

# Tura

Fizyczna implementacja maszyny Turinga

Paweł Szymański

Luty 2026

## Spis treści

|          |                                  |           |
|----------|----------------------------------|-----------|
| <b>1</b> | <b>Wstęp</b>                     | <b>2</b>  |
| <b>2</b> | <b>Opis zastosowanego modelu</b> | <b>2</b>  |
| <b>3</b> | <b>Model Działania</b>           | <b>4</b>  |
| 3.1      | Opis Ogólny . . . . .            | 4         |
| 3.2      | Opis Szczegółowy . . . . .       | 5         |
| 3.2.1    | Moduły . . . . .                 | 5         |
| 3.2.2    | Sygnały . . . . .                | 6         |
| 3.2.3    | Interfejs zewnętrzny . . . . .   | 10        |
| 3.2.4    | Programowanie . . . . .          | 11        |
| <b>4</b> | <b>Fizyczny Opis Działania</b>   | <b>11</b> |

# 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,
- $\Gamma$  - alfabet taśmy,

- $\Sigma$  - alfabet słowa wejściowego,  $\Sigma \subseteq \Gamma$ ,
- $\delta$  - 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.

[obrazek]

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 działanie 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 ( $D_0 \dots D_6$ );

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

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

**STATE REGISTER** – Rejestr przechowujący obecny stan maszyny;

HEADER POSITION – Zbiór liczników binarnych z możliwością inkrementacji oraz dekrementacji symulujący głowicę maszyny. Wartość licznika reprezentuje usytuowanie głowicy na taśmie;

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

HEADER POSITION CLOCK SYNCHRONIZATION – Bramka AND synchronizująca operację 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 interfejsu zewnętrznego od tych wewnętrznych;

IN-OPERATION CONTROL LINES BUFFER – Bufor oddzielający wewnętrzne linie kontrolne od tych generowanych przez urządzenie podczas pracy. Rozdzielenie to jest potrzebne w celu umożliwienia programowania maszyny (dostępu sygnałów zewnętrznych do wewnętrznych);

### 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 połączony koniunkcją z zegarem:

$HP\_EN\_CLK = HP\_EN \cdot CLK$ ; 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;
- $\overline{\text{RESET}}$  – Resetuje urządzenie, czyli ustawia STEP EXECUTION STATE REGISTER do stanu 0 oraz ustawia licznik HEADER POSITION na 0;
- $\overline{\text{PROGRAM}}$  – Ustala tryb pracy. Poziom niski sygnału oznacza tryb programowania maszyny, poziom wysoki zaś tryb wykonywania programu;
- PROGRAM – Przeciwnieństwo sygnału  $\overline{\text{PROGRAM}}$ ; sygnał ten jest równoważny  $\overline{\text{OPERATE}}$  – 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<sub>0</sub>, CS<sub>1</sub> na magistralę D<sub>0</sub>...7. Wysoki poziom sygnału oznacza zapis danych z interfejsu zewnętrznego do bloków pamięci wewnątrz urządzenia. Niski poziom oznacza odczyt;
- CS<sub>0</sub>, CS<sub>1</sub> – Kombinacja tych sygnałów wybiera wewnętrzny moduł do odczytu/zapisu danych z zewnętrznego interfejsu (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 taśmy S<sub>0</sub>...6 do TAPE SYMBOL REGISTER; zapis odbywa się podczas przejścia sygnału ze stanu niskiego do wysokiego;



- SR\_WR – Zapisuje dane z magistrali stanu  $Q_0 \dots 7$  do STATE REGISTER; zapis odbywa się podczas przejścia sygnału ze stanu niskiego do wysokiego;
- T\_ $\overline{OE}$  – Włącza wyjście danych z TAPE na magistralę  $S_0 \dots 6$ ; aktywny dla niskiego poziomu;
- T\_ $\overline{WE}$  – Włącza zapis danych w TAPE z magistrali  $S_0 \dots 6$ ; aktywny dla niskiego poziomu;
- SYT\_ $\overline{OE}$  – Włącza wyjście danych z SYMBOL TRANSLATOR na magistralę  $S_0 \dots 6$  oraz sygnał DIR\_B; aktywny dla niskiego poziomu;
- SYT\_ $\overline{WE}$  – Włącza zapis danych w SYMBOL TRANSLATOR z magistrali  $S_0 \dots 6$  oraz sygnału DIR\_B; aktywny dla niskiego poziomu;
- STT\_ $\overline{OE}$  – Włącza wyjście danych z STATE TRANSLATOR na magistralę  $Q_0 \dots 7$ ; aktywny dla niskiego poziomu;
- STT\_ $\overline{WE}$  – Włącza zapis danych w STATE TRANSLATOR z magistrali  $Q_0 \dots 7$ ; aktywny dla niskiego poziomu;
- QR\_ $\overline{OE}$  – Włącza wyjście danych z STATE REGISTER na magistralę  $D_0 \dots 7$ ; aktywny dla niskiego poziomu;
- T\_ $\overline{*E}_E$ , SYT\_ $\overline{*E}_E$ , ST\_ $\overline{*E}_E$ , QR\_ $\overline{OE}_E$  – Sygnały odpowiadające tym bez sufiksu „\_E”, będące połączone z nimi przez bufor U18 oraz będące wynikiem dekodera wyboru funkcji U19;
- T\_ $\overline{*E}_B$ , SYT\_ $\overline{OE}_B$ , STT\_ $\overline{OE}_B$ , 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 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.

$\text{SEST\_STATE}$  – 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 20 wyjść sterujących, 17 informujących o stanie maszyny oraz 7 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 <sub>0...7</sub>          | In/Out   |
| 9            | CLK                         | In       |
| 10           | $\overline{\text{RESET}}$   | In       |
| 11           | $\overline{\text{PROGRAM}}$ | In       |
| 12           | DDIR                        | In       |

Tabela 1: Lista sygnałów interfejsu zewnętrznego

### 3.2.4 Programowanie

Programowanie maszyny odbywa się za pomocą interfejsu zewnętrznego. Dzięki odpowiedniemu wykorzystaniu udostępnionych sygnałów

## 4 Fizyczny Opis Działania

| Num. wyjścia   | Sygnały                    | Kierunek |
|----------------|----------------------------|----------|
| 13-14          | $CS_{0,1}$                 | In       |
| 15             | $\overline{DT}$            | In       |
| 16             | $\overline{FINISH\_STATE}$ | Out      |
| 17,19,21,23,25 | $V_{CC}$                   | —        |
| 18,20,22,24,26 | $V_{SS}$                   | —        |
| 27-42          | $H_{0 \dots 15}$           | Out      |
| 43             | $SESR\_STATE$              | Out      |
| 44             | $V_{SS}$                   | —        |

Tabela 2: Lista sygnałów interfejsu zewnętrznego (cd.)