling*) rejestrów układów wejścia/wyjścia,
- przerwania.

Konstrukcje dzisiejszych mikrokomputerów są optymalizowane pod kątem obsługi przerwań. Rdzeń *Cortex-M3* posiada wbudowany kontroler przerwań *NVIC* (ang. *nested vectored interrupt controller*). Pozwala on na obsługę piętnastu przerwań (wyjątków) systemowych i 240 przerwań zewnętrznych. Faktyczna liczba obsługiwanych przerwań zależy od konkretnego mikrokontrolera. Większość wyjątków systemowych i wszystkie przerwania wewnętrzne mogą mieć ustalany programowo priorytet [10]. Niektóre priorytety zgrupowano (w kolejności malejącej) w tabeli na rysunku 5.7. Jak widać, zdarzeniem o najwyższym priorytecie w systemie jest zerowanie układu.

Możliwości kontrolera *NVIC* [10]:

1. Obsługa przerwań zagnieżdżonych (zwanymi też wielopoziomowymi) - jeśli w trakcie procedury obsługi przerwania *ISR* (ang. *interrupt service routine*) wystąpi przerwanie o wyższym poziomie,



Rysunek 5.8: Wielopoziomowa obsługa przerwań (Źródło: [10])

obsługa bieżącego jest przerywana, następuje skok do procedury obsługi przerwania o wyższym poziomie, a po jej zakończeniu powrót do obsługi zdarzenia poprzedniego (rysunek 5.8).

2. Sprzętowa obsługa wektorów przerwań - adres funkcji obsługi przerwania jest pobierany z wektora w pamięci i nie potrzeby jego programowego wyznaczania, dzięki czemu czas obsługi zdarzenia jest mniejszy.
3. Dynamiczna zmiana priorytetów przerwań - pozwala zmieniać programowo priorytet przerwania podczas wykonywania programu. Nie można jednak zmienić priorytetu przerwania przed wyjściem z procedury jego obsługi, dzięki czemu unika się sytuacji wielokrotnego obsłużenia tego samego zdarzenia podczas zmiany priorytetu.
4. Zoptymalizowane opóźnienia czasowe obsługi przerwań - uzyskano to m. in. dzięki automatycznemu zapisywaniu i odzyskiwaniu tzw. kontekstu zadania, czyli wartości kluczowych dla niego rejestrów. Inne sposoby skrócenia opóźnień w obsłudze przerwań (np. *Tail-Chaining*, przerywanie operacji POP) opisano w literaturze [10].
5. Maskowanie przerwań - wszystkie przerwania mogą być maskowane na podstawie priorytetów lub maskowane całkowicie poprzez wpisy do właściwych rejestrów maskujących.

## 5.8. Lista instrukcji *Thumb-2*

Jedną z największych zalet rdzenia *Cortex-M3* z punktu widzenia programisty, jest możliwość wykorzystania rozkazów operujących na danych 16- i 32-bitowych bez konieczności dodatkowych zabiegów (przełączanie się między trybami). Dzięki temu uzyskuje się zmniejszenie zużycia pamięci programu przy jednoczesnym wzroście szybkości wykonywania programu [10].

W celu jeszcze lepszego wykorzystania ww. cech rdzenia *Cortex-M3*, stworzono odmianę asemblera oznaczaną w skrócie jako UAL (ang. *unified assembly language*). Pozwala ona w szybszy i czytelniejszy sposób wykorzystywać w programie rozkazy 16- i 32-bitowe.

Zestaw rozkazów *Thumb-2* zawiera operacje, które mogą być wykonywane zarówno jako 16- jak i 32-bitowe. Przykładem jest dodawanie liczby do zawartości rejestru [10]:

ADDS R0, #5 ;Domyślnie zostanie użyta instrukcja 16-bitowa  
ADDS.N R0, #5 ;Jawne użycie rozkazu 16-bitowego (N = Narrow)  
ADDS.W R0, #5 ;Jawne użycie rozkazu 32-bitowego Thumb-2 (W = Wide)

Użycie przyrostka *W* lub *N* po kropce za właściwą nazwą rozkazu pozwala wyspecyfikować długość rozkazu. Mnemoniki pozbawione sufiksu są zwykle tłumaczone na instrukcje 16-bitowe.

Jak widać programowanie rdzenia *Cortex-M3* w języku assemblera niesie ze sobą spore możliwości optymalizacji. Wadą tak niskopoziomowego podejścia jest czasochłonność, duży nakład pracy i często także kiepska czytelność kodu. Decyzje o pisaniu aplikacji (bądź jej fragmentów) w assemblerze powinny być wynikiem kompromisu między względami wydajnościowymi oraz ekonomicznymi.

## Bibliografia

- [1] [http://en.wikipedia.org/wiki/File:SPI\\_timing\\_diagram2.svg](http://en.wikipedia.org/wiki/File:SPI_timing_diagram2.svg).
- [2] [http://en.wikipedia.org/wiki/File:SPI\\_three\\_slaves.svg](http://en.wikipedia.org/wiki/File:SPI_three_slaves.svg).
- [3] [http://en.wikipedia.org/wiki/File:SPI\\_three\\_slaves\\_daisy\\_chained.svg](http://en.wikipedia.org/wiki/File:SPI_three_slaves_daisy_chained.svg).
- [4] <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0337h/Behcjiic.html>.
- [5] [http://pl.wikipedia.org/wiki/Plik:Gyroscope\\_operation.gif](http://pl.wikipedia.org/wiki/Plik:Gyroscope_operation.gif).
- [6] <http://kepler.am.gdynia.pl/~karudz/zyroskop.pdf>.
- [7] <http://elektronikab2b.pl/technika/12098-zyroskopy-i-akcelerometry-mems-w-e.UuDzRhCta02>.
- [8] W. Gawędzki. *Pomiary elektryczne wielkości nieelektrycznych*. Wydawnictwa AGH, Kraków, 2010.
- [9] R. Krzyżanowski. *Układy mikroprocesorowe*. Wydawnictwo Naukowe PWN, Warszawa, 2007.
- [10] K. Paprocki. *Mikrokontrolery STM32 w praktyce*. Wydawnictwo BTC, Legionowo, 2009.
- [11] P. Zbysiński. Żarty się kończą. Nowe mikrokontrolery z rdzeniem Cortex M3 w ofercie NXP Semiconductor. *Elektronika Praktyczna*, wrzesień 2008.

## **Dodatek A**

Tu, trzeba zamieszczać treść dodatkową np. fragmenty kodu aplikacji itp.