

Python w Elektronicznej Sieci #12: Wprowadzenie do elektroniki cyfrowej

Projekt „Matematyka dla Ciekawych Świata”,

Robert Ryszard Paciorek

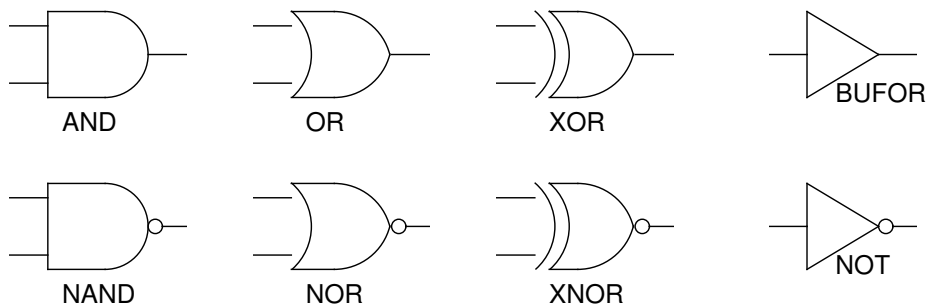
<rrp@opcode.eu.org>

2020-05-19

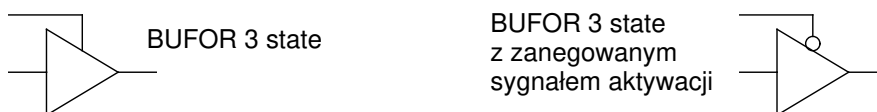
1 Bramki

Bramki są układami elektronicznymi realizującymi podstawowe, opisane powyżej funkcje logiczne. Obok zostały przedstawione podstawowe symbole poszczególnych bramek w wariantach dwu wejściowym, spotkać się można także z symbolami z zanegowanymi wejściami - w takiej konwencji np. bramka AND reprezentowana jest przez NOR z zanegowanymi wejściami. Bramki (z wyjątkiem buforów oraz bramki NOT), mogą występować także w wariantach wielo-wejściowych (ze względu na łączność podstawowych operacji nie ma wątpliwości co do wyniku jaki powinna dawać np. 8 wejściowa bramka OR). Na ogół w pojedynczym układzie scalonym znajduje się kilka jednakowych bramek.

Zobacz symulację działania różnych bramek logicznych: <http://ln.opcode.eu.org/bramki> (H oznacza stan wysoki, czyli prawdę, L stan niski czyli fałsz, klikając na H/L przy wejściach można zmieniać ich stan).



bramki (AND, OR, XOR, NAND, NOR, XNOR) mogą występować także w wariantach wielo-wejściowych



w wariantach 3 stanowych mogą występować także wszystkie pozostałe elementy

1.1 trój-stanowe

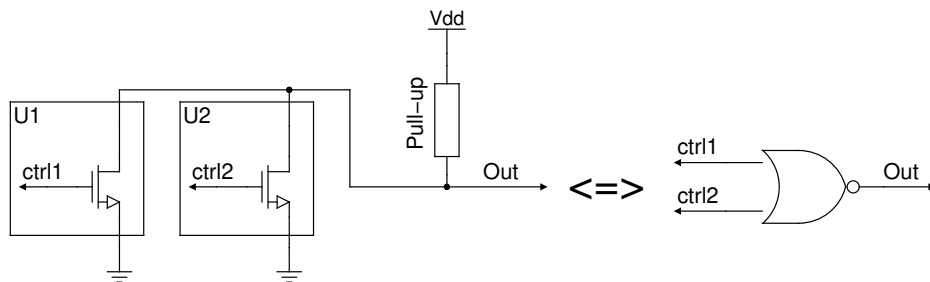
Typowa bramka wymusza (w sposób silny) na swoim wyjściu stan wysoki lub niski, co uniemożliwia bezpośrednie łączenie wyjść bramek. Bramki trój-stanowe mają możliwość skonfigurowania wyjścia w stan *wysokiej impedancji* czyli nie wymuszania żadnej jego wartości. Sterowanie załączeniem bądź wyłączeniem wyjścia (przełączeniem w stan wysokiej impedancji) odbywa się przy pomocy zewnętrznego sygnału sterującego "output enabled" ("OE"), sygnał ten może występować w postaci prostej i zanegowanej. Pozwala to na podłączanie do jednej linii wielu bramek i decydowaniu która z nich będzie nią sterować.

1.2 open collector / drain

Są kolejnym rodzajem bramek których wyjścia można podłączać do wspólnej linii. Układy te posiadają wyjście w postaci tranzystora zwierającego linię wyjściową do masy, z tego względu samodzielnie zapewniają jedynie stan niski wyjścia (są w stanie wymusić stan niski, ale nie mają możliwości wymuszenia stanu wysokiego).

Stan wysoki musi zostać zapewniony zewnętrznym rezystorem podciągającym. Pozwala to stosować na takiej linii inny poziom stanu wysokiego niż na wejściach takiej bramki oraz pozwala na sterowanie wspólną linią przez wiele bramek (czyli łączenie wyjść bramek, jednak w odróżnieniu od bramek trój-stanowych nie wymaga dodatkowych sygnałów sterujących).

Na schemacie obok przedstawiono dwa układy (U1 i U2) typu open-drain sterujące wspólną linią wyjściową w układzie *suma na drucie*. Jeżeli jeden z podłączonych do linii układów będzie miał wewnętrzne wyjście ("ctrlX") w stanie wysokim to jego wyjście OC będzie zwarte do masy (negacja na tranzystorze N-MOS lub NPN), wtedy też cała linia będzie w stanie niskim.



Zobacz symulację linii z bramkami trójstanowymi (stan wysokiej impedancji symulowany za pomocą przełącznika) oraz linii open-kolector: <http://ln.opcode.eu.org/ster>

1.3 budowa wewnętrzna

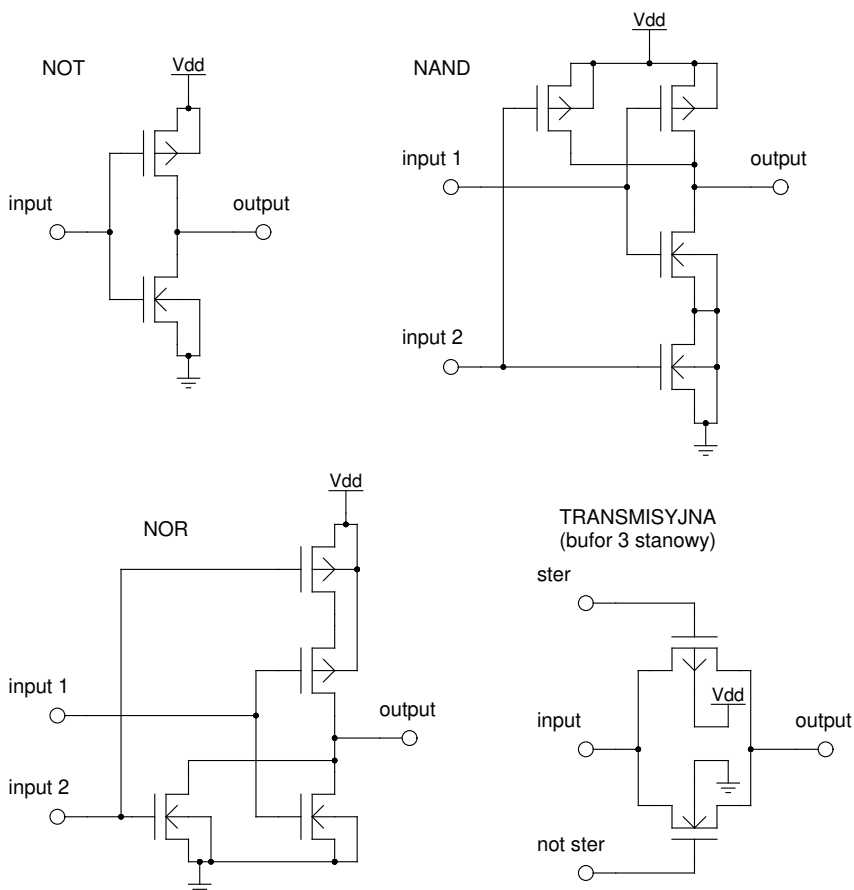
Przedstawiony powyżej układ sumy na drucie jest bardzo prostą (jedno tranzystorową) realizacją bramki logicznej realizującą funkcję logiczną NOT OR (z punktu widzenia wejść *ctrl1* i *ctrl2* oraz wyjścia *Out*). W podobny sposób można zrealizować bramkę AND (negując wejścia, np. przy pomocy jednego tranzystora). Jeszcze bardziej uproszczoną realizację można uzyskać stosując diody pozwalające na wpływanie prądu do węzła (funkcja OR) lub wypływanie z niego (funkcja AND).

Po prawej przedstawione zostały schematy ideowe inwertera, dwóch podstawowych bramek (NOR i NAND) oraz bramki transmisyjnej (bufora 3 stanowego) w technologii CMOS.

Działanie tych bramek (za wyjątkiem transmisyjnej) polega na otwieraniu tranzystorów podłączonych do napięcia które chcemy otrzymać na wyjściu, a zamykaniu prowadzących do napięcia przeciwnego. W szczególności bramka NOT stanowi półmostek H pomiędzy stanem wysokim a stanem niskim.

Dzięki zastosowaniu tranzystorów PMOS polaryzowanych Vdd oraz NMOS polaryzowanych GND obie gałęzie operują na tym samym sygnale wejściowym (nie jest wymagana jego negacja). Szeregowe łączenie tranzystorów zapewnia że należy otworzyć oba aby otworzyć daną drogę, a równoległe że otwarcie danej drogi powodowane jest otwarciem pojedynczego tranzystora. Dzięki zastosowaniu technologii MOS i podłączaniu wejść bramki tylko do bramek tranzystorów wejścia praktycznie nie pobierają prądu (istotnym wyjątkiem jest chwila zmiany sygnału).

Działanie bramki transmisyjnej polega na przepuszczaniu lub nie (w zależności od stanu wejścia sterującego) sygnału z wejścia na wyjście. Bramka taka nie regeneruje sygnału. Ponadto w uproszczonym (jedno tranzystorowym) rozwiązaniu prowadzi ona do degradacji sygnału wartość w przybliżeniu równą



napięciu progowemu tranzystora. Dlatego też na ogół występuje wraz z bramką NOT (bufor 3 stanowy z negacją) lub dwiema szeregowo połączonymi bramkami NOT (bufor 3 stanowy bez negacji).

Zobacz symulację budowy bramek: NOT (<http://ln.opcode.eu.org/not>), NAND (<http://ln.opcode.eu.org/nand>) i NOR (<http://ln.opcode.eu.org/nor>).

2 Przerzutniki i rejestry

2.1 przerzutniki i ich budowa

RS Flip-flop (RS Latch) jest podstawowym układem służącym do zapamiętania jednego bitu informacji. Posiada on dwa wejścia (set i reset) i dwa wyjścia (Q i NOT Q), wejścia mogą reagować na stan wysoki (oznaczane jako S i R) lub niski (oznaczane jako wejścia zanegowane \bar{S} i \bar{R}), jedno z wyjść może być jedynie wewnętrzne (nie wyprowadzone na zewnątrz układu). Podanie stanu wysokiego na wejście S (niskiego na \bar{S}) powoduje wystawienie stanu wysokiego na wyjściu Q, a podanie stanu wysokiego na wejście R (niskiego na \bar{R}) powoduje wystawienie stanu niskiego na wyjściu Q. Stan na wyjściu Q nie zmienia się po zmianie wejść S i R na stan niski (zostaje zapamiętany).

Zobacz i przeanalizuj symulację działania zatrzasku RS: <http://ln.opcode.eu.org/rs> z wejściami zanegowanymi.

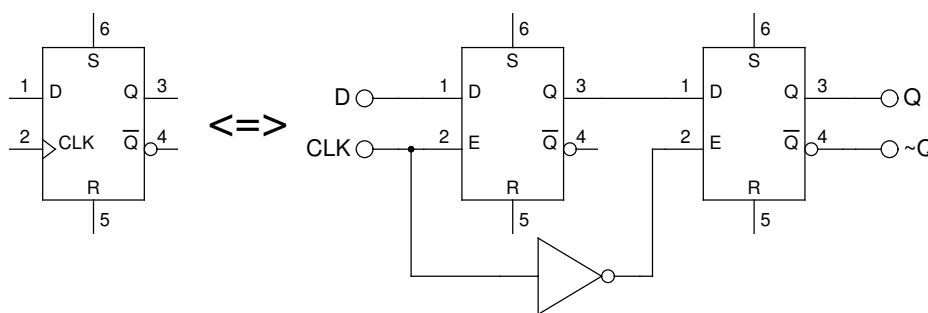
2.2 zatrzask a przerzutnik

Zatrzask jest elementem reagującym na poziom sygnału na wejściu "enable" (E). W przypadku nie zanegowanego wejścia E, jeżeli jest ono w stanie wysokim sygnał na wyjściach (Q i NOT Q) jest funkcją sygnałów wejściowych, natomiast stan niski wejścia E blokuje zmianę sygnału wyjściowego (zostaje on zapamiętany).

Przerzutnik jest elementem reagującym na zbocze sygnału na wejściu "clock" (CLK). W zależności od konstrukcji może reagować na zbocze narastające, opadające albo na oba (wtedy na jednym realizuje odczyt wejść a na drugim zmianę stanu wyjść).

2.3 zatrzask i przerzutnik D

Posiada jedno wejście sygnałowe "data" (D) oraz wejście "enable" (E) w przypadku zatrzasku lub wejście "clock" (CLK) w przypadku przerzutnika. Może także posiadać asynchroniczne (niezależne od stanu wejścia E / CLK) wejścia reset i set (zanegowane lub proste). Obniżenie sygnału E lub zbocze sygnału CLK powodują zapamiętanie (i wystawienie na wyjściu Q) stanu wejścia D.



Zobacz symulację zatrzasku typu D zbudowanego z bramek NAND: <http://ln.opcode.eu.org/zatrzask> (możesz zmieniać stan wejścia D klikając na nie, zegar zmienia się automatycznie). Zwróć uwagę iż przy wysokim stanie sygnału zegara (enable) stan wyjścia odpowiada stanowi wejścia (zatrzask jest przezroczysty), natomiast przy niskim stanie zegara wyjście nie reaguje na zmiany stanu wejścia i znajduje się w takim stanie w jakim było wejście w chwili opadającego zbocza sygnału zegarowego.

Zobacz symulację przerzutnika D złożonego z dwóch zatrzasków: <http://ln.opcode.eu.org/przerzutnik>. Zauważ że w żadnej fazie sygnału zegarowego nie jest on przezroczysty (wyjście Q nie zależy od obecnego stanu wejścia D). Zwróć uwagę że sygnał wejściowy zostanie zapamiętany i wystawiony na wyjście Q w momencie opadającego zbocza sygnału zegarowego.

2.4 rejestry

Mianem rejestru n -bitowego określa się zespół n przerzutników (rzadziej zatrzasków), często z u wspólnym sterowaniem (sygnały clock, set, reset, etc) służący do zapamiętania n -bitowej wartości (liczby). W zależności od sposobu zapisu i odczytu można wyróżnić:

2.4.1 rejestry równoległe

Posiadają taką samą liczbę wejść jak i wyjść, sygnał na i -tym wyjściu jest bezpośrednio powiązany z sygnałem z i -tego wejścia (jest sygnałem zapamiętanym z tego wejścia).

Zobacz symulację rejestru równoległego zbudowanego z przerzutników typu D: <http://ln.opcode.eu.org/rejestr1> (stan wszystkich wejść zostanie zapamiętany i przepisany na wyjścia w chwili narastającego zbocza zegara).

2.4.2 rejestry szeregowe serial-input

Z kolejnymi sygnałami zegarowymi odczytywany jest stan wejścia szeregowego, a stan poprzedni przenoszony jest do kolejnego przerzutnika w ramach rejestru. W ten sposób po n cyklach zegara n -bitowy rejestr ma zapisaną nową zawartość. Często posiada zespolony z nim rejestr równoległy zapobiegający zmianie stanu wyjść w trakcie ładowania danych z wejścia szeregowego przepisanie danych z rejestru przesuwającego do rejestru odpowiedzialnego za sterowanie wyjściami sterowane jest osobnym sygnałem zegarowym.

Zobacz symulację prostego rejestru z wejściem szeregowym (bez zatrzasku/rejestru wyjściowego): <http://ln.opcode.eu.org/rejestr2>. Zauważ że stan wyjść zmienia się na bieżąco w trakcie szeregowego wpisywania wartości do rejestru.

Zobacz symulację rejestru z wejściem szeregowym i rejestrem wyjściowym: <http://ln.opcode.eu.org/rejestr3>. Zauważ, że stan wyjść zmienia się na skutek osobnego sygnału, który może zostać wygenerowany po zakończeniu szeregowego zapisu do rejestru.

2.4.3 rejestry szeregowe parallel-input serial-output

Z kolejnymi sygnałami zegarowymi na wyjście szeregowe wystawiany jest stan kolejnego z rejestrów wejściowych. Wariant asynchroniczny posiada osobny sygnał powodujący odczyt wejść do rejestru (sygnał działa jak "enable" w zatrzaskach). Wariant synchroniczny posiada sygnał decydujący o tym czy na zboczu zegara dokonywany jest odczyt wejść czy też przesuwanie zawartości rejestru umożliwiające odczyt z wyjścia szeregowego.

2.4.4 liczniki

Z kolejnymi sygnałami zegarowymi zwiększana jest o jeden wartość rejestru. Prostszy w budowie licznik asynchroniczny ma większe (i w dodatku rosnące wraz z bitowością licznika) ograniczenia dotyczące szybkości zliczania od licznika synchronicznego, ze względu na opóźnienie z jakim dochodzi zliczany sygnał (CLK) do kolejnych stopni licznika.

3 Transmisja - sterowanie linią

3.1 bufory

Bufor jest to układ przekazujący sygnał logiczny z wejścia na wyjście. Bufor może służyć do:

- regeneracji sygnału,
- uniemożliwieniu wprowadzenia sygnału z jego strony wyjściowej na wejściową,
- decydowania o jego przepuszczeniu lub nie (trój-stanowy),
- decydowania o kierunku przepuszczenia sygnału (dwa trój-stanowe albo trój-stanowy dwukierunkowy),

- konwersji na linię open-collector / open-drain,
- negacji sygnału (niektóre bufony realizują funkcję NOT).

3.2 enkodery

Enkoder "n to m" jest to układ o n wejściach, który na swoim m bitowym wyjściu wystawia numer (typowo) wejścia o najwyższym numerze, które znajduje się w stanie niskim. Możliwe są też warianty wystawiające numer pierwszego (a nie ostatniego) w kolejności wejścia lub wybierające wejście ze stanem wysokim.

Jako że wejścia numerowane są zazwyczaj od zera do 2m to układ najczęściej posiada dodatkowe wyjście informujące że którekolwiek z wejść jest w stanie aktywnym. Typowo numer wystawiany jest w postaci NKB, ale możliwe są inne kodowania.

Układ pozwala na redukcję ilości wejść potrzebnych do obsługi n-bitowego sygnału w którym tylko jeden bit może być ustawiony lub w którym można pozwolić sobie na obsługę kolejnych linii z kasowaniem ich bitu (np. wektor przerwań z priorytetyzacją).

3.3 dekodery

Dekoder "m to n" jest układem o działaniu przeciwnym do enkodera. Aktywuje on wyjście o numerze odpowiadającym wartości na m-bitowym wejściu adresowym. Typowo posiada także wejście zezwolenia na aktywację wyjść, które może zostać użyte do podłączenia informacji że którekolwiek z wejść enkodera było w stanie aktywnym lub do podłączenia sygnału danych z multipleksowanej linii celem jej demultipleksacji.

3.4 (de)multipleksery cyfrowy

Multiplekser cyfrowy (jednokierunkowy) na wyjście kopiuje stan wskazanego (poprzez adres podany na wejście adresowe) wejścia. W przypadku braku sygnału "enable" w zależności od rozwiązania wyjście pozostanie w stanie niskim lub wysokiej impedancji.

Demultiplekser cyfrowy (jednokierunkowy) to zazwyczaj układ dekodera w którym na wejście enabled podany jest sygnał z multipleksowanej linii. Nie wybrane wyjścia pozostają w stanie niskim lub wysokim (zależnie od użycia nieodwracającego lub odwracającego dekodera). Cyfrowe demultipleksery z 3 stanowym wyjściem są rzadkością. Demultipleksację można rozwiązać także przy pomocy odpowiednio sterowanych (np. z dekodera adresu) buforów trój-stanowych lub dwu-wejściowych multiplekserów.

3.5 (de)multipleksery analogowy

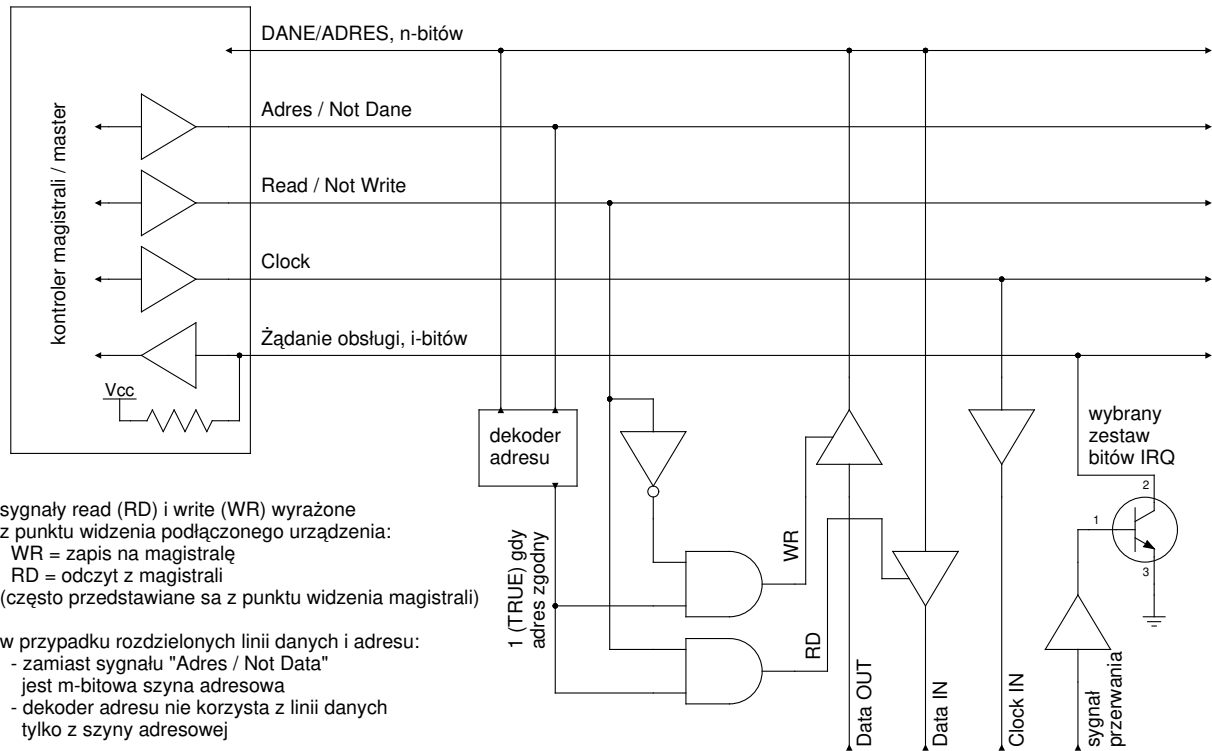
Multiplekser analogowy (dwukierunkowy) działa na zasadzie przełącznika łączącego wskazane wejście z wyjściem, dzięki elektrycznemu "zwarciu" (na ogół rezystancja takiego zwarcia to kilkadziesiąt omów) wejścia z wyjściem transmisja może odbywać się w obu kierunkach. Pozwala to także na wykorzystanie tego samego układu w roli multipleksera i demultipleksera.

4 Typy transmisji

Wyróżnić można typy transmisji:

- **simplex** – umożliwia tylko transmisję jednokierunkową
- **half-duplex** – umożliwia transmisję dwukierunkową, ale tylko w jedną stronę równocześnie
- **full-duplex** – umożliwia pełną transmisję dwukierunkową (równoczesne nadawanie i odbiór)

5 Magistrala równoległa

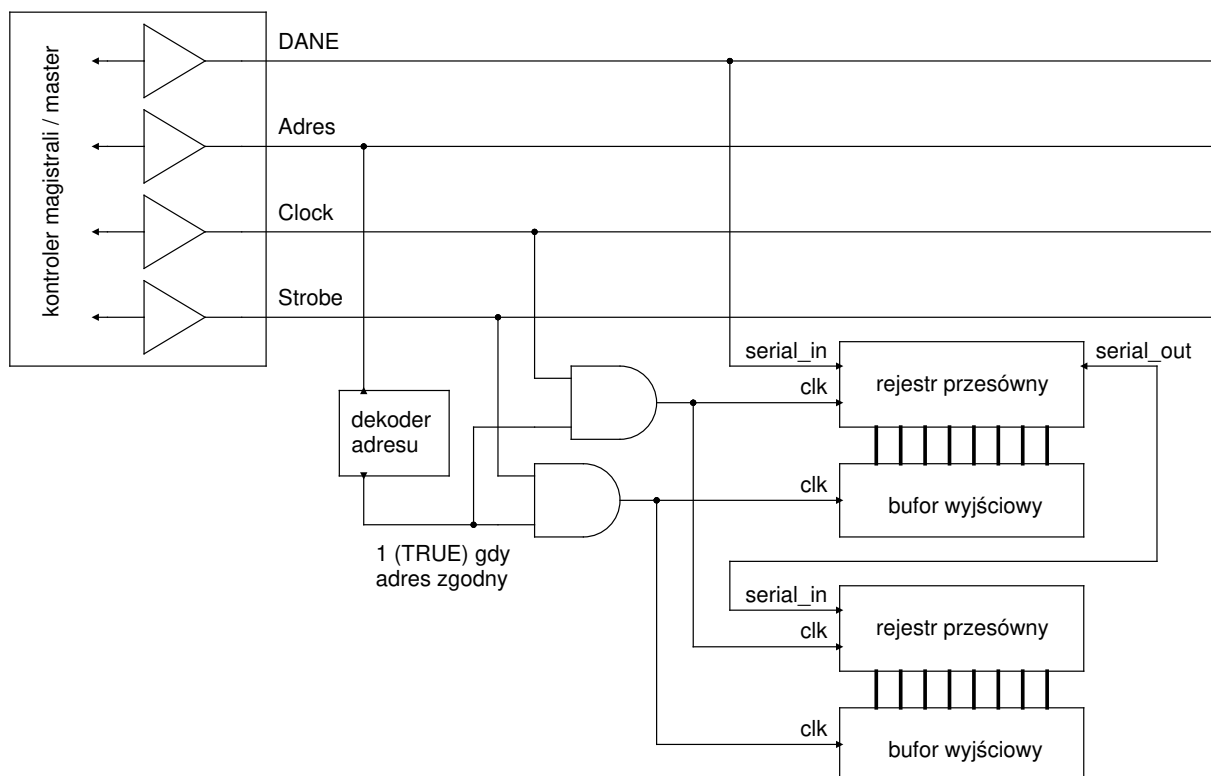


Magistrala równoległa jest zespołem linii, wraz z układami nimi sterującymi, umożliwiającym równoległe przesyłanie danych (w jednym czasie / taktie zegara na magistrali wystawiane / przesyłane jest całe n-bitowe słowo). Można wyróżnić szyny sterującą (kierunek przepływu, żądania obsługi, etc), adresową (adres układu który ma prawo nadawać) i danych (przesyłane dane). Często szyna adresowa współdzieli linie transmisyjne z szyną danych. Do realizacji magistrali (celem umożliwiania podłączenia wielu układów) stosuje się zazwyczaj bufor try-standowy, a do zapewnienia współdzielonej szyny żądania obsługi (interrupt request) często układy typu open-collector.

Typowy układ realizacji magistrali half-duplex ze współdzielonymi liniami danych i adresu przedstawiony został na schemacie zamieszczonym obok. Zadaniem dekodera adresu jest ustalenie czy wystawiony na magistrali adres (w trakcie wysokiego stanu linii "Adres / Not Dane") jest adresem danego urządzenia i zapamiętanie tej informacji do czasu wystawienia nowego adresu. Informacja ta jest wykorzystywana do sterowania dwukierunkowym buforem try-standowym (jako sygnał enable). O kierunku działania bufora decyduje sygnał "Read / Not Write". Przy magistralach o ustalonym protokole transmisyjnym sterowanie kierunkiem może być zależne od wykonywanej komendy (po ustawieniu adresu urządzenie odczytuje z magistrali polecenie i w zależności od niego steruje kierunkiem bufora - odczytuje lub zapisuje dane na magistralę). Zastosowanie kilku linii typu OC do odbierania żądań obsługi pozwala (na podstawie tego które z tych linii znalazły się w stanie niskim na identyfikację urządzenia lub grupy urządzeń, z której niektóre zgłaszają żądanie obsługi).

W przypadku prostych urządzeń wejścia / wyjścia zamiast bufora dwukierunkowego może być umieszczony np. jednokierunkowy bufor (lub n-bitowy rejestr) z wyjściami try-standowymi, który wystawia dane na magistralę w oparciu o sygnał zapisu na magistralę (WR) oraz zegar (clock) albo n-bitowy rejestr do którego zapisywane są dane z magistrali w oparciu o sygnał RD i Clock.

6 Magistrala szeregową



W magistrali szeregową dane przesyłane są bit po bicie w pojedynczej linii. Podobnie jak w magistrali równoległej oprócz linii danych mogą występować także linie sterujące. Prostą realizacją magistrali szeregową zapewniają rejestry przesuwne.

Przykładowy układ realizacji magistrali simplex (jednokierunkowej) z rozdzielonymi szynami danych i adresową został na schemacie zamieszczony obok. W prezentowanym przykładzie oprócz adresu master wystawia trzy sygnały - dane, zegar i strobe. Z każdym taktiem zegara na linii danych wystawiany jest kolejny bit który jest wczytywany do zespołu rejestrów. Sygnał strobe służy do przepisania wartości z rejestrów przesuwanych do rejestrów wyjściowych, takie rozwiązanie zapobiega zmianom wyjść w trakcie przesyłania nowych danych poprzez szynę szeregową, jest ono jednak opcjonalne.

W zależności od konstrukcji dekodera adresu szyna adresowa może być równoległa (w najprostszym przypadku - przez całą transmisję do danego urządzenia jego adres musi być wystawiony na szynie a dekodek jest układem bramek NOT i wielowejsiowej bramki AND) lub szeregową (w takim wypadku powinna posiadać własny zegar lub sygnał informujący o nadawaniu adresu z taktami zegara głównego, a dekodek powinien być wyposażony w rejestr przesuwany do odebrania i przechowywania aktualnego adresu z magistrali). Natomiast jeżeli magistrala byłaby oparta tylko na połączonych szeregowo rejestrach (wyjście serial-out do wejścia serial-in) to szyna adresowa nie jest potrzebna, ale konieczne może być każdorazowe wpisanie wszystkich wartości na szynę (czas zapisu rośnie z ilością podłączonych rejestrów).

7 Standardowe interfejsy

Istnieje wiele zestandaryzowanych interfejsów zarówno szeregowych jak i równoległych, wśród najważniejszych należy wymienić:

7.1 SPI (Serial Peripheral Interface)

jest to szeregową magistralą full-duplex działająca w układzie master-slave złożona z linii zegarowej (SCLK), nadawania przez mastera (MOSI), odbioru przez mastera (MISO) oraz linii służących do aktywacji urządzenia slave (SS / CS).

7.2 I2C (TWI)

jest to szeregową magistralą half-duplex złożoną z linii sygnałowej (SDA) i zegara (SCL) posiadającą zdefiniowany format ramki wraz z adresowaniem. Z wyjątkiem bitu startu i stopu stan linii danych może zmieniać się tylko przy niskim stanie linii zegarowej. Nadajniki są typu open-drain przez co realizowany jest iloczyn na drucie, co pozwala na wykrywanie kolizji (jeżeli dany nadajnik nie nadaje zera a linia jest w stanie zera to nadaje także ktoś inny). Pozwala to także na uzyskanie układów multimaster, pomimo iż typowo na magistrali takiej występuje tylko jeden układ master (nadający sygnał zegara i inicjujący transmisję).

7.3 1 wire (one-wire)

jest to szeregową magistralą half-duplex złożoną z jedynie z linii sygnałowej (która może także służyć do zasilania urządzeń) posiadającą zdefiniowany format ramki wraz z adresowaniem. Standardowe nadawanie jest realizowane jako open-drain (wyjątkiem jest nadawanie tzw power-byte).

7.4 USART

jest to uniwersalny synchroniczny i asynchroniczny nadajnik i odbiornik, umożliwia realizację szeregową transmisji synchronicznej (zgodnie z zegarem) lub asynchronicznej (wykrywanie początku ramki na podstawie linii danych). Interfejs korzysta z rozdzielonych linii nadajnika i odbiornika (wyjście danych TxD oraz wejście danych RxD, co umożliwia realizację transmisji full-duplex) oraz może korzystać z dodatkowych sygnałów sterujących (wyjście RTS informujące o gotowości do odbioru oraz wejście CTS informacji o gotowości odbioru / zezwolenia na nadawanie). Niekiedy dostępne jest także wyjście załączenia nadajnika używane do pracy w trybie half-duplex (linie TxD i RxD połączone buforem trój-stanowym nadajnika).

Interfejs ten najczęściej wykorzystywany jest w trybie asynchronicznym jako UART. W połączeniach UART zarówno nadajnik jak i odbiornik muszą mieć ustawione takie same parametry transmisji (szybkość, znaczenie 9 bitu (typowo bit parzystości, ale może także oznaczać np. pole adresowe), itp). Głównymi standardami elektrycznymi dla UART są: poziomy napięcie układów elektronicznych używających tych portów (3.3V, 5V), RS-232 (w pełnym wariacie używa sygnałów kontroli przepływu, poziom logiczny 1 wynosi od -15V do -3V, a poziom logiczny 0 od +3V do +15V), RS-422 (transmisja różnicowa full-duplex pomiędzy dwoma urządzeniami) i RS-485 (transmisja różnicowa half-duplex w oparciu o szynę łączącą wiele urządzeń, kompatybilny elektrycznie z RS-422), możliwa jest też transmisja światłowodowa i bezprzewodowa.

8 Układy programowalne

8.1 układy o programowalnej strukturze (PLD)

Są to układy w których programowany jest układ bramek, przerzutników, itp. "umieszczanych" w kości oraz ich połączeń.

Program dla takich układów tworzony jest w Hardware Description Language (najczęściej VHDL lub Verilog) i zamiast wykonywanego kodu opisuje strukturę układu logicznego (połączenia bramek, tablice prawdy, etc), która następnie jest programowana w fizycznej kości.

Najprostszym przykładem układu o programowalnej strukturze logicznej jest układ pamięci 2^n bitowej z n-bitową szyną adresową adresującą pojedyncze bity - pozwala on na realizację dowolnej funkcji logicznej o n wejściach i pojedynczym wyjściu.

Do kategorii tej zaliczają się układy typu:

- SPLD
 - PLE - programowalna matryca bramek OR
 - PAL i GAL - programowalna matryca AND z dodatkowymi bramkami OR (często także obudowana rejestrami i multiplekserami na wyjściach)

– PLA - programowalne matryce AND i OR

- CPLD
- FPGA - programowalny element pamięciowy (możliwość zdefiniowania dowolnej - na ogół 4 wejściowej - funkcji w każdym elemencie logicznym, programowalne połączenia między elementami logicznymi i pinami, itd)

8.2 systemy procesorowe

Są to systemy realizujące ciąg instrukcji pobieranych z jakiejś pamięci.

System taki składa się z procesora odpowiedzialnego za interpretację i wykonywanie kolejnych instrukcji oraz pamięci z której pobierane są instrukcje i dane (może to być jedna pamięć, mogą to być rozdzielone pamięci). Do kategorii tej zaliczają się zarówno typowe systemy komputerowe, systemy obliczeniowe jak i różnego rodzaju programowalne mikrokontrolery.

Procesor pracuje w cyklach rozkazowych, w ramach których przetwarza pojedynczą instrukcję. Cykl taki może trwać od 1 do kilku lub więcej cykli zegarowych i typowo składa się z następujących kroków:

1. pobranie instrukcji z pamięci - realizowane jest poprzez wystawienie na szynę adresową zawartości *licznika programu* (zawierające adres instrukcji do wykonania) oraz wygenerowanie cyklu odczytu z pamięci, po wykonaniu odczytu danych następuje ich zapamiętanie w *rejestrze instrukcji* oraz zwiększenie wartości *licznika programu* o jeden;
(zawartość rejestru *licznika programu* po resecie procesora określa skąd pobierana będzie pierwsza instrukcja, pod takim adresem zazwyczaj umieszczana jest jakaś pamięć typu ROM lub flash)
2. dekodowanie instrukcji - układ dekodera (np. oparty o PLA) dokonuje zdekodowania instrukcji znajdującej się w *rejestrze instrukcji* i konfiguracji procesora w zależności od jej kodu i (opcjonalnie) jej argumentów; może to być np.:
 - odpowiednie ustawienie multiplexerów pomiędzy rejestrami a jednostką ALU oraz wystawienie odpowiedniego kod operacji dla ALU (celem wykonania operacji arytmetycznej na wartościach rejestrów)
 - wystawienie zawartości wskazanego rejestru na szynę adresową, podłączenie wskazanego rejestru do szyny danych oraz skonfigurowanie operacji odczytu/zapisu (celem wykonania odczytu lub zapisu wartości rejestru z/do pamięci)
3. wykonanie instrukcji - realizacja wcześniej zdekodowanej instrukcji zgodnie z ustawioną konfiguracją procesora

Instrukcje skoku polegają na załadowaniu nowej wartości do *licznika programu*, w przypadku skoków warunkowych jest to uzależnione od stanu *rejestru flag*, które ustawiane są w oparciu o wynik ostatniej operacji wykonywanej przez ALU.

Przedstawiony model działania jest przykładowym i w rzeczywistym procesorze może to wyglądać odmiennie - np. długość instrukcji może być większa niż długość słowa używanego przez procesor / szerokość szyny danych co rozbudowuje fazę pobierania instrukcji z pamięci, mogą występować instrukcje bardziej złożone (np. operacje wykonywane z argumentem pobieranym z pamięci a nie rejestru), może także występować więcej faz (np. poprzez wydzielenie faz dostępu do pamięci, czy zapisywania wyników działania instrukcji). Procesor może także działać potokowo, czyli nakładać na siebie kolejne fazy wykonywania różnych instrukcji (np. w czasie wykonywania jednej instrukcji realizować pobieranie kolejnej).

8.2.1 Mikrokontrolery

Mikrokontroler jest układem typu "System on Chip" zawierającym w jednym układzie procesor, pamięć RAM, układy wejścia-wyjścia (np. GPIO, porty szeregowo typu USART, SPI, I2C, przetworniki ADC), pamięć typu Flash (dla programu).

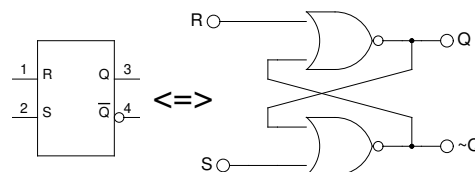
9 Literatura dodatkowa

- *Vademecum informatyki praktycznej* (<http://vip.opcode.eu.org/>) - zbiór materiałów na temat elektroniki i programowania.
- *Kurs elektroniki w serwisie forbot.pl* (<https://forbot.pl/blog/kurs-elektroniki-dla-poczatkujacych-id5151>).
- *Kurs elektroniki w serwisie robotykadlapoczatkujacych.pl* (<http://robotykadlapoczatkujacych.pl/kurs-elektroniki-dla-poczatkujacych/>).

10 Zadania

Zadanie 10.0.1

Na schemacie przedstawiono dwubramkową budowę przerzutnika RS w wariacie z wejściami nie zanegowanymi (zastosowanie bramek NAND w miejsce NOR spowoduje zanegowanie wejść). Zbuduj taki układ i sprawdź jego działanie.



Zadanie 10.0.2

Podłącz do kolejnych wyjść układu rejestru przesuwnego z buforem wyjściowym (np. CD4094 lub 74HC595) 4 diody LED (pamiętaj o rezystorach). Zapisz do rejestru i ustaw na wyjściach taką wartość aby świeciły się dwie pierwsze i ostatnia dioda, użyj w tym celu ręcznego manipulowania sygnałami:

- wejścia szeregowego (SERIAL IN), służącego do wprowadzania danych
- zegara danych (CLOCK, CLK), determinującego chwilę odczytu kolejnego bitu z wejścia szeregowego
- zegara wyjść (STROBE), determinującego chwilę przepisania danych z rejestru przesuwnego do rejestru wyjściowego

Wskazówka: zapoznaj się z dokumentacją posiadanego układu, ustal nazewnictwo używane do określania poszczególnych sygnałów (może się różnić nawet w zależności od producenta układu) oraz numery nóżek układu z nimi związane (mogą się różnić w zależności od modelu / wariantu obudowy).

Zadanie 10.0.3

Spróbuj zbudować własną bramkę logiczną w oparciu o tranzystory NPN i PNP. Pamiętaj że w odróżnieniu od pokazanych na powyższym schemacie tranzystorów NMOS i PMOS wymagane jest stosowanie rezystora na bramce.

11 Praca domowa

11.1 Instrukcja wysyłania rozwiązań

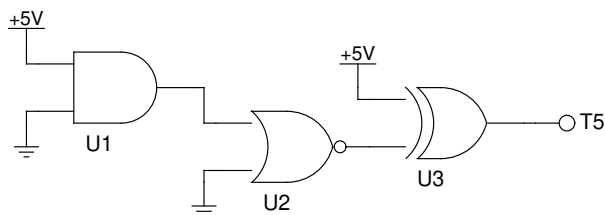
Rozwiązania zadań domowych należy przesłać na adres ciekawi.pracownia@icm.edu.pl wpisując jako temat wiadomości g2.x_PD12, gdzie x to numer grupy, np. g2.1_PD12 dla grupy nr. 1, itd. Zadania domowe są nie obowiązkowe, jednak zachęcamy do ich robienia i wysyłania rozwiązań (nawet niekompletnych).

Termin nadsyłania zdań domowych to 2020-05-26 godzina 16⁰⁰. Jeżeli wysłałeś rozwiązania w terminie, ale nie były one w 100% poprawne i dostałeś od sprawdzającego możliwość wysłania poprawki masz na to dodatkowe 4 dni.

Na ten adres można także nadsyłać ewentualne pytania do zadań (zarówno domowych jak i innych zamieszczonych w skrypcie), w tym wypadku także prosimy o umieszczenie w temacie wiadomości g2.x,

gdzie x to numer grupy.

11.2 Zadania domowe



Zadanie 11.2.1 — 1 pkt

Podaj wartość napięcia (względem GND) w punkcie T5. Przyjmujemy iż użyte bramki działają na poziomie napięć 5V (prawda) / 0V (fałsz). Odpowiedź krótko uzasadnij.

Zadanie 11.2.2 — 2 pkt

Zapoznaj się z dokumentacją układu 74HC574 (np. https://assets.nexperia.com/documents/data-sheet/74HC_HCT574.pdf) i opisz sposób jego użycia (wraz z sposobem sterowania) jako modułu podłączonego do 8 bitowej magistrali równoległej w roli układu wejściowego oraz w roli układu wyjściowego.

Zadanie 11.2.3 — 2 pkt

Zapoznaj się z dokumentacją układu 74HC595 (np. <http://www.ti.com/lit/ds/symlink/sn74hc595.pdf>) i opisz sposób jego użycia (wraz z sposobem sterowania) w roli układu wyjściowego podłączonego do magistrali szeregowej.