

Laboratorium 2

Układ odliczający

Łukasz Kwinta, Kacper Kozubowski, Ida Ciepiela

kwiecień 2024

Spis treści

1	Cel zadania	3
2	Idea rozwiązania	3
3	Układ timer	4
3.1	Black box	4
3.1.1	Wejścia	4
3.1.2	Wyjścia	5
3.2	Diagram załączania układów	6
3.3	Kontrola działania licznika	6
3.3.1	Tablice prawdy	6
3.4	Wyprowadzenie formuł	7
3.5	Realizacja formuł	8
3.6	Podukład ustawiający czas licznika	9
4	Zastosowania	10
5	Wnioski	10

1 Cel zadania

Korzystając wyłącznie z wybranych przerzutników oraz dowolnych bramek logicznych, proszę zaprojektować czterobitowy układ TIMER, odmierzający ustawiany za pomocą przełączników czas (od 0 do 15).

Po wciśnięciu przycisku STRAT, układ rozpoczyna odmierzanie czasu do tyłu (proszę dobrać częstotliwość tak, aby efekt był dobrze widoczny na ekranie).

Po wyzerowaniu się licznika czasu, układ powinien się zatrzymać i włączyć alarm świetlny wykorzystujący diodę LED. Po ponownym wciśnięciu przycisku START, układ powinien wyłączyć alarm i ponownie rozpocząć odmierzanie ustawionego na przełącznikach czasu. Aktualny wskazywany przez układ czas proszę pokazywać na wyświetlaczach siedmiosegmentowych.

2 Idea rozwiązania

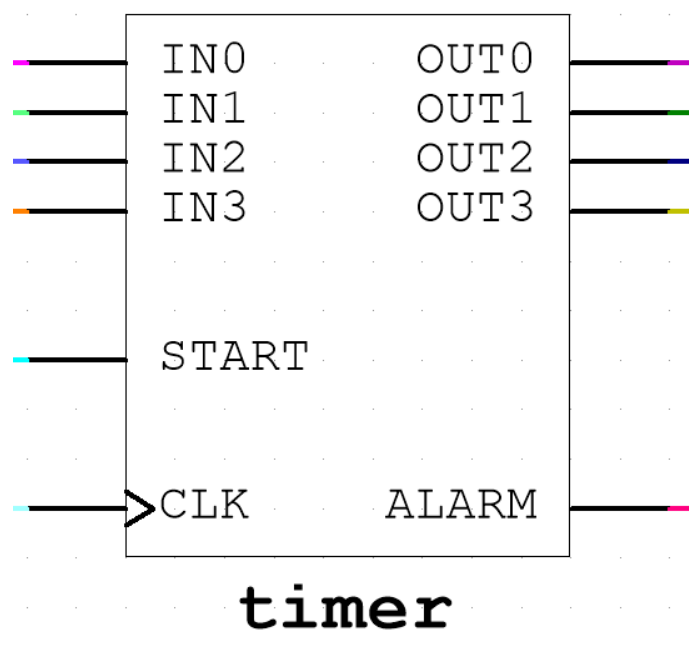
Do rozwiązania zadania wybraliśmy przerzutniki T - z powodu łatwości sterowania takim układem. Do sterowania przerzutnikami w trybie synchronicznym wykorzystaliśmy transkoder uruchamiający wejścia T kolejnych przerzutników na bazie obecnego stanu wyjścia układu. Dodatkowo dodaliśmy pojedynczy sygnał kontrolujący włączenie/wyłączenie układu.

Do początkowego zaprogramowania czasu odliczania na liczniku wykorzystaliśmy możliwość asynchronicznego ustawienia przerzutników w konkretny stan, również tutaj zaprojektowaliśmy transkoder, który porównuje stan przerzutników z wejściem do programowania czasu i odpowiednio ustawia układ.

3 Układ timer

3.1 Black box

Pierwszym krokiem w projektowaniu układu było zaprojektowanie czarnej skrzynki i określenie wejść i wyjść układu.



Poniżej przedstawimy specyfikację wejść i wyjść układu

3.1.1 Wejścia

- **INx** - wejścia programujące czas odliczania licznika - binarny zapis liczby od której licznik powinien zacząć odliczać. 4 wejścia łącznie pozwalają na odliczanie w zakresie 0-15. IN0 oznacza najmniej znaczący bit, IN3 oznacza najbardziej znaczący bit. Wejście jest używane do zaprogramowania w momencie gdy na wejściu START pojawi się stan wysoki.

Numer bitu	3	2	1	0
Bit	IN3	IN2	IN1	IN0
Mnożnik	2^3	2^2	2^1	2^0

Tabela 3.1: Kodowanie pinów wejściowych

- **START** - wejście aktywujące układ. Stan wysoki oznacza aktywację licznika, stan niski oznacza, że licznik dokończy liczenie do wyzerowania licznika.

Jeśli wejście **START** będzie miało stan wysoki w czasie dojścia licznika do zera, na wyjściu **ALARM** pojawi się puls, a następnie licznik zostanie zaprogramowany obecnym wejściem, a następnie uruchomiony ponownie.

Zmiany stanu na wejściu **START** w czasie gdy licznik jest w stanie liczenia, nie mają żadnego efektu.

- **CLK** - wejście zegara stanowiącego podstawę czasu licznika - określa jak szybko następować będą zmiany wyjścia i odliczanie licznika do zera.

3.1.2 Wyjścia

- **OUTx** - wyjścia stanowiące kolejne bity aktualnego stanu licznika. Zmiana wartości licznika, następuje na wznoszącym zboczu zegara wejściowego. **OUT0** stanowi najmniej znaczący bit, a **OUT3** najbardziej znaczący bit.

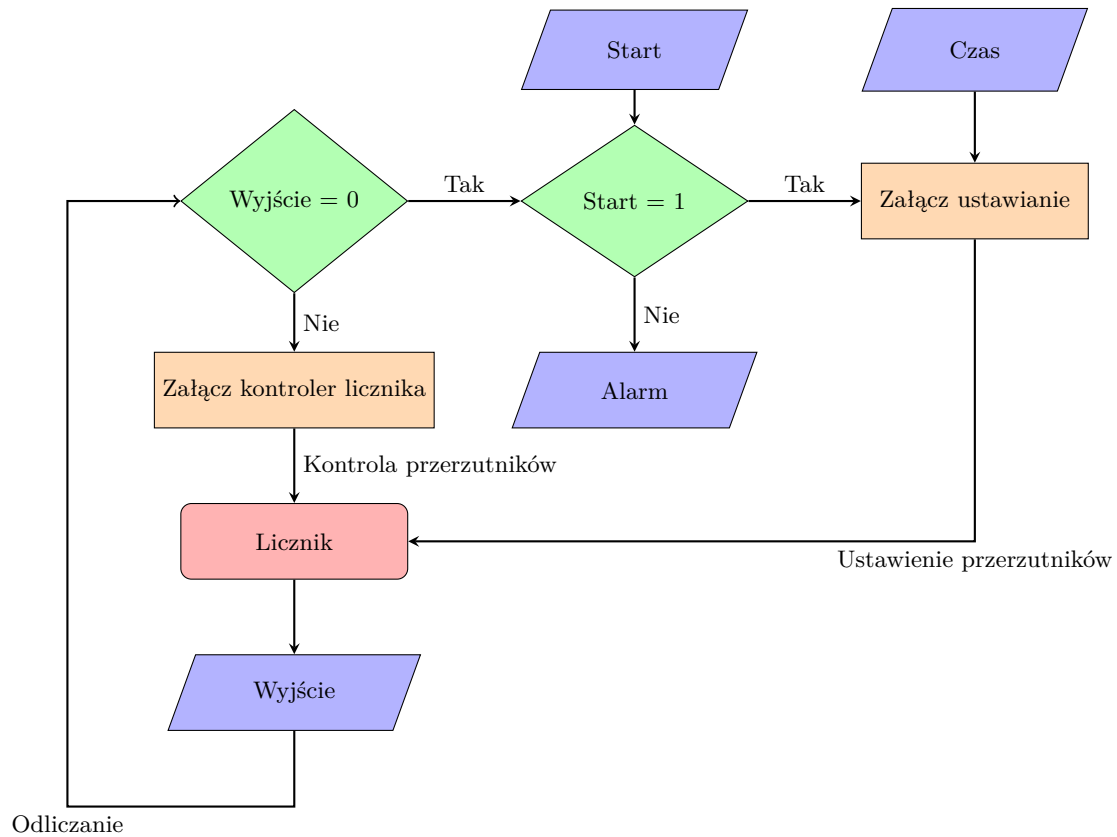
Numer bitu	3	2	1	0
Bit	OUT3	OUT2	OUT1	OUT0
Mnożnik	2^3	2^2	2^1	2^0

Tabela 3.2: Kodowanie pinów wejściowych

- **ALARM** - wyjście sygnalizujące zakończenie odliczania licznika. Stan wysoki oznacza, że obecny stan licznika jest równy 0.

3.2 Diagram załączania układów

Poniżej rozpisaliśmy diagram zależności stanu załączenia poszczególnych układów od siebie z którego wynikać będą tabele prawdy.



3.3 Kontrola działania licznika

Na najwyższym poziomie nasz układ **timer** składa się z dwóch podukładów: **timer_setter** - układu ustawiającego czas odliczania oraz **timer_driver** układu kontrolującego wejścia T przerzutników. Na tym samym poziomie znajdują się przerzutniki stanowiące faktyczny licznik oraz implementacja formuł załączających te układy opisanych tutaj.

Dla czytelności poniżej przyjmujemy następujące oznaczenia:

- **EN_SET** - wejście aktywujące w układzie **timer_setter**
- **EN_DRV** - wejście aktywujące w układzie **timer_driver**
- **EQ0** - wyjście układu **timer_driver** mówiące o tym czy obecny stan licznika to 0 (stan wysoki).
- **START** - wejście startowe timera

3.3.1 Tablice prawdy

Tabela prawdy wynika z schematu kontroli przedstawionego powyżej.

Wejście		Wyjście	
EQ0	START	EN_SET	EN_DRV
0	0	0	1
0	1	0	1
1	0	0	0
1	1	1	0

Tabela 3.3: Tabela prawdy dla stanów aktywacji podukładów

3.4 Wyprowadzenie formuł

Dla wyjścia EN_SET możemy odczytać formułę wprost z tabeli:

$$\text{EN_SET} = \text{EQ0} \cdot \text{START}$$

Dla wyjścia EN_DRV możemy pokusić się o próbę optymalizacji formuły przy pomocy tablicy Karnaugh:

		START	
		0	1
EQ0	0	1	0
	1	1	0

Tabela 3.4: Tablica Karnaugh dla formuły aktywującej układ kontrolujący licznik

Możemy z niej odczytać zoptymalizowaną formułę:

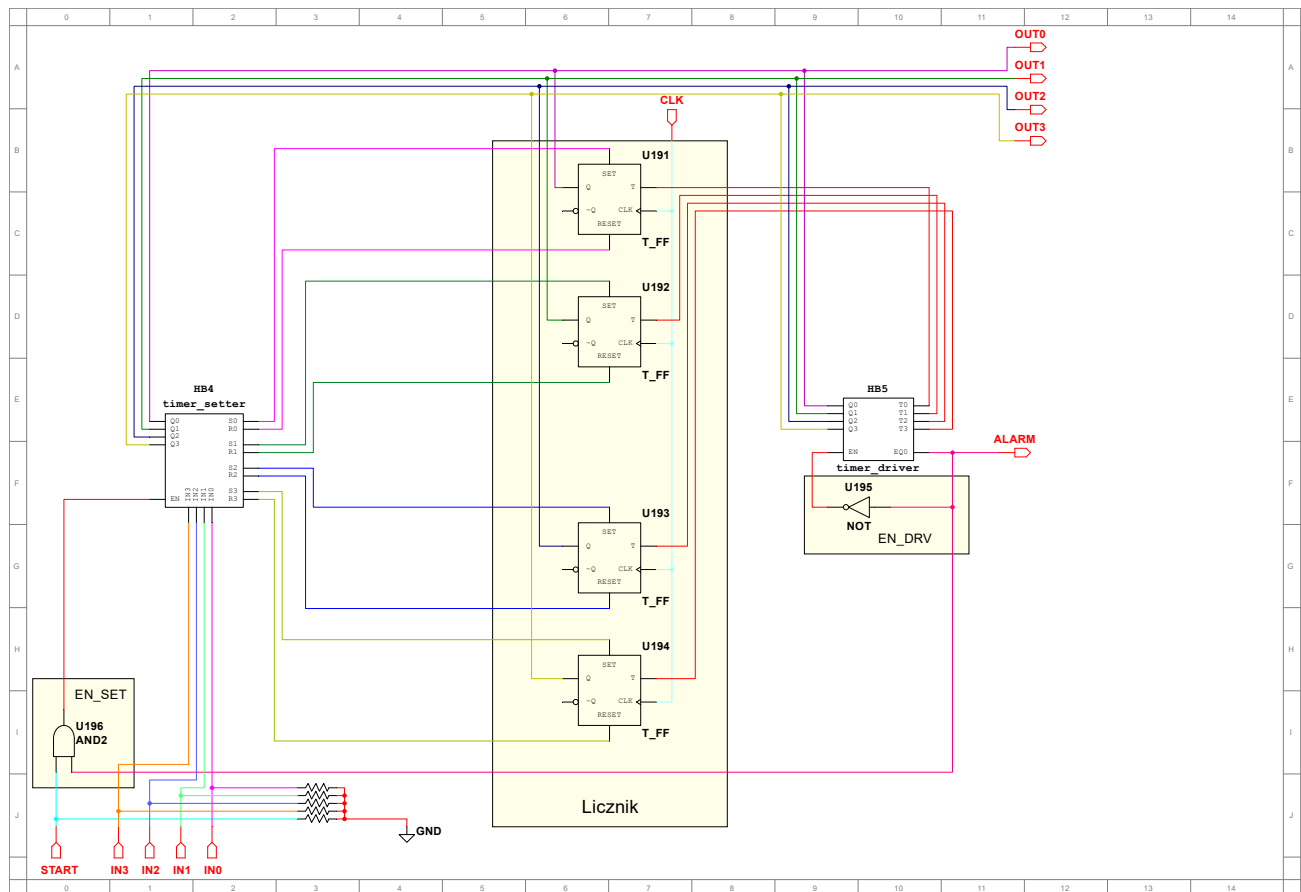
$$\text{EN_DRV} = \overline{\text{EQ0}}$$

3.5 Realizacja formuł

Poniżej przedstawiamy realizację wcześniej wyprowadzonych formuł:

$$\text{EN_SET} = \text{EQ0} \cdot \text{START}$$

$$\text{EN_DRV} = \overline{\text{EQ0}}$$



Na schemacie znajdują się również rezystory pull-down zabezpieczające wejścia układu przed nieokreślonym stanem wejść.

3.6 Podukład ustawiający czas licznika

Układ nazwany na naszych schematach `timer_setter` programuje

4 Zastosowania

- Sprzętowa obsługa timerów na mikroprocesora, wraz z mechanizmem sprzętowych przerwań stanowią przydatny mechanizm pozwalający na cykliczne wywoływanie zadanego kodu bez konieczności zużywania czasu procesora na kontrolę upływu czasu. Mechanizm taki ma zastosowanie od wspomnianego cyklicznego wykonywania kodu do sprzężenia ze sobą różnych peryferiów mikroprocesora ze sobą pozwalające na automatyczne dokonywanie cyklicznych pomiarów czy generowanie sygnałów jak np. sygnał Alarm timera podpięty jako sygnał wyzwalający konwersję przetwornika ADC który odczyt zapisuje bezpośrednio do pamięci poprzez mechanizm DMA czy timer generujący sygnał PWM przez porównanie obecnego stanu licznika z zadaną wartością - ponownie w sprzężeniu z mechanizmem DMA pozwalającym odczytywać kolejne wartości do porównania w każdym okresie timera bezpośrednio z pamięci można nadawać praktycznie dowolny sygnał cyfrowy.

5 Wnioski

Dzięki przerzutnikom możemy projektować złożone układy z pamięcią swojego stanu.

W toku rozwiązywania postawionego problemu opracowaliśmy kilka możliwych rozwiązań. Ostatecznie zdecydowaliśmy się na przedstawienie tego konkretnego rozwiązania ze względu na jego prostotę i największą niezawodność. Zastosowanie przerzutników działających synchronicznie pozwoliło na zredukowanie ilości hazardów przy odliczaniu czasu.

Innym sposobem było: <tu opis rozwiązanie Kacpra i schemat>

Pierwotnym pomysłem było użycie przerzutników T asynchronicznie - łącząc je w szeregowo dzięki czemu nie było konieczne projektowanie układu kontrolującego przerzutniki. Największą wadą tego rozwiązania było to, że otoczka sterowania tak połączonych przerzutników była trudna do przedstawienia wzorami i uzasadnienia w prosty sposób. Rozwiązanie opierało się na idei generowania kolejnych impulsów synchronizujących kolejne etapy przygotowania i uruchomienia układu.

<tu mój schemat>