Układy sekwencyjne

Wszystkie do tej pory poznane układy logiczne były układami **kombinacyjnymi**, tzn. wartości na wyjściach układu zależały tylko i wyłącznie od wartości na wejściach. Aby skonstruować pewne elementy komputera będziemy potrzebowali układów **sekwencyjnych**. Układy takie będą miały swój wewnętrzny stan. Kolejny stan układu oraz jego wyjście zależy nie tylko od tego co jest na wejściach, ale także od poprzedniego stanu.

1 Podstawowe układy sekwencyjne

1.1 Przerzutnik S-R

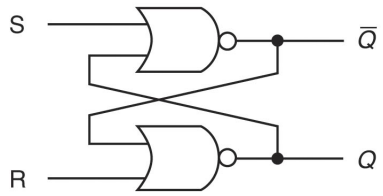
Każdy przerzutnik ma dwa możliwe stany wewnętrzne: 1 i 0. Jest zatem rodzajem pamięci 1-bitowej. Stan układu jest reprezentowany przez wyjście Q. Zaznaczamy zazwyczaj także wyjście dopełniające $\overline{Q}$.

Przerzutnik S-R ma dwa wyjścia S i R pozwalające ustawiać Q odpowiednio na 1 i 0. Rys. 1 przedstawia realizację przerzutnika S-R za pomocą bramek NOR i jego tabelę przejść. Pełna tabela przejść przedstawiona jest poniżej:

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

Kombinacja wejść 10 ustawia zatem przerzutnik w stanie 1, kombinacja 01 - w stanie 0, 00 jest kombinacją podtrzymującą stan układu, a 11 - kombinacją zabronioną (co się dzieje jak podamy takie wejście?)

Przedstawiony układ ma pewną wadę: po zmianie wejścia układ od razu działa i zmienia wyjście. Zachowanie takie może być nieporządane w systemie komputerowym - chcielibyśmy synchronizować działanie różnych jego elementów. W tym celu użyjemy sygnału zegarowego (na rysunkach oznaczanego zazwyczaj literką C). Sygnał taki na zmianę co pewnien okres czasu (połowę cyklu zegarowego) zmienia się z 0 na 1 i odwrotnie. Zegar o częstotliwośći 1GHz zmienia swój stan z 0 na 1 miliard razy na sekundę. Pierwsze usprawnienie przerzutnika S-R będzie polegało na podłączeniu go do sygnału zegarowego i zezwoleniu na zmianę stanu tylko w czasie gdy sygnał zegarowy jest równy 1. Wystarczy w tym celu połączyć wejścia S i R z sygnałem zegarowym bramkami AND (możemy synchronizować wiele przerzutników lub innych układów komputera podłączając je do tego samego sygnału zegarowego). Uzyskany układ nazywany jest **przerzutnikiem S-R sterowanym poziomem sygnału**.



(a)

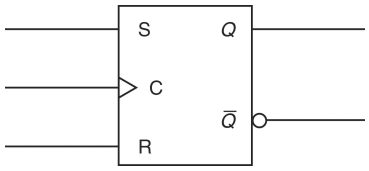
S	R	Q (t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

(b)

Rysunek 1: Przerzutnik S-R zrealizowany za pomocą bramek NOR

Układ zachowuje się już trochę lepiej, ale wciąż nie jest idealny: w czasie jednego cyklu zegarowego stan Q może zmienić się wielokrotnie. Spróbujemy wymusić, aby nasz przerzutnik mógł reagować zmianą stanu tylko raz w czasie pojedynczego cyklu zegarowego. Idea polega na użyciu połączonych szeregowo dwóch przerzutników S-R sterowanych poziomem sygnału i podpięciu do drugiego negacji sygnału zegarowego pierwszego przerzutnika. Odpowiedni rysunek i analiza działania zostały przedstawione na wykładzie. Uzyskany układ będziemy nazywali **przerzutnikiem S-R sterowanym zboczem (opadającym) sygnału**.

Symbol synchronizowanego przerzutnika S-R sterowanego zboczem przedstawia rysunek 2. W przypadku przerzutnika sterowanego poziomem sygnału trójkącik zastępowany jest zazwyczaj półokręgiem.

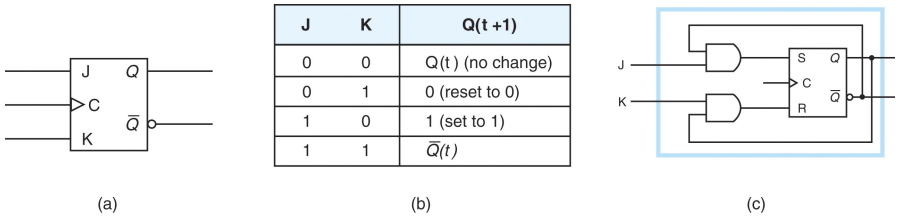


Rysunek 2: Symbol synchronizowanego przerzutnika S-R

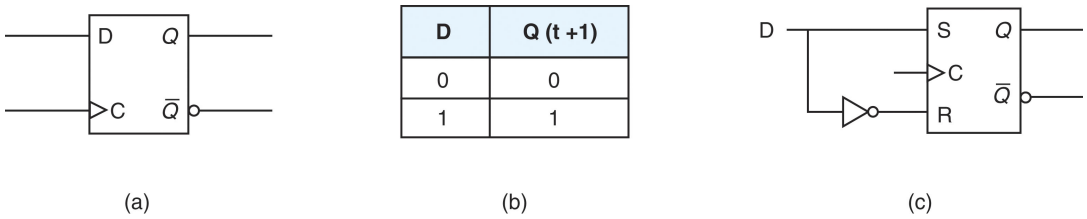
1.2 Inne przerzutniki

Przerzutnik J-K to nieco usprawniona wersja przerzutnika S-R – rozwiązuje problem zabronionego wejścia 11. Symbol, tabelę przejść oraz realizację za pomocą przerzutnika S-R przedstawia rysuneke 3.

Przerzutnik D (przerzutnik danych) to rzeczywisty odpowiednik jednostki pamięci przechowującej pojedynczy bit informacji. Ma jedno wejście D, które mówi co powinno być zapamiętane. Symbol, tabelę przejść oraz realizację za pomocą przerzutnika S-R przedstawia rysunek 4.



Rysunek 3: Przerzutnik J-K

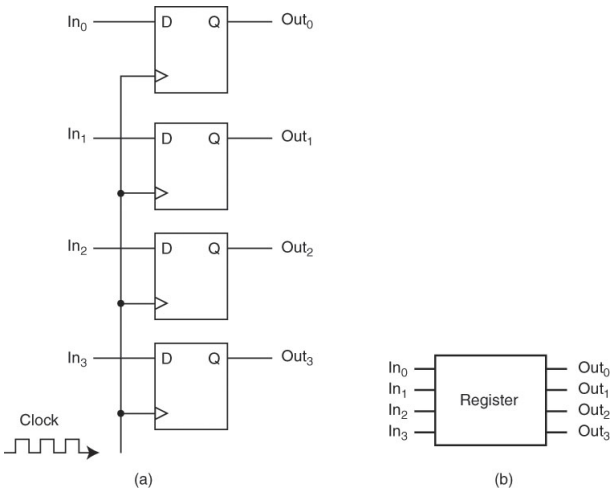


Rysunek 4: Przerzutnik D

2 Przykłady układów sekwencyjnych

2.1 Rejestr

Rejestry służą do przechowywania danych, np. liczb. Na rysunku 5 przedstawiamy rejestr 4-bitowy. W praktyce rejestry są większe (16, 32, 64 bity). Sygnał zegarowy synchronizujący cały układ zapewnia, że wszystkie bity rejestru będą modyfikowane w tym samym czasie.



Rysunek 5: Rejestr 4-bitowy i jego blokowe oznaczenie

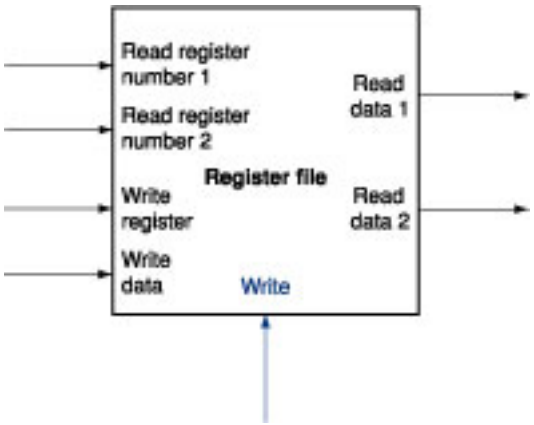
Rzeczywiste rejestry mają jeszcze dodatkowe linie zasilania i uziemienia oraz linie zerowania. Pomijamy je w naszych rozważaniach.

2.2 Zestaw rejestrów

Budując procesor realizujący listę rozkazów MIPS użyjemy m.in. zestawu (pliku, ang. *register file*) 32 rejestrów 32-bitowych. Na rysunku 6 przestawiamy schemat blokowy takiego zestawu. Układ ma następujące wejścia:

- dwa wejścia pięciobitowe, wskazujące numery rejestrów, które mają być odczytane (`Read register number 1`, `Read register number 2`),
- pięciobitowe wejście, wskazujące numer rejestru, który ma być zapisany (`Write register`),
- 32-bitową daną, która ma być zapisana (`Write data`),
- 1-bitowy sygnał zezwolenia na zapis (`Write`).

Układ umieszcza na dwóch 32-bitowych wyjściach dane odczytane z rejestów o wskazywanych numerach. Dodatkowo, jeśli sygnał `Write` jest ustawiony na 1 układ zapisuje do rejestru o wskazywanym numerze dane z wejścia `Write data`. Na rysunkach 7 i 8 przedstawiamy realizacją odpowiednio części odpowiedzialnych za odczyt oraz za zapis.



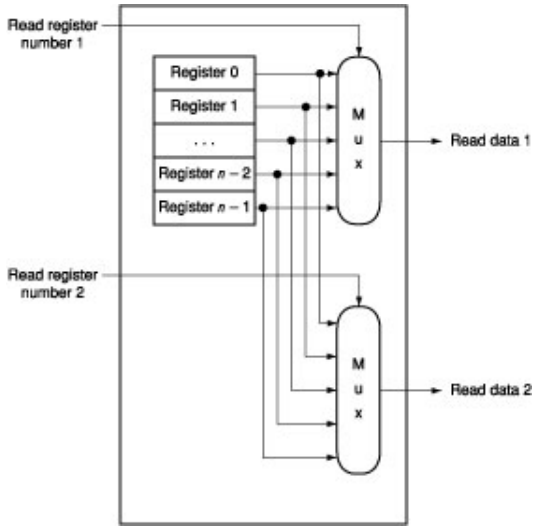
Rysunek 6: Zestaw rejestrów

2.3 Licznik binarny

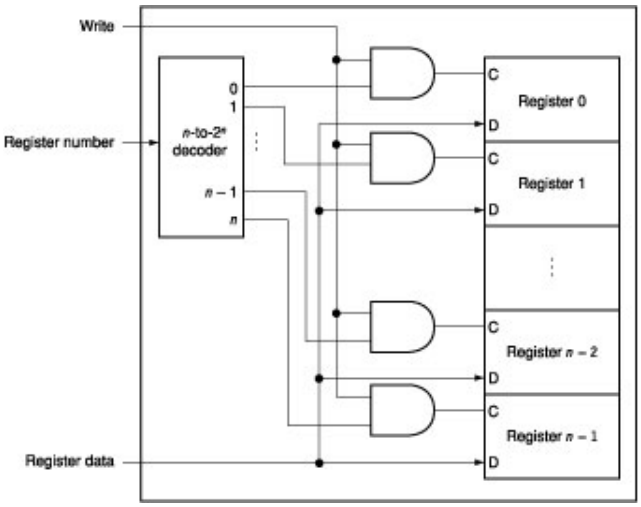
Licznik binarny to układ, który w takt zegara przyjmuje kolejno ściśle określone stany (zazwyczaj sekwencje kolejnych liczb binarnych). Rozważmy sekwencje: 0000, 0001, 0010, 0011, ..., 1111, 0000, 0001, ... Najmniej znaczący bit zmienia się w każdym cyklu zegarowym, kolejne bity: dokładnie wtedy, gdy wszystkie bity na prawo od nich są jedynkami. Najprościej zrealizować tego typu układ z przerzutników J-K.

Wersja asynchroniczna. Na wejścia J i K każdego z przerzutników podajemy 1 (na stałe). Na wejście zegarowe pierwszego przerzutnika podajemy sygnał zegarowy; na wejście zegarowe każdego z pozostałych przerzutników: wyjście Q poprzedniego przerzutnika. Dlaczego taki układ jest nazywany asynchronicznym (rozważ przejście ze stanu 1111 do 0000)?

Wersja synchroniczna. Na rysunku 9 przedstawiona jest realizacja 4-bitowego licznika synchronicznego. Zmiana stanu przerzutników następuje w nim równolegle. Licznik zaczyna



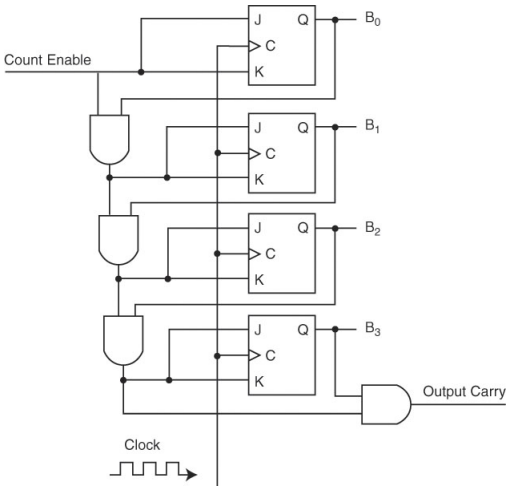
Rysunek 7: Układy odpowiedzialne za odczyt rejestrów



Rysunek 8: Układy odpowiedzialne za zapis rejestru

pracować po ustawieniu linii *count enable* na 1. Dodatkowo wyjście *output carry* pozwala wychwycić moment, gdy licznik doszedł do stanu 1111.

Na kolejnym wykładzie poznamy algorytm pozwalający konstruować liczniki oraz nieco bardziej skomplikowane układy sekwencyjne.



Rysunek 9: Licznik synchroniczny