

Tura

Fizyczna implementacja maszyny Turinga

Paweł Szymański

Luty 2026

Spis treści

1	Wstęp	2
2	Opis zastosowanego modelu	2
3	Model Działania	5
3.1	Opis Ogólny	5
3.2	Opis Szczegółowy	5
3.2.1	Moduły	5
3.2.2	Sygnały	7
3.2.3	Interfejs zewnętrzny	11
3.2.4	Programowanie	12
3.2.5	Wykonanie Programu	18
3.3	Odczyt Wyniku Programu	18
4	Fizyczny Opis Działania	18

1 Wstęp

Niewątpliwie jednym z najważniejszych osiągnięć ludzkości XX wieku jest wynalazek będący abstrakcyjnym modelem matematycznym. Mowa oczywiście o maszynie Turinga. Dzięki temu rewolucyjnemu wynalazkowi dziś jako ludzkość jesteśmy w stanie automatyzować zaawansowane zagadnienia, jak na przykład obróbka i montaż produktów w fabrykach czy rozwiązywanie skomplikowanych obliczeń matematycznych. Bez maszyn Turinga nie mielibyśmy dziś komputerów, urządzeń tak uniwersalnych i wszechstronnych jak te które znamy, z których korzystamy i z którego najprawdopodobniej czytelnik czyta tę dokumentację. W ramach urzeczywistnienia matematycznego modelu maszyny Turinga powstał projekt „Tura”, którego zadaniem jest fizyczna implementacja pewnego modelu maszyny. Z racji, iż podstawowy model zakłada nieograniczoną długość taśmy, implementacja maszyny jako takiej nie jest możliwa. Jednak pewien podzbiór maszyn zakłada skończoną długość taśmy. Tym modelem jest maszyna Turinga ograniczona liniowo. Implementowany model, w celu uproszczenia elektronicznych układów, dodatkowo posiada własność akceptującego stanu zatrzymującego.

2 Opis zastosowanego modelu

Definicja maszyny Turinga M jest następująca:

$$M = (Q, \Gamma, \Sigma, \delta, q_0, B, F),$$

gdzie:

- Q - zbiór stanów,
- Γ - alfabet taśmy,

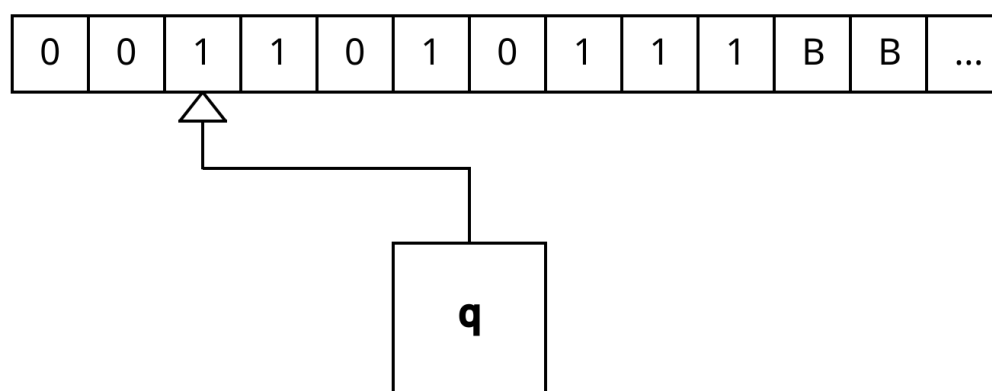
- Σ - alfabet słowa wejściowego, $\Sigma \subseteq \Gamma$,
- δ - funkcja przejścia, $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times D$, gdzie D jest zbiorem możliwych ruchów głowicy maszyny, $D = \{L, R\}$,
- q_0 - stan początkowy, $q_0 \in Q$,
- B - symbol pusty, $B \in \Gamma \setminus \Sigma$,
- F - zbiór stanów akceptujących.

Żeby znaleźć jakieś intuicyjne wyobrażenie jak taka maszyna może wyglądać i funkcjonować możemy wyobrazić sobie skrzynkę, w której zamknięty jest układ sterujący urządzeniem oraz przede wszystkim przechowujący jego obecny stan. Obok skrzynki umieszczona jest taśma z symbolami. Ma ona swój początek z jednej strony, jednak z drugiej nie ma końca - jest jednostronnie nieskończona. Taśmę i skrzynkę łączy głowica - czyta ona lub nadpisuje jeden symbol na taśmie oraz przesuwa się po niej. Na początku działania maszyny głowica znajduje się nad skrajnym symbolem taśmy.

Działanie takiej maszyny można porównać do następującej sekwencji czynów:

1. Głowica odczytuje symbol $s \in \Gamma$ z taśmy.
2. Skrzynka na podstawie obecnego stanu $q \in Q$ oraz s oblicza nowy stan, symbol i kierunek przesunięcia głowicy za pomocą funkcji przejścia: $\delta(q, s) = (q', s', d)$.
3. Skrzynka zmienia obecny stan z q na q' .
4. Głowica nadpisuje symbol na taśmie s na s' , po czym przesuwa się o jeden symbol się zgodnie z kierunkiem d .

5. Sekwencja ta jest powtarzana dopóty, dopóki maszyna nie osiągnie stanu akceptującego.



Rysunek 1: Przykładowa wizualizacja maszyny Turinga

Fizyczne urządzenie cechuje się dodatkowo ograniczoną długością taśmy oraz jednym stanem akceptującym zatrzymującym działanie maszyny. Formalnie ograniczoną długość taśmy można uzyskać poprzez wstawienie wartowników na jej początku oraz końcu, jednak te działania pozostawione jest dla użytkownika maszyny do samodzielnej implementacji. Zaś stan akceptujący zatrzymujący działanie maszyny oznacza, że F ma dokładnie jeden stan akceptujący q_A , po którego osiągnięciu urządzenie zatrzymuje pracę.

3 Model Działania

3.1 Opis Ogólny

Działanie maszyny w skrócie opiera się o zaprogramowanie urządzenia, wykonanie obliczeń i pobranie wyniku.

- Programowanie odbywa się poprzez zapisanie wejściowej zawartości taśmy oraz funkcji przejścia w trzech blokach pamięci (bloku symboli taśmy, bloku funkcji przejścia symbolu taśmy oraz bloku funkcji przejścia stanu).
- Wykonanie obliczeń to uruchomienie urządzenia i pozostawienie w działaniu do czasu osiągnięcia stanu akceptującego.
- Pobranie wyniku jest odczytem zawartości taśmy po zakończeniu działania maszyny.

3.2 Opis Szczegółowy

Opis szczegółowy wchodzi już w fizyczną implementację urządzenia. Jego celem jest staranny opis modułów, sygnałów, ich wzajemnej interakcji oraz dokładne wytłumaczenie działania maszyny. Opis w znacznej mierze opiera się o schematy KiCad i korzysta ze stosowanych w nich oznaczeń oraz nazw.

3.2.1 Moduły

Maszyna została podzielona na 11 modułów w celu oznakowania jednostek odpowiedzialnych za poszczególne funkcje. Moduły w schematach KiCad zostały otoczone ciemnożółtymi ramkami i podpisane niebieskim tekstem lub jedynie podpisane niebieskim tekstem w przypadku modułów jednoelementowych.

TAPE – Blok pamięci symulujący taśmę. Każda komórka pamięci reprezentuje jeden symbol taśmy. Pojedynczy symbol taśmy składa się z 7 bitów;

TAPE SYMBOL REGISTER – Rejestr przechowujący jeden symbol taśmy. Jest on potrzebny do chwilowego przechowania odczytanego symbolu na czas zapisu nowego do **TAPE**;

SYMBOL TRANSLATOR – Blok pamięci symulujący funkcję przejścia dla zadanego symbolu oraz stanu zwracający wynik w formie symbolu taśmy (7 bitów) oraz kierunku ruchu głowicy (1 bit);

STATE TRANSLATOR – Blok pamięci symulujący funkcję przejścia dla zadanego symbolu oraz stanu zwracający wynik w formie stanu (8 bitów);

STATE REGISTER – Rejestr przechowujący obecny stan maszyny;

HEADER POSITION – Zbiór liczników binarnych z możliwością inkrementacji oraz dekrementacji reprezentujący usytuowanie głowicy na taśmie (wskazuje obecnie wybrany symbol w **TAPE**);

HEADER POSITION DIRECTION DISASSEMBLY – Prosty multiplekser sygnału zmiany wartości **HEADER POSITION** w zależności od kierunku zmiany;

HEADER POSITION CLOCK SYNCHRONIZATION – Bramka **AND** synchronizująca sygnał zmiany wartości **HEADER POSITION** z zegarem;

STEP EXECUTION STATE REGISTER – Rejestr przechowujący obecny stan wykonywania ruchu maszyny. Ze względu na brak praktycznego rozwiązania jednoczesnego odczytu i zapisu symbolu taśmy, przeprowadzenie przejścia zostało podzielone na

dwa ruchy rozdzielające te operacje; stąd dwa stany wykonania ruchu;

IN-PROGRAM CONTROL LINES BUFFERS – Zestaw buforów oddzielający kontrolne linie sygnałowe generowane przez interfejs zewnętrzny od tych wewnętrznych;

IN-OPERATION CONTROL LINES BUFFER – Bufor oddzielający wewnętrzne linie kontrolne od tych generowanych przez urządzenie podczas pracy;

3.2.2 Sygnały

Oznaczenie sygnałów jest takie same jak oznaczenia zastosowane w schematach KiCad. Sygnały z poprzeczką nad nazwą oznaczają, że sygnał jest aktywny dla niskiego poziomu, zaś nieaktywny dla wysokiego.

Lista sygnałów:

$D_{0...7}$ – Magistrala danych używana do programowania i odczytu danych z maszyny;

$S_{0...6}$ – Magistrala symbolu taśmy;

$SR_{0...6}$ – Magistrala symbolu taśmy zapisanego w TAPE SYMBOL REGISTER;

$Q_{0...7}$ – Magistrala stanu;

$QR_{0...7}$ – Magistrala stanu zapisanego w STATE REGISTER;

$H_{0...15}$ – Magistrala z wartością licznika HEADER POSITION;

CLK – Sygnał zegarowy wyznaczający rytm pracy; każda zmiana sygnału ze stanu niskiego do wysokiego powoduje przejście STEP EXECUTION STATE REGISTER do następnego stanu; działa pod warunkiem aktywnego sygnału CLK_EN;

CLK_EN – Kontroluje dopływ sygnału CLK do STEP EXECUTION STATE REGISTER;

DIR – Ustala kierunek zmiany wartości HEADER POSITION. Wysoki stan sygnału znaczy inkrementację, niski stan dekrementację;

HP_EN – Dopuszcza możliwość zmiany wartości HEADER POSITION;

HP_EN_CLK – sygnał HP_EN zsynchronizowany z zegarem (CLK):
$$HP_EN_CLK = HP_EN \cdot CLK;$$
 synchronizuje zmianę wartości HEADER POSITION z zegarem;

DIR_UP – Decyduje o inkrementacji wartości HEADER POSITION; działa przy przejściu ze stanu niskiego do wysokiego;

DIR_DOWN – Decyduje o dekrementacji wartości HEADER POSITION; działa przy przejściu ze stanu niskiego do wysokiego;

\overline{RESET} – Resetuje urządzenie, czyli ustawia STEP EXECUTION STATE REGISTER do stanu 0 oraz ustawia wartość HEADER POSITION na 0;

$\overline{PROGRAM}$ – Ustala tryb pracy. Poziom niski sygnału oznacza tryb programowania maszyny, poziom wysoki zaś tryb wykonywania programu;

PROGRAM – Przeciwnieństwo sygnału $\overline{PROGRAM}$; sygnał ten jest równoważny $\overline{OPERATE}$ – podany został jedynie dla czytelności schematu;

$\overline{\text{OPERATE}}$ – Przeciwnieństwo sygnału $\overline{\text{PROGRAM}}$. Niski poziom oznacza pracę urządzenia;

DDIR – Ustala kierunek przepływu danych z wewnętrznych modułów wybranych sygnałami CS_0 , CS_1 na magistralę $\text{D}_0 \dots 7$. Wysoki poziom sygnału oznacza zapis danych z interfejsu zewnętrznego do modułów wewnątrz urządzenia. Niski poziom oznacza odczyt;

CS_0 , CS_1 – Kombinacja tych sygnałów wybiera wewnętrzną magistralę, która zostanie połączona z zewnętrzną $\text{D}_0 \dots 7$ oraz wewnętrzny moduł do odczytu/zapisu danych z wybranej magistrali (kierunek zależny od DDIR);

$\overline{\text{DT}}$ – Włącza przesył danych między wewnętrznymi modułami a interfejsem zewnętrznym; aktywny dla niskiego poziomu;

TSR_WR – Zapisuje dane z magistrali symbolu taśmy $\text{S}_0 \dots 6$ do $\text{TAPE SYMBOL REGISTER}$; zapis odbywa się podczas przejścia sygnału ze stanu niskiego do wysokiego;

SR_WR – Zapisuje dane z magistrali stanu $\text{Q}_0 \dots 7$ do STATE REGISTER ; zapis odbywa się podczas przejścia sygnału ze stanu niskiego do wysokiego;

$\text{T_}\overline{\text{OE}}$ – Włącza wyjście danych z TAPE na magistralę $\text{S}_0 \dots 6$; aktywny dla niskiego poziomu;

$\text{T_}\overline{\text{WE}}$ – Włącza zapis danych w TAPE z magistrali $\text{S}_0 \dots 6$; aktywny dla niskiego poziomu;

$\text{SYT_}\overline{\text{OE}}$ – Włącza wyjście danych z SYMBOL TRANSLATOR na magistralę $\text{S}_0 \dots 6$ oraz sygnał DIR_B ; aktywny dla niskiego poziomu;

$\text{SYT_}\overline{\text{WE}}$ – Włącza zapis danych w SYMBOL TRANSLATOR z magistrali $\text{S}_0 \dots 6$ oraz sygnału DIR_B ; aktywny dla niskiego poziomu;

$\overline{\text{STT_OE}}$ – Włącza wyjście danych ze STATE TRANSLATOR na magistralę $Q_0 \dots 7$; aktywny dla niskiego poziomu;

$\overline{\text{STT_WE}}$ – Włącza zapis danych w STATE TRANSLATOR z magistrali $Q_0 \dots 7$; aktywny dla niskiego poziomu;

$\overline{\text{QR_OE}}$ – Włącza wyjście danych ze STATE REGISTER na magistralę $D_0 \dots 7$; aktywny dla niskiego poziomu;

$\overline{\text{T_E_E}}, \overline{\text{SYT_E_E}}, \overline{\text{ST_E_E}}, \overline{\text{QR_OE_E}}$ – Sygnały odpowiadające tym bez sufiksu „_E”, będące połączone z nimi przez bufor U18 oraz będące wynikiem dekodera wyboru funkcji U19;

$\overline{\text{T_E_B}}, \overline{\text{SYT_OE_B}}, \overline{\text{STT_OE_B}}, \overline{\text{DIR_B}}$ – Sygnały odpowiadające tym bez sufiksu „_B”, będące połączone z nimi przez bufor U17. Są one sygnałami generowanymi przez maszynę podczas obliczeń potrzebnymi do poprawnej jej samodzielnej pracy;

$\overline{\text{SYMB_BUF}}$ – Włącza przepływ danych z TAPE lub SYMBOL TRANSLATOR (w zależności od wybranego urządzenia sygnałami $\text{CS}_{0,1}$) na magistralę $D_0 \dots 7$; aktywny dla niskiego poziomu;

$\overline{\text{STAT_BUF}}$ – Włącza przepływ danych ze STATE TRANSLATOR na magistralę $D_0 \dots 7$; aktywny dla niskiego poziomu;

$\overline{\text{FINISH_STATE}}$ – Informuje o osiągnięciu przez maszynę stanu akceptującego i zarazem zatrzymującego pracę; aktywny dla niskiego poziomu.

SEST_STATE – Wskazuje obecny stan STEP EXECUTION STATE REGISTER;

3.2.3 Interfejs zewnętrzny

W celu zasilenia urządzenia oraz komunikacji zastosowany został zewnętrzny interfejs składający się z 16 wyjść sterujących, 17 informujących o stanie maszyny oraz 11 zasilających. W sumie daje to 44 wyjścia. Wszystkie wspomniane sygnały (poza V_{CC} oraz V_{SS}) zostały wyjaśnione w poprzednim podrozdziale. Wyjścia V_{CC} oraz V_{SS} są wyjściami odpowiednio zasilania i uziemienia. V_{CC} powinno mieć napięcie +5V względem V_{SS} .

By uniknąć zbyt podobnych oznaczeń, kierunki sygnałów zamiast „Wej/Wyj” zostały oznaczone angielskimi „In/Out”.

Tabela sygnałów interfejsu:

Num. wyjścia	Sygnały	Kierunek
1-8	$D_0 \dots 7$	In/Out
9	CLK	In
10	$\overline{\text{RESET}}$	In
11	$\overline{\text{PROGRAM}}$	In
12	DDIR	In
13-14	$CS_{0,1}$	In
15	\overline{DT}	In
16	$\overline{\text{FINISH_STATE}}$	Out

Tabela 1: Lista sygnałów interfejsu zewnętrznego

Num. wyjścia	Sygnały	Kierunek
17,19,21,23,25	V _{CC}	—
18,20,22,24,26	V _{SS}	—
27-42	H _{0...15}	Out
43	SESR_STATE	Out
44	V _{SS}	—

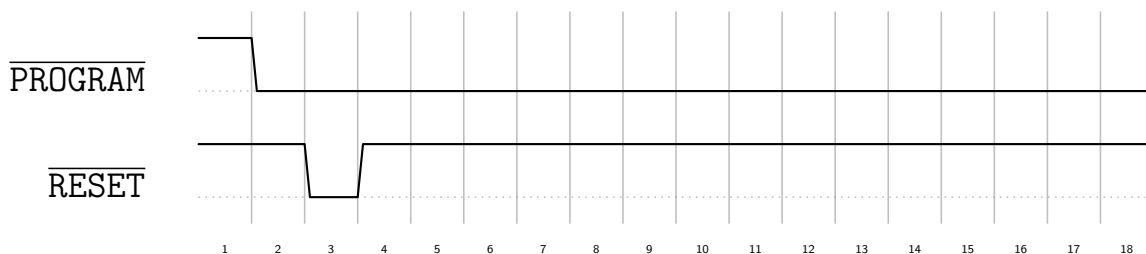
Tabela 2: Lista sygnałów interfejsu zewnętrznego (cd.)

3.2.4 Programowanie

Programowanie maszyny odbywa się za pomocą interfejsu zewnętrznego. Dzięki odpowiedniemu wykorzystaniu udostępnionych sygnałów użytkownik jest w stanie manipulować zawartością bloków pamięci TAPE, SYMBOL TRANSLATOR oraz STATE TRANSLATOR.

3.2.4.1 Przygotowanie maszyny do programowania

W celu rozpoczęcia programowania maszyny należy wpierw ją do tego przygotować. Zadanie to jest dość proste - wystarczy ustawić sygnał $\overline{\text{PROGRAM}}$ na niski poziom, po czym puścić puls sygnału $\overline{\text{RESET}}$.

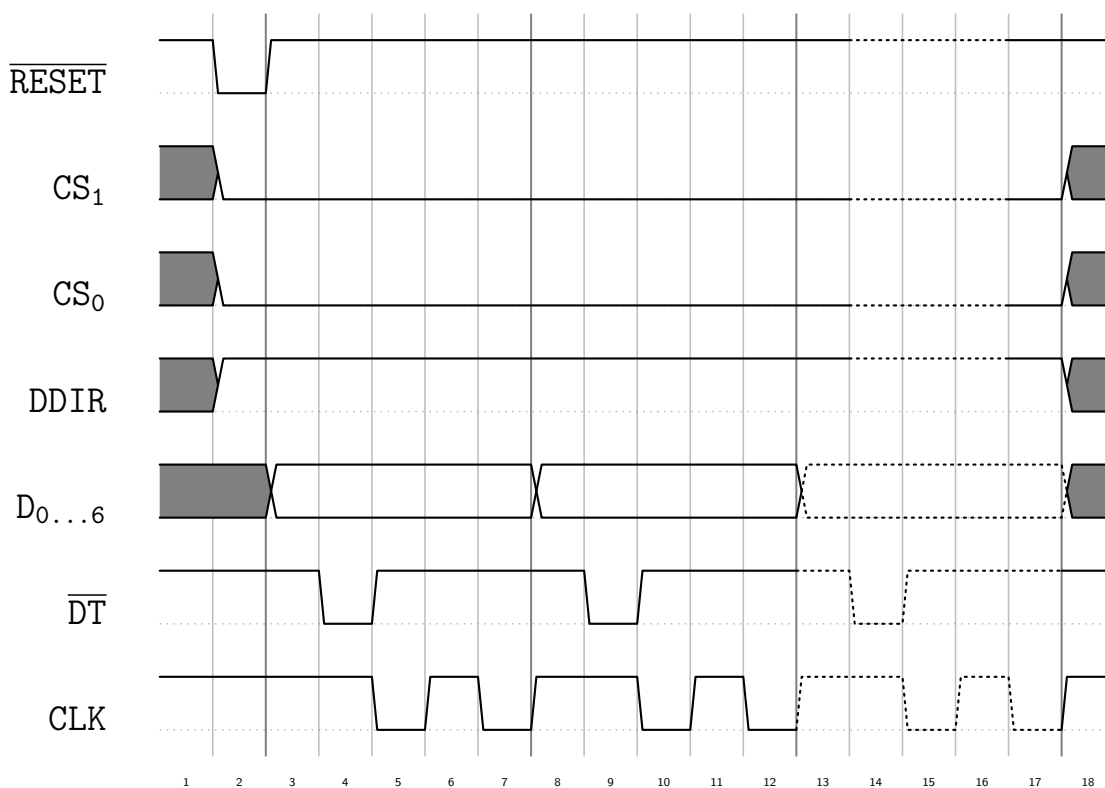


Rysunek 2: Wykres czasowy przygotowania maszyny do programowania.

3.2.4.2 Programowanie modułu TAPE (taśmy)

Programowanie modułu TAPE zaczyna się przygotowaniem go do programowania, czyli wybraniem modułu, z użyciem sygnałów $CS_{0,1}$, ustaleniem kierunku przepływu danych sygnałem $DDIR$ - poziom wysoki dla zapisu - oraz puszczeniem pulsu sygnału \overline{RESET} .

Następnie zapis słowa wejściowego taśmy odbywa się w pętli: ustala się sybol do zapisu na magistrali $D_0 \dots 6$ (proszę zwrócić uwagę na indeks $0 \dots 6$ - symbol ma 7 bitów), aktywuje się przesył danych (w tym przypadku zapis) do modułu pulsem sygnału \overline{DT} oraz wysła się dwa pulsy zegara CLK w celu przejścia do następnej komórki pamięci przechowującej zawartość taśmy.



Rysunek 3: Wykres czasowy programowania TAPE.

Na wykresie czasowym kolumny 1-2 reprezentują przygotowanie TAPE do programowania. 2-6 i 7-11 przedstawiają dwa pojedyncze zapisy symboli do dwóch kolejnych komórek pamięci w TAPE. Sygnały w kolumnach 13-17 zostały wykropkowane w celu wyeksponowania wielokrotnego powielenia operacji zapisu symbolu.

3.2.4.3 Programowanie modułu SYMBOL TRANSLATOR

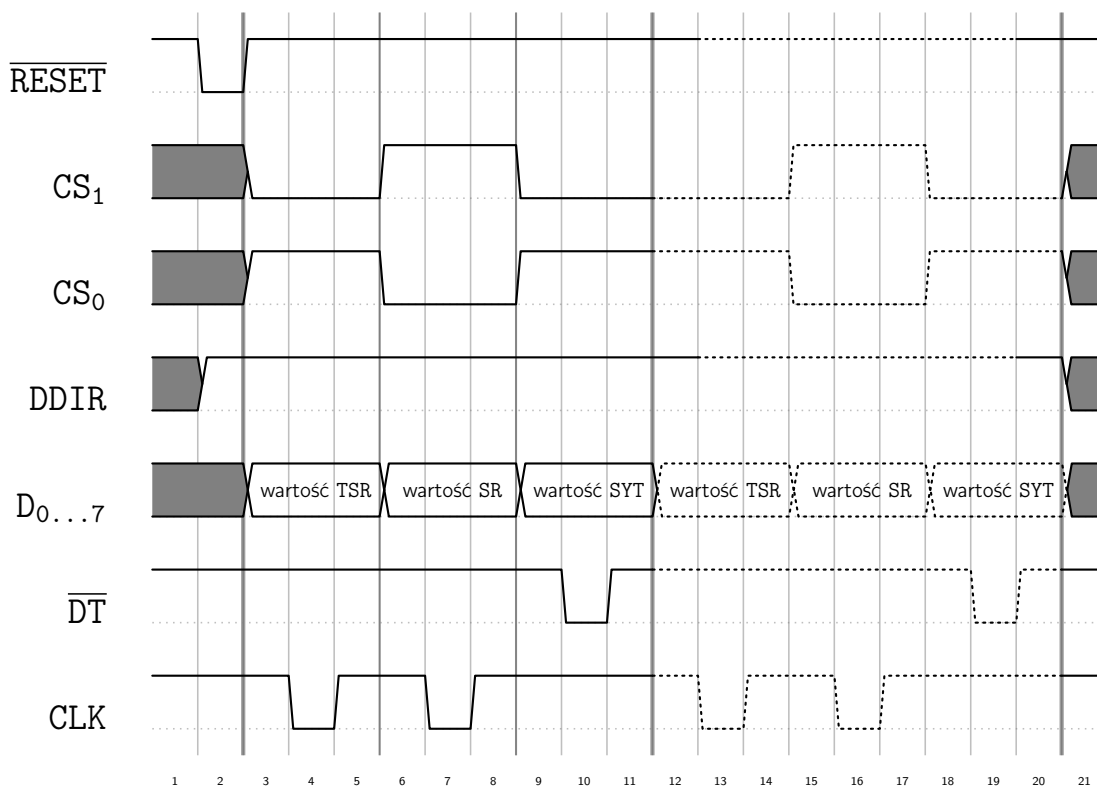
Podobnie jak w przypadku programowania TAPE, SYMBOL TRANSLATOR również należy przygotować do programowania. W tym celu trzeba wybrać jedynie kierunek DDIR oraz puścić puls sygnału $\overline{\text{RESET}}$.

Zapis pojedynczej wartości funkcji przejścia z wynikiem w postaci symbolu oraz kierunku przesunięcia głowicy jest bardziej skomplikowany w porównaniu do zapisu pojedynczego symbolu taśmy. Główny problem leży w określeniu, która wartość funkcji ma zostać wgrana do modułu, czyli która komórka pamięci ma być zapisana. Aby to sprecyzować należy dokonać zapisu w dwóch rejestrach, które bezpośrednio określają wskazywane miejsce w pamięci: TAPE SYMBOL REGISTER oraz STATE REGISTER. Te dwa rejestry odpowiadają argumentom funkcji przejścia: stanowi oraz symbolowi.

Zatem, żeby wskazać miejsce w pamięci modułu SYMBOL TRANSLATOR do którego należy zapisać wartość, trzeba wpierw wysłać symbol będący jednym z argumentów funkcji przejścia na magistralę $D_0 \dots 6$, wybrać wewnętrzną magistralę $S_0 \dots 6$ sygnałami $CS_{0,1}$ oraz wysłać puls zegara w celu zapisania wartości magistrali w TAPE SYMBOL REGISTER. Następnie trzeba wysłać stan będący drugim z argumentów funkcji przejścia na magistralę $D_0 \dots 7$, wybrać wewnętrzną magistralę $Q_0 \dots 7$ sygnałami $CS_{0,1}$ oraz wysłać puls zegara w celu zapisania wartości magistrali w STATE REGISTER.

Po tym miejsce w pamięci modułu SYMBOL TRANSLATOR jest wybrane i można przystąpić do zapisu symbolu oraz kierunku wartości funkcji przejścia. Osiąga się to za pomocą wyboru modułu sygnałami $CS_{0,1}$,

umieszczeniu danych do zapisu na magistrali $D_0...7$ oraz puszczeniem pulsu sygnału \overline{DT} .



Rysunek 4: Wykres czasowy programowania SYMBOL TRANSLATOR.

Oznaczenia:

TSR – TAPE SYMBOL REGISTER,

SR – STATE REGISTER,

SYT – SYMBOL TRANSLATOR.

Na wykresie czasowym kolumny 1-2 reprezentują przygotowanie do programowania; 3-11 przedstawiają programowanie pojedynczej wartości: 3-5 odpowiada za zapis do TAPE SYMBOL REGISTER, 6-8 za zapis do STATE REGISTER, 9-11 za zapis do SYMBOL TRANSLATOR. Kolumny 12-20 również przedstawiają programowanie wartości do

SYMBOL TRANSLATOR, jednak zostały one wykropkowane w celu wyeksponowania wielokrotnego powtórzenia tej operacji.

W przypadku programowania wartości funkcji dla argumentu ze stanem akceptującym po wykonaniu zapisu do STATE REGISTER należy puścić puls sygnału $\overline{\text{RESET}}$ w celu wyjścia maszyny ze stanu zatrzymania pracy (zatrzymania wpływu zegara na przejścia między stanami STEP EXECUTION STATE REGISTER). Jednak można zauważyć, że programowanie tych przejść nie ma sensu ze względu na zatrzymanie pracy urządzenia zaraz po przejściu do stanu akceptującego.

3.2.4.4 Programowanie modułu STATE TRANSLATOR

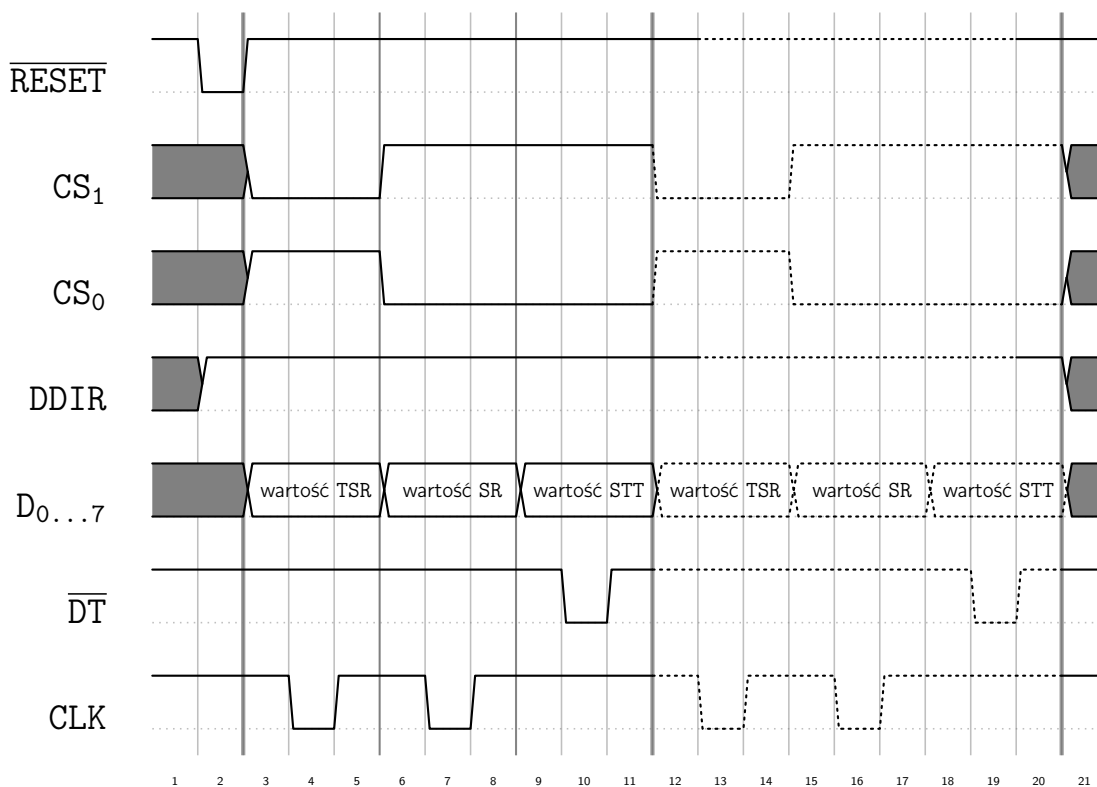
Programowanie STATE TRANSLATOR odbywa się w sposób bardzo zbliżony do SYMBOL TRANSLATOR. Wpierw należy przygotować moduł do programowania co dzieje się poprzez ustawienie kierunku sygnałem DDIR oraz puszczenie pulsu sygnału $\overline{\text{RESET}}$.

Zapis pojedynczej wartości funkcji przejścia z wynikiem w postaci stanu wymaga dokładnie tego samego procesu wybrania komórki pamięci w STATE TRANSLATOR co w SYMBOL TRANSLATOR. Z tego tytułu opis tego zadania nie został tu omówiony (jednak wciąż znajduje się on na wykresie czasowym).

Po wybraniu miejsca w pamięci modułu STATE TRANSLATOR następuje zapis stanu wartości funkcji przejścia: wybierany jest moduł sygnałami $\text{CS}_{0,1}$, umieszczana jest wartość na magistrali $\text{D}_0 \dots 7$ oraz puszczone zostaje puls sygnału $\overline{\text{DT}}$ w celu zapisania danych w pamięci.

Podobnie jak na wykresie czasowym programowania SYMBOL TRANSLATOR kolumny 1-2 reprezentują przygotowanie do programowania; 3-11 przedstawiają programowanie pojedynczej wartości: 3-5 odpowiada za zapis do TAPE SYMBOL REGISTER, 6-8 za zapis do STATE REGISTER, 9-11 za zapis do STATE TRANSLATOR. Kolumny 12-20 również przedstawiają programowanie wartości do

STATE TRANSLATOR, jednak zostały one wykropkowane w celu wyeksponowania wielokrotnego powtórzenia tej operacji.



Rysunek 5: Wykres czasowy programowania SYMBOL TRANSLATOR.

Oznaczenia:

TSR – TAPE SYMBOL REGISTER,

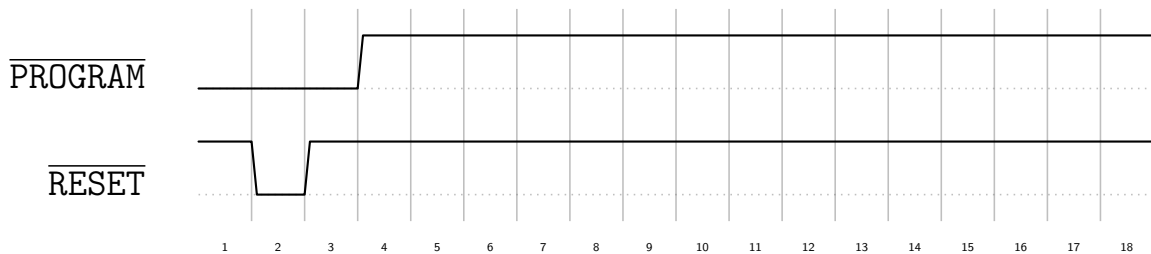
SR – STATE REGISTER,

STT – STATE TRANSLATOR.

Po wykonaniu powyższych czynności, tj. zaprogramowaniu modułów TAPE, SYMBOL TRANSLATOR oraz STATE TRANSLATOR urządzenie jest gotowe do wykonania zapisanego programu.

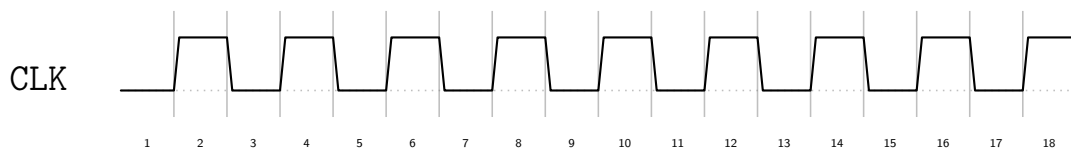
3.2.5 Wykonanie Programu

Mając zaprogramowaną maszynę w celu uruchomienia i wykonania jej programu należy ją przygotować. Odbywa się to poprzez wysłanie pulsu sygnału $\overline{\text{RESET}}$ oraz przejście sygnału $\overline{\text{PROGRAM}}$ do stanu wysokiego.



Rysunek 6: Wykres czasowy przygotowania maszyny do pracy.

Następnie całe działanie maszyny opiera się o sygnał zegara CLK. Każde dwa pulsy tego sygnału oznaczają wykonanie jednego ruchu urządzenia. Zegar powinien być dostarczony do momentu osiągnięcia przez maszynę stanu akceptującego zatrzymującego jej działanie.



Rysunek 7: Wykres czasowy przykładowego działania zegara.

3.2.6 Odczyt Wyniku Programu

Po osiągnięciu przez maszynę stanu akceptującego

4 Fizyczny Opis Działania